



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Termografía infrarroja activa aplicada al análisis de defectos
internos simulados en materiales poliméricos fabricados por
impresión 3D*

Grado en Ingeniería Mecánica

ALUMNO: Gampon Naseeto

DIRECTORES: Iván Garrido González

CURSO ACADÉMICO: 2025-2026

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Termografía infrarroja activa aplicada al análisis de defectos
internos simulados en materiales poliméricos fabricados por
impresión 3D*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

Este Trabajo de Fin de Grado evalúa la eficacia de la termografía infrarroja activa como Ensayo No Destructivo (END) para detectar y caracterizar defectos internos en componentes fabricados por impresión 3D. El estudio compara dos polímeros con propiedades termofísicas diferentes: ácido poliláctico (PLA) y tereftalato de polietileno glicolizado (PETG) reforzado con fibra de carbono (PETG-CF).

La metodología abarca el diseño paramétrico de probetas con defectos controlados mediante Autodesk Inventor, su fabricación por modelado por deposición fundida en una impresora Bambu Lab y la ejecución de ensayos térmicos con una cámara FLIR E85 y diversas fuentes de excitación. Los datos extraídos se procesaron aplicando algoritmos avanzados de tratamiento de imagen: Reconstrucción de la Señal Termográfica (TSR), Termografía de Componentes Principales (PCT) y su variante dispersa (SPCT).

Los resultados demuestran que el PLA usado ofrece un comportamiento ligeramente superior para la inspección térmica. Además, el algoritmo SPCT destacó como el método más robusto por su excelente relación señal-ruido (SNR) y capacidad de segmentación, aunque no son excluyentes, aportando información añadida los otros dos algoritmos.

En conclusión, se valida la termografía infrarroja activa como una técnica altamente eficaz para inspeccionar polímeros 3D, destacando su gran potencial para la Industria 4.0.

PALABRAS CLAVE

Termografía infrarroja activa, fabricación aditiva (*Fused Deposition Modeling*, FDM), termografía de componentes principales (*Principal Component Thermography*, PCT), termografía dispersa de componentes principales (*Sparse Principal Component Thermography*, SPCT), Reconstrucción de Señales Termográficas (*Thermography Signal Reconstruction*, TSR)

AGRADECIMIENTOS

En primer lugar, deseo expresar mi más sincero agradecimiento a Iván Garrido González, tutor de este Trabajo Fin de Grado, por su constante apoyo, orientación y dedicación a lo largo de todo el desarrollo del proyecto. Su disponibilidad para resolver dudas, así como sus valiosas aportaciones, han sido fundamentales para la correcta realización de este trabajo.

Asimismo, quisiera agradecer a Antonio Eirís Barca por su ayuda y asesoramiento en el uso de la impresora 3D, así como por compartir sus conocimientos técnicos durante la fase de fabricación de las probetas.

También quiero expresar mi gratitud a mis compañeros de promoción, con quienes he compartido estos cinco años de formación. A pesar de las dificultades iniciales derivadas de ser estudiante internacional procedente de Tailandia, su apoyo, compañerismo y cercanía han sido esenciales para mi adaptación, haciéndome sentir como en una segunda casa.

Por último, deseo agradecer profundamente a mi familia, a mi pareja y a mis amigos, por su apoyo incondicional, comprensión y motivación constante durante todo este proceso, lo que ha hecho posible alcanzar los objetivos de este Trabajo Fin de Grado.

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	9
1 Introducción y objetivos	10
1.1 Introducción	10
1.2 Objetivos	11
1.3 Estructura del trabajo	11
2 Estado del arte	13
2.1 Materiales poliméricos y problemática asociada a defectos internos	13
2.1.1 Definición y características generales de los polímeros	13
2.1.2 Importancia y aplicaciones de los polímeros en entornos navales y de Defensa	16
2.1.3 Defectos internos en materiales poliméricos y su origen	19
2.1.4 Consecuencias de los defectos internos y necesidad de inspección	23
2.2 Técnicas de Ensayo No Destructivo aplicadas a la detección de defectos	26
2.2.1 Concepto y relevancia de los Ensayos No Destructivos.....	26
2.2.2 Técnicas END empleadas en materiales poliméricos	29
2.2.3 Ventajas de la termografía infrarroja como técnica END	33
2.3 Termografía infrarroja activa y procesamiento de datos térmicos.....	34
2.3.1 Fundamentos físicos de la termografía infrarroja	34
2.3.2 Termografía infrarroja activa: concepto y modalidades	37
2.3.3 Fase de adquisición térmica.....	41
2.3.4 Procesado	42
2.4 Aplicaciones de la termografía infrarroja activa en materiales poliméricos impresos en 3D.....	47
2.4.1 Fabricación aditiva de polímeros y problemática asociada	47
2.4.2 Aplicación de la termografía infrarroja activa en polímeros comunes	51
3 Desarrollo del TFG.....	54
3.1 Materiales usados	54
3.1.1 PETG-CF	54
3.1.2 PLA.....	54
3.1.3 Comparación entre PETG-CF y PLA	55
3.2 Cámara infrarroja empleada: FLIR E85.....	56
3.3 Preparación de las probetas.....	57
3.3.1 Modelado 3D mediante Autodesk Inventor.....	57

3.3.2 Impresión 3D mediante Bambu Lab	64
3.4 Campaña de las mediciones termográficas	68
3.5 Aplicaciones de los algoritmos de procesamiento de imagen.....	70
4 Resultados y discusión	86
4.1 Configuración de los resultados	86
4.2 Resultados en PETG-CF	88
4.2.1 Caso de estudio 1: PETG-CF con lámpara halógena sin tapa	88
4.2.2 Caso de estudio 2: PETG-CF con lámpara halógena con tapa	93
4.2.3 Caso de estudio 3: PETG-CF con lámpara infrarroja sin tapa.....	96
4.2.4 Caso de estudio 4: PETG-CF con lámpara infrarroja con tapa	100
4.2.5 Comparación de los resultados en el PETG-CF	104
4.3 Resultados en PLA.....	105
4.3.1 Caso de estudio 5: PLA con lámpara halógena sin tapa	105
4.3.2 Caso de estudio 6: PLA con lámpara halógena con tapa.....	110
4.3.3 Caso de estudio 7: PLA con lámpara infrarroja sin tapa	113
4.3.4 Caso de estudio 8: PLA con lámpara infrarroja con tapa	119
4.3.5 Comparación de los resultados en el PLA	124
4.4 Comparación resultados entre PETG-CF y PLA	125
5 Conclusiones y líneas futuras	127
5.1 Conclusión	127
5.2 Líneas futuras	128
6 Bibliografía.....	129
Anexo I: Implicaciones Sociales, y/o Económicas, y/o Ambientales	138
Anexo II: Reflexiones Éticas y Sociales	140
Anexo III: Código de Procesamiento y Algoritmos Técnicos	141
Anexo IV: Declaración sobre la Utilización de Herramientas de Inteligencia Artificial.....	180

ÍNDICE DE FIGURAS

Figura 2-1 Estructura macromolecular de los polímeros (Extraído de [4]).....	13
Figura 2-2 Clasificación de los polímeros (Extraído de [5]).....	14
Figura 2-3 Polímeros como matriz en materiales compuestos a) CFRP (Extraído de [7]), b) GFRP (Extraído de [8]).....	14
Figura 2-4 Defectos internos en polímeros fabricados por impresión 3D (Extraído de [11])	15
Figura 2-5 Uso de materiales compuestos en estructuras navales (Extraído de [12])	16
Figura 2-6 Corbetas de la clase Visby (Extraído de [17]).....	17
Figura 2-7 La corbeta de la clase Pohjanmaa de la Marina finlandesa, actualmente en construcción, están equipadas con el Mástil Integrado Ligero de Saab (SLIM) (Extraído de [19]).....	18
Figura 2-8 Nueva nota de orientación para hélices fabricadas con materiales compuestos (Extraído de [22]).....	19
Figura 2-9 Defectos internos en polímeros (Extraído de [26])	20
Figura 2-10 Defectos inducidos por fabricación aditiva (FDM) a) diagrama de formación de vacíos subperimetrales. La región marcada con ▲ corresponde a un vacío subperimetral formado entre el perímetro y las trayectorias de relleno (rasters), b) diagrama de una capa de la fabricación por filamento fundido (Fused Filament Fabrication, FFF) rectangular genérica, c) micrografía SEM de la superficie superior de una pieza FFF con vacíos subperimetrales (s), d) secciones transversales de las probetas. Las flechas rojas señalan los vacíos subperimetrales. (Extraído de [30])	21
Figura 2-11 Micrografía óptica de la delaminación entre los grupos de capas a 0° y 90° en un laminado dañado. (Extraído de [33])	21
Figura 2-12 Esquema del daño por impacto apenas visible (BVID) en una estructura compuesta. (Extraído de [34]).....	22
Figura 2-13 Abaqus/CAE que muestran la variación de la tensión principal máxima / concentración de tensiones para diferentes ubicaciones de un poro con un diámetro de 0,3 mm (Extraído de [37]) ...	23
Figura 2-14 Patrones de fractura de componentes en el fallo por fatiga (Extraído de [40]).....	24
Figura 2-15 Un gran buque en dique seco con andamios y equipos que muestran detalles de su casco y hélice durante una sesión de reparación (Extraído de [46]).....	25
Figura 2-16 Sistema de detección de defectos mediante termografía infrarroja (Extraído de [49])	25
Figura 2-17 Ensayo por ultrasonidos (UT) (Extraído de [62]).....	27
Figura 2-18 Termografía infrarroja frente a otras técnicas END a) equipos y accesorios para inspección visual b) ensayo por emisión acústica c) ensayo mediante termografía infrarroja d) ensayo por ultrasonidos (Extraído de [68]).....	28
Figura 2-19 Equipo de C-scan por ultrasonidos para la evaluación de daños por impacto (Extraído de [70]).....	29
Figura 2-20 Configuración del ensayo de emisión acústica (AE) (Extraído de [74]).....	30
Figura 2-21 Inspección por corrientes inducidas de estructuras aeronáuticas en servicio (Extraído de [79]).....	30

Figura 2-22 Defectos de delaminación en cilindros de almacenamiento de hidrógeno compuestos: escaneo por tomografía computarizada (TC) y medición mediante shearografía (Extraído de [85]).....	31
Figura 2-23 Radiografía industrial y XCT a) configuración típica de un equipo XCT de laboratorio, b) representación esquemática del sistema y c) procedimiento paso a paso para el análisis de datos XCT. (Extraído de [88]).....	32
Figura 2-24 Esquema conceptual de la termografía infrarroja activa: aplicación de una excitación térmica externa y detección de anomalías superficiales asociadas a defectos internos (Extraído de [94])	32
Figura 2-25 Naturaleza sin contacto y remota (Extraído de [95]).....	33
Figura 2-26 Relación entre energía, frecuencia y longitud de onda de la radiación electromagnética según la ecuación de Planck. (Extraído de [98]).....	35
Figura 2-27 La intensidad de la radiación del cuerpo negro en función de la longitud de onda de la radiación emitida. Cada curva corresponde a una temperatura diferente del cuerpo negro, desde una temperatura baja (curva inferior) hasta una temperatura alta (curva superior) (Extraído de [99]).....	36
Figura 2-28 Comparación entre cuerpo negro y cuerpo gris, ilustrando la influencia de la emisividad en la radiación emitida (Extraído de [100])	36
Figura 2-29 Esquema del balance radiactivo real en termografía infrarroja, incluyendo radiación emitida por el objeto, radiación reflejada y contribución atmosférica (Extraído de [101]).....	37
Figura 2-30 Ejemplo que muestra imágenes de delaminación en un muro de un edificio durante el ciclo de calentamiento a) imagen visual, b) imagen infrarroja (IR). Las flechas indican las zonas con delaminación. (Extraído de [102])	37
Figura 2-31 Principio de la detección mediante imagen térmica por ondas infrarrojas (Extraído de [103]).....	38
Figura 2-32 Diagrama esquemático de la termografía pulsada en configuración de reflexión (Extraído de [104]).....	38
Figura 2-33 Configuración del sistema de termografía infrarroja (IRT) en modo transmisión (Extraído de [105]).....	39
Figura 2-34 Configuración para termografía pulsada con cuatro lámparas flash (Extraído de [106])	39
Figura 2-35 Esquema del sistema de termografía Lock-in (LIT) (Extraído de [108]).....	40
Figura 2-36 Ensayo de termografía por calentamiento escalonado a) posición esquemática de los vacíos y de las barras de refuerzo en la placa, b) configuración experimental 1-cámara IR, 2-lámparas halógenas, 3-placa de hormigón ensayada, 4-ordenador, c) termo gramas de cada defecto a los 10 minutos (Extraído de [110]).....	40
Figura 2-37 Procedimiento de calibración radiométrica mediante cuerpo negro de referencia (Extraído de [111]).....	41
Figura 2-38 Comparación entre termogramas RAW y termogramas reconstruidos tras aplicar el algoritmo PCT a) muestra el termograma original (RAW), donde el defecto apenas se distingue debido al bajo contraste térmico, b) presenta la imagen procesada mediante el algoritmo PCT (Transformada de Componentes Principales), en la que el defecto se visualiza con mayor claridad gracias al aumento del contraste y la reducción del ruido, c) muestra el termograma original (RAW), d) corresponde a la imagen obtenida tras aplicar el algoritmo PCT, donde el defecto aparece más definido y con mejor diferenciación respecto al fondo (Extraído de [112])	43

Figura 2-39 Ajuste polinómico de la curva térmica mediante TSR y cálculo de derivadas para mejorar detección de defectos (Extraído de [113]).....	44
Figura 2-40 Resultados obtenidos mediante la aplicación de algunos algoritmos de procesamiento de datos de termografía infrarroja (IRT) de última generación a las marqueterías bajo inspección (Extraído de [114]).....	45
Figura 2-41 Esquema de la Transformada Discreta de Fourier aplicada a una secuencia de termogramas (Extraído de [115]).....	45
Figura 2-42 Resultado visual de la detección de la peor, intermedia y mejor posición del defecto, así como de la segmentación del área del defecto, aplicando el conjunto de pesos y sesgos obtenidos en el mejor proceso de entrenamiento de Mask R-CNN sobre el conjunto de validación (Extraído de [114])	46
Figura 2-43 Simulación mediante el método de elementos finitos de la conducción térmica del sensor durante 3 horas a) variación de la temperatura en la superficie superior de la carcasa, b) datos de temperatura en la superficie superior (Extraído de [116])	47
Figura 2-44 a) proceso de seccionado de modelos 3D en capas y trayectorias para su impresión b) proceso de fabricación e ilustración del ancho de corte (slicing width) (Extraído de [118])	48
Figura 2-45 a) representación esquemática del proceso de extrusión de filamento. Los fragmentos de material compuesto amasado son impulsados a través de un cilindro calefactado mediante un tornillo sin fin, extruidos a través de una boquilla de diámetro definido y posteriormente bobinados b) proceso de impresión FDM: un sistema motorizado introduce el filamento en el cabezal de impresión con control de temperatura, montado sobre una plataforma XY. Tras pasar por la boquilla, el material se deposita sobre la placa de fabricación, que desciende progresivamente para permitir la conformación de la siguiente capa (Extraído de [121]).....	49
Figura 2-46 Esquemática del proceso de estereolitografía (SLA) (Imagen cortesía de 3D Systems) (Extraído de [122]).....	49
Figura 2-47 Efecto escalera en superficies inclinadas debido a la deposición capa a capa (Extraído de [124]).....	50
Figura 2-48 Imagen de una pieza FFF con relleno tipo gyroid a) seccionado del modelo, b) sección transversal en el plano XZ de la pieza, c) pieza impresa, d) sección transversal en el plano XY de la pieza (Extraído de [30]).....	50
Figura 2-49 Vacíos en piezas impresas mediante FDM a) Imagen obtenida con Microscopio Electrónico de Barrido (SEM) que muestra los vacíos típicos inherentes al proceso FDM, b) Imagen óptica de una probeta tipo “dog-bone” donde se observan claramente los vacíos (Extraído de [126])..	52
Figura 2-50 Comparación de los resultados obtenidos a partir de datos en bruto y mediante el uso de PCT a) termograma, b) EOF1, c) EOF2 (Extraído de [129])	53
Figura 2-51 Extracción de características mediante una red neuronal convolucional con arquitectura codificador-decodificador (Extraído de [130])	53
Figura 3-1 Cámara FLIR E85 (Extraído de [139]).....	56
Figura 3-2 El modelo tridimensional de la base principal (Elaboración propia)	57
Figura 3-3 El boceto bidimensional de la base principal (Elaboración propia)	58
Figura 3-4 El proceso de extrusión (Elaboración propia)	58
Figura 3-5 El resultado final tras aplicar la operación de extrusión que genera las cavidades internas (Elaboración propia)	59

Figura 3-6 El boceto de las paredes laterales (Elaboración propia)	59
Figura 3-7 La extrusión generada de las paredes laterales (Elaboración propia).....	60
Figura 3-8 El boceto de la tapa (Elaboración propia)	60
Figura 3-9 La extrusión generada de la tapa (Elaboración propia)	61
Figura 3-10 los planos de la base (a) y de la tapa (b) (Sistema Europeo, escala 1:1 y unidades en mm) (Elaboración propia).....	63
Figura 3-11 Impresora 3D Bambu Lab P1S empleada para la fabricación de las probetas experimentales mediante tecnología de deposición de filamento fundido (FFF) (Elaboración propia).	64
Figura 3-12 La tapa en el software Bambu Studio antes del proceso de laminado (Elaboración propia)	65
Figura 3-13 Las trayectorias generadas por el software durante el proceso de laminado (Elaboración propia)	65
Figura 3-14 Impresión de la base principal (Elaboración propia).....	66
Figura 3-15 Probeta Base principal vista desde la superficie posterior (Elaboración propia)	66
Figura 3-16 Probeta Base principal vista desde la superficie frontal (Elaboración propia).....	67
Figura 3-17 Probeta Tapa vista desde la superficie frontal (Elaboración propia).....	67
Figura 3-18 Montaje experimental utilizando la lámpara halógena de 500 W (Elaboración propia)	69
Figura 3-19 Montaje experimental utilizando la lámpara infrarroja de 2000 W (Elaboración propia)	69
Figura 3-20 Recorte de la región de interés de la probeta utilizando el software FLIR Tools (Elaboración propia)	70
Figura 3-21 El diagrama de flujo del código (Elaboración propia)	72
Figura 4-1 Referencia utilizada para identificar la posición y el tamaño relativo de los defectos presentes en la placa analizada, visto desde la superficie frontal, la superficie grabada con la cámara (Elaboración propia)	87
Figura 4-2 La comparación entre el SNR general y el máximo SNR por defecto para cada resultado algorítmico. Caso de estudio 1 (Elaboración propia).....	89
Figura 4-3 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	90
Figura 4-4 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	90
Figura 4-5 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	91
Figura 4-6 Comparación visual entre los resultados obtenidos a) SC_1 y b) SC_4 (Elaboración propia)	91
Figura 4-7 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia).....	92
Figura 4-8 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia)	92
Figura 4-9 La comparación entre el SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)	93
Figura 4-10 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	94

Figura 4-11 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	95
Figura 4-12 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	95
Figura 4-13 Comparación visual entre los resultados obtenidos a) SC_3 y b) EOF_4 (Elaboración propia)	96
Figura 4-14 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)	97
Figura 4-15 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	97
Figura 4-16 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	98
Figura 4-17 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	99
Figura 4-18 Comparación visual entre los resultados obtenidos a) SC_1 y b) skew_2 (Elaboración propia)	99
Figura 4-19 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)	100
Figura 4-20 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	101
Figura 4-21 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	102
Figura 4-22 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	102
Figura 4-23 Comparación visual entre los resultados obtenidos a) SC_1 y b) SC_5 (Elaboración propia)	103
Figura 4-24 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)	103
Figura 4-25 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia) ..	104
Figura 4-26 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)	106
Figura 4-27 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	106
Figura 4-28 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	107
Figura 4-29 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	107
Figura 4-30 Comparación visual entre los resultados obtenidos a) SC_1, b) SC_5 y c) EOF_1 (Elaboración propia)	108
Figura 4-31 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)	109
Figura 4-32 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia) ..	109
Figura 4-33 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)	110
Figura 4-34 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	111
Figura 4-35 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	112
Figura 4-36 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	112

Figura 4-37 Comparación visual entre los resultados obtenidos a) SC_3, b) EOF_2, c) EOF_3 y d) EOF_5 (Elaboración propia).....	113
Figura 4-38 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)	114
Figura 4-39 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	114
Figura 4-40 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	115
Figura 4-41 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	116
Figura 4-42 Comparación visual entre los resultados obtenidos a) SC_0, b) SC_3 y c) EOF_1 (Elaboración propia)	117
Figura 4-43 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia).....	118
Figura 4-44 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia).....	118
Figura 4-45 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)	119
Figura 4-46 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)	120
Figura 4-47 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)	120
Figura 4-48 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)	121
Figura 4-49 Comparación visual entre los resultados obtenidos a) SC_0, b) SC_1 y c) EOF_2 (Elaboración propia)	122
Figura 4-50 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia).....	123
Figura 4-51 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia) ..	123

ÍNDICE DE TABLAS

Tabla 2-1 Resumen de las diferencias entre la construcción en acero y la construcción con materiales compuestos (Extraído de [15]).....	17
Tabla 2-2 Tipos de defectos internos en polímeros y causas asociadas (tomada de [21])	23
Tabla 2-3 Clasificación de los distintos métodos (Extraído de [58])	26
Tabla 2-4 END frente a ensayos destructivos (Extraído de [60])	27
Tabla 2-5 Categorías de las técnicas de ensayo y evaluación no destructiva (Extraído de [66]).....	28
Tabla 2-6 Comparación general entre distintas técnicas END y su aplicabilidad a materiales poliméricos. (Extraído de [48]).....	33
Tabla 4-1 Resumen comparativo de los resultados obtenidos para PETG-CF en las distintas configuraciones experimentales (Elaboración propia).....	105
Tabla 4-2 Resumen comparativo de resultados para PLA en diferentes configuraciones experimentales (Elaboración propia)	125
Tabla 4-3 Resumen la comparación resultados entre PETG-CF y PLA (Elaboración propia).....	126

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

En las últimas décadas, los materiales poliméricos han adquirido un papel cada vez más relevante en numerosos sectores industriales gracias a su elevada relación resistencia-peso, facilidad de procesamiento, buena resistencia química y coste relativamente reducido frente a otros materiales estructurales tradicionales. Este crecimiento también se refleja en el mercado global de polímeros, cuyo valor superó los 750.000 millones de dólares en 2023, con previsiones de crecimiento a una tasa anual compuesta (*Compound Annual Growth Rate*, CAGR) de entre el 5,3 % y el 5,9 % durante la próxima década, pudiendo superar los 1,2 billones de dólares hacia 2033-2035. Esta expansión está impulsada principalmente por su creciente aplicación en sectores como la automoción, la aeronáutica, la industria naval, la electrónica y la fabricación aditiva, donde los polímeros se emplean cada vez con mayor frecuencia en componentes estructurales y funcionales de alto valor tecnológico [1].

Como consecuencia de esta expansión, los materiales poliméricos se utilizan cada vez con mayor frecuencia en aplicaciones estructurales y funcionales dentro de sectores tecnológicos avanzados, como la automoción, la aeronáutica, la industria naval y el ámbito de la defensa. En estos campos, los componentes fabricados con polímeros o materiales compuestos deben cumplir requisitos estrictos de fiabilidad mecánica, durabilidad y seguridad operativa, ya que suelen trabajar bajo condiciones exigentes de carga, temperatura o impacto. Por este motivo, resulta fundamental garantizar la integridad estructural de los componentes a lo largo de toda su vida útil mediante técnicas adecuadas de inspección y control.

Paralelamente, el desarrollo de la fabricación aditiva, y en particular de la impresión 3D de polímeros, ha permitido la producción de piezas con geometrías complejas y un alto grado de personalización. Sin embargo, la naturaleza capa a capa de estos procesos de fabricación favorece la aparición de defectos internos no visibles externamente, tales como porosidad, vacíos, inclusiones o delaminaciones entre capas. Estos defectos pueden afectar de manera significativa al comportamiento mecánico y térmico de los componentes, reduciendo su resistencia, su durabilidad y su fiabilidad en servicio.

Ante esta problemática, resulta imprescindible disponer de técnicas de inspección capaces de detectar y caracterizar defectos internos sin alterar la funcionalidad del componente. En este contexto, las técnicas de Ensayo No Destructivo (END) se presentan como herramientas fundamentales para el control de calidad, el mantenimiento preventivo y la evaluación de la integridad estructural de materiales y componentes. Entre ellas, la termografía infrarroja ha despertado un interés creciente debido a su carácter no invasivo, su rapidez de inspección y su capacidad para evaluar grandes superficies sin contacto físico.

En particular, la termografía infrarroja activa permite inducir un flujo de calor controlado sobre el material y analizar su respuesta térmica mediante una cámara infrarroja. La presencia de defectos internos altera la difusión del calor, generando huellas térmicas que pueden ser detectadas y analizadas. Esta técnica se presenta, por tanto, como una solución prometedora para la detección de defectos internos en materiales poliméricos, especialmente en piezas fabricadas mediante impresión 3D.

El presente Trabajo Fin de Grado se enmarca en este contexto y tiene como finalidad evaluar la aplicabilidad y el rendimiento de la termografía infrarroja activa como técnica de ensayo no destructivo para la detección de defectos internos, en este caso simulados, en materiales poliméricos fabricados por impresión 3D, contribuyendo al desarrollo de metodologías de inspección eficaces y accesibles para aplicaciones reales.

1.2 Objetivos

El objetivo de este Trabajo Fin de Grado es evaluar el rendimiento de la termografía infrarroja activa como técnica de ensayo no destructivo (END) para la detección y caracterización de defectos internos simulados en materiales poliméricos fabricados mediante tecnología de impresión 3D. Para ello, se diseñarán y fabricarán probetas poliméricas con defectos internos controlados, se aplicarán distintos métodos de excitación térmica artificial y se analizará la respuesta térmica obtenida mediante una cámara infrarroja y técnicas de procesamiento de imágenes térmicas, con el fin de valorar la eficacia, las limitaciones y el potencial de aplicación de esta técnica en contextos reales de inspección y mantenimiento.

En este estudio se emplearán específicamente dos materiales poliméricos utilizados habitualmente en fabricación aditiva: polietileno tereftalato glicolado (*Polyethylene Terephthalate Glycol*, PETG) reforzado con fibra de carbono (PETG-CF) y ácido poliláctico (*Polylactic Acid*, PLA). Los defectos internos simulados en las probetas consistirán en huecos internos llenos de aire (vacíos) de diferentes tamaños y profundidades, diseñados con el objetivo de reproducir discontinuidades internas que puedan aparecer durante los procesos de fabricación o durante el servicio de los componentes.

Las probetas serán fabricadas mediante una impresora 3D Bambu Lab P1S, utilizando tecnología de modelado por deposición de filamento fundido (*Fused Filament Fabrication*, FFF). Para la adquisición de la respuesta térmica se empleará una cámara termográfica FLIR E85, mientras que la excitación térmica se realizará mediante una lámpara halógena de 500W (configuración foco) en un escenario, y en otro escenario una lámpara infrarroja de 2000 W (configuración estufa), con el objetivo de analizar la influencia de la potencia térmica aplicada en la detectabilidad de los defectos internos (con lo que tiene puesto a continuación).

Finalmente, los datos térmicos obtenidos serán procesados mediante diferentes algoritmos avanzados de análisis de imágenes, incluyendo *Principal Component Thermography* (PCT), *Sparse Principal Component Thermography* (SPCT) y *Thermographic Signal Reconstruction* (TSR). Estos métodos permitirán mejorar la relación señal-ruido de las secuencias térmicas y facilitar la identificación y caracterización de los defectos internos presentes en los materiales estudiados.

1.3 Estructura del trabajo

El presente Trabajo Fin de Grado se organiza en cinco capítulos principales que permiten desarrollar el estudio de forma progresiva, partiendo de los fundamentos teóricos hasta llegar a la validación experimental de los resultados obtenidos.

El capítulo 1, titulado Introducción y objetivos, presenta el contexto general del trabajo y expone la motivación que justifica el estudio. En esta sección se introduce el papel creciente de los materiales poliméricos fabricados mediante impresión 3D en diferentes sectores industriales, así como la importancia de las técnicas de ensayo no destructivo para garantizar la integridad estructural de estos

componentes. Además, se definen los objetivos principales del trabajo y el enfoque seguido durante el desarrollo del proyecto.

El capítulo 2, correspondiente al Estado del arte, recoge los fundamentos teóricos necesarios para comprender el marco tecnológico del estudio. En este apartado se revisan los principios de funcionamiento de la termografía infrarroja activa, su aplicación como técnica de ensayo no destructivo y los trabajos más relevantes relacionados con la detección de defectos en materiales poliméricos y compuestos. Este capítulo proporciona la base científica necesaria para interpretar los resultados obtenidos en el desarrollo experimental.

El capítulo 3, denominado Desarrollo experimental del TFG, describe detalladamente el procedimiento experimental seguido durante el trabajo. En esta sección se explica el diseño de las probetas mediante el software Autodesk Inventor, su fabricación mediante impresión 3D utilizando una impresora Bambu Lab P1S y materiales poliméricos PETG-CF y PLA, así como la preparación del montaje experimental. También se describe el sistema de adquisición térmica basado en una cámara termográfica FLIR E85, el uso de lámparas como fuente de excitación térmica, la configuración de los distintos escenarios de ensayo y explicación de la aplicación de cada algoritmo de procesamiento de imagen utilizado.

El capítulo 4, titulado Resultados y discusión, presenta el análisis de los datos obtenidos durante los ensayos experimentales. En este apartado se estudia la respuesta térmica de las probetas y se evalúa la capacidad de detección de los defectos internos simulados mediante las técnicas de procesamiento de imágenes térmicas comentadas en el apartado 3.

Finalmente, el capítulo 5 recoge las Conclusiones del trabajo. En esta sección se sintetizan los principales resultados obtenidos, se evalúa el cumplimiento de los objetivos planteados y se discute el potencial de la termografía infrarroja activa como herramienta de inspección para materiales poliméricos fabricados mediante impresión 3D. Asimismo, se plantean posibles líneas de trabajo futuro orientadas a mejorar la detección y caracterización de defectos en este tipo de materiales.

2 ESTADO DEL ARTE

2.1 Materiales poliméricos y problemática asociada a defectos internos

2.1.1 Definición y características generales de los polímeros

Desde una perspectiva macromolecular, los materiales poliméricos se definen como compuestos orgánicos de elevado peso molecular, formados por la repetición sucesiva de unidades estructurales básicas denominadas monómeros [2],[3]. En la Figura 2-1 se muestra la estructura macromolecular de los polímeros. La naturaleza y disposición de estas cadenas moleculares confieren a los polímeros un comportamiento mecánico, térmico y químico diferenciado frente a otros materiales de ingeniería, como los metales o las cerámicas. En el ámbito de la ciencia e ingeniería de materiales, los polímeros se clasifican habitualmente en tres grandes familias atendiendo a su estructura interna y a su respuesta frente al incremento de temperatura: termoplásticos, termoestables y elastómeros [3]. En la Figura 2-2 se muestra la clasificación de los polímeros. Esta clasificación resulta esencial para el ingeniero mecánico, ya que condiciona tanto los procesos de fabricación disponibles como las limitaciones de uso en condiciones reales de servicio.

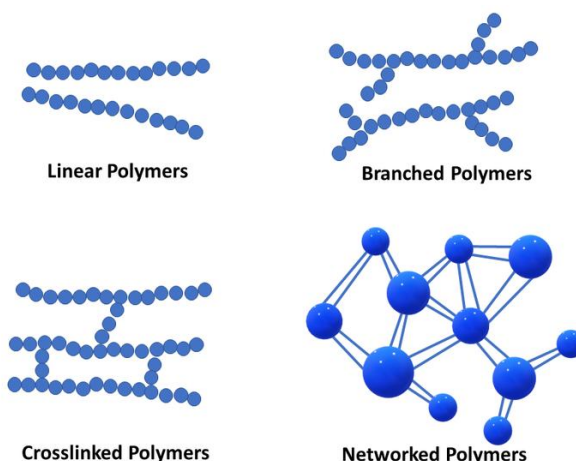


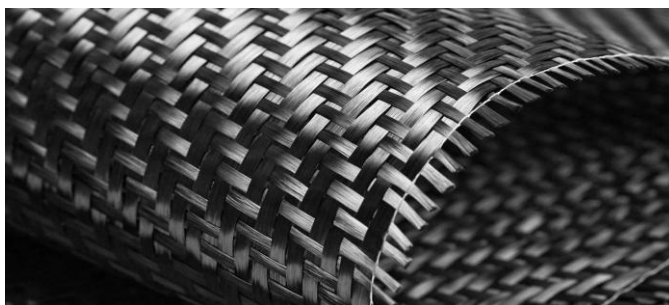
Figura 2-1 Estructura macromolecular de los polímeros (Extraído de [4])



Figura 2-2 Clasificación de los polímeros (Extraído de [5])

Los termoplásticos están constituidos por cadenas moleculares lineales o ligeramente ramificadas, unidas mediante enlaces secundarios relativamente débiles. Esta configuración permite que el material se funda al aplicar calor y se solidifique nuevamente al enfriarse, posibilitando su procesamiento de forma reversible mediante técnicas industriales como la extrusión, la inyección o la fabricación aditiva [3]. Por el contrario, los polímeros termoestables, como las resinas epoxi o fenólicas, experimentan durante su curado una transformación química irreversible que da lugar a una red tridimensional de enlaces covalentes fuertes. Como consecuencia, estos materiales no pueden volver a fundirse una vez curados, presentando una elevada rigidez estructural y una buena estabilidad térmica [3]. Finalmente, los elastómeros se caracterizan por cadenas moleculares altamente flexibles y parcialmente entrecruzadas, lo que les permite soportar grandes deformaciones elásticas bajo carga y recuperar prácticamente su geometría original cuando cesa el esfuerzo aplicado [3]. Esta diversidad estructural explica la amplia gama de aplicaciones de los polímeros y subraya la importancia de su correcta selección en función de los requisitos mecánicos y térmicos del diseño.

Desde un punto de vista funcional y de diseño estructural, los materiales poliméricos han adquirido una relevancia estratégica frente a los materiales metálicos tradicionales debido a una combinación de propiedades altamente competitivas, entre las que destacan su baja densidad, su elevada resistencia a la corrosión y su notable versatilidad para ser procesados en geometrías de elevada complejidad [2], [3]. En aplicaciones de alto rendimiento, como la industria aeroespacial, naval o de Defensa, los polímeros desempeñan con frecuencia el papel de matriz en materiales compuestos, cuya función principal es envolver, proteger y cohesionar los materiales de refuerzo, como fibras de carbono (*Carbon Fiber Reinforced Polymer*, CFRP) o de vidrio (*Glass Fiber Reinforced Polymer*, GFRP) [3], [6]. En la Figura 2-3 se muestra polímeros como matriz en materiales compuestos (CFRP / GFRP). La matriz polimérica actúa como medio de transmisión de las cargas mecánicas hacia el refuerzo, distribuyendo las tensiones y protegiendo a las fibras frente a agentes ambientales agresivos, como la humedad, la corrosión química o los impactos mecánicos [3].



(a)



(b)

Figura 2-3 Polímeros como matriz en materiales compuestos a) CFRP (Extraído de [7]), b) GFRP (Extraído de [8])

Una característica especialmente relevante de la matriz polimérica es su capacidad para frenar o desviar la propagación de grietas durante un proceso de fallo, actuando como un elemento disipador de

energía que reduce la probabilidad de una fractura súbita y catastrófica [3]. A diferencia de los metales, que generalmente presentan un comportamiento mecánico isotrópico, los materiales poliméricos reforzados muestran una marcada anisotropía, de modo que sus propiedades mecánicas y térmicas dependen de la orientación del refuerzo interno [3]. Esta particularidad introduce ventajas en el diseño estructural optimizado, pero también exige un mayor control del proceso de fabricación y una evaluación rigurosa de la integridad interna del material.

En los últimos años, la evolución tecnológica ha impulsado de manera significativa el desarrollo de la fabricación aditiva o impresión 3D de polímeros, una técnica que permite construir componentes mediante la deposición sucesiva de capas a partir de un modelo digital [9]. No obstante, esta estrategia de fabricación capa "a capa" introduce una problemática técnica relevante relacionada con la aparición de defectos internos que, en la mayoría de los casos, no son detectables mediante inspección visual externa [3], [10]. En la Figura 2-4 se muestra defectos internos en polímeros fabricados por impresión 3D. Entre los defectos más frecuentes se encuentran la porosidad asociada a aire atrapado, los vacíos debidos a una extrusión insuficiente de material y, de forma especialmente crítica para la integridad estructural, la delaminación o falta de adhesión adecuada entre capas consecutivas [3], [9].

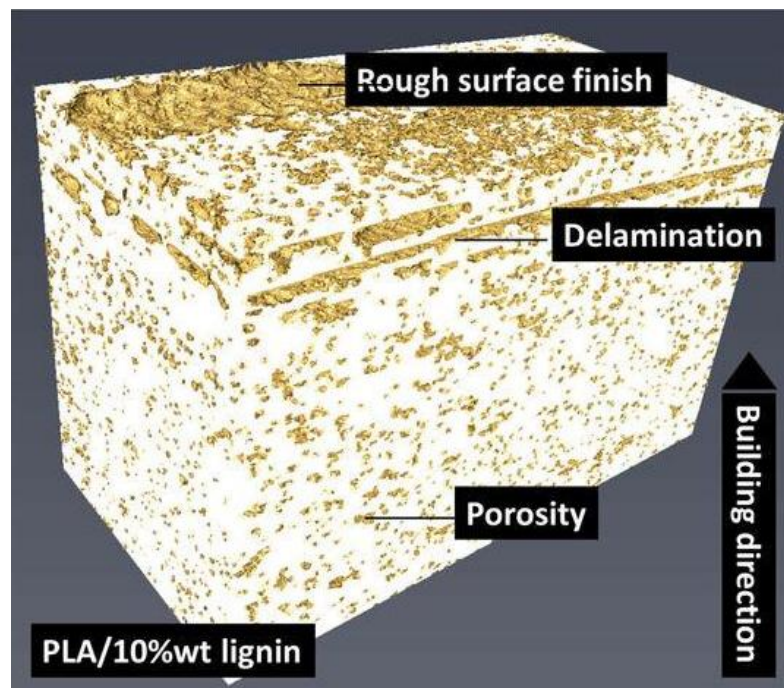


Figura 2-4 Defectos internos en polímeros fabricados por impresión 3D (Extraído de [11])

Estas discontinuidades internas actúan como concentradores de tensiones que comprometen la capacidad portante del componente, reducen de manera significativa su resistencia a fatiga y modifican el comportamiento térmico del material al alterar su difusividad [3], [10]. En condiciones de servicio reales, la presencia de estos defectos puede provocar fallos prematuros y pérdidas de funcionalidad, especialmente en aplicaciones estructurales exigentes. Por este motivo, surge la necesidad imperativa de emplear técnicas de Ensayo No Destructivo (END) capaces de evaluar la calidad interna de los componentes poliméricos sin comprometer su uso. En este contexto, la termografía infrarroja activa se presenta como una herramienta de especial interés, al permitir inducir un contraste térmico controlado que revela la presencia de defectos subsuperficiales sin alterar la integridad ni la funcionalidad del componente inspeccionado [2], [10].

2.1.2 Importancia y aplicaciones de los polímeros en entornos navales y de Defensa

La integración de materiales poliméricos y de materiales compuestos de matriz polimérica en la ingeniería naval y de Defensa ha supuesto un cambio significativo en los criterios de diseño estructural y en la operatividad de los sistemas modernos. Históricamente, la transición progresiva desde materiales tradicionales como la madera, el acero o incluso el aluminio hacia polímeros reforzados con fibras de vidrio (GFRP) y de carbono (CFRP) ha estado motivada por la necesidad de reducir el peso total de las plataformas navales y mejorar su eficiencia energética y maniobrabilidad [2], [3]. En la Figura 2-5 se muestra el uso de materiales compuestos en estructuras navales. Esta reducción de masa permite disminuir el consumo de combustible, aumentar la autonomía operativa y mejorar el comportamiento dinámico del buque, aspectos críticos en aplicaciones militares y de patrulla prolongada.

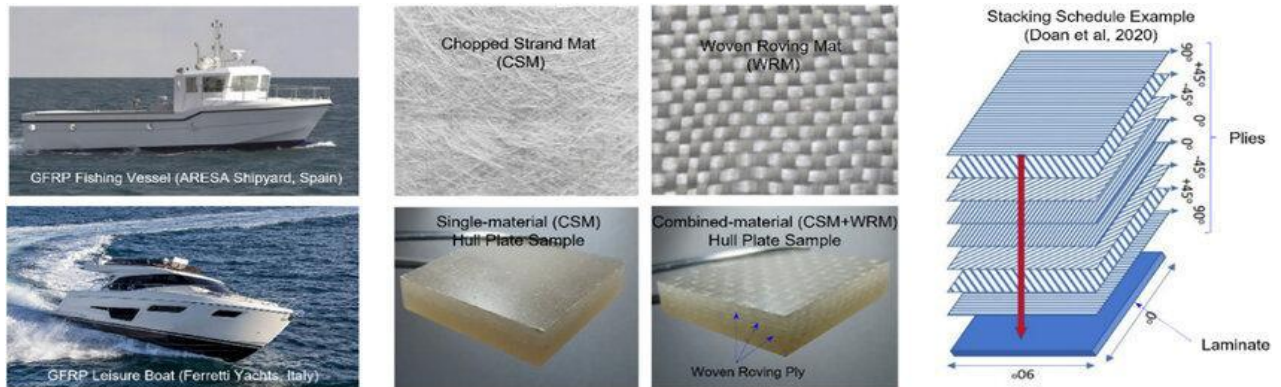


Figura 2-5 Uso de materiales compuestos en estructuras navales (Extraído de [12])

Uno de los principales desafíos del entorno marino es la degradación de los materiales por corrosión galvánica y ambiental, fenómeno que afecta de forma severa a las estructuras metálicas sometidas a atmósferas salinas y a inmersión prolongada. En este contexto, las matrices poliméricas presentan una ventaja fundamental, ya que son intrínsecamente resistentes a la corrosión, lo que se traduce en una mayor durabilidad de los componentes y en una reducción significativa de los costes de mantenimiento y de los periodos de inactividad en dique seco [13]. Además, el empleo de materiales poliméricos permite la fabricación de cascos y superestructuras no magnéticas, una característica de vital importancia para los buques dedicados a medidas contra minas (*Mine Countermeasure Vessels*, MCMV). En este tipo de plataformas, la ausencia de firma magnética reduce de manera drástica el riesgo de activación de minas sensibles al campo magnético generado por cascos de acero [3], [14]. En la Tabla 2-1 se muestra el resumen de las diferencias entre la construcción en acero y la construcción con materiales compuestos. De este modo, la adopción de polímeros no responde únicamente a criterios de optimización de peso con reducciones típicas de entre el 30% y el 50% respecto a estructuras equivalentes en acero o aluminio, sino también a requisitos estratégicos directamente relacionados con la supervivencia y la seguridad en combate [9], [13].

Propiedad	Construcción en acero	Construcción con materiales compuestos
Peso	Alto	Permite una reducción significativa del peso estructural
Corrosión	Se oxida en ambientes marinos, lo que conlleva un alto coste de mantenimiento	Muy duradera en ambientes marinos, requiere poco mantenimiento
Combustibilidad	No combustible; no contribuye al fuego ni genera humos tóxicos	Combustible, la superficie debe protegerse en zonas con riesgo de incendio
Conductividad térmica	Alta, debe aislarse para evitar la propagación del fuego y controlar la firma infrarroja	Baja, el aislamiento inherente es más que suficiente
Conductividad eléctrica	Alta, proporciona de forma inherente apantallamiento electromagnético	Baja, es necesario incorporar capas conductoras si se requiere apantallamiento electromagnético

Tabla 2-1 Resumen de las diferencias entre la construcción en acero y la construcción con materiales compuestos (Extraído de [15])

En el ámbito de la tecnología furtiva o *stealth*, los materiales compuestos de matriz polimérica desempeñan un papel clave debido a su baja firma radar y a su capacidad para interactuar de forma controlada con las ondas electromagnéticas. A diferencia de las superficies metálicas, que reflejan la radiación de radar de manera intensa y predecible, las estructuras fabricadas con polímeros reforzados, combinadas con geometrías complejas y configuraciones tipo sándwich frecuentemente con núcleos de espuma de PVC o estructuras de nido de abeja, permiten reducir de forma significativa la sección transversal de radar (*Radar Cross Section*, RCS) de los buques de guerra modernos. Ejemplos representativos de esta filosofía de diseño pueden observarse en plataformas como las corbetas de la clase Visby o las fragatas de la clase La Fayette [9], [10]. En la Figura 2-6 se muestra el diseño futuro (*stealth*) y reducción de firma radar. Esta capacidad de reducción de firma se extiende también al ámbito térmico, donde la baja conductividad térmica de los polímeros contribuye a una disipación más homogénea del calor generado por la maquinaria interna, dificultando su detección mediante sistemas de vigilancia infrarroja [16].



Figura 2-6 Corbetas de la clase Visby (Extraído de [17])

Desde el punto de vista estructural, el uso de polímeros y compuestos se ha generalizado en superestructuras, mástiles integrados, domos de sonar y carenados, donde se requiere simultáneamente una elevada rigidez específica, resistencia ambiental y, en algunos casos, transparencia acústica o electromagnética [3], [6]. En la Figura 2-7 se muestra aplicaciones estructurales específicas: mástiles. La naturaleza anisotrópica de estos materiales ofrece al ingeniero mecánico una herramienta de diseño avanzada, ya que permite orientar las fibras de refuerzo en función de las trayectorias principales de carga. Esta capacidad permite optimizar componentes críticos, como cubiertas de hangares, mástiles o sistemas de lanzamiento, alcanzando elevados niveles de resistencia mecánica sin incrementar innecesariamente la masa estructural del buque [13], [18].



Figura 2-7 La corbeta de la clase Pohjanmaa de la Marina finlandesa, actualmente en construcción, están equipadas con el Mástil Integrado Liger de Saab (SLIM) (Extraído de [19])

Más allá de las estructuras principales, los materiales poliméricos han extendido su aplicación a sistemas auxiliares y de propulsión, donde el comportamiento dinámico y vibracional resulta determinante. El desarrollo de ejes de transmisión y hélices fabricados en materiales compuestos ha demostrado reducciones significativas del ruido y de las vibraciones estructurales del orden del 20% respecto a soluciones tradicionales en acero, lo que mejora tanto el confort de la tripulación como la discreción acústica frente a sistemas de detección submarina [18], [20]. En la Figura 2-8 se muestra la reducción de vibraciones y aplicaciones en propulsión. En el interior de las plataformas navales, la sustitución de tuberías metálicas de cuproníquel por conducciones de polímeros reforzados ha permitido resolver problemas recurrentes de erosión y corrosión en sistemas de refrigeración que emplean agua de mar [2]. De forma análoga, en aplicaciones de Defensa terrestre y aeroespacial, el empleo de fibras de aramida, como el Kevlar, integradas en matrices termoestables proporciona soluciones de protección balística ligeras y altamente resistentes para vehículos y personal [21].



Figura 2-8 Nueva nota de orientación para hélices fabricadas con materiales compuestos (Extraído de [22])

No obstante, la criticidad de estas aplicaciones exige la implementación de protocolos de inspección rigurosos, ya que defectos internos como delaminaciones inducidas por impacto, porosidad o degradación progresiva de la matriz pueden comprometer de forma severa la integridad estructural y el cumplimiento de la misión [23]. A diferencia de los metales, los materiales poliméricos y compuestos no suelen manifestar señales visibles de daño, como grietas superficiales evidentes, lo que dificulta la detección temprana de fallos. Por este motivo, el empleo de técnicas de Ensayo No Destructivo, y en particular de la termografía infrarroja activa, resulta indispensable para garantizar la seguridad operativa y la fiabilidad de los sistemas en entornos navales y de Defensa de alta responsabilidad [24], [25].

2.1.3 Defectos internos en materiales poliméricos y su origen

Los defectos internos en los materiales poliméricos se definen, desde un punto de vista técnico, como discontinuidades u heterogeneidades presentes en la estructura molecular o en la morfología macroscópica del material que rompen su continuidad física y afectan directamente a su capacidad de servicio y a su fiabilidad operativa [2]. En el ámbito de la ingeniería mecánica, estas imperfecciones se clasifican habitualmente en patologías volumétricas, entre las que destacan la porosidad asociada a pequeños vacíos gaseosos ocluidos, las delaminaciones, que implican la separación de capas adyacentes, y las grietas internas, capaces de propagarse desde el núcleo hacia la superficie o viceversa [3]. La relevancia de estos defectos radica en que actúan como concentradores de tensiones, es decir, regiones donde los esfuerzos locales superan ampliamente el esfuerzo nominal aplicado, provocando una reducción significativa del límite de fatiga y de la resistencia a la fractura del componente [10]. En la Figura 2-9 se muestra la tipología general de defectos internos en polímeros.

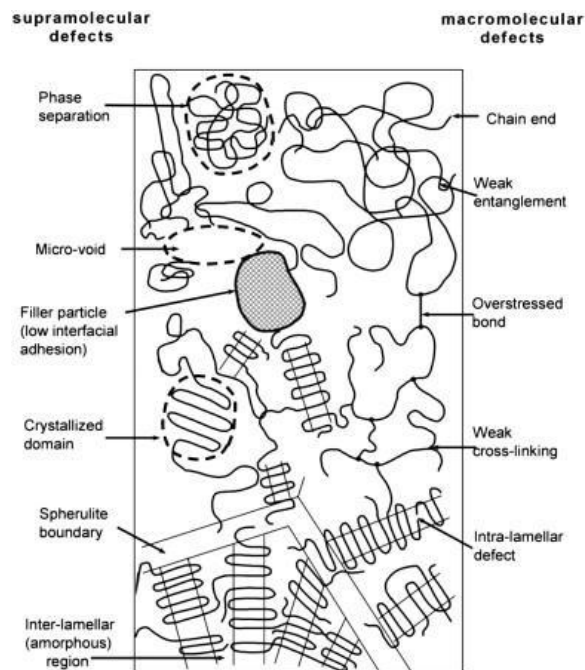


Figura 2-9 Defectos internos en polímeros (Extraído de [26])

Estas patologías subsuperficiales resultan especialmente peligrosas porque, a diferencia de los daños externos, permanecen ocultas a la inspección visual convencional. Como consecuencia, el componente puede aparentar un estado externo adecuado mientras el daño interno evoluciona progresivamente hasta alcanzar un tamaño crítico, dando lugar a fallos inesperados y, en ocasiones, de carácter catastrófico [5]. En estructuras de alta responsabilidad, como las empleadas en aplicaciones navales y aeroespaciales, la detección temprana de vacíos internos, delaminaciones y variaciones locales de densidad se convierte en una prioridad estratégica, ya que estas plataformas están sometidas a cargas dinámicas, vibraciones y ambientes agresivos que pueden acelerar la evolución del daño [2], [27].

El origen de estas patologías está estrechamente relacionado con las etapas de procesado y fabricación de las piezas poliméricas, especialmente en tecnologías avanzadas como la fabricación aditiva y la laminación de materiales compuestos. En procesos de impresión 3D mediante deposición de filamento fundido, conocida como modelado por deposición fundida (*Fused Deposition Modeling*, FDM) donde *Fused* hace referencia al material termoplástico fundido, *Deposition* al proceso de deposición del material y *Modeling* a la construcción de la geometría tridimensional capa a capa es habitual que se produzcan defectos derivados de errores en la discretización del modelo digital. Estos modelos suelen almacenarse en formato *Standard Tessellation Language* (STL), en el cual *Standard* indica que se trata de un formato ampliamente aceptado, *Tessellation* o triangulación describe la representación de la superficie mediante una malla de triángulos, y *Language* hace referencia al conjunto de instrucciones geométricas que permiten interpretar el modelo en los sistemas de fabricación.

Durante este proceso, errores en la discretización del modelo digital o en la estrategia de deposición de capas pueden dar lugar a vacíos internos o a una adhesión deficiente entre trayectorias de extrusión consecutivas [9], [28]. Asimismo, una selección inadecuada de los parámetros de proceso como velocidades de extrusión excesivas, temperaturas de boquilla insuficientes o una gestión inadecuada del enfriamiento puede provocar inestabilidades en el flujo del material fundido, introduciendo microdefectos tanto en la superficie como en el interior del material [29]. En la Figura 2-10 se muestra defectos inducidos por fabricación aditiva (FDM).

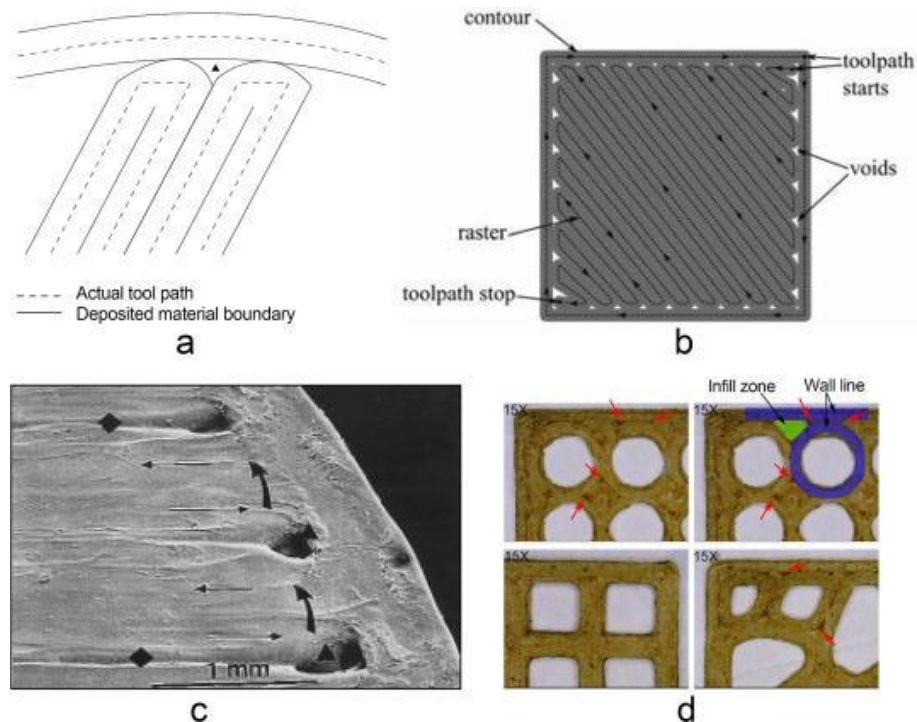


Figura 2-10 Defectos inducidos por fabricación aditiva (FDM) a) diagrama de formación de vacíos subperimetrales. La región marcada con ▲ corresponde a un vacío subperimetral formado entre el perímetro y las trayectorias de relleno (rasters), b) diagrama de una capa de la fabricación por filamento fundido (*Fused Filament Fabrication*, FFF) rectangular genérica, c) micrografía SEM de la superficie superior de una pieza FFF con vacíos subperimetrales (s), d) secciones transversales de las probetas. Las flechas rojas señalan los vacíos subperimetrales. (Extraído de [30])

En el caso de los polímeros reforzados con fibras, como los CFRP y GFRP, las delaminaciones pueden originarse durante las fases de fabricación y curado debido a la presencia de aire atrapado, a una impregnación incompleta de las fibras o a la segregación local de la matriz polimérica. Estas imperfecciones impiden una transferencia de carga eficiente entre el refuerzo y la matriz, comprometiendo de forma significativa la capacidad portante del material compuesto [6]. En la Figura 2-11 se muestra delaminaciones y defectos en CFRP/GFRP. Como consecuencia, se introduce una anisotropía no deseada que hace que el comportamiento mecánico del componente dependa no solo de la orientación de las fibras, sino también de la localización y severidad de las discontinuidades internas generadas durante el proceso de solidificación [31], [32].

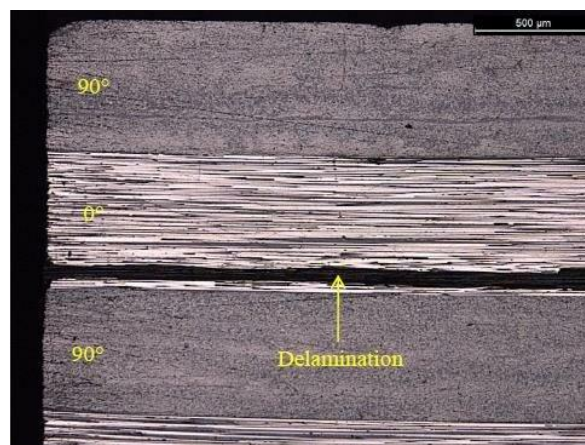


Figura 2-11 Micrografía óptica de la delaminación entre los grupos de capas a 0° y 90° en un laminado dañado. (Extraído de [33])

Además de los defectos inducidos durante la fabricación, los materiales poliméricos pueden desarrollar daños internos a lo largo de su vida operativa como resultado de factores externos. Entre ellos, los impactos de baja velocidad constituyen una de las causas más recurrentes y preocupantes en los sectores naval y de Defensa. Este tipo de solicitaciones da lugar al denominado daño por impacto apenas visible (*Barely Visible Impact Damage*, BVID), caracterizado por una apariencia externa prácticamente intacta mientras el interior del material presenta microfracturas de la matriz, delaminaciones y, en materiales reforzados, rotura o desalineación de fibras [3], [6]. En la Figura 2-12 se muestra daño por BVID.

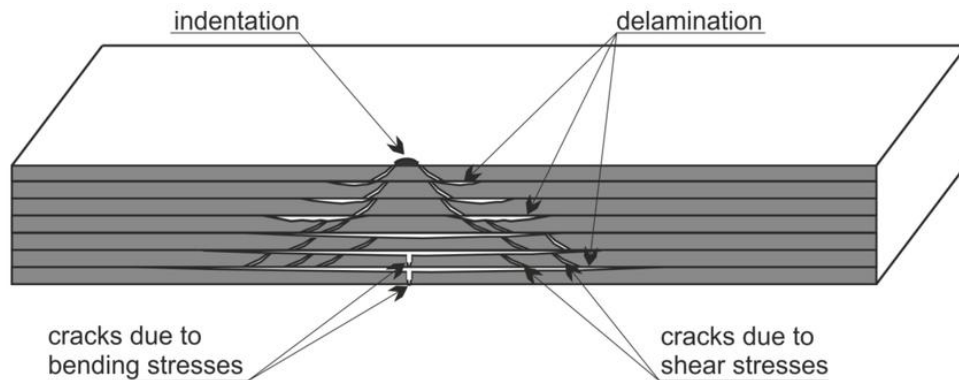


Figura 2-12 Esquema del daño por impacto apenas visible (BVID) en una estructura compuesta. (Extraído de [34])

Adicionalmente, las condiciones ambientales de servicio pueden acelerar la degradación del material. La absorción de humedad y el sobre estrés térmico favorecen procesos de envejecimiento químico de la matriz polimérica y facilitan la iniciación de grietas por fatiga [35]. La humedad absorbida puede actuar como lubricante en las interfaces fibra-matriz o provocar hinchamientos localizados que generan tensiones residuales internas, incrementando la probabilidad de delaminación y pérdida de rigidez estructural [29]. Dado que estas patologías alteran de manera local las propiedades termofísicas del material, como la conductividad y la difusividad térmica, se hace imprescindible el uso de métodos avanzados de inspección no destructiva capaces de caracterizar la extensión y profundidad del daño sin comprometer la integridad del componente [23], [24].

Por tanto, comprender la relación entre los mecanismos de generación del defecto, la energía asociada a impactos o cargas cíclicas y la evolución del daño interno resulta clave para predecir la vida útil remanente de los materiales poliméricos y compuestos en condiciones reales de servicio. Este conocimiento constituye la base para el desarrollo de estrategias de inspección, mantenimiento preventivo y evaluación estructural basadas en técnicas no destructivas avanzadas [6], [36].

La Tabla 2-2 recoge de forma resumida los principales tipos de defectos internos que pueden aparecer en materiales poliméricos, así como las causas más comunes asociadas a su generación durante los procesos de fabricación. La identificación y comprensión de estos defectos resulta fundamental para evaluar su impacto en la integridad estructural de los componentes y para justificar la necesidad de aplicar técnicas de inspección no destructiva capaces de detectarlos de forma temprana.

Defecto	Causa
Porosidad	Aire atrapado en el material
Vacíos	Falta de material durante la fabricación
Delaminación	Mala adhesión entre capas
Inclusiones	Contaminación de la materia

Tabla 2-2 Tipos de defectos internos en polímeros y causas asociadas (tomada de [21])

2.1.4 Consecuencias de los defectos internos y necesidad de inspección

La presencia de defectos internos en estructuras de ingeniería conlleva consecuencias mecánicas severas que comprometen directamente la seguridad, la fiabilidad y la estabilidad global de los sistemas. Estas patologías, ya sean porosidades, vacíos, delaminaciones o grietas internas, actúan como concentradores de tensiones donde los esfuerzos locales superan ampliamente la resistencia nominal del material, facilitando la iniciación y propagación de fallos estructurales [2], [3]. En la Figura 2-13 se muestra, concentración de tensiones y fallo estructural por defectos internos.

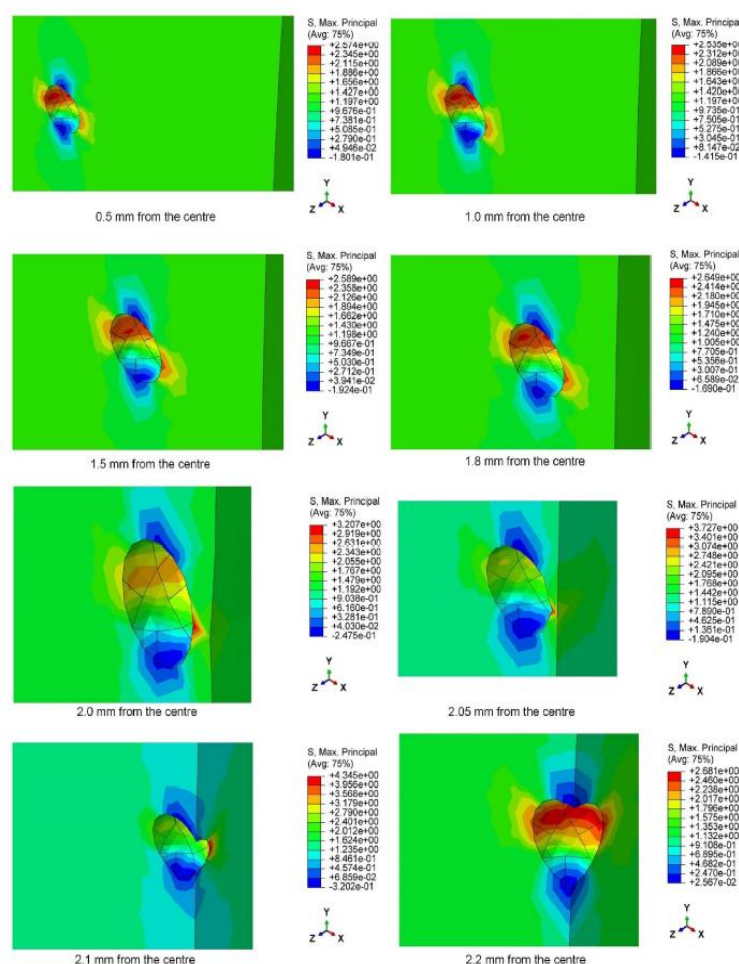


Figura 2-13 Abaqus/CAE que muestran la variación de la tensión principal máxima / concentración de tensiones para diferentes ubicaciones de un poro con un diámetro de 0,3 mm (Extraído de [37])

En componentes mecánicos sometidos a sollicitaciones cíclicas, como ejes navales, elementos ferroviarios o estructuras sometidas a vibraciones continuas, las discontinuidades subsuperficiales no detectadas pueden evolucionar rápidamente hacia fracturas frágiles o fallos por fatiga de carácter catastrófico. Este proceso reduce drásticamente la vida útil remanente (*Remaining Useful Life*, RUL) del

activo y aumenta de manera exponencial el riesgo de fallo en servicio [38], [39]. En la Figura 2-14 se muestra patrones de fractura de componentes en el fallo por fatiga.

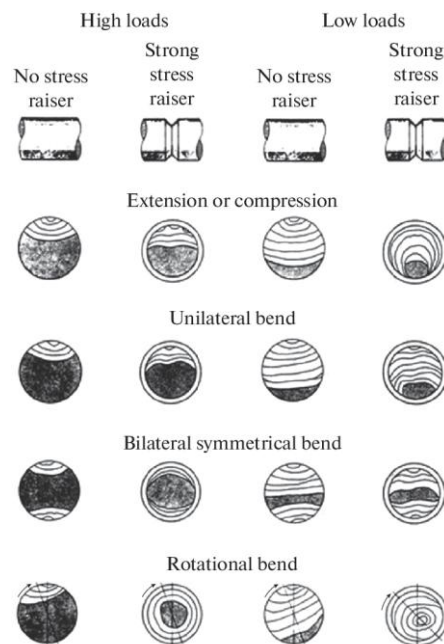


Figura 2-14 Patrones de fractura de componentes en el fallo por fatiga (Extraído de [40])

Desde una perspectiva económica y operativa, la ausencia de una detección temprana de estas patologías internas conduce a una gestión de activos puramente reactiva, caracterizada por paradas no programadas, reparaciones de emergencia y un incremento sustancial de los costes de mantenimiento. La degradación progresiva de los materiales, provoca una pérdida notable de eficiencia técnica y energética [29], [41]. En el sector de la edificación, por ejemplo, defectos como puentes térmicos, infiltraciones de aire o discontinuidades en los cerramientos pueden suponer entre el 10 % y el 50 % de la demanda energética total de una estructura, afectando además al confort de los usuarios y acelerando el deterioro de los elementos constructivos [16], [42], [43].

En entornos navales y de Defensa, las consecuencias de un fallo estructural imprevisto durante el servicio son aún más críticas. La necesidad de realizar reparaciones en dique seco implica costes extremadamente elevados y largos periodos de inactividad operativa. En los casos más severos, si el defecto interno supera el tamaño crítico de fractura antes de ser detectado, puede producirse la pérdida total del componente o incluso del sistema completo [13], [44]. En la Figura 2-15 se muestra una serie de consecuencias económicas y operativas de fallos no detectados. Asimismo, la presencia de inclusiones, vacíos o delaminaciones en elementos de alta responsabilidad, como depósitos de gas a alta presión o sistemas de propulsión, representa un riesgo de seguridad inaceptable, lo que exige la implementación de estrategias de mantenimiento predictivo basadas en la monitorización continua de la salud estructural (*Structural Health Monitoring*, SHM) [3], [45].



Figura 2-15 Un gran buque en dique seco con andamios y equipos que muestran detalles de su casco y hélice durante una sesión de reparación (Extraído de [46])

La necesidad imperativa de inspección se fundamenta en la limitada capacidad de los métodos tradicionales, como la inspección visual externa, para identificar defectos que permanecen ocultos bajo la superficie del material. Este tipo de inspección resulta lenta, subjetiva y restringida a patologías externas evidentes, dejando un margen de riesgo inasumible en sectores de alta responsabilidad técnica [16], [47]. Por este motivo, el empleo de técnicas END se convierte en un elemento esencial para evaluar la integridad interna de los componentes sin alterar su funcionalidad ni su estructura física [16], [48].

Dentro del conjunto de técnicas END, la termografía infrarroja (*InfraRed Thermography*, IRT) se ha consolidado como una de las tecnologías más prometedoras debido a su capacidad de inspección rápida, su carácter no invasivo y su posibilidad de proporcionar mapas bidimensionales de temperatura que revelan anomalías termofísicas invisibles al ojo humano [14], [16], [18]. En la Figura 2-16 se muestra la necesidad de END y termografía infrarroja. No obstante, la interpretación de los datos termográficos en bruto puede verse afectada por ruido térmico, variaciones ambientales y efectos de emisividad, lo que hace necesario el desarrollo de herramientas avanzadas de procesamiento y análisis automático.

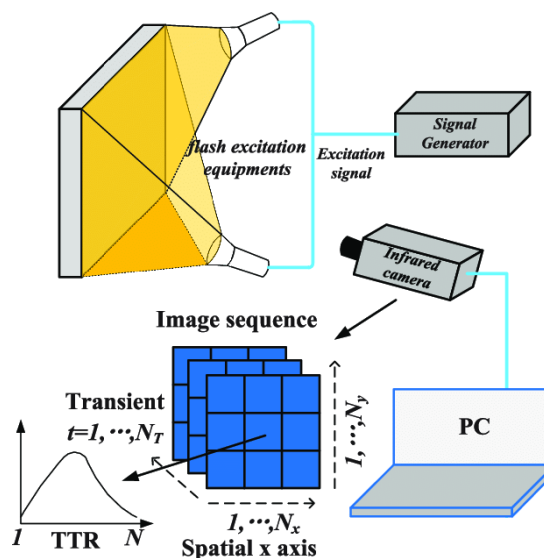


Figura 2-16 Sistema de detección de defectos mediante termografía infrarroja (Extraído de [49])

En este contexto, la integración de técnicas de inteligencia artificial y aprendizaje profundo (*Deep Learning*) ha permitido avanzar significativamente en la automatización de la detección de defectos.

Algoritmos avanzados de visión artificial, como Mask R-CNN, permiten localizar, segmentar y clasificar automáticamente patologías internas a partir de imágenes termográficas, reduciendo la dependencia de la experiencia del operador y minimizando el riesgo de errores de diagnóstico o de omisión de defectos críticos [24], [50], [51], [52]. Solo mediante la combinación de sensores avanzados, técnicas END y algoritmos de post-procesado inteligentes es posible garantizar, de forma fiable, la seguridad, la durabilidad y la operatividad de las infraestructuras modernas sometidas a condiciones de servicio exigentes [53], [54], [55].

2.2 Técnicas de Ensayo No Destructivo aplicadas a la detección de defectos

2.2.1 Concepto y relevancia de los Ensayos No Destructivos

Los Ensayos No Destructivos (END), conocidos internacionalmente como *Non-Destructive Testing* (NDT), constituyen una disciplina técnica fundamental orientada a la inspección, ensayo y evaluación de materiales, componentes y estructuras con el objetivo de detectar discontinuidades, defectos internos o variaciones en sus propiedades físicas sin comprometer su integridad estructural ni su capacidad de servicio futura [2], [3]. En la Tabla 2-3 se muestra el concepto general de END. El principio esencial que define a estas técnicas es su carácter no invasivo, lo que permite que, una vez finalizada la inspección, el componente analizado pueda reincorporarse a su función operativa sin haber sufrido daños permanentes o alteraciones en su comportamiento mecánico [10], [56], [57].

VISUALES	SUPERFICIALES	VOLUMÉTRICOS
Observación directa a simple vista	Partículas magnéticas (MT)	Ultrasonidos (UT)
Inspección visual asistida:	Líquidos penetrantes (PT)	Radiografía industrial (RT)
- Espejos	Corrientes inducidas (ET)	- Rayos X
- Lentes de aumento	Otros métodos basados en:	- Isótopos radiactivos
- Microscopios	- Campos eléctricos	- Aceleradores
- Boroscopios	- Campos magnéticos	Ensayos de fugas
- Endoscopios		Emisión acústica
Inspección visual indirecta:		Otros métodos avanzados
- Videoscopios		
- Cámaras remotas		
Otros métodos ópticos		

Tabla 2-3 Clasificación de los distintos métodos (Extraído de [58])

A diferencia de los ensayos mecánicos destructivos tradicionales en los que la pieza debe ser sometida a cargas hasta su rotura para caracterizar sus propiedades, los END ofrecen la posibilidad de evaluar la calidad y fiabilidad de cada componente individual sin necesidad de inutilizarlo. Esta capacidad resulta especialmente relevante en sectores industriales con elevados requisitos de seguridad y certificación, como la industria automotriz, aeronáutica, naval y de Defensa, donde la integridad estructural de los componentes es un factor crítico [10]. En la Tabla 2-4 se muestra END frente a ensayos destructivos. Asimismo, la aplicación sistemática de estas técnicas contribuye a reducir la dependencia de métodos de inspección visual subjetivos y poco reproducibles, proporcionando datos cuantitativos, objetivos y contrastables que permiten una evaluación técnica rigurosa del estado del material [2], [44], [59].

Aspecto	Ensayos Destructivos (ED)	Ensayos No Destructivos (END)
Daño en la pieza	La muestra se daña o destruye.	No se produce daño permanente.
Objetivo	Medir propiedades mecánicas del material.	Detectar defectos e integridad estructural.
Uso posterior	No puede reutilizarse.	Puede seguir en servicio si es apta.
Aplicación	Diseño y validación en laboratorio.	Inspección y mantenimiento en servicio.
Ejemplos	Tracción, compresión, dureza, impacto.	Ultrasonidos, radiografía, penetrantes, termografía.

Tabla 2-4 END frente a ensayos destructivos (Extraído de [60])

Desde un enfoque funcional y estratégico, la relevancia de los END en la ingeniería mecánica reside en su capacidad para operar de forma remota y, en muchos casos, sin contacto directo con el componente inspeccionado. Esta característica permite realizar diagnósticos precisos en elementos de difícil acceso o sometidos a condiciones de operación extremas, como altas temperaturas, elevados niveles de vibración o presencia de campos eléctricos intensos, garantizando en todo momento la seguridad del operador [2], [10], [59]. En la Figura 2-17 se muestra una aplicación de END en sectores críticos (naval, aeroespacial, defensa). Además de la detección de defectos superficiales, los END resultan esenciales para la caracterización de parámetros críticos del comportamiento de los materiales, como la resistencia a la fractura, la ductilidad, la resiliencia al impacto o la evolución de grietas internas por fatiga [31], [59], [61].



Figura 2-17 Ensayo por ultrasonidos (UT) (Extraído de [62])

Durante las fases de fabricación y ensamblaje, los Ensayos No Destructivos desempeñan un papel clave en el control de calidad, ya que permiten verificar la correcta ejecución de procesos de unión, conformado o laminación. La detección temprana de defectos en etapas iniciales del proceso productivo posibilita el descarte de componentes defectuosos antes de su integración en sistemas más complejos, reduciendo de manera significativa los costes de producción y retrabajo [3], [10]. Por otro lado, en la fase de servicio, los END constituyen la base de los sistemas de Monitorización de la Salud Estructural (*Structural Health Monitoring*, SHM), facilitando la transición desde estrategias de mantenimiento correctivo hacia modelos de mantenimiento predictivo que anticipan el fallo antes de que este ocurra [55], [63], [64].

La variedad de técnicas END disponibles es amplia, existiendo actualmente hasta dieciséis métodos reconocidos por organizaciones internacionales como la Sociedad Americana de Ensayos No

Destructivos (ASNT), los cuales se clasifican en función del tipo de señal física utilizada o del principio de funcionamiento del equipo empleado [3], [65]. Entre las modalidades más relevantes en la práctica ingenieril se encuentran las técnicas visuales, radiográficas, ultrasónicas, electromagnéticas como las corrientes inducidas y las técnicas térmicas o infrarrojas [10], [55]. Cada una de estas metodologías presenta capacidades específicas que deben seleccionarse cuidadosamente en función del material inspeccionado (metales, polímeros o materiales compuestos), el espesor de la pieza y el tipo de defecto subsuperficial que se desea caracterizar, como porosidades, delaminaciones o inclusiones [6], [10]. En la Tabla 2-5 se muestra la clasificación de técnicas END.

Categoría	Principio físico	Técnicas principales
Inspección visual	Observación directa o asistida	Inspección visual (VT), boroscopia, videoscopia
Métodos acústicos	Propagación de ondas mecánicas	Ultrasonidos (UT), emisión acústica (AE), técnicas no lineales
Métodos ópticos / térmicos	Interacción luz-materia / radiación térmica	Termografía infrarroja (IRT), shearografía, correlación digital de imágenes (DIC), terahercios
Técnicas radiográficas (imagen)	Radiación ionizante	Rayos X, radiografía gamma, radiografía con neutrones
Métodos electromagnéticos	Interacción con campos eléctricos o magnéticos	Corrientes inducidas (ET), partículas magnéticas (MT), fuga de flujo magnético (MFL), Ensayo por campo remoto (RFT).
Métodos superficiales químicos	Capilaridad y contraste	Líquidos penetrantes (PT), tintes penetrantes

Tabla 2-5 Categorías de las técnicas de ensayo y evaluación no destructiva (Extraído de [66])

Mientras que los ensayos por ultrasonidos destacan por su elevada precisión en la detección de discontinuidades internas mediante el análisis de la propagación de ondas elásticas, la termografía infrarroja (IRT) se ha consolidado como una herramienta especialmente eficaz para el escaneo rápido de grandes superficies. Esta técnica permite visualizar anomalías termofísicas en tiempo real de forma intuitiva, lo que resulta particularmente ventajoso en inspecciones de componentes de geometría compleja o de gran tamaño [16], [35], [67]. En la Figura 2-18 se muestra la termografía infrarroja frente a otras técnicas END. En última instancia, la integración sistemática de distintas técnicas END, apoyada por algoritmos avanzados de procesamiento de datos, garantiza que la integridad mecánica de los componentes cumpla con los exigentes estándares de seguridad, fiabilidad y sostenibilidad que demanda la ingeniería moderna [16], [24], [55].

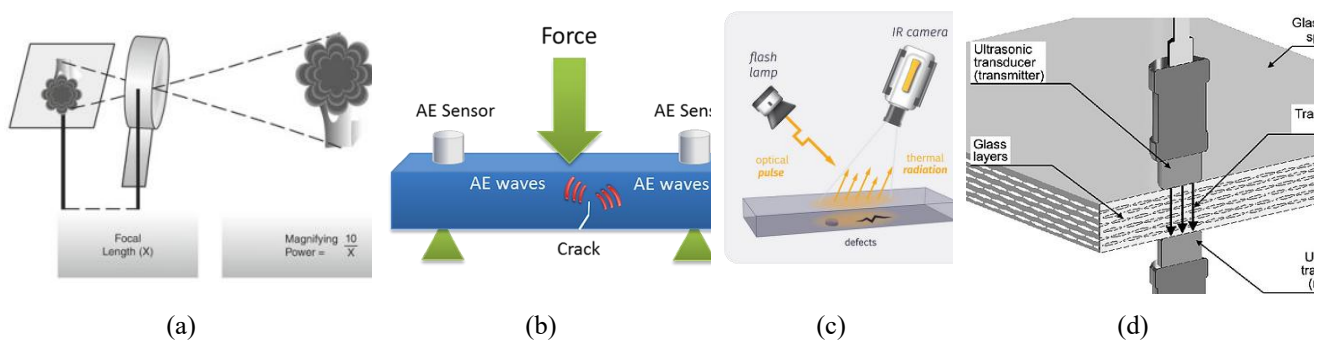


Figura 2-18 Termografía infrarroja frente a otras técnicas END a) equipos y accesorios para inspección visual b) ensayo por emisión acústica c) ensayo mediante termografía infrarroja d) ensayo por ultrasonidos (Extraído de [68])

2.2.2 Técnicas END empleadas en materiales poliméricos

La selección de las técnicas de Ensayo No Destructivo aplicables a materiales poliméricos y a materiales compuestos de matriz polimérica está fuertemente condicionada por las propiedades termofísicas, mecánicas y acústicas inherentes a estas matrices orgánicas [16], [69]. A diferencia de los metales, los polímeros presentan generalmente una menor conductividad térmica, una elevada atenuación acústica y un comportamiento anisótropo cuando están reforzados con fibras, lo que obliga a adaptar los métodos de inspección a estas particularidades. Entre las técnicas más empleadas destaca la inspección por ultrasonidos (*Ultrasonic Testing*, UT), basada en la detección de discontinuidades internas mediante variaciones en la propagación de ondas elásticas de alta frecuencia a través del material [32]. En la Figura 2-19 se muestra un ejemplo.

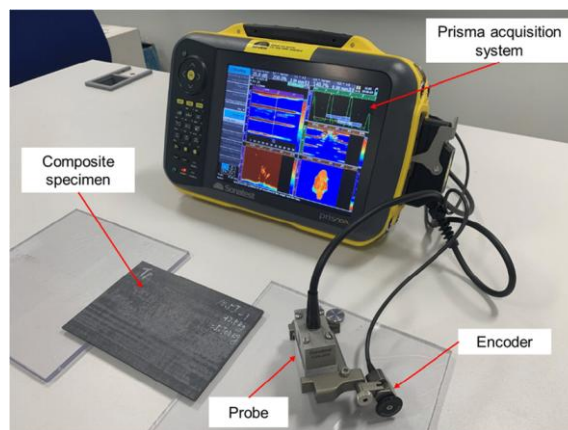


Figura 2-19 Equipo de C-scan por ultrasonidos para la evaluación de daños por impacto (Extraído de [70])

En el análisis de polímeros reforzados con fibras, como los CFRP y GFRP, el método C-Scan resulta especialmente valioso, ya que permite generar mapas bidimensionales de la pérdida de transmisión ultrasónica a lo largo del espesor del laminado. Este tipo de representación facilita la identificación de defectos como delaminaciones, vacíos volumétricos y daños por BVID, proporcionando una visión global de la extensión del daño interno [3], [32], [55]. No obstante, desde el punto de vista de la ingeniería mecánica, uno de los principales retos de la inspección ultrasónica en polímeros es el elevado desajuste de impedancia acústica entre el aire y los sólidos, lo que hace necesario el uso de técnicas por inmersión en medios acuosos o sistemas de acoplamiento directo para asegurar una transmisión eficiente del pulso ultrasónico [71], [72].

De manera complementaria, la Emisión Acústica (*Acoustic Emission*, AE) se emplea para detectar la liberación de energía elástica asociada a procesos de daño activo, como la fractura de la matriz polimérica o la rotura de fibras de refuerzo. Esta técnica permite una monitorización continua en tiempo real de la salud estructural del componente bajo condiciones de carga, lo que resulta especialmente útil en aplicaciones de monitorización de la salud estructural. Mediante el análisis de los eventos acústicos generados durante el servicio, es posible identificar fallos incipientes antes de que se manifiesten mediante inspección visual externa o técnicas convencionales [55], [73]. En la Figura 2-20 se muestra una configuración de AE en monitorización estructural.

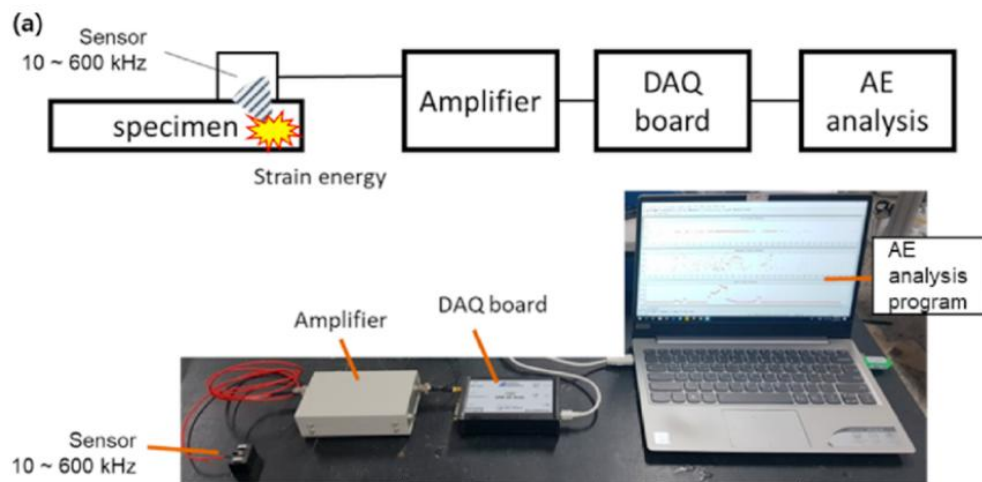


Figura 2-20 Configuración del ensayo de emisión acústica (AE) (Extraído de [74])

Otra alternativa tecnológica relevante es el empleo de corrientes inducidas (*Eddy Current Testing*, ECT), cuya aplicabilidad en materiales poliméricos depende estrictamente de la conductividad eléctrica del refuerzo. En los compuestos de carbono-epoxi (CFRP), la naturaleza conductora de las fibras de carbono permite detectar fracturas de fibras, delaminaciones subsuperficiales o variaciones locales de conductividad. Sin embargo, en los polímeros reforzados con fibras de vidrio (GFRP), que actúan como aislantes eléctricos, el uso convencional de esta técnica resulta ineficaz, salvo que se incorporen elementos conductores durante el proceso de fabricación [75], [76], [77]. En aplicaciones aeronáuticas y de Defensa, el uso de frecuencias bajas, generalmente inferiores a 250 kHz, permite inspeccionar estructuras tipo sándwich o paneles de gran espesor, mejorando la capacidad de detección en profundidad respecto al empleo de frecuencias elevadas [76], [78]. En la Figura 2-21 se muestra una aplicación del método de corrientes inducidas (ECT) en materiales compuestos conductores (CFRP) para la detección de discontinuidades subsuperficiales.

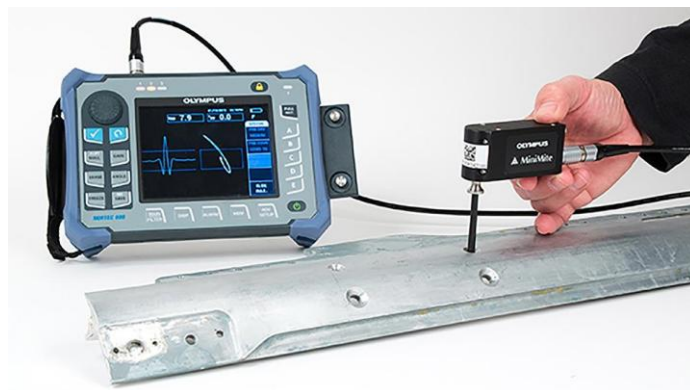


Figura 2-21 Inspección por corrientes inducidas de estructuras aeronáuticas en servicio (Extraído de [79])

La shearografía láser (*Laser Shearography*, LS) ha ganado una notable relevancia en la inspección de materiales poliméricos y compuestos debido a su capacidad para evaluar grandes superficies en tiempo real mediante técnicas interferométricas basadas en patrones de *speckle* [80]. El principio de funcionamiento de esta técnica consiste en aplicar una carga externa controlada como vacío, excitación térmica o vibración mecánica para inducir microdeformaciones en la superficie del componente. La comparación de imágenes antes y después de la carga permite visualizar patrones de franjas que delatan la presencia de delaminaciones, separaciones de núcleos o defectos adherenciales que permanecen invisibles al ojo humano [81], [82]. No obstante, una de las principales limitaciones de la shearografía es la reducción de sensibilidad a medida que aumenta la profundidad del defecto, lo que hace recomendable su combinación con otras técnicas de inspección subsuperficial [83], [84]. En la Figura 2-

22 se muestra una inspección mediante shearografía láser mostrando patrones interferométricos asociados a delaminaciones internas en materiales compuestos.

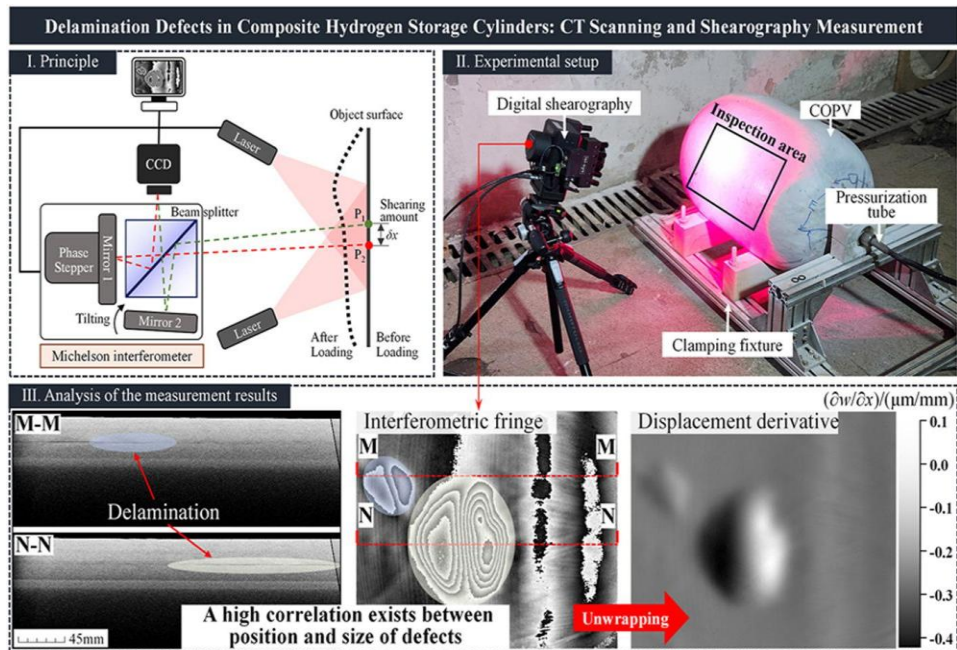


Figura 2-22 Defectos de delaminación en cilindros de almacenamiento de hidrógeno compuestos: escaneo por tomografía computarizada (TC) y medición mediante shearografía (Extraído de [85])

Para obtener una caracterización volumétrica más completa, se recurre habitualmente a la radiografía industrial (*Industrial Radiography*, RT) y a la termografía infrarroja (IRT). La radiografía mediante rayos X emplea fotones de alta energía para generar imágenes de sombra, siendo especialmente eficaz en la detección de inclusiones sólidas, vacíos volumétricos y desalineaciones de fibras. Sin embargo, la estimación precisa de la profundidad del defecto resulta limitada sin el uso de técnicas avanzadas como la tomografía computarizada (*X-ray Computed Tomography*, XCT), que permite reconstrucciones tridimensionales de alta resolución [86], [87]. En la Figura 2-23 se muestra una radiografía industrial y XCT.

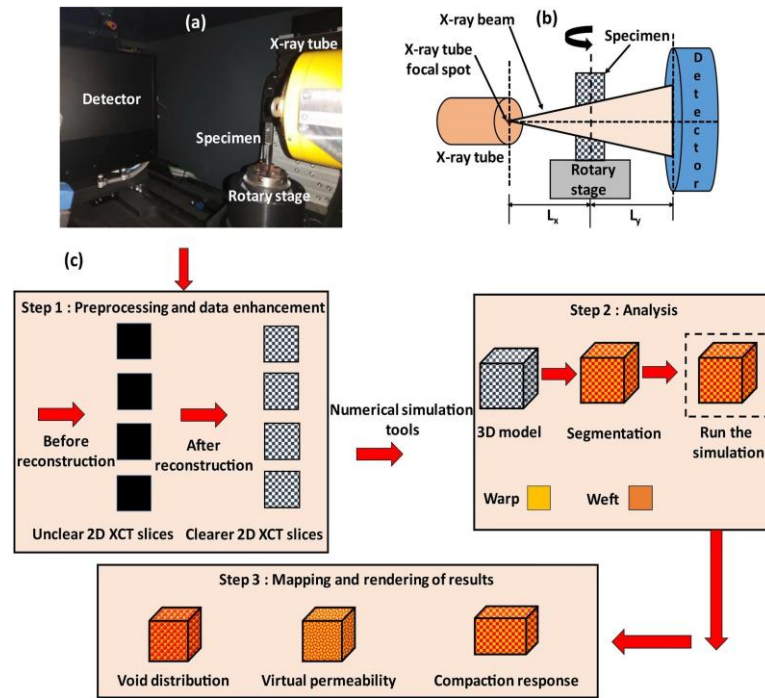


Figura 2-23 Radiografía industrial y XCT a) configuración típica de un equipo XCT de laboratorio, b) representación esquemática del sistema y c) procedimiento paso a paso para el análisis de datos XCT. (Extraído de [88])

En contraste, la termografía infrarroja se posiciona como una de las técnicas más eficientes para el escaneo rápido de grandes superficies de materiales poliméricos y compuestos, gracias a su carácter no invasivo y a su elevada velocidad de inspección [89]. Dado que estos materiales no suelen presentar contrastes térmicos naturales significativos, se emplea la termografía infrarroja activa, en la que se aplica una excitación térmica artificial mediante flashes o lámparas halógenas para inducir un flujo de calor que permita revelar defectos internos a partir de anomalías en la respuesta térmica superficial [18], [90], [91]. En la Figura 2-24 se muestra un ejemplo de termografía infrarroja activa [89], [92], [93].

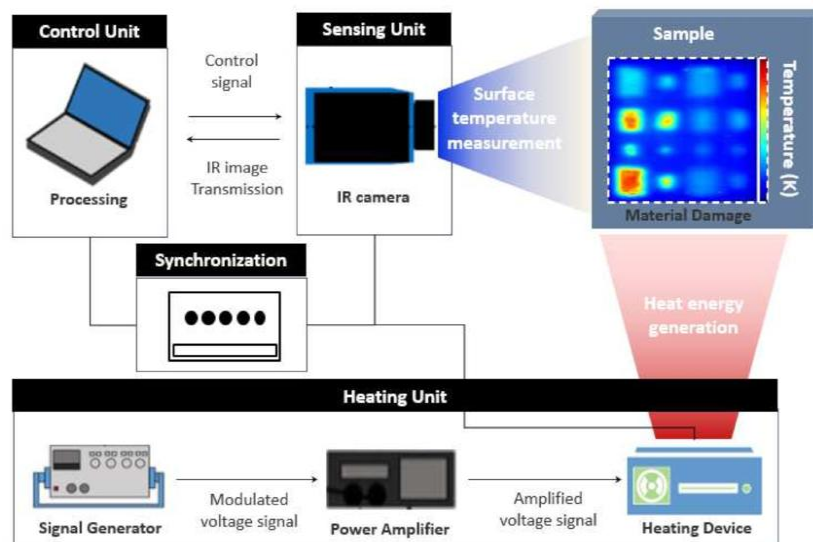


Figura 2-24 Esquema conceptual de la termografía infrarroja activa: aplicación de una excitación térmica externa y detección de anomalías superficiales asociadas a defectos internos (Extraído de [94])

Finalmente, la integración de sensores termográficos con algoritmos avanzados de aprendizaje profundo (*Deep Learning*) representa el estado del arte en la automatización de la detección de defectos. Esta convergencia entre END e inteligencia artificial constituye una herramienta clave para garantizar

la seguridad operativa y la fiabilidad estructural en aplicaciones navales y militares de alta responsabilidad [16], [24], [35].

La Tabla 2-6 presenta una comparación general entre distintas técnicas END y su aplicabilidad a materiales poliméricos, destacando la elevada adecuación de la termografía infrarroja para este tipo de aplicaciones

Técnica	Contacto	Profundidad	Adecuada polímeros
Ultrasonidos	Sí	Alta	Media
Radiografía	No	Alta	Media
Termografía IR	No	Media	Alta

Tabla 2-6 Comparación general entre distintas técnicas END y su aplicabilidad a materiales poliméricos.
(Extraído de [48])

2.2.3 Ventajas de la termografía infrarroja como técnica END

La termografía infrarroja (IRT) se ha consolidado como una de las técnicas de Ensayo No Destructivo más eficaces y versátiles para la inspección de componentes mecánicos y estructuras complejas, especialmente en materiales poliméricos y materiales compuestos de baja conductividad térmica. Su creciente implantación en el ámbito industrial, naval y de Defensa se debe a una combinación de características técnicas que la diferencian de otras metodologías END tradicionales y que la convierten en una herramienta especialmente adecuada para la detección temprana de defectos internos [2], [48]. En la Figura 2-25 se muestra su naturaleza sin contacto y remota. Estas ventajas se relacionan tanto con su principio físico de funcionamiento como con su elevada capacidad operativa y su integración con sistemas modernos de análisis de datos.



Figura 2-25 Naturaleza sin contacto y remota (Extraído de [95])

Las principales ventajas de la termografía infrarroja como técnica de Ensayo No Destructivo pueden resumirse en los siguientes puntos:

- Naturaleza sin contacto y remota: La IRT permite realizar inspecciones a distancia, evitando el acceso directo a zonas peligrosas o de difícil alcance sin alterar el estado físico del material inspeccionado. Esta característica resulta especialmente relevante en componentes sometidos a altas temperaturas, atmósferas agresivas o condiciones operativas complejas [2], [48].

- Inspección no invasiva y no destructiva: La técnica no introduce daños mecánicos ni químicos en la pieza analizada, lo que posibilita que el componente sea devuelto inmediatamente a servicio tras la inspección, manteniendo intacta su funcionalidad y capacidad estructural [18], [48].
- Resultados bidimensionales (2D): La obtención de mapas térmicos o termogramas permite una visualización global y espacial de la distribución de temperaturas sobre grandes superficies, facilitando la comparación directa entre zonas sanas y áreas potencialmente defectuosas [2], [96].
- Alta velocidad de inspección y análisis en tiempo real: Las cámaras termográficas modernas son capaces de adquirir imágenes a altas frecuencias, lo que permite el escaneo rápido de grandes superficies y la detección de fenómenos térmicos transitorios asociados a defectos subsuperficiales [3], [18].
- Ausencia de radiaciones peligrosas: A diferencia de técnicas como la radiografía industrial, la termografía infrarroja no emplea radiación ionizante, lo que incrementa la seguridad del operador y permite su uso en entornos de trabajo convencionales sin necesidad de medidas de protección especiales [2], [3].
- Versatilidad frente a distintos materiales: La IRT puede aplicarse eficazmente a la inspección de polímeros, metales, cerámicas y materiales compuestos avanzados, como los CFRP, adaptándose a una amplia variedad de configuraciones estructurales y aplicaciones industriales [10], [97].
- Capacidad de detección subsuperficial: Mediante el empleo de enfoques activos, la termografía infrarroja permite revelar defectos ocultos bajo la superficie, tales como delaminaciones, porosidades y vacíos internos, que no pueden ser detectados mediante inspección visual convencional [97].
- Reducción de costes operativos y mejora del mantenimiento: La detección temprana de defectos internos favorece la implantación de estrategias de mantenimiento preventivo y predictivo, reduciendo el mantenimiento reactivo, prolongando la vida útil de los activos y minimizando los costes asociados a fallos imprevistos [10], [16].

En conjunto, estas ventajas explican el creciente interés por la termografía infrarroja como técnica END en aplicaciones de alta responsabilidad, donde la fiabilidad, la rapidez de diagnóstico y la seguridad operativa son requisitos esenciales para la ingeniería mecánica moderna [24], [96].

2.3 Termografía infrarroja activa y procesamiento de datos térmicos

2.3.1 Fundamentos físicos de la termografía infrarroja

La termografía infrarroja se fundamenta en la capacidad de detectar y cuantificar la radiación electromagnética emitida por los cuerpos como consecuencia directa de su temperatura. De acuerdo con los principios básicos de la transferencia de calor y de la física de la radiación, todo objeto real su temperatura es superior al cero absoluto, presentando una estructura atómica en constante agitación, lo que da lugar a una emisión espontánea de energía en forma de radiación térmica [3], [10], [18]. En los rangos de temperatura habituales en la ingeniería mecánica y en aplicaciones industriales, la máxima intensidad de dicha radiación se localiza predominantemente en la banda del espectro infrarrojo, comprendida aproximadamente entre 740 nm y 1 mm, lo que permite su captación mediante sensores termográficos especializados capaces de transformar la energía radiante en señales eléctricas procesables [10], [24].

La energía asociada a esta radiación electromagnética puede describirse mediante la relación fundamental entre la energía del fotón, su frecuencia y su longitud de onda, expresada matemáticamente en la Ecuación (1)

$$e = h \cdot v = \frac{(h \times c)}{\lambda} \quad \text{Ecuación (1)}$$

En esta expresión, e representa la energía del fotón (J), h es la constante de Planck ($6,626 \times 10^{-34} \text{ J} \cdot \text{s}$), ν la frecuencia de la radiación (1/s), c la velocidad de la luz en el vacío ($\approx 3 \times 10^8 \text{ m/s}$) y λ la longitud de onda (m) [10], [24]. Esta relación pone de manifiesto que, dentro del espectro infrarrojo, pequeñas variaciones de temperatura en un material provocan cambios detectables en la distribución espectral de la radiación emitida. Para el ingeniero mecánico, esta base física resulta esencial, ya que explica cómo el sensor de la cámara termográfica convierte la intensidad de radiación detectada en valores de temperatura aparente, permitiendo identificar heterogeneidades térmicas asociadas a defectos internos en componentes estructurales [2], [10], [18]. En la Figura 2-26 se muestra la relación entre la energía, la frecuencia y la longitud de onda.

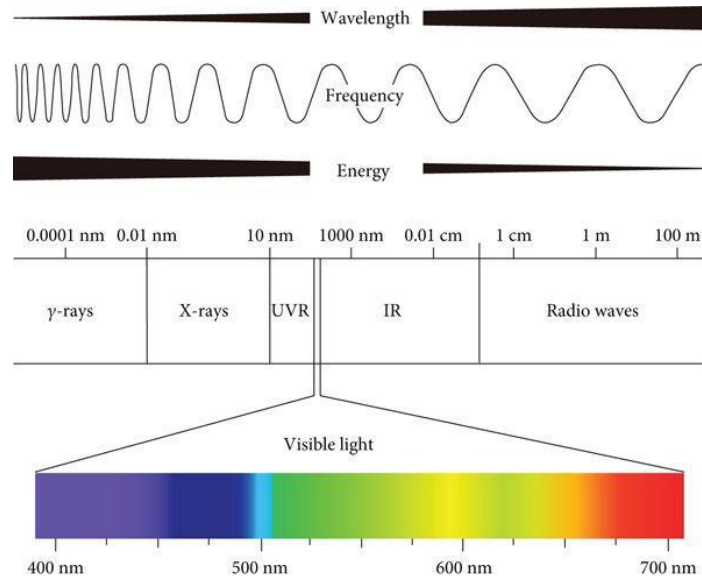


Figura 2-26 Relación entre energía, frecuencia y longitud de onda de la radiación electromagnética según la ecuación de Planck. (Extraído de [98])

Desde un enfoque teórico ideal, la potencia radiada por una superficie se describe mediante el concepto de cuerpo negro, definido como una lámpara perfecta capaz de absorber y emitir toda la energía incidente a una temperatura dada [10], [24]. La potencia total emitida por unidad de superficie de un cuerpo negro se rige por la ley de Stefan–Boltzmann, tal como se expresa en la Ecuación (2):

$$E_b = \sigma \cdot T_b^4 \quad \text{Ecuación (2)}$$

Donde E_b es la energía radiada por unidad de superficie (W/m^2), σ es la constante de Stefan–Boltzmann ($5,67 \times 10^{-8} \text{ W/m}^2 \cdot \text{K}^4$) y T_b es la temperatura absoluta del cuerpo (K) [10], [24]. Esta ecuación evidencia la fuerte dependencia no lineal entre la radiación emitida y la temperatura, lo que constituye el principio físico que permite a la termografía detectar variaciones térmicas incluso pequeñas, siempre que exista un contraste suficiente respecto al entorno. En la Figura 2-27 se muestra la curva de radiación de cuerpo negro (Stefan–Boltzmann).

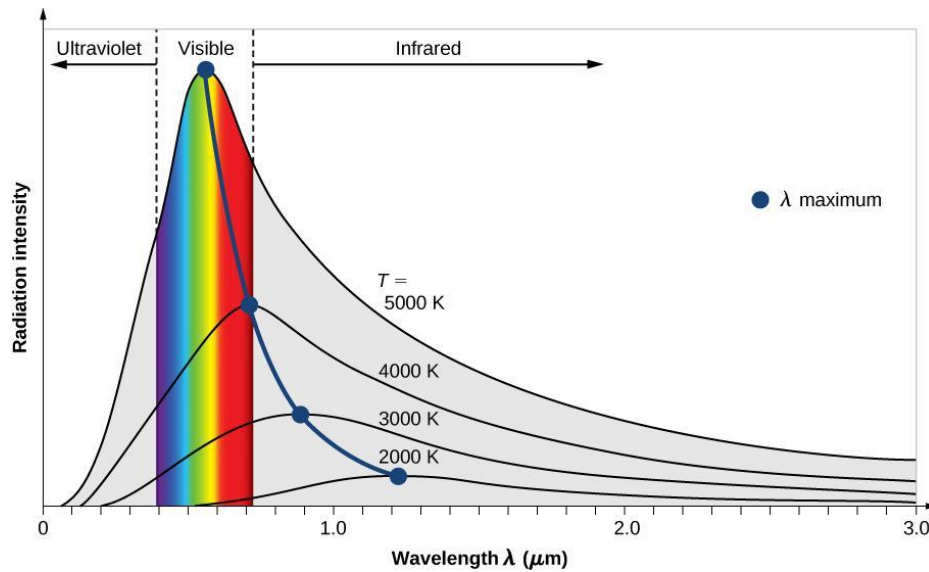


Figura 2-27 La intensidad de la radiación del cuerpo negro en función de la longitud de onda de la radiación emitida. Cada curva corresponde a una temperatura diferente del cuerpo negro, desde una temperatura baja (curva inferior) hasta una temperatura alta (curva superior) (Extraído de [99])

Sin embargo, en aplicaciones reales de ingeniería, los materiales no se comportan como cuerpos negros ideales, sino como cuerpos grises, cuya capacidad de emitir radiación depende de una propiedad superficial denominada emisividad (ε). Para estos materiales, la radiación emitida se describe mediante la Ecuación (3):

$$E_g = \varepsilon \cdot \sigma \cdot T_g^4 \quad \text{Ecuación (3)}$$

En esta expresión, ε es la emisividad del material, con valores comprendidos entre 0 y 1, y T_g es la temperatura absoluta del cuerpo (K) [10], [24]. Este parámetro resulta especialmente relevante en la inspección de materiales poliméricos y compuestos, ya que estos suelen presentar valores de emisividad elevados y relativamente estables generalmente superiores a 0,9, lo que reduce la influencia de reflexiones térmicas parásitas procedentes del entorno y favorece la aplicación de la termografía infrarroja para la detección de defectos subsuperficiales [3], [10], [24]. Una emisividad elevada garantiza que la mayor parte de la energía detectada por la cámara provenga realmente del objeto inspeccionado, aumentando la fiabilidad del diagnóstico estructural. En la Figura 2-28 se muestra la emisividad y comportamiento de cuerpo gris.

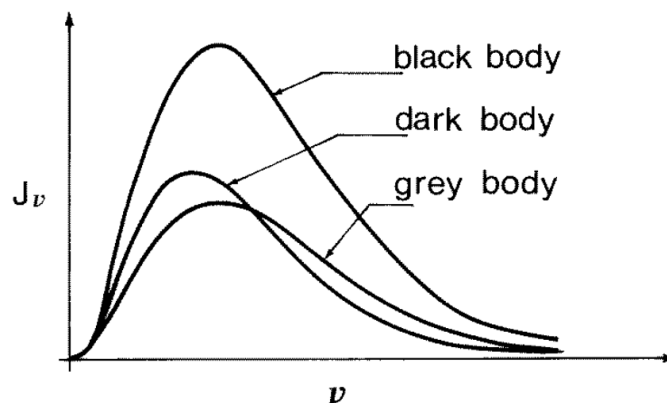


Figura 2-28 Comparación entre cuerpo negro y cuerpo gris, ilustrando la influencia de la emisividad en la radiación emitida (Extraído de [100])

Finalmente, es fundamental considerar que la señal captada por el detector de una cámara termográfica no depende únicamente de la radiación emitida por el objeto, sino que es el resultado de

una interacción compleja entre el objeto, el entorno y la atmósfera. En condiciones reales de inspección, la radiación total recibida por el sensor está influenciada por la emisión propia del objeto, la radiación reflejada desde su superficie y la radiación emitida por la columna de aire situada entre el objeto y la cámara [10], [18], [24]. Esta interacción se describe mediante la Ecuación (4):

$$W_{tot} = \varepsilon \cdot T \cdot W_{obj} + (1 - \varepsilon) \cdot T \cdot W_{refl} + (1 - T) \cdot W_{atm} \quad \text{Ecuación (4)}$$

Donde W_{tot} es la radiación total registrada por el sensor (W/m^2), W_{obj} la radiación emitida por el objeto (W/m^2), W_{refl} la radiación reflejada por su superficie (W/m^2), W_{atm} la radiación emitida por la atmósfera (W/m^2) y T la transmisividad del medio [10], [24]. Esta expresión pone de manifiesto la importancia crítica de considerar factores ambientales como la humedad relativa, la distancia de inspección y la temperatura de fondo para evitar errores radiométricos significativos [2], [10], [18]. En la Figura 2-29 se muestra un balance radiactivo real.

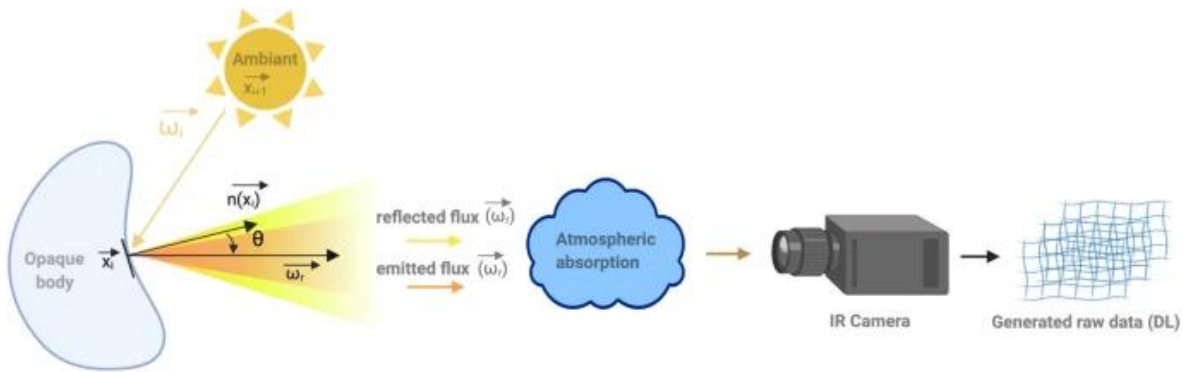


Figura 2-29 Esquema del balance radiactivo real en termografía infrarroja, incluyendo radiación emitida por el objeto, radiación reflejada y contribución atmosférica (Extraído de [101])

En el contexto de la ingeniería mecánica, este rigor analítico resulta indispensable para garantizar que el flujo de calor detectado permita caracterizar correctamente defectos subsuperficiales, como delaminaciones o porosidades, cuya huella térmica podría quedar enmascarada si no se compensan adecuadamente las variables ambientales descritas en la ecuación anterior [3], [6], [10]. En la Figura 2-30 se muestra una huella térmica de defectos subsuperficiales.

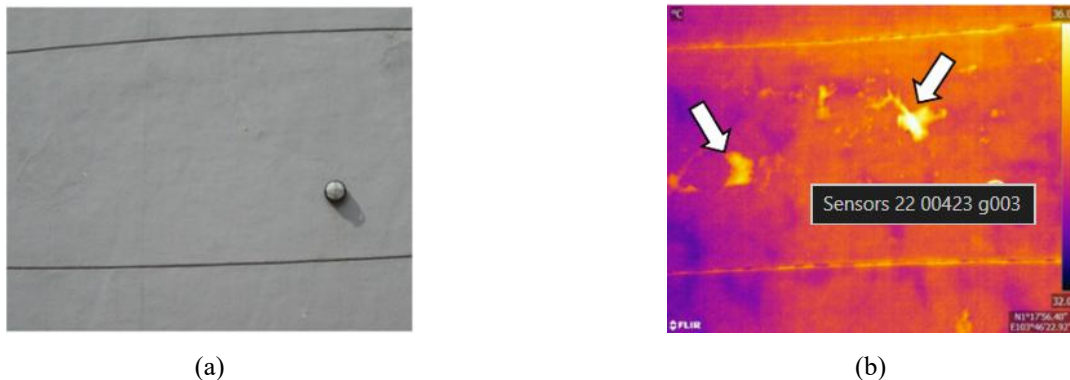


Figura 2-30 Ejemplo que muestra imágenes de delaminación en un muro de un edificio durante el ciclo de calentamiento a) imagen visual, b) imagen infrarroja (IR). Las flechas indican las zonas con delaminación. (Extraído de [102])

2.3.2 Termografía infrarroja activa: concepto y modalidades

A diferencia de la termografía pasiva, que se limita a captar la radiación natural emitida por objetos que no se encuentran en equilibrio térmico con su entorno, la modalidad activa resulta indispensable cuando el componente inspeccionado y sus posibles defectos se hallan en equilibrio termodinámico,

situación en la que los gradientes de temperatura naturales son inexistentes o insuficientes para una detección fiable [3], [10].

El fundamento físico de la termografía infrarroja activa radica en que la presencia de discontinuidades subsuperficiales como delaminaciones, porosidades, vacíos o grietas internas modifica localmente las propiedades termofísicas del material, en particular su conductividad y su difusividad térmica. Estas alteraciones actúan como barreras parciales a la propagación del calor, distorsionando el flujo térmico que se difunde desde la superficie hacia el interior del sólido [3], [6], [10]. En la Figura 2-31 se muestra una alteración del flujo térmico por defecto subsuperficial. Como consecuencia, se genera un contraste térmico medible en la superficie del componente, que es captado por el sensor de la cámara infrarroja y posteriormente analizado mediante algoritmos de procesamiento para inferir la localización geométrica, el tamaño y, en determinados casos, la profundidad de la patología interna [2], [3]. En estructuras fabricadas con polímeros reforzados con fibras (CFRP y GFRP), esta técnica adquiere especial relevancia para la identificación de daños por BVID, los cuales pueden comprometer gravemente la integridad mecánica sin dejar señales evidentes en la superficie externa [3].

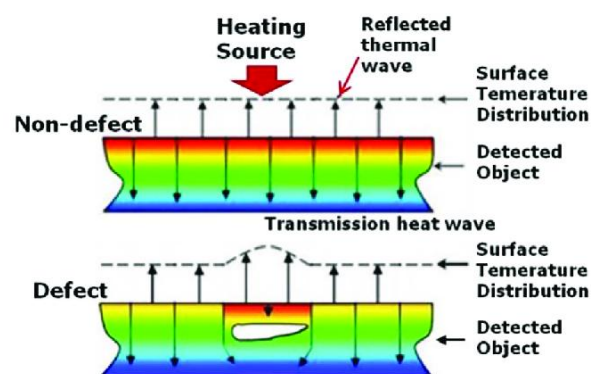


Figura 2-31 Principio de la detección mediante imagen térmica por ondas infrarrojas (Extraído de [103])

En función de la disposición relativa de la fuente de excitación térmica y de la cámara infrarroja respecto a la pieza inspeccionada, la termografía infrarroja activa puede aplicarse en dos configuraciones principales: modo de reflexión y modo de transmisión [2], [3], [10]. En el modo de reflexión, tanto la fuente térmica como la cámara se sitúan en el mismo plano frontal del componente. Esta configuración resulta especialmente adecuada para la inspección de estructuras de gran tamaño o de acceso unilateral, como cascos de buques, fuselajes de aeronaves o depósitos a presión, donde no es posible acceder a la cara posterior de la pieza [2], [3], [10]. En la Figura 2-32 se muestra una configuración modo reflexión. Desde el punto de vista de la transferencia de calor, este modo presenta una elevada sensibilidad a defectos situados a profundidades reducidas, ya que permite registrar el denominado “eco térmico” reflejado por la discontinuidad antes de que el calor se disipe completamente en el espesor del material [3], [10].

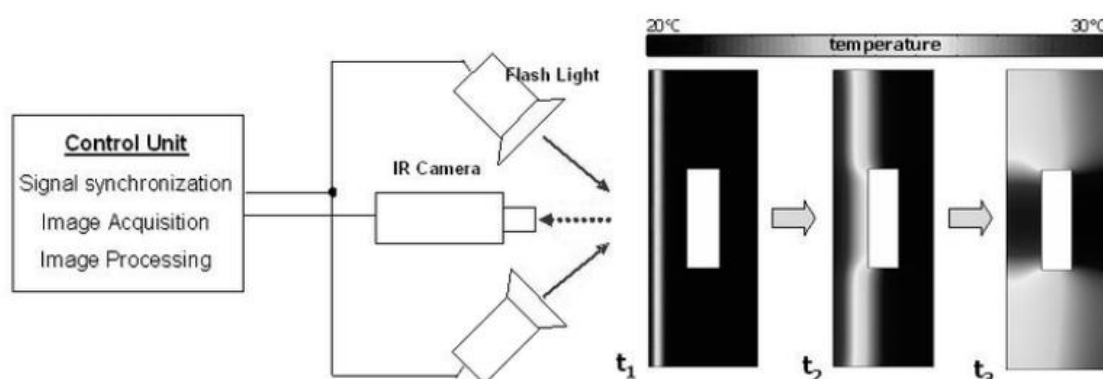


Figura 2-32 Diagrama esquemático de la termografía pulsada en configuración de reflexión (Extraído de [104])

Por el contrario, en el modo de transmisión, la excitación térmica se aplica en una cara del material mientras que la cámara infrarroja registra la respuesta térmica en la cara opuesta, evaluando la energía que ha atravesado todo el volumen del componente [2], [10]. En la Figura 2-33 se muestra una configuración modo transmisión. Esta configuración resulta especialmente adecuada para la detección y caracterización de defectos profundos o inclusiones internas, ya que permite analizar cómo la patología retrasa o atenúa la llegada del frente térmico a la superficie posterior [3], [10]. No obstante, su aplicación práctica en ingeniería se ve limitada por la necesidad de acceso a ambas caras del componente y por la pérdida progresiva de resolución espacial a medida que aumenta el espesor del material inspeccionado [3].

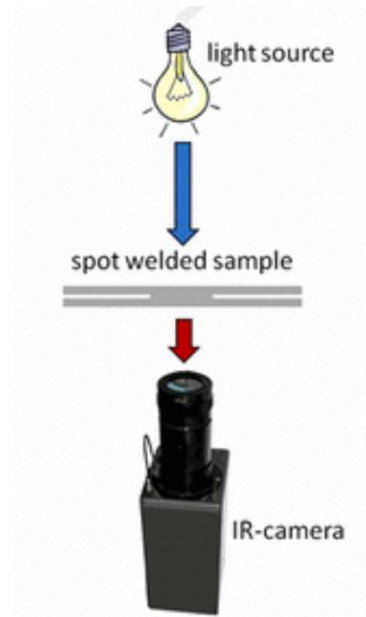


Figura 2-33 Configuración del sistema de termografía infrarroja (IRT) en modo transmisión (Extraído de [105])

Atendiendo a la naturaleza temporal y a la forma de aplicación del estímulo energético, la termografía infrarroja activa se clasifica en tres modalidades principales: termografía pulsada (*Pulsed Thermography*, PT), termografía *lock-in* (LT) y calentamiento escalonado o *step-heating* [2], [3], [10]. La termografía pulsada emplea pulsos de energía de muy corta duración, generalmente del orden de milisegundos, producidos habitualmente mediante lámparas flash de xenón. Esta excitación genera un frente térmico que se difunde rápidamente hacia el interior del material, permitiendo caracterizar un amplio rango de profundidades en un único ensayo a partir del análisis del decaimiento temporal de la temperatura superficial [3], [10]. En la Figura 2-34 se muestra una termografía Pulsada (PT).



Figura 2-34 Configuración para termografía pulsada con cuatro lámparas flash (Extraído de [106])

Por su parte, la termografía *lock-in* utiliza una excitación térmica modulada de forma sinusoidal, introduciendo ondas térmicas periódicas en el material. Esta modalidad requiere una sincronización precisa entre la fuente de excitación y el sensor infrarrojo y destaca por su elevada relación señal-ruido (*Signal to Noise Ratio*, SNR). El análisis de la amplitud y de la fase de la señal térmica permite obtener imágenes menos sensibles a calentamientos no uniformes y a variaciones de emisividad superficial, lo que mejora la fiabilidad en la detección de defectos subsuperficiales [3], [10], [107]. En la Figura 2-35 se muestra una termografía *Lock-in* (LT).

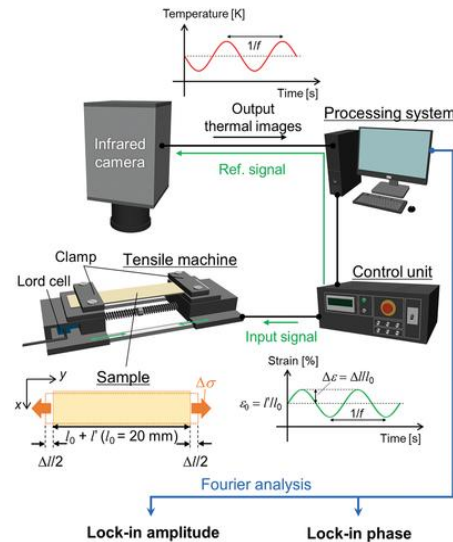


Figura 2-35 Esquema del sistema de termografía Lock-in (LIT) (Extraído de [108])

Finalmente, la técnica de *step-heating* o calentamiento convencional consiste en la aplicación de un flujo de energía constante de baja densidad durante periodos prolongados. Esta modalidad resulta especialmente adecuada para la inspección de estructuras multicapa de gran espesor o de materiales altamente aislantes, ya que permite una penetración térmica profunda sin riesgo de degradación térmica de la matriz polimérica superficial [3], [109]. En la Figura 2-36 se muestra un *Step-heating*. En última instancia, la elección de la modalidad de termografía infrarroja activa más adecuada dependerá del equilibrio entre la resolución requerida, la profundidad esperada del defecto y las propiedades termofísicas del material bajo inspección, en particular su difusividad térmica [3], [10].

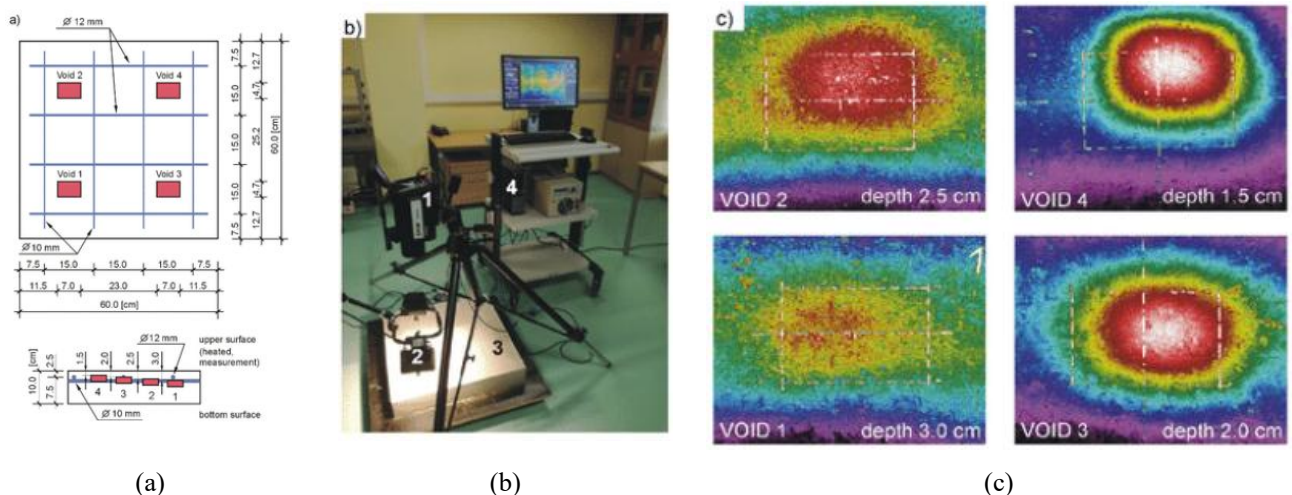


Figura 2-36 Ensayo de termografía por calentamiento escalonado a) posición esquemática de los vacíos y de las barras de refuerzo en la placa, b) configuración experimental 1-cámara IR, 2-lámparas halógenas, 3-placa de hormigón ensayada, 4-ordenador, c) termogramas de cada defecto a los 10 minutos (Extraído de [110])

2.3.3 Fase de adquisición térmica

La fase de adquisición de datos en la termografía infrarroja activa constituye el eslabón fundamental para garantizar la integridad y fiabilidad del análisis estructural posterior, ya que en esta etapa se capturan las perturbaciones del flujo térmico transitorio provocadas por la presencia de patologías subsuperficiales. En el ámbito de la ingeniería mecánica, este proceso se basa en la aplicación controlada de un estímulo energético externo sobre la superficie del componente, generando un gradiente térmico que se propaga hacia el interior del material mediante conducción física [2], [10]. La correcta caracterización de esta fenómeno resulta esencial, puesto que cualquier error en la adquisición puede comprometer la detección y cuantificación de los defectos internos.

Los sistemas de adquisición actuales emplean una amplia variedad de estimuladores térmicos en función de la técnica de termografía activa seleccionada. Entre ellos destacan las lámparas flash de xenón, utilizadas en termografía pulsada (PT) por su elevada densidad energética y corta duración del pulso, así como las lámparas halógenas moduladas empleadas en termografía lock-in (LT). En aplicaciones específicas, también se recurre a técnicas de excitación mediante corrientes de inducción o ultrasonidos, especialmente en materiales con propiedades particulares [2], [3], [35]. En todos los casos, resulta imprescindible que el ingeniero defina correctamente la disposición geométrica del montaje experimental, manteniendo una distancia cámara objeto adecuado y un ángulo de incidencia preferiblemente inferior a 50° , con el fin de minimizar las reflexiones parásitas y evitar pérdidas de resolución espacial en los termogramas adquiridos [2], [16].

La conversión precisa de la radiación electromagnética detectada por el sensor en valores térmicos cuantitativos exige la integración de múltiples variables ambientales y termofísicas que condicionan la exactitud de los mapas de temperatura. Durante la adquisición de los termogramas, es necesario calibrar y configurar en el software de adquisición parámetros críticos como la emisividad superficial (ε), la temperatura de fondo reflejada (T_{refl}), distancia cámara objeto (d), la temperatura atmosférica (T_{atm}) y la humedad relativa (RH), los cuales los tres últimos determinan el coeficiente de atenuación de la señal térmica registrada [2], [16]. En la Figura 2-37 se muestra parámetros radiométricos y calibración. En materiales de alta responsabilidad, como los polímeros reforzados con fibras de carbono (CFRP) o de vidrio (GFRP), que suelen presentar valores de emisividad elevados y estables (*generalmente* $\varepsilon \geq 0.9$), es habitual realizar calibraciones in situ mediante cuerpos negros de referencia o cintas adhesivas de emisividad conocida para compensar errores radiométricos [2], [3], [16].

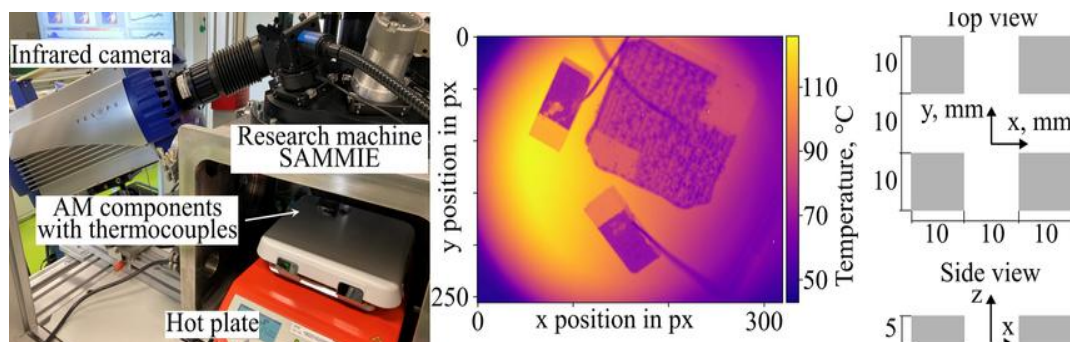


Figura 2-37 Procedimiento de calibración radiométrica mediante cuerpo negro de referencia (Extraído de [111])

Para la corrección de la temperatura reflejada, se emplean comúnmente reflectores difusos, como láminas de aluminio rugoso, que permiten al sensor identificar la radiación procedente del entorno y restarla de la señal total recibida [2], [16]. Asimismo, la transmisividad del medio atmosférico (T) adquiere especial relevancia cuando la distancia entre la cámara y la pieza inspeccionada es considerable, siendo necesario aplicar ajustes basados en la ley de Stefan–Boltzmann para transformar las intensidades de brillo detectadas en temperaturas reales y no simplemente aparentes [3], [16]. Solo mediante este

control riguroso de las condiciones experimentales puede garantizarse que las anomalías registradas correspondan efectivamente a variaciones en la difusividad térmica local del material y no a interferencias externas o ruido de fondo [2], [10], [18].

En lo relativo a la estrategia de muestreo y a los parámetros temporales del ensayo, la elección de la frecuencia de adquisición y del tiempo total de grabación depende intrínsecamente de las propiedades térmicas del material y de la resolución requerida para caracterizar el defecto. De acuerdo con el teorema del muestreo de Shannon, para recuperar la señal térmica sin pérdida de información, la frecuencia de muestreo de la cámara debe ser al menos el doble de la frecuencia máxima presente en la respuesta térmica del material [3]. En la termografía pulsada, se requieren frecuencias de adquisición elevadas, típicamente comprendidas entre 50 Hz y 100 Hz, para capturar el rápido decaimiento de la temperatura tras el pulso inicial. En cambio, en la termografía lock-in, el análisis se centra en la amplitud y la fase de una excitación periódica, proporcionando una mayor robustez frente a calentamientos no uniformes [2], [35].

El tiempo total de adquisición (T_{acq}) debe ajustarse en función de la conductividad térmica del objeto inspeccionado. Materiales como el CFRP, con mayor difusividad térmica, presentan respuestas térmicas más rápidas que materiales más aislantes como el GFRP, lo que obliga a establecer un compromiso entre la velocidad de disparo de la cámara y la capacidad de almacenamiento del sistema [3]. Finalmente, la secuencia de termogramas resultante se almacena como una matriz tridimensional de datos (espacial y temporal), que constituye la base para la aplicación posterior de algoritmos de post-procesado y técnicas de inteligencia artificial orientadas a la segmentación y reconstrucción de patologías internas [6], [10], [24].

2.3.4 Procesado

La efectividad de la termografía infrarroja activa como técnica de evaluación no destructiva depende de manera crítica de la etapa de procesado de los datos adquiridos, ya que los termogramas capturados en bruto suelen estar degradados por diversas fuentes de ruido y artefactos térmicos [2]. Durante la fase de adquisición, la señal térmica puede verse afectada por múltiples factores, entre ellos el ruido electrónico propio del equipo, calentamientos no uniformes derivados de imperfecciones en la fuente de excitación, variaciones locales de la emisividad superficial y reflexiones parásitas procedentes del entorno circundante [2], [18]. En la Figura 2-38 se muestra una comparación entre termograma bruto y termograma tras procesado digital, mostrando mejora en el contraste del defecto.

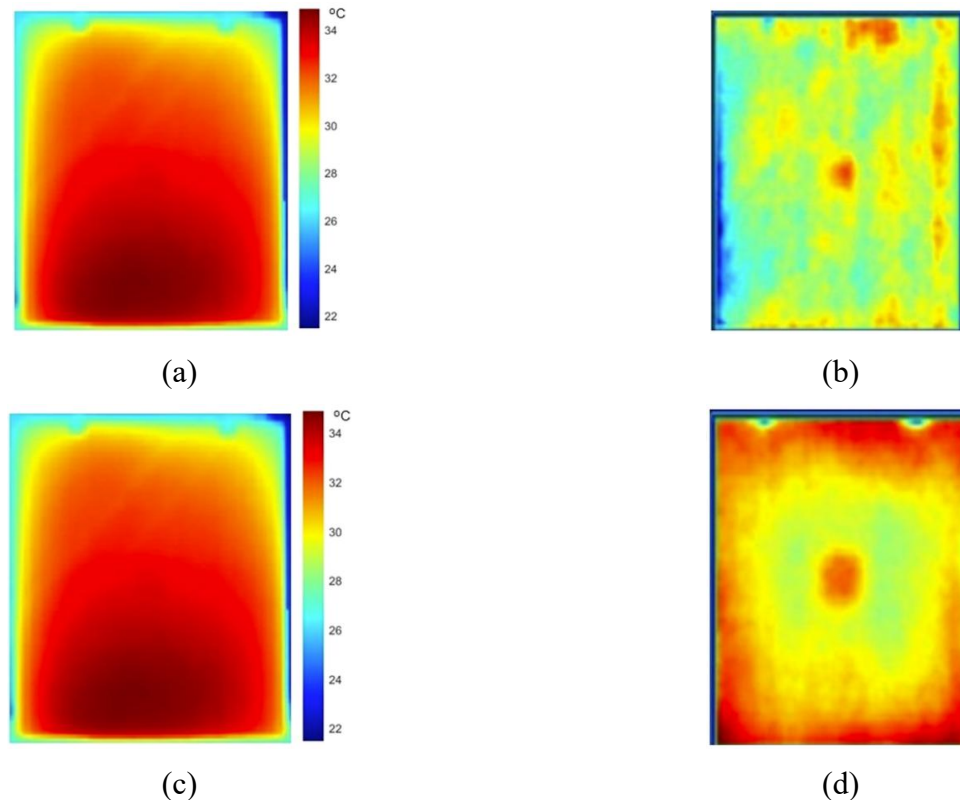


Figura 2-38 Comparación entre termogramas RAW y termogramas reconstruidos tras aplicar el algoritmo PCT a) muestra el termograma original (RAW), donde el defecto apenas se distingue debido al bajo contraste térmico, b) presenta la imagen procesada mediante el algoritmo PCT (Transformada de Componentes Principales), en la que el defecto se visualiza con mayor claridad gracias al aumento del contraste y la reducción del ruido, c) muestra el termograma original (RAW), d) corresponde a la imagen obtenida tras aplicar el algoritmo PCT, donde el defecto aparece más definido y con mejor diferenciación respecto al fondo (Extraído de [112])

Desde un punto de vista metodológico, el análisis de los datos termográficos puede clasificarse en dos vertientes principales. Por un lado, el análisis cualitativo se centra en la evaluación de la distribución relativa de temperaturas para localizar anomalías espaciales asociadas a defectos. Por otro lado, el análisis cuantitativo busca determinar propiedades termofísicas precisas del material y estimar parámetros geométricos como la profundidad o el espesor de las patologías detectadas [2], [3], [10], [18]. Sin este rigor analítico, la interpretación de los resultados quedaría sujeta a la experiencia subjetiva del operador, comprometiendo la fiabilidad de las inspecciones en aplicaciones de alta responsabilidad técnica [2], [97].

Dentro del conjunto de metodologías avanzadas orientadas a maximizar la relación señal-ruido (SNR), la Reconstrucción de la Señal Termográfica (TSR), propuesta inicialmente por Shepard, constituye una herramienta fundamental para el tratamiento de datos termografía [2], [92]. En la Figura 2-39 se muestra un ejemplo de TSR. Este método se basa en el ajuste de la evolución temporal del enfriamiento de cada píxel a un modelo polinómico de bajo orden generalmente entre grado 4 y 8 en el dominio logarítmico, apoyándose en la solución unidimensional de la ecuación de difusión del calor [2], [6], [10].

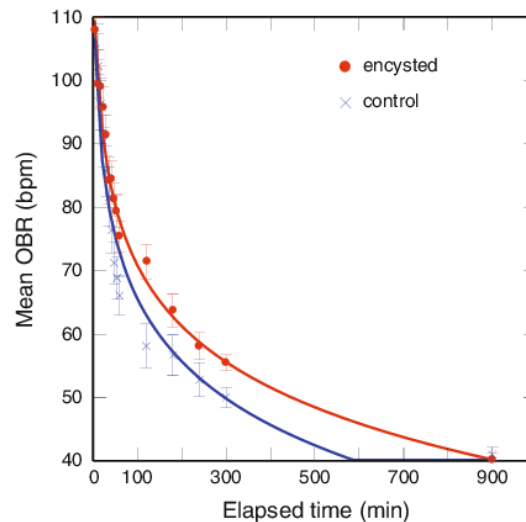


Figura 2-39 Ajuste polinómico de la curva térmica mediante TSR y cálculo de derivadas para mejorar detección de defectos (Extraído de [113])

La técnica TSR no solo reduce de manera significativa el ruido aleatorio de alta frecuencia, sino que además permite obtener derivadas temporales de primer y segundo orden de la temperatura. Estas derivadas son especialmente sensibles a perturbaciones en el flujo térmico provocadas por defectos subsuperficiales sutiles que, en muchas ocasiones, permanecen enmascarados en la secuencia original [2], [6], [10]. Además de mejorar la detección de delaminaciones o porosidades, la TSR optimiza la gestión de datos al posibilitar el almacenamiento exclusivo de los coeficientes del polinomio ajustado en lugar de la secuencia completa de imágenes térmicas. Esta característica representa una ventaja relevante en entornos industriales donde se manejan grandes volúmenes de información, como en el contexto de la Industria 4,0 [2], [10].

De forma complementaria, el ingeniero mecánico recurre a técnicas estadísticas multivariantes de alta eficacia como la Termografía de Componentes Principales (PCT) y su variante avanzada, la Termografía de Componentes Principales Dispersa (SPCT) [2], [3]. Introducida por Rajic en 2002 [2], [10], [93], la PCT aplica el Análisis de Componentes Principales (*Principal Component Analysis*, PCA) mediante el algoritmo de Descomposición de Valores Singulares (*Singular Value Decomposition*, SVD) sobre la secuencia temporal de imágenes térmicas. Este procedimiento transforma la matriz tridimensional de datos (dos dimensiones espaciales y una temporal) en un conjunto reducido de Funciones Ortogonales Empíricas (*Empirical Orthogonal Function*, EOF), también conocidas en termografía infrarroja como autoimágenes o eigenimágenes. Las primeras funciones capturan la mayor parte de la varianza espacial y temporal de la respuesta térmica de un objeto, lo que facilita la identificación de defectos de distintos tamaños y profundidades en un pequeño grupo de imágenes de alta claridad [2], [10], [93].

Por su parte, la técnica SPCT introduce una penalización de dispersión (*sparsity penalty*) sobre las EOF obtenidas, permitiendo separar con mayor nitidez las huellas térmicas correspondientes a defectos que se encuentran superpuestos o muy próximos entre sí. De este modo, se reduce la dependencia estadística entre componentes y se mejora la resolución en escenarios complejos [2], [3]. En la Figura 2-40 se muestra comparación de resultados obtenidos mediante datos brutos (*raw*), PCT y SPCT. Estos métodos, combinados con la Termografía de Fase Pulsada (*Pulse Phase Thermography*, PPT), que traslada el análisis temporal al dominio de la frecuencia, que traslada el análisis al dominio de la frecuencia mediante la Transformada de Fourier, permiten discriminar con mayor precisión entre el ruido estructural del material y las señales reales asociadas a fallos críticos en polímeros reforzados [2], [6], [10]. En la Figura 2-41 se muestra un ejemplo del algoritmo PPT.

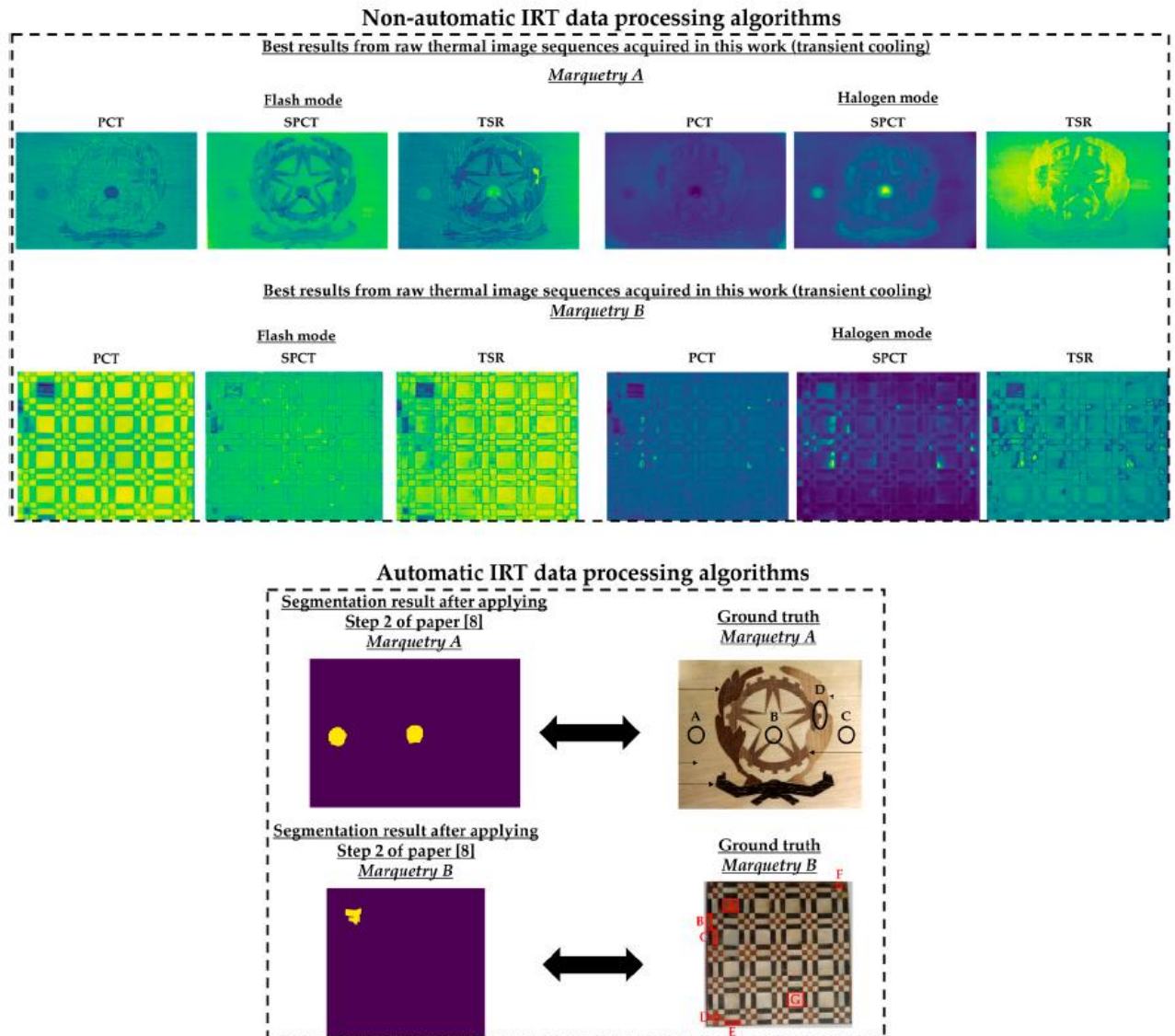


Figura 2-40 Resultados obtenidos mediante la aplicación de algunos algoritmos de procesamiento de datos de termografía infrarroja (IRT) de última generación a las marqueterías bajo inspección (Extraído de [114])

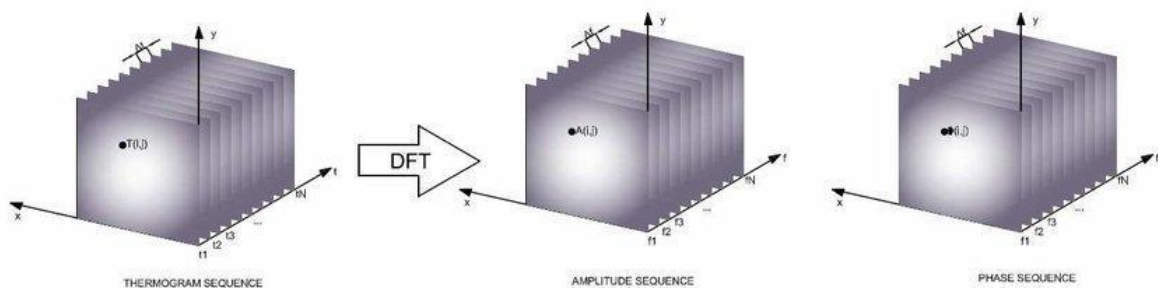


Figura 2-41 Esquema de la Transformada Discreta de Fourier aplicada a una secuencia de termogramas (Extraído de [115])

En la vanguardia del procesado, la integración de la Inteligencia Artificial (IA) y el *Deep Learning* (DL) ha permitido avanzar hacia la automatización completa del diagnóstico estructural [2], [24], [93]. El empleo de redes neuronales convolucionales (*Convolutional Neural Networks*, CNN), y en particular arquitecturas de segmentación de instancias como el modelo Mask R-CNN, facilita la localización, clasificación y delimitación automática del contorno de cada defecto mediante la

generación de máscaras binarias de alta resolución sobre los termogramas [6], [24]. En la Figura 2-42 se muestra un ejemplo de segmentación automática de defectos mediante red neuronal convolucional. La limitación principal de la IA y el DL en el análisis de materiales poliméricos, es la necesidad de tener un banco de entrada de datos de gran tamaño con distintos materiales, tamaños, geometrías y diferentes condiciones aplicadas de excitación térmica en caso de que el aprendizaje de esos modelos se desea que sea el óptimo.

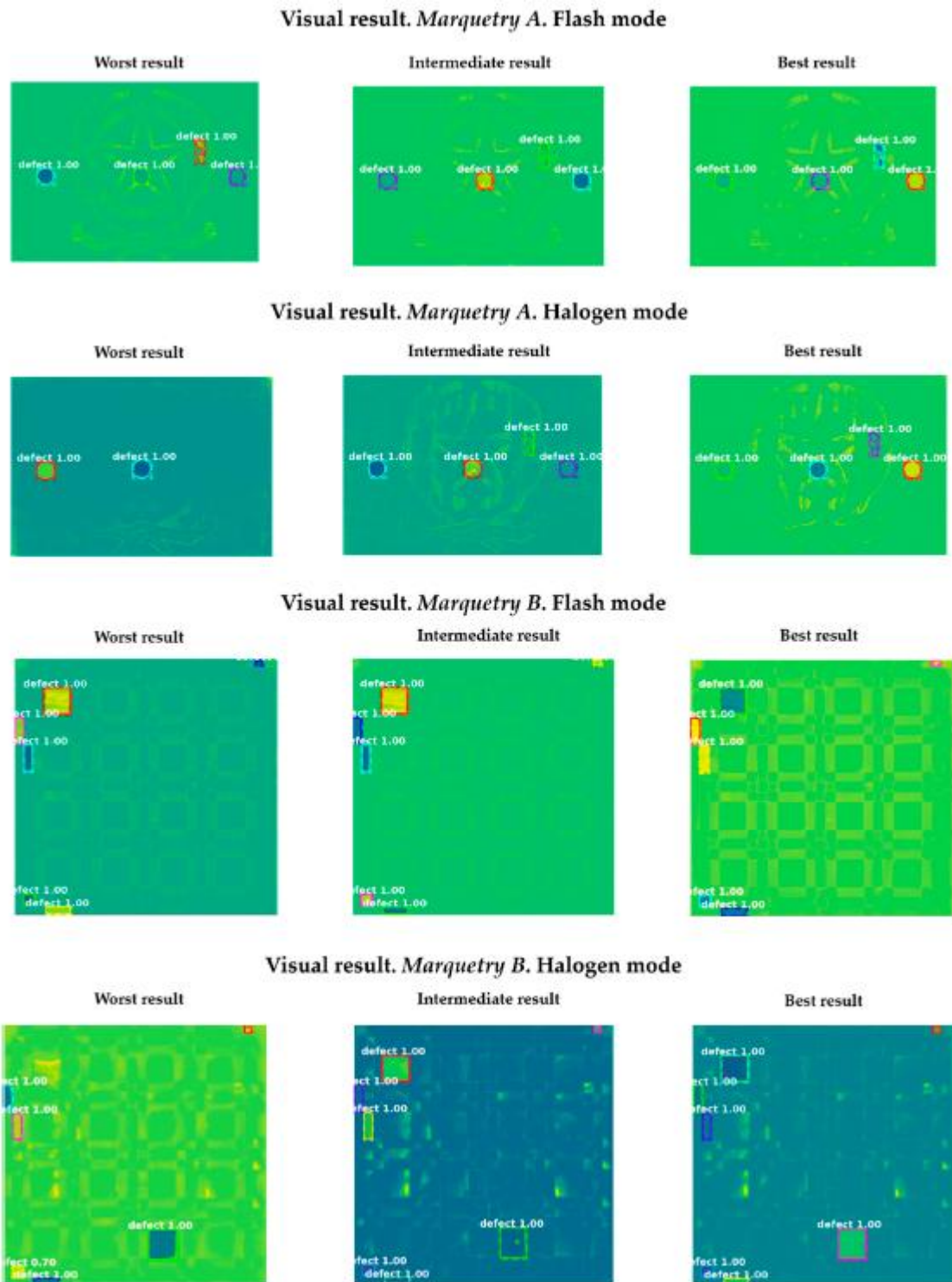


Figura 2-42 Resultado visual de la detección de la peor, intermedia y mejor posición del defecto, así como de la segmentación del área del defecto, aplicando el conjunto de pesos y sesgos obtenidos en el mejor proceso de entrenamiento de Mask R-CNN sobre el conjunto de validación (Extraído de [114])

Para evaluar el rendimiento de los algoritmos de procesamiento de imágenes, se emplean métricas de prestación como la precisión, el *recall* y el *F-score*, junto con el cálculo de la métrica *Signal to Noise Ratio* (SNR), que cuantifica el contraste entre la región defectuosa y la zona libre de daños [6], [24], [90]. Asimismo, el modelado numérico mediante diferencias finitas permite realizar la reconstrucción tridimensional de las patologías detectadas, estimando parámetros como el volumen, la orientación y la profundidad exacta del defecto dentro del material a partir de representaciones en forma de malla [10]. En la Figura 2-43 se muestra un ejemplo conceptual de reconstrucción tridimensional de defecto interno mediante modelado numérico térmico.

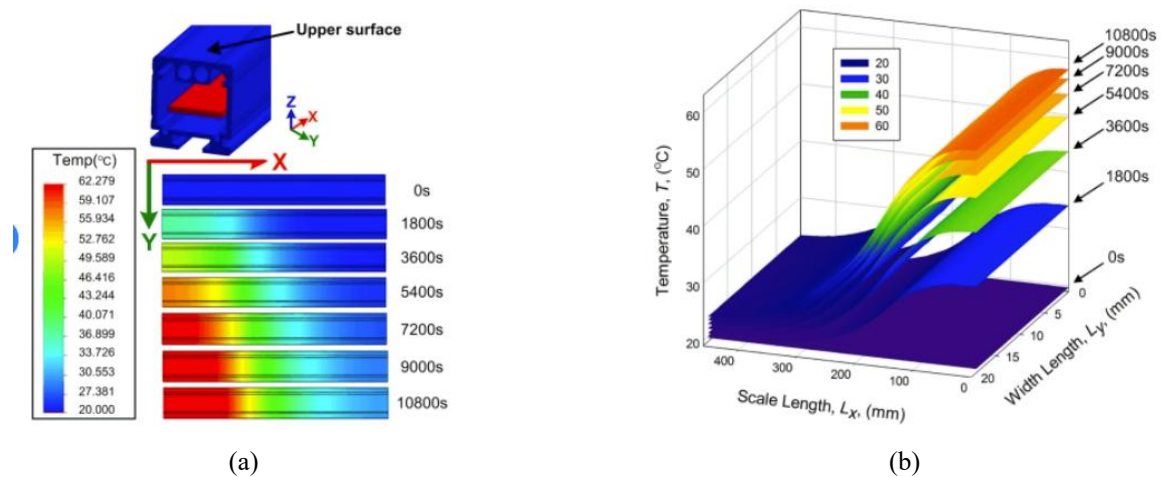


Figura 2-43 Simulación mediante el método de elementos finitos de la conducción térmica del sensor durante 3 horas a) variación de la temperatura en la superficie superior de la carcasa, b) datos de temperatura en la superficie superior (Extraído de [116])

En conjunto, la sinergia entre sensores infrarrojos avanzados, técnicas de procesamiento matemático y algoritmos de visión artificial define el estándar moderno en la monitorización de la salud estructural (SHM). Este enfoque integrado garantiza una mayor fiabilidad diagnóstica y contribuye a asegurar la seguridad operativa de activos mecánicos en sectores estratégicos como la Defensa y la ingeniería naval [6], [24], [55].

2.4 Aplicaciones de la termografía infrarroja activa en materiales poliméricos impresos en 3D

2.4.1 Fabricación aditiva de polímeros y problemática asociada

La fabricación aditiva (*Additive Manufacturing*, AM) se define técnicamente como el conjunto de procesos destinados a la fabricación de componentes mediante la unión progresiva de material a partir de información digital tridimensional, generalmente siguiendo una estrategia de deposición capa a capa. Esta tecnología ha supuesto una transformación profunda en los modelos de producción industrial, al permitir la generación de geometrías altamente complejas que resultan inviables mediante métodos de fabricación sustractivos convencionales [29], [117]. El principio operativo de la AM se basa en la conversión de diseños digitales desarrollados mediante herramientas de diseño asistido por ordenador (*Computer-Aided Design*, CAD) en archivos de triangulación estándar (STL), los cuales son posteriormente segmentados computacionalmente para generar instrucciones precisas de deposición que gobiernan el movimiento y la operación de la máquina [2], [3]. En la Figura 2-44 se muestra una representación esquemática del proceso de fabricación aditiva mediante deposición sucesiva de capas.

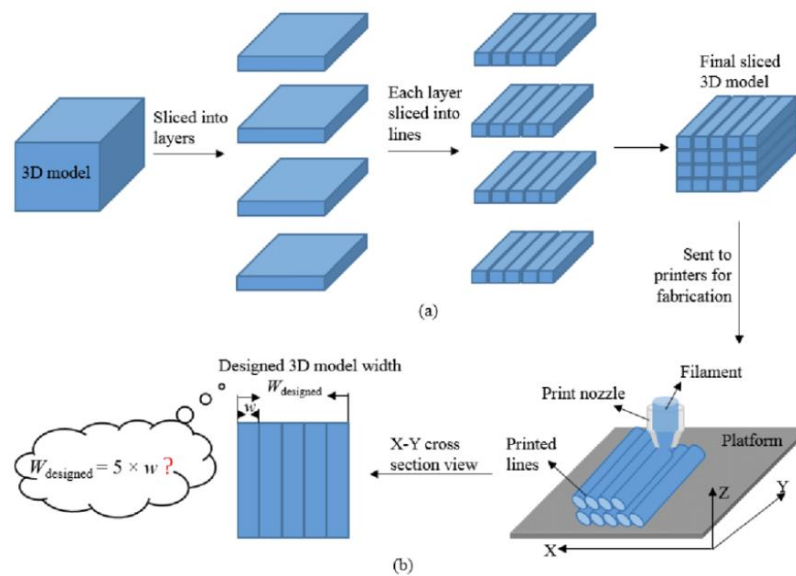


Figura 2-44 a) proceso de seccionado de modelos 3D en capas y trayectorias para su impresión b) proceso de fabricación e ilustración del ancho de corte (*slicing width*) (Extraído de [118])

En el ámbito específico de los materiales poliméricos, la técnica de Modelado por Deposición Fundida (FDM) o fabricación con filamento fundido (FFF) se ha consolidado como la modalidad más extendida, debido a su elevada flexibilidad operativa, bajo coste relativo y amplia disponibilidad de materiales termoplásticos comerciales como el PLA, el acrilonitrilo butadieno estireno (*Acrylonitrile Butadiene Styrene*, ABS) y el PETG [6], [9], [10]. En la Figura 2-45 se muestra un proceso FDM mostrando la extrusión del filamento fundido capa a capa. No obstante, la fabricación aditiva de polímeros no se limita a procesos basados en extrusión térmica, sino que incluye tecnologías avanzadas de fotopolimerización como la Estereolitografía (*Stereolithography*, SLA) y la Fabricación Aditiva Volumétrica Tomográfica (*Tomographic Volumetric Additive Manufacturing*, TVAM), las cuales emplean radiación ultravioleta o láseres de alta energía para curar resinas líquidas con resoluciones geométricas significativamente superiores [3], [28], [55]. En la Figura 2-46 se muestra un ejemplo de SLA. La transición de estas tecnologías desde entornos de prototipado hacia redes de producción descentralizadas exige un control exhaustivo de las propiedades mecánicas finales del componente, dado que la integridad estructural depende de forma crítica de la cohesión entre las capas sucesivas depositadas [6], [119], [120].

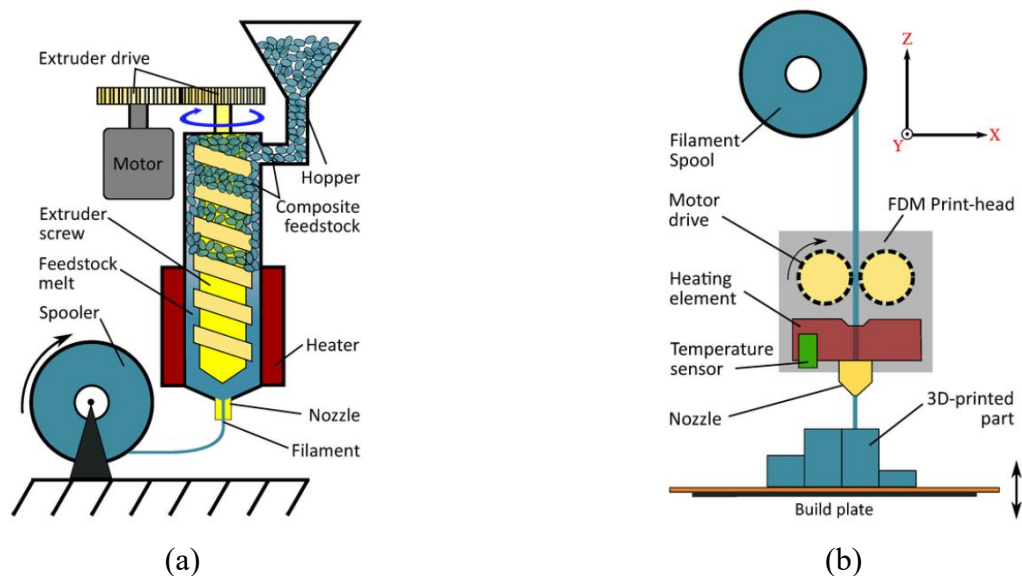


Figura 2-45 a) representación esquemática del proceso de extrusión de filamento. Los fragmentos de material compuesto amasado son impulsados a través de un cilindro calefactado mediante un tornillo sin fin, extruidos a través de una boquilla de diámetro definido y posteriormente bobinados b) proceso de impresión FDM: un sistema motorizado introduce el filamento en el cabezal de impresión con control de temperatura, montado sobre una plataforma XY. Tras pasar por la boquilla, el material se deposita sobre la placa de fabricación, que desciende progresivamente para permitir la conformación de la siguiente capa (Extraído de [121])

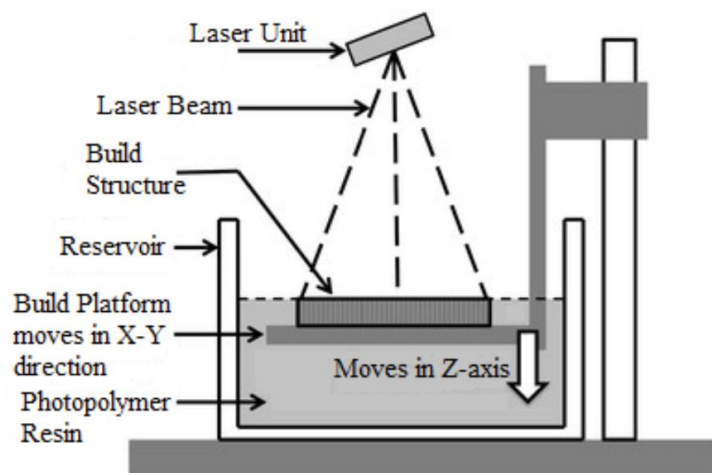


Figura 2-46 Esquemática del proceso de estereolitografía (SLA) (Imagen cortesía de 3D Systems) (Extraído de [122])

La problemática técnica asociada a la fabricación aditiva de polímeros se origina principalmente en las limitaciones inherentes al proceso de estratificación, el cual introduce defectos geométricos y estructurales que pueden comprometer la funcionalidad del activo. Uno de los fenómenos más comunes es el efecto de escalera (*stepping effect*), derivado de la deposición ortogonal de capas discretas, que se traduce en una rugosidad superficial elevada y dificulta la obtención de curvas continuas en piezas de alta fidelidad geométrica [123]. Figura 2-47 se muestra ejemplo de un efecto escalera en superficies inclinadas debido a la deposición capa a capa. Además, durante la fase de conversión del modelo CAD a formato STL y el posterior proceso de *slicing*, pueden generarse errores volumétricos y de relleno que alteran la densidad nominal prevista en el diseño original [28], [123]. Desde una perspectiva mecánica, la aparición de patologías internas como delaminaciones, vacíos, porosidad y desalineación de fibras en materiales reforzados constituye una preocupación constante, ya que estas discontinuidades actúan como concentradores de tensiones y reducen significativamente la resistencia estructural del componente [10],

[109]. En la Figura 2-48 se muestra la sección transversal de una pieza impresa mostrando porosidad y defectos interlaminares.



Figura 2-47 Efecto escalera en superficies inclinadas debido a la deposición capa a capa (Extraído de [124])

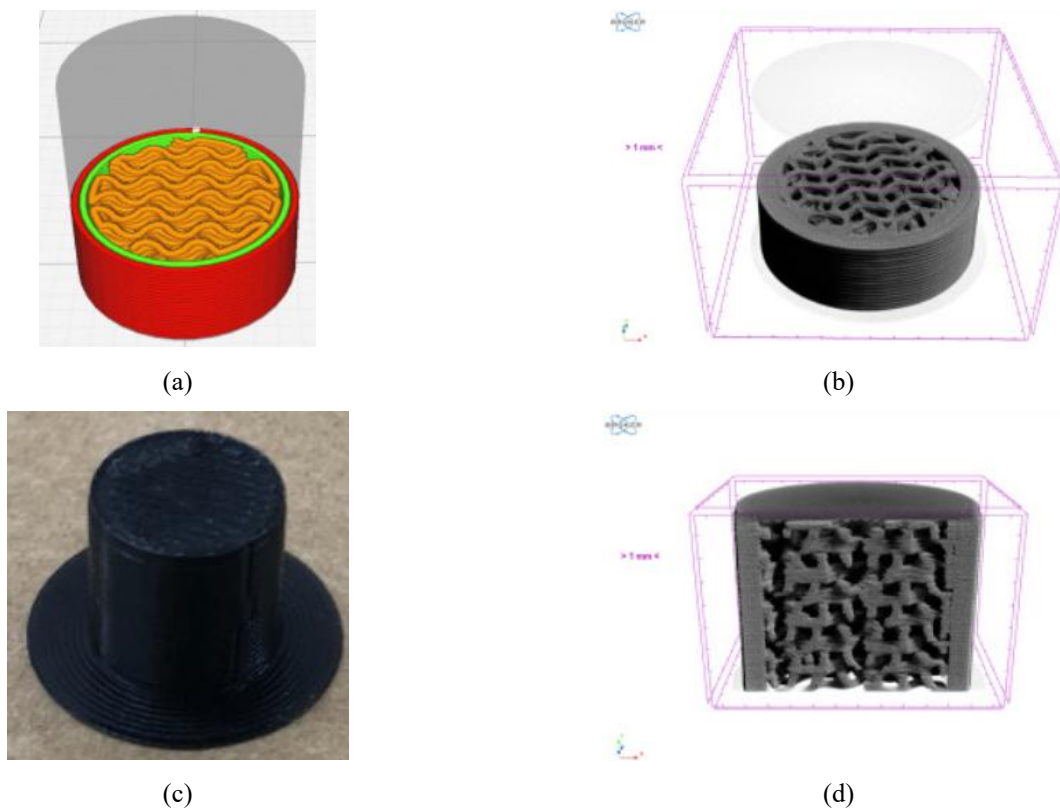


Figura 2-48 Imagen de una pieza FFF con relleno tipo gyroid a) seccionado del modelo, b) sección transversal en el plano XZ de la pieza, c) pieza impresa, d) sección transversal en el plano XY de la pieza (Extraído de [30])

Estas patologías suelen estar asociadas a gradientes térmicos no controlados durante el proceso de fabricación, los cuales inducen tensiones residuales, deformaciones fuera de plano (*warping*) e inestabilidad dimensional, especialmente en polímeros con elevada contracción térmica como el ABS [9], [10]. Adicionalmente, en compuestos biopoliméricos o reforzados con fibras naturales, la naturaleza hidrofílica de las fibras puede provocar una adhesión interfacial deficiente con matrices poliméricas hidrofóbicas, reduciendo la eficiencia en la transferencia de esfuerzos y promoviendo fallos prematuros por fatiga o impacto [10], [21], [109]. Estos efectos comprometen de forma directa la fiabilidad mecánica de los componentes fabricados y limitan su aplicación en entornos industriales exigentes.

Desde una perspectiva de desempeño operativo, los componentes poliméricos fabricados mediante AM presentan un comportamiento mecánico marcadamente anisótropo, caracterizado por una resistencia significativamente inferior en el eje de construcción (dirección Z) en comparación con los planos longitudinales. Esta debilidad estructural se debe a la limitada calidad de las uniones térmicas entre capas

sucesivas [10], [38]. Esta limitación se acentúa en entornos industriales severos, donde muchos polímeros comerciales presentan una resistencia química reducida frente a agentes agresivos y una estabilidad térmica inferior a la de las aleaciones metálicas tradicionales [21]. En tecnologías de alta resolución como el TVAM o la polimerización de dos fotones (2PP), surge además un compromiso crítico entre la productividad volumétrica y la resolución espacial, ya que un exceso de energía láser puede provocar fenómenos de degradación térmica o “quemado” de la resina antes de completar el curado [3], [10]. Para mitigar estas limitaciones y alcanzar los requisitos de rigidez y fiabilidad mecánica exigidos en aplicaciones de alta responsabilidad, muchos componentes fabricados mediante AM requieren etapas adicionales de post-procesado, tales como curado Ultravioleta (*Ultraviolet*, UV) complementario, tratamientos térmicos o acabados superficiales específicos. Estas operaciones incrementan los tiempos y los costes de fabricación, reduciendo la competitividad del proceso en entornos productivos a gran escala [2], [77]. En consecuencia, la implementación efectiva de polímeros impresos en sectores críticos como el naval o el de Defensa exige el desarrollo de protocolos de inspección rigurosos basados en Ensayos No Destructivos, que permitan certificar la ausencia de defectos subsuperficiales críticos derivados de las complejas variables de proceso propias de la fabricación aditiva [9], [55], [83].

2.4.2 Aplicación de la termografía infrarroja activa en polímeros comunes

La aplicación de la termografía infrarroja activa en polímeros de uso generalizado en fabricación aditiva, como el ácido poliláctico (PLA), el tereftalato de polietileno glicolado (PETG) y el acrilonitrilo butadieno estireno (ABS), responde a una necesidad técnica derivada de la propia naturaleza termodinámica de estos materiales. En condiciones normales de servicio, estos termoplásticos suelen encontrarse en equilibrio térmico con su entorno operativo, lo que impide la detección fiable de defectos internos mediante técnicas de termografía pasiva [2]. En este contexto, la aplicación de una excitación térmica externa controlada permite generar gradientes de temperatura artificiales cuya evolución temporal es sensible a la presencia de discontinuidades internas.

Desde el punto de vista radiométrico, estos polímeros presentan una emisividad intrínsecamente elevada, con valores típicos próximos a $\varepsilon \approx 0,9$, sobre todo los que tengan una aplicación de pintura superficial de color negro, lo que favorece la captación eficiente de la radiación térmica emitida por la superficie y reduce significativamente los errores asociados a reflexiones parásitas del entorno en comparación con superficies metálicas altamente reflectantes [2], [16]. Esta característica convierte a los polímeros impresos en 3D en candidatos especialmente adecuados para la inspección mediante termografía infrarroja activa, ya que la señal registrada por la cámara está dominada por la emisión propia del material. En el caso específico del ABS, su elevada rigidez, buena resistencia mecánica y durabilidad lo posiciona como un material ampliamente utilizado en componentes funcionales, presentando valores de módulo de tracción del orden de 2.300 MPa y una temperatura de reblandecimiento Vicat cercana a 95 °C [9], [125]. No obstante, el proceso de fabricación por deposición fundida (FDM) en ABS suele inducir tensiones residuales y fenómenos de alabeo que pueden dar lugar a delaminaciones internas y daños por impacto apenas visibles (BVID), los cuales actúan como barreras térmicas que alteran localmente la difusividad del material, facilitando su detección mediante el análisis de anomalías en el flujo de calor inducido externamente [9], [69], [97]. En la Figura 2-49 se muestra defectos interlaminares típicos en ABS impreso mediante FDM.

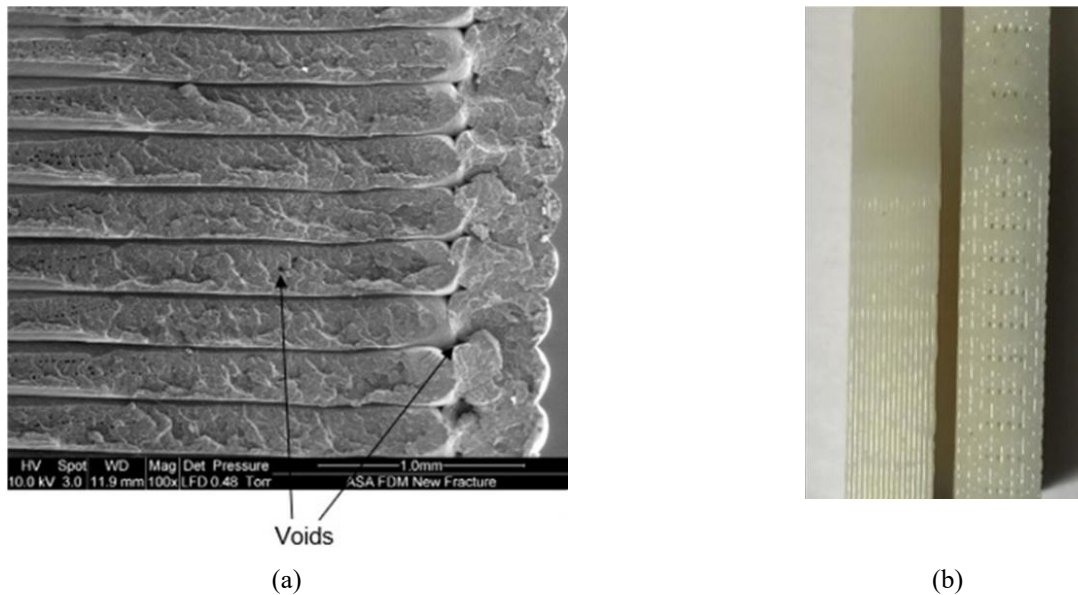


Figura 2-49 Vacíos en piezas impresas mediante FDM a) Imagen obtenida con Microscopio Electrónico de Barrido (SEM) que muestra los vacíos típicos inherentes al proceso FDM, b) Imagen óptica de una probeta tipo “dog-bone” donde se observan claramente los vacíos (Extraído de [126])

Desde la perspectiva del comportamiento termofísico, el PETG destaca por su buena fluidez, elevada resistencia química y estabilidad dimensional, siendo ampliamente empleado en la impresión de utillaje industrial, carcasas y componentes sometidos a ambientes agresivos [3], [9]. Al igual que otros termoplásticos, el PETG presenta una baja conductividad térmica, lo que favorece la persistencia temporal de los gradientes térmicos generados durante la excitación, permitiendo una ventana de observación más amplia para la detección de defectos subsuperficiales mediante lámparas halógenas o fuentes de pulso térmico [2], [6]. Esta característica resulta especialmente útil para la identificación de vacíos internos y zonas de mala adhesión interlaminar, frecuentes en procesos de impresión con parámetros no optimizados.

Por su parte, el PLA se utiliza de forma extensiva en estructuras ligeras, configuraciones tipo sándwich y paneles alveolares, donde la integridad de las uniones entre capas resulta crítica para el comportamiento mecánico global. En estos casos, la termografía infrarroja activa permite monitorizar la continuidad interfacial y detectar la presencia de humedad infiltrada, un factor que degrada significativamente las propiedades mecánicas del biopolímero y acelera su envejecimiento [127]. La termografía pulsada (PT) se ha mostrado particularmente eficaz para la identificación de porosidades y vacíos de aire atrapados durante la extrusión del filamento, ya que el frente térmico se difunde de forma no uniforme al encontrar estas discontinuidades, cuya conductividad térmica es drásticamente inferior ($k \approx 0,02 \text{ W/m} \cdot \text{K}$) a la de la matriz polimérica circundante [97], [128]. En la Figura 2-50 se muestra un ejemplo de porosidad interna en PLA y contraste térmico asociado durante ensayo activo.

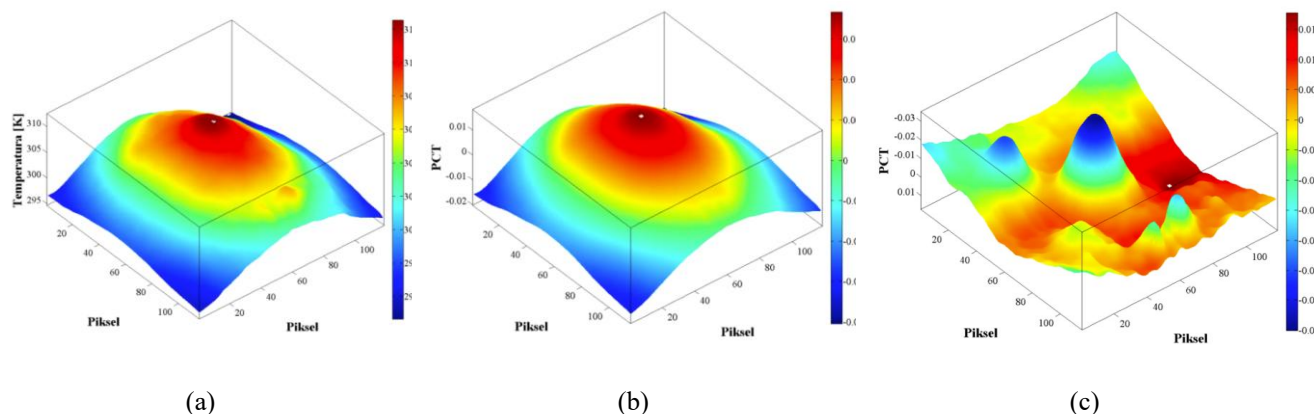


Figura 2-50 Comparación de los resultados obtenidos a partir de datos en bruto y mediante el uso de PCT a) termograma, b) EOF1, c) EOF2 (Extraído de [129])

En la Figura 2-51 se muestra una segmentación automática de defectos en polímero impreso (en este caso CFRP) mediante red neuronal convolucional aplicada a imagen termográfica. Esta capacidad de automatización, combinada con el uso de cámaras termográficas de bajo coste (*low-cost*), representa un avance significativo para la monitorización de la salud estructural (SHM) en el marco de la Industria 4.0, garantizando que los componentes poliméricos cumplan con las estrictas tolerancias mecánicas exigidas en sectores estratégicos como la defensa y la ingeniería naval [2], [24].

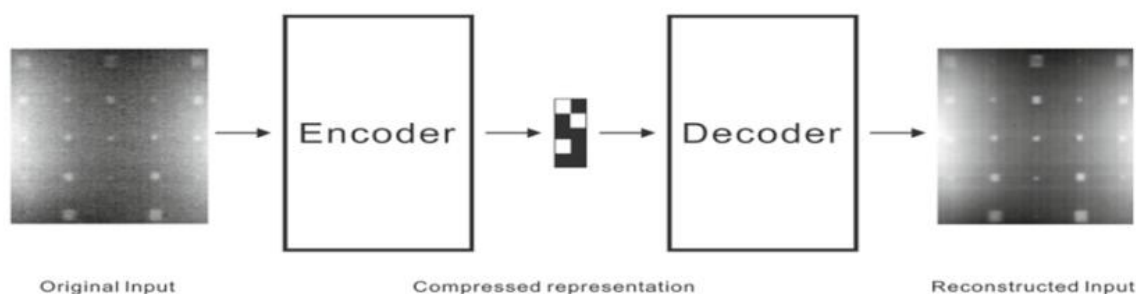


Fig. 10 Feature extraction of encoder and decoder format convolutional neural network

Figura 2-51 Extracción de características mediante una red neuronal convolucional con arquitectura codificador-decodificador (Extraído de [130])

3 DESARROLLO DEL TFG

3.1 Materiales usados

3.1.1 PETG-CF

El primer material seleccionado para la fabricación de las probetas experimentales fue el tereftalato de polietileno glicolado reforzado con fibra de carbono (PETG-CF). Se trata de un material compuesto que combina la procesabilidad de un polímero termoplástico con la mejora mecánica proporcionada por fibras cortas de carbono. Desde el punto de vista de la ingeniería mecánica, el PETG presenta ventajas relevantes como buena resistencia química, baja absorción de humedad y elevada estabilidad dimensional en comparación con materiales de uso común en impresión 3D como el PLA o el ABS. La incorporación de fibra de carbono incrementa notablemente la rigidez y el módulo elástico del material, reduciendo además la contracción térmica durante el proceso de fabricación. Esta estabilidad dimensional fue un factor determinante en el presente trabajo, ya que las probetas incluían defectos internos geoméricamente definidos cuya precisión debía mantenerse constante para garantizar la validez del estudio experimental.

Desde una perspectiva termofísica, el PETG-CF ofrece un comportamiento favorable para ensayos de termografía infrarroja activa. Aunque el material sigue siendo mayoritariamente aislante, la presencia de fibras de carbono contribuye a aumentar ligeramente su conductividad térmica efectiva en comparación con el PETG convencional. Estas fibras actúan como microcanales que facilitan la difusión del calor dentro del volumen del sólido, permitiendo que el frente térmico inducido se propague de manera más controlada. Esto resulta especialmente importante en la detección de discontinuidades internas, ya que los defectos modelados mediante agujeros ciegos con aire atrapado presentan una conductividad térmica mucho menor que la del material circundante, generando alteraciones locales en la tasa de enfriamiento superficial. Además, el PETG-CF posee una emisividad elevada ($\epsilon \approx 0,9$) debido a su acabado mate y opaco, lo que minimiza reflexiones parásitas y mejora la calidad radiométrica de los termogramas sin necesidad de aplicar tratamientos superficiales adicionales, sobre todo si tiene un acabado negro, ya que es el color con mayor valor de emisividad entre todos los abanicos de colores.

La fabricación de las probetas se realizó mediante tecnología FDM utilizando una impresora Bambu Lab. Tras el diseño en Autodesk Inventor, el modelo se exportó en formato STL y se procedió al proceso de laminado (*slicing*), ajustando parámetros como temperatura de extrusión, temperatura de la base y velocidad de deposición para garantizar una adecuada adhesión interlaminar. Debido a la naturaleza abrasiva de las fibras de carbono, fue necesario emplear boquillas de acero endurecido para evitar el desgaste prematuro y mantener un diámetro de extrusión constante. Asimismo, se controlaron cuidadosamente las condiciones de enfriamiento en las zonas donde se encontraban los defectos internos, evitando el colapso del material fundido sobre las cavidades. Este control permitió asegurar que las variaciones térmicas observadas durante los ensayos se debieran exclusivamente a la geometría de los defectos diseñados y no a imperfecciones derivadas del proceso de fabricación. El modelado 3D del prototipo y el proceso de la impresión 3D se detalla en los siguientes apartados.

3.1.2 PLA

El segundo material utilizado en la fase experimental fue el ácido poliláctico (PLA), un polímero termoplástico biodegradable derivado de recursos renovables que se ha consolidado como estándar en procesos de fabricación con filamento fundido (FFF). En este trabajo, la elección del PLA se fundamenta en su excelente estabilidad dimensional y su baja contracción térmica, lo que permite obtener piezas con tolerancias geométricas estrictas. Esta característica resulta esencial para asegurar el correcto ajuste entre la plancha base y la tapa superior, especialmente en un estudio donde se pretende simular defectos internos con profundidades bien definidas. El diseño de las probetas se realizó igualmente en Autodesk Inventor, incorporando seis defectos subsuperficiales en forma de agujeros ciegos distribuidos en tres

diámetros distintos, con el objetivo de evaluar la sensibilidad del sistema termográfico en función del tamaño de la discontinuidad.

Desde el punto de vista mecánico, el PLA presenta valores de módulo y resistencia inferiores a los del PETG-CF, aunque es más ligero; sin embargo, su comportamiento homogéneo y su facilidad de impresión lo convierten en un material adecuado para estudios experimentales comparativos. Termofísicamente, el PLA se caracteriza por una baja conductividad térmica, lo que favorece la persistencia de los gradientes térmicos inducidos durante los ensayos. Esta propiedad resulta ventajosa en termografía activa, ya que las perturbaciones generadas por defectos internos permanecen visibles durante un mayor intervalo de tiempo. La diferencia de difusividad térmica entre el PLA sólido y el aire contenido en los defectos actúa como una barrera que altera localmente la propagación del calor, produciendo contrastes detectables en la superficie. Durante el proceso de fabricación con impresora Bambu Lab, se optimizaron parámetros de laminado como densidad de relleno y resolución de capa para minimizar el efecto escalera y evitar porosidades no deseadas que pudieran interferir en la interpretación de los resultados.

3.1.3 Comparación entre PETG-CF y PLA

El PETG reforzado con fibra de carbono (PETG-CF) se posiciona como un material técnico orientado a aplicaciones funcionales que requieren mayor rigidez y estabilidad estructural, mientras que el ácido poliláctico (PLA) continúa siendo uno de los materiales más utilizados en impresión 3D debido a su facilidad de procesamiento, precisión dimensional y bajo coste [9], [138]. A continuación, se presentan las principales características comparativas de ambos materiales.

1) Ventajas PETG-CF frente al PLA

- Alta resistencia y rigidez estructural: ofrece una mayor resistencia mecánica y rigidez que el PLA, siendo adecuado para aplicaciones funcionales como componentes de robótica, drones o estructuras ligeras en ingeniería [9], [10], [138].
- Acabado superficial técnico: proporciona un mayor acabado mate que disimula las líneas de capa generadas durante el proceso de fabricación por deposición de filamento fundido (FFF).
- Baja contracción térmica: presenta menor deformación (*warping*) que el PLA convencional, lo que mejora la estabilidad dimensional en piezas de mayor tamaño [3], [93].
- Mayor resistencia térmica: puede soportar temperaturas de funcionamiento de aproximadamente 70–80 °C, superando los límites térmicos del PLA que es generalmente entre 50 y 60 °C.

2) Contras PETG-CF frente PLA

- Naturaleza abrasiva: las fibras de carbono pueden desgastar rápidamente las boquillas de latón de las impresoras 3D, por lo que se recomienda el uso de boquillas de acero endurecido.
- Facilidad de impresión: PLA es uno de los materiales más sencillos de imprimir, ya que no requiere cámaras impresoras cerradas ni configuraciones térmicas complejas para obtener buenos resultados.
- Material biodegradable: el PLA se obtiene a partir de recursos renovables, como el almidón de maíz, lo que lo convierte en una opción más sostenible dentro del mercado de filamentos [24], [63].
- Adhesión entre capas: la presencia de fibras puede reducir ligeramente la unión entre capas si los parámetros de impresión no se ajustan correctamente con respecto al PLA.
- Coste más elevado: el refuerzo con fibra de carbono aumenta el precio del filamento respecto a materiales poliméricos convencionales como el PLA.

3.2 Cámara infrarroja empleada: FLIR E85

La caracterización térmica y la adquisición de datos de las probetas fabricadas en PETG-CF y PLA se realizaron mediante la cámara termográfica FLIR E85, seleccionada como instrumento principal de medición debido a su elevada resolución espacial y precisión radiométrica. Este dispositivo utiliza un sensor de microbolómetro no refrigerado que opera en la banda espectral de infrarrojo de onda larga (*Long-Wave InfraRed*, LWIR), permitiendo detectar radiación térmica emitida por los objetos sin necesidad de contacto físico. La cámara dispone de un rango de medición aproximado entre $-20\text{ }^{\circ}\text{C}$ y $1200\text{ }^{\circ}\text{C}$, lo que la hace adecuada para aplicaciones industriales y experimentales donde es necesario analizar variaciones térmicas en materiales estructurales. Su resolución nativa de 384×288 píxeles proporcionan más de cien mil puntos de medición independientes por imagen, lo que permite obtener una representación detallada de la distribución de temperatura sobre la superficie de las probetas. Esta densidad de información resulta fundamental en un estudio basado en termografía activa, ya que facilita la identificación de pequeñas perturbaciones térmicas generadas por defectos subsuperficiales. En la Figura 3-1 se muestra el modelo utilizado para la realización del trabajo.



Figura 3-1 Cámara FLIR E85 (Extraído de [139])

Uno de los parámetros más importantes de la FLIR E85 es su sensibilidad térmica inferior a 40 mK, lo que significa que el sensor es capaz de detectar variaciones de temperatura extremadamente pequeñas. Esta característica resulta esencial para los ensayos realizados en este trabajo, donde los defectos internos generan diferencias térmicas muy sutiles durante la fase de calentamiento y enfriamiento del material. Cuando el flujo de calor se ve alterado por una cavidad interna llena de aire, se produce un contraste térmico en la superficie que puede ser captado por la cámara antes de que la difusión lateral del calor lo disipe completamente. Además, la cámara integra sistemas de asistencia que mejoran la calidad de la imagen, como el enfoque automático asistido por láser, que permite obtener imágenes nítidas manteniendo una distancia adecuada entre la cámara y la muestra. También incorpora la tecnología MSX (*Multi-Spectral Dynamic Imaging*), que superpone detalles visuales de la imagen visible sobre la imagen térmica, facilitando la localización precisa de las zonas de interés durante la inspección.

Durante el desarrollo experimental, la FLIR E85 se utilizó para registrar las secuencias térmicas generadas durante los ensayos de termografía activa, donde las probetas eran calentadas mediante las lámparas detalladas previamente. Los archivos radiométricos obtenidos fueron almacenados en la tarjeta SD integrada en el equipo y posteriormente transferidos al ordenador para su procesamiento. Gracias a la precisión del sistema, estimada en $\pm 2\text{ }^{\circ}\text{C}$ o $\pm 2\%$ de la lectura, la más precisa, fue posible obtener datos fiables para el análisis posterior mediante algoritmos de procesamiento de señal. Estas secuencias térmicas constituyeron la base experimental para aplicar los métodos TSR, PCT y SPCT, utilizados para

mejorar la relación señal-ruido y facilitar la detección de defectos internos en las probetas fabricadas mediante manufactura aditiva.

3.3 Preparación de las probetas

3.3.1 Modelado 3D mediante Autodesk Inventor

La primera fase del desarrollo experimental consistió en el diseño geométrico de las probetas utilizando el software de diseño asistido por ordenador Autodesk Inventor Professional. Este software CAD permite la creación de modelos tridimensionales paramétricos con gran precisión dimensional, lo cual resulta especialmente importante en aplicaciones de ingeniería mecánica donde la exactitud de las dimensiones influye directamente en el comportamiento físico del sistema analizado. En el presente trabajo, Autodesk Inventor se utilizó para diseñar el modelo completo de la pieza experimental que posteriormente será fabricada mediante impresión 3D. El diseño se compone de dos elementos principales: una base principal (Base ppl) y una tapa superior (Tapa). La base principal contiene los defectos internos simulados, mientras que la tapa permite cubrir dichas cavidades con el fin de reproducir una situación similar a la de defectos subsuperficiales reales en materiales sólidos, es decir, defectos internos sin estar en contacto con el entorno en ninguna de sus caras. En la Figura 3-2, se muestra el modelo tridimensional de la base de la probeta. Destacar que la configuración geométrica es la misma tanto para el material polimérico PETG-CF como para el PLA utilizados.

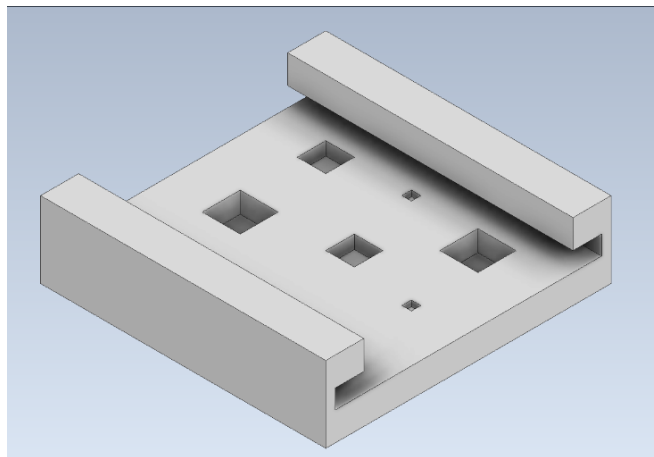


Figura 3-2 El modelo tridimensional de la base principal (Elaboración propia)

El proceso de diseño comenzó con la creación de un boceto bidimensional (boceto, *sketch*) sobre uno de los planos principales del sistema de coordenadas de Autodesk Inventor. En este boceto inicial se definió la geometría base de la pieza mediante un rectángulo de 15 cm $\times$ 15 cm, que representa la superficie principal de la probeta. Posteriormente, este boceto fue transformado en un sólido tridimensional mediante la herramienta Extrusión (Extrude), generando el volumen principal de la base. La altura de extrusión aplicada para la creación de la base fue de 1,5 cm, lo que proporciona suficiente espesor estructural para el estudio de la propagación térmica durante los ensayos experimentales. En la Figura 3-3 se muestra el boceto bidimensional utilizado para definir la geometría inicial de la base, mientras que en la Figura 3-4 se observa el proceso de extrusión que permite generar el volumen tridimensional correspondiente a la base principal.

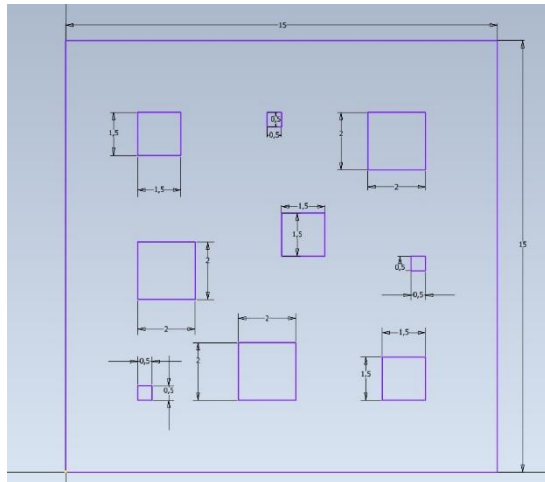


Figura 3-3 El boceto bidimensional de la base principal (Elaboración propia)

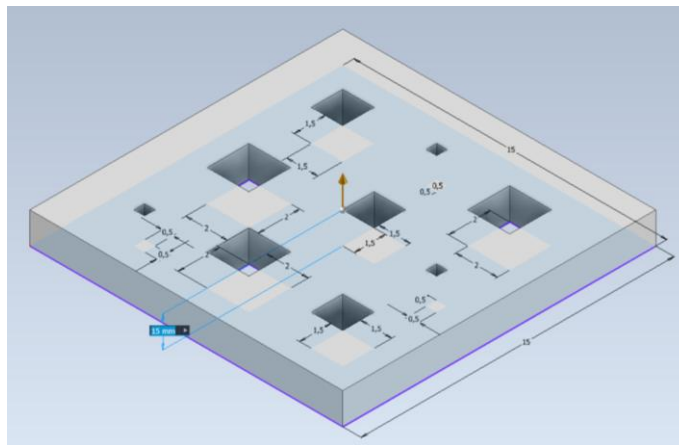


Figura 3-4 El proceso de extrusión (Elaboración propia)

Una vez creada la base sólida, se procedió al modelado de los defectos subsuperficiales que forman parte del objeto de estudio de este trabajo. Estos defectos fueron diseñados como cavidades cuadradas internas (agujeros ciegos) distribuidas sobre la superficie superior de la base. Con el objetivo de analizar la capacidad de detección del sistema de termografía infrarroja en función del tamaño del defecto, se definieron tres tamaños diferentes de cavidades. En concreto, se diseñaron defectos pequeños de $0,5 \text{ cm} \times 0,5 \text{ cm}$, defectos medianos de $1,5 \text{ cm} \times 1,5 \text{ cm}$ y defectos grandes de $2 \text{ cm} \times 2 \text{ cm}$. En total se tiene tres defectos por tipo, colocando en diagonal los defectos medianos, y en forma de tresbolillo tanto los defectos pequeños como los grandes, para tener la mayor variabilidad posible en cuanto a la ubicación se refiere en la base del prototipo. A cada tipo de defecto asimismo se aplicó una profundidad diferente: profundidad de 0,4, 0,8 y 1,2 cm para los defectos pequeños, medianos y grandes, respectivamente. Esos tres valores de profundidad se aplicaron para permitir simular defectos internos sin perforar completamente la pieza. Destacar que la cámara infrarroja se enfocará en la superficie inferior de la base, por tanto, se tendrá defectos a una profundidad real en cuanto al experimento se refiere de 0,3, 0,7 y 1,1 cm, recordando que el espesor de la base es de 1,5 cm. Por tanto, en este estudio se quiso valorar la prestación de la termografía infrarroja enfocando la cámara a tres escenarios diferentes, de menos a más exigente: defectos grandes ($2 \text{ cm} \times 2 \text{ cm}$) a profundidades de 0,3 cm, defectos medianos ($1,5 \text{ cm} \times 1,5 \text{ cm}$) a profundidades de 0,7 cm y defectos pequeños ($0,5 \text{ cm} \times 0,5 \text{ cm}$) a profundidades de 1,1 cm. En la Figura 3-5 se puede observar el resultado final tras aplicar la operación de extrusión que genera las cavidades internas.

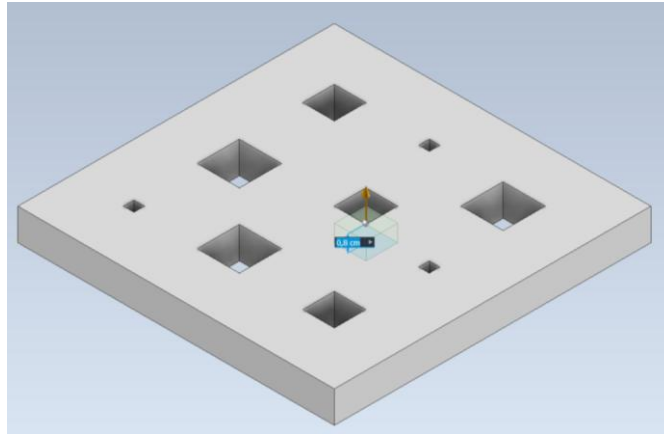


Figura 3-5 El resultado final tras aplicar la operación de extrusión que genera las cavidades internas (Elaboración propia)

Posteriormente, se añadieron dos paredes laterales elevadas situadas en lados opuestos de la base. Estas paredes se generaron a partir de nuevos bocetos creados sobre la superficie superior de la base y posteriormente extruidos para formar las estructuras laterales. La función de estas paredes es actuar como soporte para la tapa superior, permitiendo cerrar el volumen que contiene los defectos internos. Gracias a esta configuración, se crea una cámara de aire entre la base y la tapa, donde el aire queda atrapado en las cavidades que simulan los defectos. Desde el punto de vista térmico, este aire funciona como una barrera de baja conductividad térmica, lo que provoca perturbaciones en la difusión del calor cuando la pieza es sometida a un estímulo térmico externo. Estas perturbaciones generan contrastes térmicos en la superficie del material que pueden ser detectados mediante técnicas de termografía infrarroja activa. En la Figura 3-6 se muestra el boceto de las paredes laterales mientras en la Figura 3-7 se muestra el proceso de extrusión utilizado para generar las paredes laterales de la estructura.

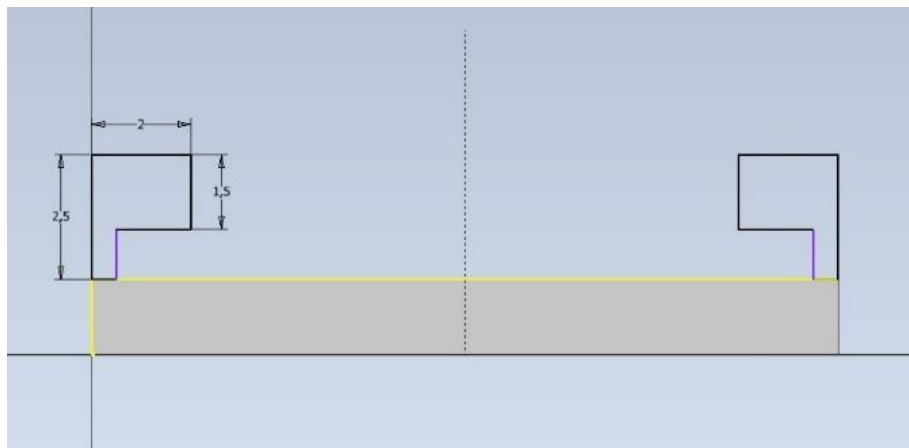


Figura 3-6 El boceto de las paredes laterales (Elaboración propia)

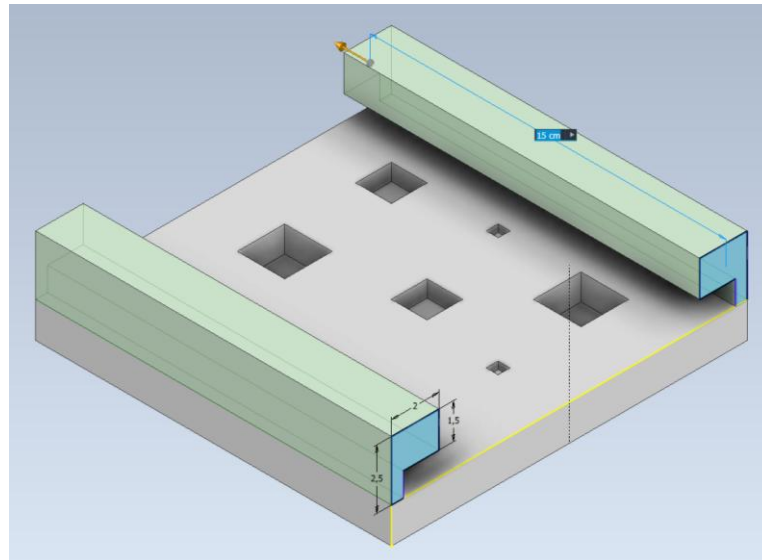


Figura 3-7 La extrusión generada de las paredes laterales (Elaboración propia)

Finalmente, se diseñó la tapa superior (Tapa) con el objetivo de cubrir completamente los defectos internos. La tapa fue modelada como una placa rectangular con dimensiones de 14 cm $\times$ 15 cm y un espesor a 1 cm, generada mediante una operación de extrusión a partir de un boceto rectangular simple. Esta tapa se apoya sobre las paredes laterales de la base, cerrando el volumen interno donde se encuentran las cavidades que simulan los defectos. De esta forma, los defectos quedan completamente ocultos bajo una capa sólida de material, reproduciendo condiciones más realistas de inspección no destructiva, donde las discontinuidades se encuentran bajo la superficie del componente. En la Figura 3-8 se muestra el boceto utilizado para el diseño de la tapa, mientras que en la Figura 3-9 se observa el modelo tridimensional final de la tapa una vez aplicada la operación de extrusión.

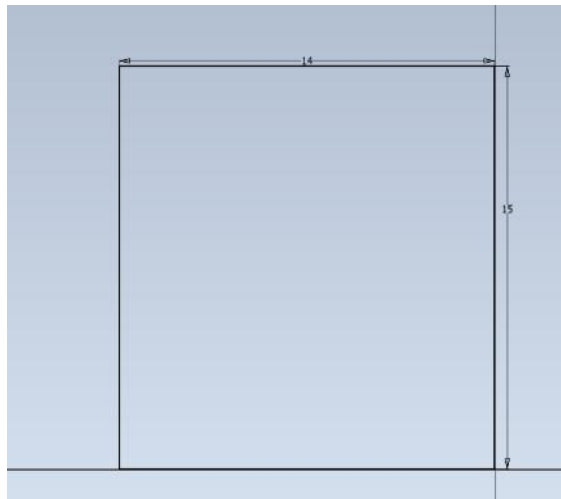


Figura 3-8 El boceto de la tapa (Elaboración propia)

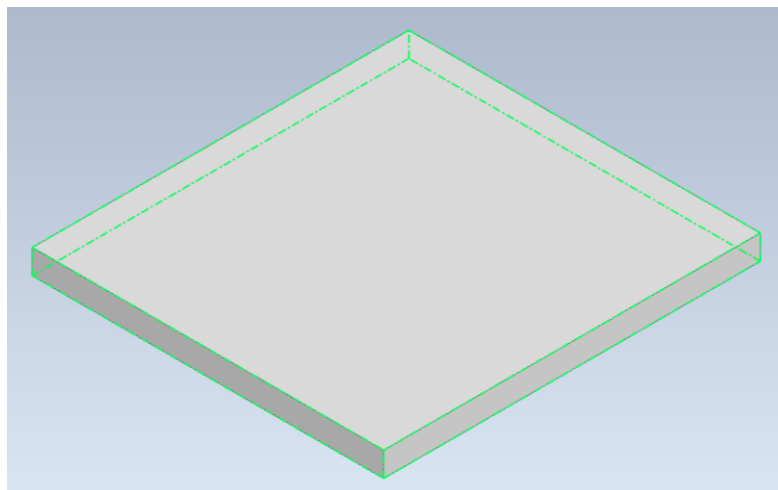
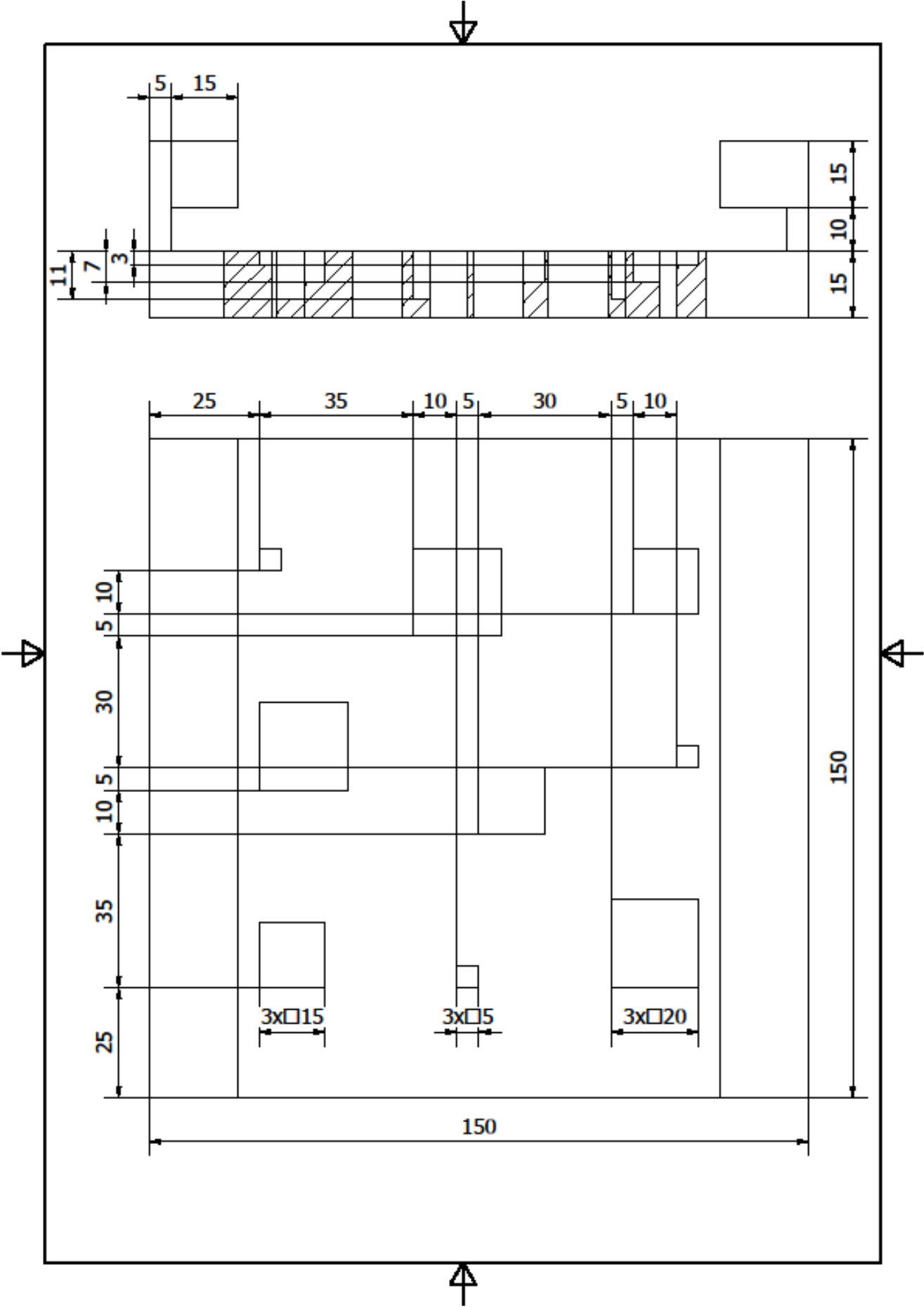


Figura 3-9 La extrusión generada de la tapa (Elaboración propia)

Una vez completado el diseño de todos los elementos, se realizó una revisión final del modelo para comprobar la coherencia geométrica y la correcta definición de todas las dimensiones. Tras esta verificación, los modelos fueron exportados en formato STL (*Standard Tessellation Language*), que es el formato estándar utilizado en procesos de fabricación aditiva. Estos archivos STL se utilizaron posteriormente en el software de laminado de la impresora 3D Bambu Lab para generar las trayectorias de impresión necesarias para la fabricación física de las probetas experimentales. En la Figura 3-10 se muestra los planos de la base (a) y de la tapa (b) (Sistema Europeo, escala 1:1 y unidades en mm).



(a)

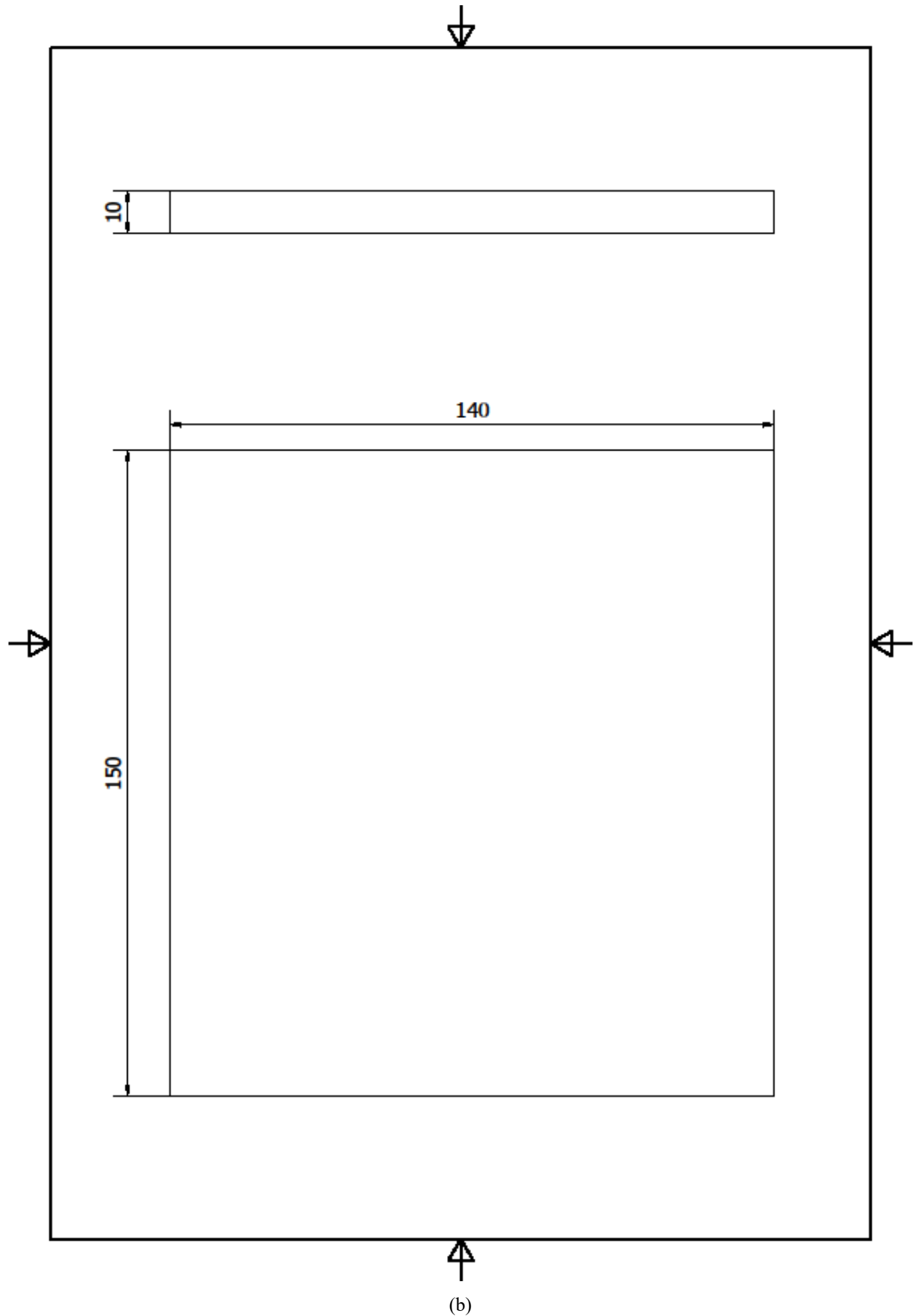


Figura 3-10 los planos de la base (a) y de la tapa (b) (Sistema Europeo, escala 1:1 y unidades en mm)
(Elaboración propia)

3.3.2 Impresión 3D mediante Bambu Lab

Una vez finalizado el diseño tridimensional de las piezas en Autodesk Inventor, el siguiente paso del desarrollo experimental consistió en la fabricación física de las probetas mediante tecnología de fabricación aditiva. Para ello se utilizó una impresora 3D Bambu Lab P1S, la cual trabaja mediante tecnología *Fused Filament Fabrication* (FFF). Este tipo de tecnología se basa en la deposición controlada de material termoplástico fundido capa por capa hasta formar el objeto tridimensional completo. La impresora Bambu Lab P1S fue seleccionada debido a su alta precisión, estabilidad en la deposición de material y compatibilidad con materiales técnicos reforzados, lo que la convierte en una herramienta adecuada para la fabricación de probetas utilizadas en estudios experimentales. En la Figura 3-11 se muestra la impresora 3D Bambu Lab P1S utilizada durante el proceso de fabricación de las probetas experimentales.



Figura 3-11 Impresora 3D Bambu Lab P1S empleada para la fabricación de las probetas experimentales mediante tecnología de deposición de filamento fundido (FFF) (Elaboración propia)

El proceso de fabricación comenzó con la exportación del modelo diseñado en Autodesk Inventor al formato STL (*Standard Tessellation Language*). Posteriormente, este archivo STL fue importado en el software de preparación de impresión Bambu Studio, donde se realizó la conversión al formato 3MF, formato utilizado por el ecosistema de impresión de Bambu Lab. Este formato permite almacenar no solo la geometría del modelo, sino también parámetros de impresión como el tipo de material, la orientación de la pieza, la altura de capa y otras configuraciones necesarias para el proceso de fabricación. En la Figura 3-12 se muestra la interfaz del software Bambu Studio durante el proceso de preparación del modelo antes de iniciar la impresión.

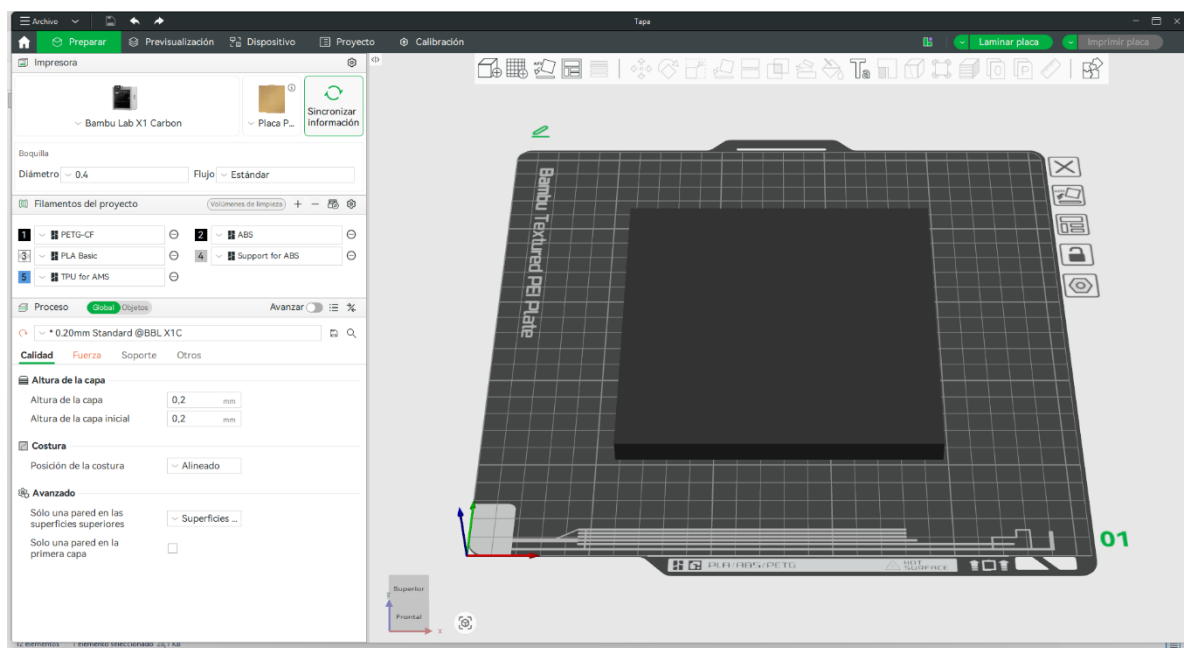


Figura 3-12 La tapa en el software Bambu Studio antes del proceso de laminado (Elaboración propia)

Una vez importado el modelo, se procedió a definir los parámetros de impresión necesarios para garantizar una correcta fabricación de las piezas. Entre los parámetros configurados destacan la altura de capa de 0,2 mm, el diámetro de boquilla de 0,4 mm y la correcta orientación del modelo sobre la plataforma de impresión. Estos parámetros permiten obtener una buena resolución dimensional y asegurar una correcta adhesión entre las capas depositadas. En la Figura 3-13 se muestra la etapa de preparación y laminado del modelo correspondiente a la tapa, donde el software genera la simulación de las trayectorias de impresión que seguirá el extrusor durante el proceso de fabricación.

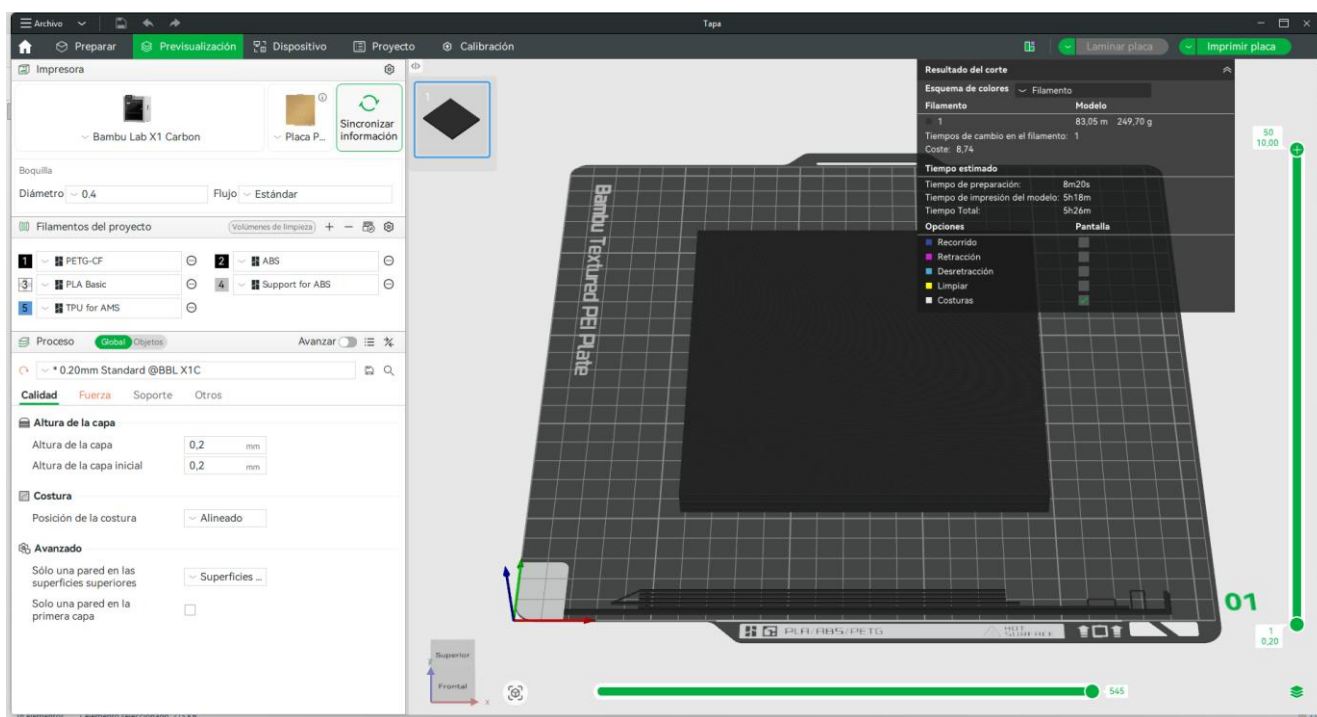


Figura 3-13 Las trayectorias generadas por el software durante el proceso de laminado (Elaboración propia)

Posteriormente, se realizó el mismo procedimiento para la preparación de la base principal (Base ppl). En este caso, debido a la mayor complejidad geométrica de la pieza, el software generó una

trayectoria de impresión más extensa, lo que se refleja en un mayor tiempo estimado de fabricación. En la Figura 3-14 se muestra la simulación del proceso de laminado de la base principal, donde se puede observar el modelo colocado sobre la plataforma de impresión antes de iniciar el proceso de fabricación.

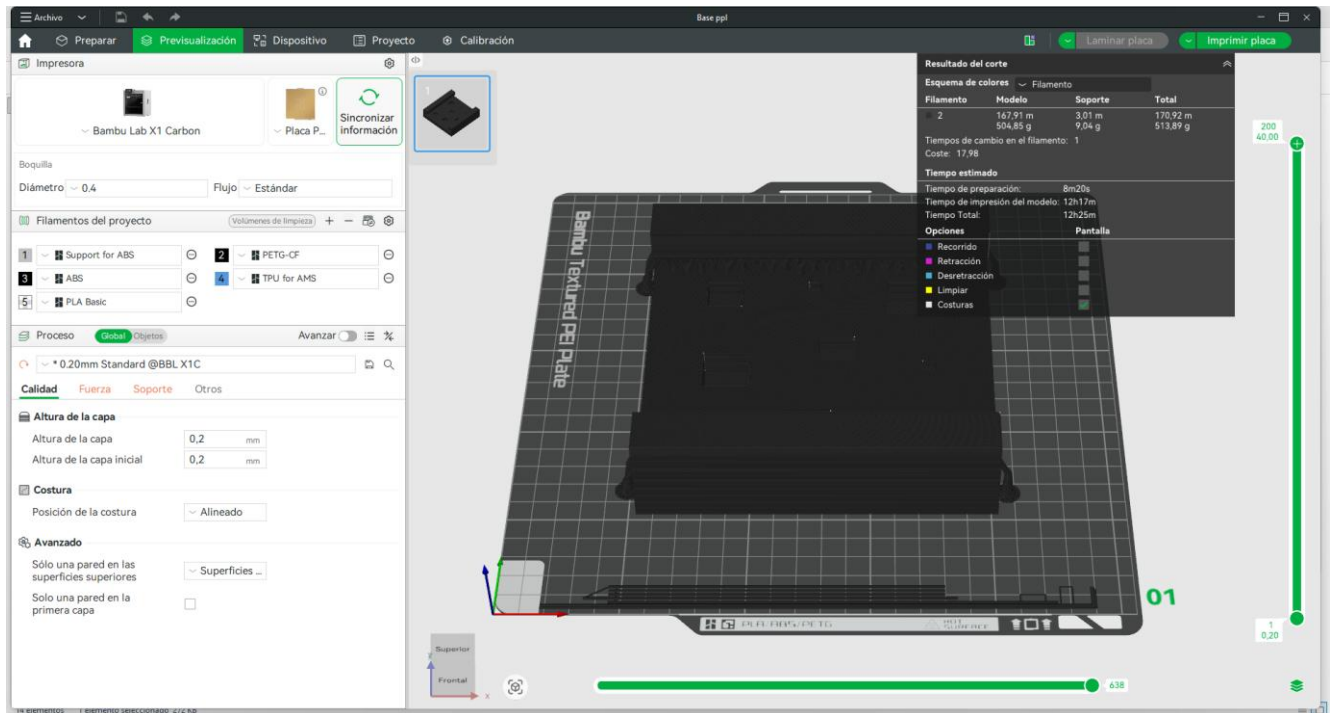


Figura 3-14 Impresión de la base principal (Elaboración propia)

Después de completar el proceso de impresión 3D, las piezas fueron retiradas cuidadosamente de la plataforma de fabricación y se realizó una inspección visual para verificar la calidad de impresión y la fidelidad geométrica respecto al modelo diseñado en el software CAD. Esta verificación es importante para asegurar que los defectos internos modelados se hayan fabricado correctamente y que no existan irregularidades adicionales producidas durante el proceso de impresión que puedan afectar a los resultados del análisis termográfico posterior.

En la Figura 3-15 se muestra la probeta Base principal observada desde la superficie posterior, donde se puede apreciar la cara externa de la pieza con una geometría plana y uniforme. Esta superficie corresponde a la zona donde posteriormente se realizará la observación mediante la cámara termográfica durante los ensayos experimentales. Resaltar que esa textura aceitosa central debido a que era la superficie de apoyo durante la impresión 3D no afecta en las medidas térmicas, ya que su conductividad térmica es igual o prácticamente igual al resto del material.

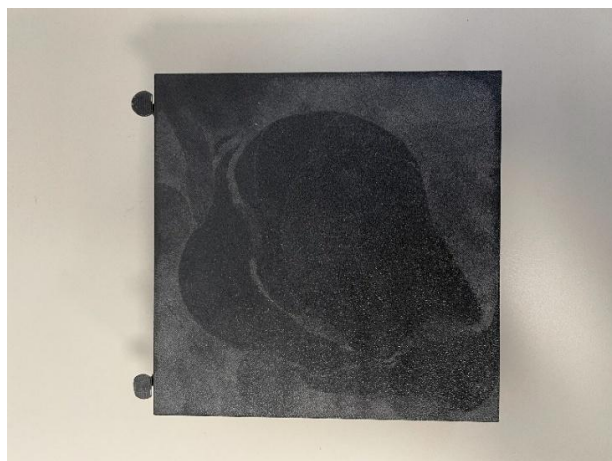


Figura 3-15 Probeta Base principal vista desde la superficie posterior (Elaboración propia)

Por otro lado, en la Figura 3-16 se presenta la cara posterior de la misma pieza Base principal, donde pueden observarse claramente los seis defectos internos simulados. Estos defectos corresponden a cavidades internas diseñadas con tres tamaños diferentes (pequeño, mediano y grande), cuyo objetivo es evaluar la capacidad de detección de la técnica de termografía infrarroja frente a discontinuidades de distinto volumen.

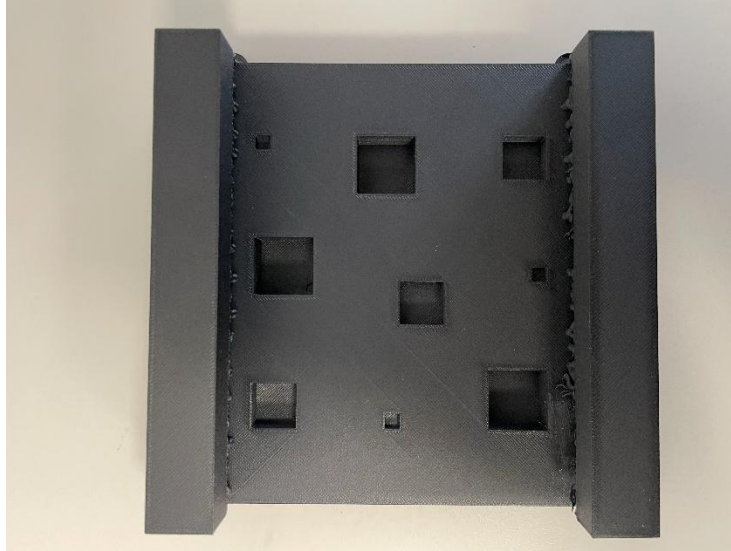


Figura 3-16 Probeta Base principal vista desde la superficie frontal (Elaboración propia)

Finalmente, en la Figura 3-17 se muestra la pieza correspondiente a la tapa superior (Tapa), la cual se utiliza para cubrir la base principal durante algunos de los ensayos experimentales, ya que se realiza las mediciones con las cavidades internas, y también con las cavidades con sus caras posteriores descubiertas con el fin de comparar las prestaciones de la termografía infrarroja activada en ambos escenarios. La presencia de esta tapa permite simular condiciones más realistas de defectos internos profundos, ya que el flujo de calor debe atravesar una capa adicional de material antes de alcanzar la superficie observada por la cámara termográfica.



Figura 3-17 Probeta Tapa vista desde la superficie frontal (Elaboración propia)

3.4 Campaña de las mediciones termográficas

En el protocolo experimental, las probetas tanto de PETG-CF como PLA fueron sometidas a ensayos de termografía infrarroja activa utilizando la cámara FLIR 85 comentada en el apartado previo y lámparas de dos potencias distintas, una de 500 W siendo una lámpara halógena con configuración de foco, y la otra una lámpara infrarroja de potencia 2000 W con configuración de estufa. Se realizaron cuatro configuraciones para cada polímero:

- 1) Configuración 1: Inspección de la base sin tapa con la lámpara halógena.
- 2) Configuración 2: Inspección de la base con tapa con la lámpara halógena.
- 3) Configuración 3: Inspección de la base sin tapa con la lámpara infrarroja.
- 4) Configuración 4: Inspección de la base con tapa con la lámpara infrarroja.

Cada ensayo tuvo una duración de 2 horas con una frecuencia de adquisición de 10 FPS (*frames por second*), generando una matriz tridimensional de datos espaciales y temporales para su posterior análisis. Se ha iniciado con un calentamiento durante 30 min, y, por tanto, un posterior enfriamiento de 1h y 30 min. Debido a posibles variaciones en la emisividad, tanto la bobina del PETG-CF como del PLA son de filamento de color negro mate con valores cercanos a $\varepsilon \approx 0,98$.

Para la correcta adquisición de los datos termográficos, se definió una configuración geométrica controlada del sistema experimental. La cámara termográfica, la probeta y la fuente de excitación térmica se situaron alineadas frontalmente, formando una línea recta entre los tres componentes, lo que corresponde a una configuración de termografía infrarroja activa en modo transmisión. En esta disposición, la fuente térmica se coloca enfocando la superficie posterior de la muestra mientras que la cámara se ubica en el lado opuesto, enfocando la superficie frontal de muestra y permitiendo registrar la distribución de temperatura generada por la propagación del calor a través del material. Durante los ensayos, la distancia entre la cámara termográfica y la probeta se fijó en 100 cm, mientras que la distancia entre la probeta y la lámpara de excitación térmica se estableció en 50 cm. Esta configuración se mantuvo constante en todos los ensayos con el objetivo de garantizar la repetibilidad de las mediciones y la comparabilidad de los resultados obtenidos.

Para asegurar la estabilidad de la adquisición térmica durante los ensayos prolongados, la cámara termográfica fue montada sobre un trípode, lo que permitió mantener una posición fija y evitar vibraciones o desplazamientos que pudieran introducir ruido en la secuencia de imágenes térmicas capturadas. La utilización del trípode resulta especialmente importante en experimentos de larga duración, como en este caso, donde cada ensayo se extendió durante dos horas, ya que cualquier variación en la posición de la cámara podría afectar a la coherencia espacial de los datos registrados.

A continuación, se muestra el montaje experimental empleado durante la campaña de mediciones termográficas. En la Figura 3-18 se presenta la configuración del sistema utilizando la lámpara halógena de 500 W, mientras que en la Figura 3-19 se muestra el montaje correspondiente al ensayo realizado con la lámpara infrarroja de 2000 W.



Figura 3-18 Montaje experimental utilizando la lámpara halógena de 500 W (Elaboración propia)



Figura 3-19 Montaje experimental utilizando la lámpara infrarroja de 2000 W (Elaboración propia)

Una vez finalizada la fase de adquisición, las secuencias de vídeo termográficas fueron procesadas para aislar exclusivamente la región correspondiente a la probeta, es decir, su superficie

frontal. Para ello se utilizó el software de análisis térmico FLIR Tools [140], el cual permite visualizar, analizar y exportar los datos radiométricos registrados por la cámara infrarroja. En esta etapa se empleó la herramienta de recorte de región de interés (*Region of Interest*, ROI) con el fin de eliminar el fondo de la imagen y conservar únicamente la zona correspondiente a la probeta analizada. Posteriormente, los datos térmicos fueron exportados en formato *Comma Separated Values* (CSV), lo que permitió convertir la secuencia de termogramas en una matriz de datos numéricos adecuada para su posterior tratamiento mediante algoritmos de procesamiento de señal. Figura 3-20 se muestra un ejemplo del proceso de recorte de la región de interés de la probeta utilizando el software FLIR Tools.

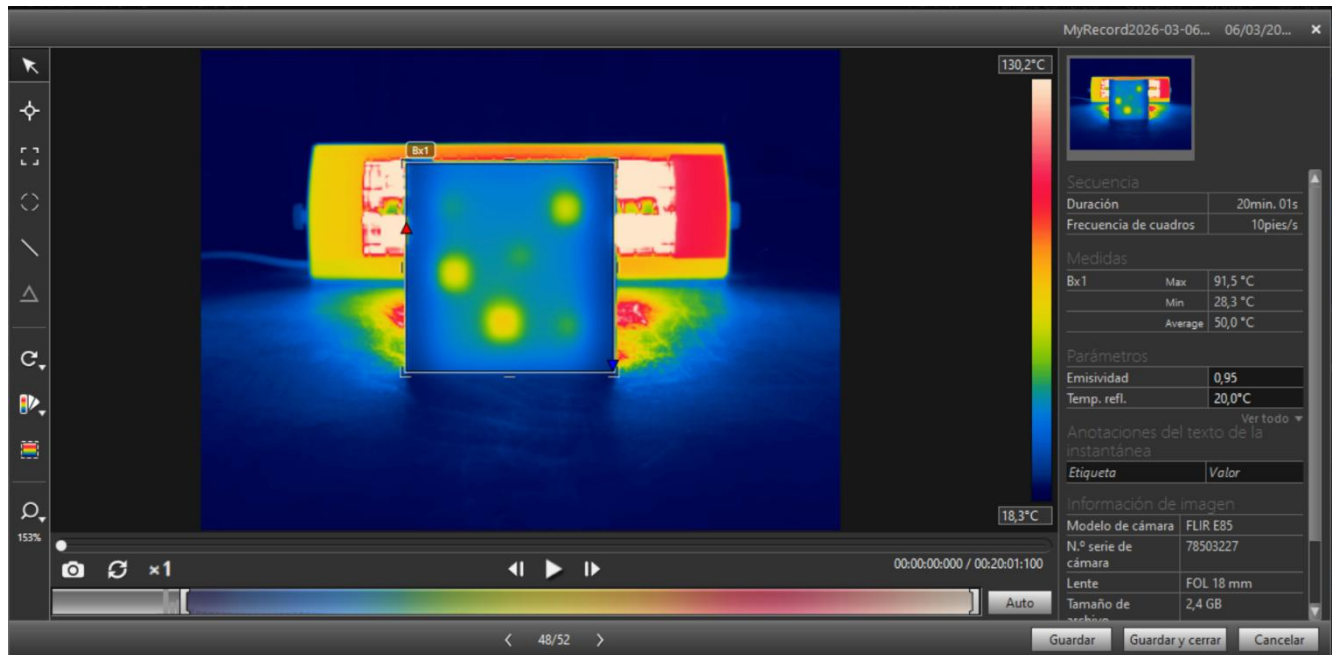


Figura 3-20 Recorte de la región de interés de la probeta utilizando el software FLIR Tools (Elaboración propia)

Esta preparación aseguró que los datos obtenidos fueran adecuados para la aplicación posterior de los algoritmos TSR, PCT y SPCT, permitiendo una comparación objetiva entre el comportamiento térmico del PLA y del PETG-CF en la detección de defectos internos.

3.5 Aplicaciones de los algoritmos de procesamiento de imagen

Para la aplicación de los algoritmos de procesamiento de imagen se ha utilizado el entorno de programación Google Colab [141], el cual permite ejecutar código en Python dentro de un entorno interactivo basado en la nube. Esta plataforma facilita el manejo de grandes volúmenes de datos y la implementación de bibliotecas científicas especializadas para el análisis numérico y el procesamiento de imágenes térmicas.

A partir de los datos exportados previamente desde el software FLIR Tools en formato CSV, se desarrolló un conjunto de scripts en Python con el objetivo de procesar las secuencias termográficas obtenidas durante la campaña experimental, cada una por separado. Estos scripts permiten organizar los datos térmicos, realizar operaciones de preprocesado y aplicar diferentes algoritmos de análisis utilizados en termografía infrarroja activa para mejorar la detección de defectos internos.

En particular, en este trabajo se han implementado los algoritmos TSR, PCT y SPCT, los cuales permiten mejorar la relación señal-ruido de las secuencias térmicas y resaltar las anomalías asociadas a discontinuidades internas en el material.

A continuación, se describe el funcionamiento de cada uno de los bloques de código desarrollados para el procesamiento de los datos experimentales. Con el fin de facilitar la comprensión del

procedimiento seguido, el código completo utilizado en este trabajo se presenta en el Anexo III, mientras que en este apartado se explica el objetivo y la función de cada bloque dentro del flujo general de procesamiento (ver Figura 3-21 que es el diagrama de flujo del código, diagrama obtenido mediante la herramienta *Canvas* de *Gemini*, en donde la explicación de cada bloque está a continuación de la figura).

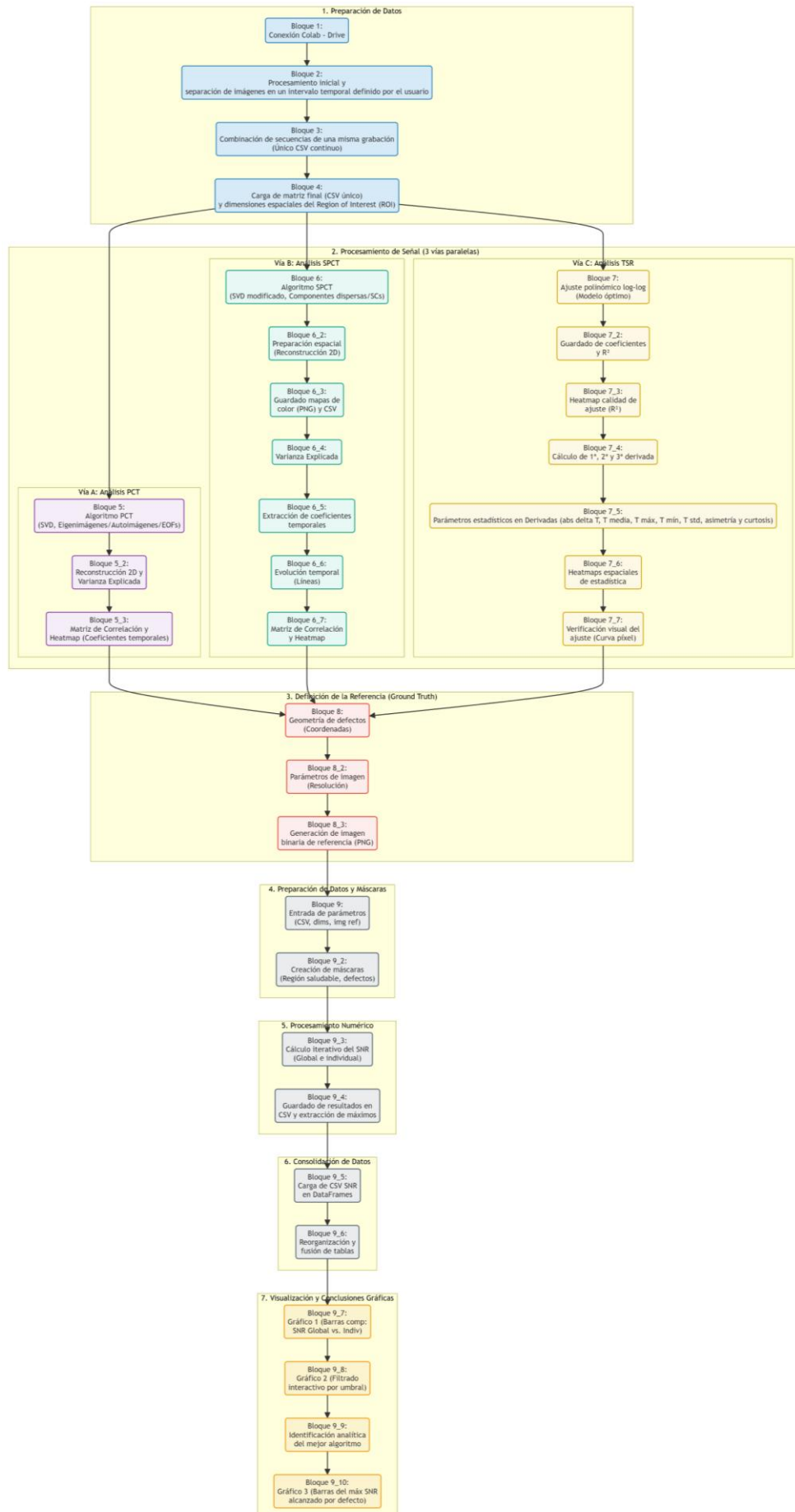


Figura 3-21 El diagrama de flujo del código (Elaboración propia)

- El **Bloque 1** tiene como objetivo establecer la conexión entre el entorno de trabajo de Google Colab y el sistema de almacenamiento Google Drive. Esta etapa permite acceder a los archivos y carpetas donde se encuentran almacenados los datos experimentales utilizados en el estudio, facilitando su carga y utilización en las etapas posteriores del procesamiento de las imágenes termográficas.
- El **Bloque 2** se encarga del procesamiento inicial de los datos térmicos obtenidos a partir de los archivos CSV exportados desde el software de la cámara infrarroja. En esta etapa se realiza la lectura del archivo de entrada y la organización de los datos en función de la resolución de ROI aplicado y de los parámetros experimentales definidos durante la adquisición (FPS).

Posteriormente, el script separa las imágenes térmicas individuales que se encuentran almacenadas de forma consecutiva en el archivo CSV, permitiendo reconstruir cada termograma correspondiente a un instante temporal específico, designado por el usuario. A partir de estas imágenes, se generan representaciones visuales en formato de imagen y se organiza la información numérica en una matriz estructurada donde cada columna representa el estado térmico de la probeta en un momento determinado, donde las filas y columnas del ROI de un instante temporal determinado se apilan verticalmente, es decir, la primera fila se convierte en columna, y justo tras ella por la parte inferior se concatena con la segunda fila rotada también y así sucesivamente.

El resultado final de este bloque es un nuevo archivo CSV procesado, en el que los datos térmicos quedan organizados de forma adecuada para su análisis posterior. Esta estructura permite aplicar de manera eficiente los algoritmos de procesamiento de imagen utilizados en este trabajo, como TSR, PCT y SPCT, facilitando el estudio de la evolución térmica del material a lo largo del tiempo.

- El **Bloque 3** tiene como objetivo combinar los diferentes archivos CSV generados durante una misma grabación en modo cronológico, ya que la grabación durante la campaña experimental, aunque fue de 2h, se ha tenido que fragmentar en grabaciones de 20 minutos para no superar el límite de RAM del entorno de ejecución (del *Google Colab*).

En esta etapa, los archivos CSV almacenados en una misma carpeta son identificados y cargados de forma automática. Posteriormente, las matrices de datos se concatenan horizontalmente, de manera que las columnas de cada archivo se incorporan sucesivamente a una única matriz final. En esta matriz resultante, cada columna representa una imagen térmica correspondiente a un instante temporal diferente, manteniendo un orden temporal continuo entre las diferentes grabaciones.

Finalmente, la matriz combinada se guarda como un nuevo archivo CSV, que contiene la secuencia completa de datos térmicos organizada en función del tiempo. Este archivo constituye la base de datos final utilizada en las etapas posteriores del análisis, permitiendo aplicar de forma eficiente los algoritmos de procesamiento de imagen empleados en este trabajo.

- El **Bloque 4** tiene como finalidad cargar la matriz de datos térmicos previamente generada y combinada en las etapas anteriores del procesamiento. Para ello, se accede al archivo CSV final, que contiene la secuencia completa de datos térmicos organizados en función del tiempo.

Una vez localizada la ruta del archivo en el sistema de almacenamiento, la matriz es importada al entorno de trabajo como una estructura de datos adecuada para su análisis posterior. Este paso permite verificar que los datos han sido cargados correctamente y que la matriz resultante mantiene la estructura esperada, donde cada columna representa la información térmica correspondiente a un instante temporal específico, designado por el usuario.

Finalmente, se introducen las dimensiones originales del ROI (por tanto, en cada fragmentación de la grabación original de 2h, el ROI utilizado tiene que ser el mismo tanto en ubicación como en dimensiones) utilizado en la grabación que se esté analizando en el código en ese momento. Esta información es necesaria para reconstruir la distribución espacial de los píxeles y poder aplicar correctamente los algoritmos de análisis termográfico, como PCT, SPCT y TSR, los cuales requieren conocer la geometría original de los termogramas para interpretar adecuadamente la evolución térmica del material.

- El **Bloque 5** corresponde a la aplicación del algoritmo PCT, una técnica ampliamente utilizada en termografía infrarroja activa para mejorar la detección de defectos mediante el análisis estadístico de la evolución térmica registrada durante el experimento.

En primer lugar, la matriz de datos térmicos obtenida en el bloque anterior se transforma en una estructura numérica adecuada para el análisis matemático. Posteriormente, los datos se estandarizan, de modo que todas las variables presenten una media nula y una desviación estándar unitaria. Este procedimiento permite evitar que diferencias de escala en los valores de temperatura influyan en el análisis estadístico y garantiza que todas las variaciones térmicas tengan el mismo peso durante el procesamiento.

A continuación, se aplica la Descomposición en Valores Singulares (SVD), que constituye el núcleo matemático del método PCT. Este procedimiento descompone la matriz de datos en tres componentes principales: las Eigenimágenes o Funciones Ortogonales Empíricas (EOF), que representan los patrones espaciales de variación térmica; los valores singulares, que indican la importancia relativa de cada EOF; y los componentes principales temporales, que describen la evolución de estos patrones a lo largo del tiempo. Finalmente, el sistema permite seleccionar y guardar las eigenimágenes más relevantes, las cuales suelen concentrar la mayor parte de la información relacionada con la presencia de defectos internos en el material, hasta aproximadamente la eigenimagen 6 según el estado del arte.

- El **Bloque 5_2** tiene como objetivo almacenar y representar los resultados obtenidos tras la aplicación del algoritmo PCT. En esta etapa se gestionan los principales elementos generados por la descomposición matemática realizada previamente: las eigenimágenes o Funciones Ortogonales Empíricas (EOF), los valores singulares y los componentes temporales asociados a cada modo térmico.

En primer lugar, las eigenimágenes seleccionadas se reconstruyen con las dimensiones originales de la ROI para poder representarlas visualmente. Estas imágenes permiten identificar los patrones espaciales más relevantes de variación térmica en la probeta, facilitando la localización de posibles anomalías o defectos internos. Además, los datos numéricos correspondientes a estas eigenimágenes se almacenan en archivos CSV, lo que permite realizar análisis posteriores de carácter cuantitativo (véase desde el **Bloque 9**). Posteriormente, se guardan los valores singulares, los cuales indican la importancia relativa cada EOF obtenido durante el análisis. A partir de estos valores se calcula el porcentaje de varianza explicada por cada modo, permitiendo identificar qué eigenimágenes contienen la mayor parte de la información térmica relevante. Esta información se presenta tanto en formato de tabla como mediante representaciones gráficas.

Finalmente, se almacenan los componentes temporales, que describen la evolución de cada patrón espacial (cada EOF) a lo largo del tiempo. De este modo, se obtiene una representación continua de la evolución térmica durante todo el ensayo experimental.

En conjunto, este bloque permite organizar y visualizar los resultados del análisis PCT, proporcionando tanto representaciones gráficas como datos numéricos que facilitan la interpretación de los fenómenos térmicos asociados a la presencia de defectos en los materiales estudiados.

- El **Bloque 5_3** tiene como objetivo analizar la relación existente entre las distintas componentes temporales de las EOFs obtenidas mediante el análisis PCT. Para ello, se calcula una matriz de correlación de Pearson entre los diferentes modos temporales, lo que

permite evaluar el grado de dependencia o independencia entre los patrones térmicos identificados durante el procesamiento de los datos.

A partir de esta matriz de correlación se genera una representación visual mediante un mapa de calor (*heatmap*), que facilita la interpretación de los resultados. En esta representación gráfica es posible identificar de forma rápida las correlaciones positivas o negativas entre las distintas componentes temporales, así como detectar posibles redundancias o relaciones significativas entre los modos principales.

Finalmente, tanto la matriz de correlación como su representación gráfica se almacenan en archivos de salida para su posterior análisis. Esta información resulta útil para comprender la estructura de los datos térmicos obtenidos y para evaluar la independencia de los distintos patrones espaciales y temporales detectados durante el análisis termográfico.

- El **Bloque 6** corresponde a la aplicación del algoritmo SPCT, una variante avanzada del análisis de componentes principales utilizada en termografía infrarroja para mejorar la identificación de patrones térmicos relevantes. Este método introduce un criterio de dispersión (*sparsity*) en la descomposición de los datos, lo que permite obtener componentes más independientes entre ellos y facilitar la detección de anomalías térmicas asociadas a defectos internos.

En primer lugar, se define el número de componentes dispersas que se desea extraer del conjunto de datos térmicos. Posteriormente, los datos previamente estandarizados se reorganizan para adaptarse al formato requerido por el algoritmo, de modo que cada instante temporal se considere una observación y cada píxel de la imagen térmica una característica.

A continuación, se aplica el algoritmo SPCT, que descompone la matriz de datos en dos tipos de resultados principales: los componentes temporales dispersos, que describen la evolución temporal de los patrones térmicos identificados, y los componentes espaciales dispersos, que representan las distribuciones espaciales de temperatura asociadas a dichos patrones. Gracias a la naturaleza dispersa del método, estos patrones suelen resaltar de forma más clara las regiones de interés relacionadas con posibles defectos en el material analizado.

- El **Bloque 6_2** tiene como objetivo preparar los resultados espaciales obtenidos mediante el algoritmo SPCT para su posterior visualización y almacenamiento. En esta etapa se utilizan las componentes espaciales dispersas calculadas en el bloque anterior (los equivalentes a los EOFs), que representan los patrones térmicos localizados identificados durante el análisis.

En primer lugar, cada una de estas componentes espaciales, inicialmente almacenadas en formato vectorial, se transforma nuevamente a las dimensiones originales de la ROI utilizada durante el experimento. Este proceso permite reconstruir cada componente en forma de imagen bidimensional, facilitando su interpretación visual como patrones espaciales de variación térmica sobre la superficie de la probeta.

Finalmente, las componentes espaciales reconstruidas se almacenan en una estructura de datos que permitirá su representación gráfica y su exportación en etapas posteriores del análisis (mismo protocolo con respecto al PCT para su aplicación desde el **Bloque 9**). Este procedimiento es fundamental para poder visualizar los patrones térmicos dispersos obtenidos mediante el método SPCT y evaluar su capacidad para resaltar defectos internos en los materiales analizados.

- El **Bloque 6_3** tiene como finalidad almacenar y representar los componentes espaciales dispersos obtenidos mediante el algoritmo SPCT (tras la preparación en el **Bloque 6_2**). Estos componentes corresponden a los patrones térmicos localizados identificados durante el análisis y permiten resaltar regiones de interés relacionadas con posibles defectos en el material.

En primer lugar, se crea una estructura organizada de carpetas destinada a almacenar los resultados del análisis SPCT. Dentro de esta estructura se genera una carpeta principal para

los resultados y una subcarpeta específica para las imágenes correspondientes a los componentes espaciales dispersos.

Posteriormente, cada componente espacial disperso reconstruido se representa gráficamente en forma de imagen bidimensional, utilizando un mapa de color que permite visualizar la distribución espacial de los valores asociados a cada componente. Estas representaciones se guardan en formato PNG (*Portable Network Graphics*), lo que facilita su inclusión posterior en el análisis y discusión de resultados (mismo proceso con respecto a cómo se ha hecho con los EOFs).

Finalmente, los datos numéricos correspondientes a los componentes espaciales dispersos también se almacenan en un archivo CSV, permitiendo su análisis cuantitativo posterior. De esta forma, el bloque proporciona tanto una representación visual de los patrones térmicos detectados como una base de datos estructurada para futuras etapas de análisis (desde **Bloque 9**).

- El **Bloque 6_4** tiene como objetivo analizar la varianza explicada por cada uno de los componentes dispersos obtenidos mediante el algoritmo SPCT. Este análisis permite evaluar la importancia relativa de cada componente identificado durante el proceso de descomposición de los datos térmicos.

Dado que el método SPCT no proporciona directamente el porcentaje de varianza explicada como ocurre en el análisis PCT clásico, en este bloque se calcula una estimación basada en la varianza de las componentes temporales dispersas. A partir de estos valores se determina el porcentaje de contribución de cada componente respecto a la varianza total del conjunto de datos.

Posteriormente, los resultados se almacenan en un archivo CSV, donde se recogen los porcentajes de varianza explicada por cada componente disperso. Además, se genera un gráfico de barras que representa visualmente la contribución de cada componente al comportamiento térmico global observado durante el experimento.

Finalmente, tanto los datos numéricos como las representaciones gráficas se guardan en una carpeta específica dentro del directorio de resultados del análisis SPCT. Este procedimiento permite identificar de forma clara qué componentes dispersos contienen la mayor parte de la información relevante para la detección de defectos en los materiales analizados.

- El **Bloque 6_5** tiene como objetivo extraer y almacenar los coeficientes temporales dispersos obtenidos mediante el algoritmo SPCT. Estos coeficientes describen la evolución temporal de cada uno de los componentes espaciales identificados durante el análisis.

En primer lugar, los datos correspondientes a los coeficientes temporales se organizan en una estructura tabular adecuada para su tratamiento posterior. Cada columna representa la evolución temporal de un componente disperso específico (como recordatorio, el equivalente a los EOFs del PCT), mientras que cada fila corresponde a un instante temporal del experimento.

Posteriormente, se incorpora una columna adicional que contiene el tiempo acumulado en minutos, lo que permite relacionar directamente la evolución de los coeficientes con la secuencia temporal del ensayo termográfico. Esta información resulta especialmente útil para analizar la dinámica térmica del material durante las fases de calentamiento y enfriamiento.

Finalmente, todos los datos se almacenan en un archivo CSV dentro de una carpeta específica destinada a los resultados del análisis SPCT. Este archivo proporciona una base de datos estructurada que facilita el estudio cuantitativo de la evolución temporal de los patrones térmicos identificados en las probetas analizadas.

- El **Bloque 6_6** tiene como objetivo representar gráficamente la evolución temporal de los coeficientes dispersos obtenidos mediante el algoritmo SPCT. Estos coeficientes describen

cómo varía la contribución de cada componente disperso a lo largo del tiempo durante el ensayo termográfico.

Para ello, se utilizan los datos organizados previamente en el bloque anterior, donde cada columna del conjunto de datos corresponde a un componente disperso específico y cada fila representa un instante temporal del experimento. A partir de esta información se genera una gráfica de líneas en la que el eje horizontal indica el tiempo transcurrido en minutos, mientras que el eje vertical representa la amplitud de cada coeficiente temporal.

Esta visualización permite analizar el comportamiento dinámico de los distintos componentes identificados por el algoritmo SPCT y facilita la interpretación de los patrones térmicos asociados a la presencia de posibles defectos en el material. Finalmente, la gráfica generada se guarda en formato PNG dentro de la carpeta correspondiente a los resultados del análisis, permitiendo su utilización posterior en la sección de resultados y discusión del trabajo.

- El **Bloque 6_7** tiene como finalidad analizar la relación existente entre las diferentes componentes temporales dispersas obtenidas mediante el algoritmo SPCT. Este análisis permite evaluar el grado de independencia entre los distintos patrones temporales identificados durante el procesamiento de los datos térmicos.

Para ello, se calcula una matriz de correlación de Pearson entre los coeficientes temporales correspondientes a cada componente disperso. Esta matriz cuantifica la relación lineal entre los diferentes componentes, permitiendo identificar posibles dependencias o redundancias en la información extraída por el algoritmo.

Posteriormente, la matriz de correlación se almacena en un archivo CSV, lo que facilita su análisis cuantitativo posterior. Además, se genera una representación visual mediante un mapa de calor (*heatmap*) que permite interpretar de forma rápida la intensidad y el signo de las correlaciones entre los distintos componentes temporales.

Finalmente, tanto la matriz de correlación como su representación gráfica se guardan en la carpeta correspondiente a los resultados del análisis SPCT. Este procedimiento permite evaluar si los componentes dispersos capturan patrones térmicos independientes, lo que resulta fundamental para interpretar correctamente los resultados del análisis termográfico.

- El **Bloque 7** implementa el algoritmo TSR, cuyo objetivo es modelar la evolución temporal de la temperatura registrada en cada píxel mediante un ajuste polinómico de la señal térmica. Este método permite reconstruir y suavizar la respuesta térmica obtenida durante el proceso de grabación, facilitando posteriormente la identificación de defectos internos.

En primer lugar, se prepara la información temporal asociada a las mediciones termográficas, convirtiendo los tiempos a segundos y aplicando transformaciones logarítmicas tanto al tiempo como a la temperatura. Esta transformación permite linearizar parcialmente el comportamiento de la difusión térmica, lo que facilita el ajuste matemático de la señal.

Posteriormente, para cada píxel de la matriz de datos térmicos se realiza un ajuste polinómico en el dominio log-log, evaluando distintos grados de polinomio hasta un máximo predefinido (en este estudio 5). Para cada caso se calcula el coeficiente de determinación R^2 , que permite determinar la calidad del ajuste obtenido.

El grado polinómico que presenta el mayor valor de R^2 se selecciona como el modelo óptimo para describir la evolución térmica de ese píxel. A partir de este ajuste se almacenan tanto los coeficientes del polinomio óptimo como el correspondiente valor de R^2 , lo que permite caracterizar la respuesta térmica de cada punto de la probeta.

Finalmente, los resultados obtenidos se almacenan en estructuras de datos destinadas a su análisis posterior y se genera una carpeta específica para guardar los resultados asociados al procesamiento mediante TSR. Este procedimiento permite describir de forma más robusta el comportamiento térmico del material y mejorar la detección de anomalías internas en las probetas analizadas.

- El **Bloque 7_2** se encarga de almacenar los resultados obtenidos tras la aplicación del algoritmo TSR descrito en el bloque anterior. En particular, se guardan los coeficientes del ajuste polinómico óptimo y los correspondientes valores del coeficiente de determinación R^2 calculados para cada píxel de la imagen termográfica.

En primer lugar, los coeficientes óptimos obtenidos para cada píxel se organizan en una estructura tabular, donde cada columna representa uno de los coeficientes del polinomio ajustado. Posteriormente, esta información se almacena en un archivo CSV, lo que permite su utilización en análisis posteriores.

De forma análoga, los valores del coeficiente de determinación R^2 , que indican la calidad del ajuste polinómico para cada píxel, se agrupan en otro archivo CSV independiente. Este parámetro resulta fundamental para evaluar la fiabilidad del modelo utilizado para describir la evolución térmica.

Finalmente, ambos archivos se guardan dentro de la carpeta destinada a los resultados del análisis TSR, garantizando así una organización estructurada de los datos obtenidos durante el procesamiento. Estos resultados constituyen la base para los posteriores análisis de detección de defectos.

- El **Bloque 7_3** tiene como objetivo generar una representación visual de la calidad del ajuste polinómico obtenido mediante el algoritmo TSR. Para ello se utilizan los valores del coeficiente de determinación R^2 calculados previamente para cada píxel de la imagen termográfica.

En primer lugar, los valores de R^2 , originalmente almacenados en forma de vector unidimensional, se reorganizan según las dimensiones originales de la imagen térmica (filas y columnas). De este modo se reconstruye una matriz bidimensional que representa espacialmente la calidad del ajuste en cada píxel.

Posteriormente, se genera un mapa de calor (*heatmap*) utilizando una escala de colores que permite identificar visualmente las zonas donde el modelo TSR describe de forma más precisa la evolución térmica (valores elevados de R^2) y aquellas regiones donde el ajuste presenta menor calidad (en todos los casos, el R^2 fue mayor a 0,9).

Finalmente, la visualización generada se guarda en formato PNG dentro de una carpeta específica destinada a las representaciones de R^2 dentro de los resultados del análisis TSR. Este tipo de representación facilita la interpretación espacial de la calidad del modelo.

- El **Bloque 7_4** se encarga de calcular la primera, segunda y tercera derivada de la señal térmica reconstruida mediante el algoritmo TSR. El análisis de estas derivadas permite estudiar con mayor detalle la evolución temporal de la temperatura en cada píxel, lo que resulta especialmente útil para resaltar variaciones térmicas asociadas a posibles defectos internos.

En primer lugar, se reconstruye el vector temporal en escala logarítmica a partir del tiempo en segundos, filtrando los valores no válidos para asegurar la correcta aplicación del logaritmo. Posteriormente, se inicializan las estructuras de datos necesarias para almacenar los resultados de las derivadas.

A continuación, el procedimiento se aplica de forma iterativa a cada píxel de la imagen térmica. Para cada caso, se recuperan los coeficientes del polinomio óptimo previamente obtenidos durante el ajuste TSR. A partir de estos coeficientes se calculan matemáticamente la primera, segunda y tercera derivada del polinomio, evaluando posteriormente dichas funciones en los instantes de tiempo considerados.

Finalmente, los valores obtenidos para cada orden de derivada se almacenan en tres matrices independientes, organizadas en forma de *DataFrame*. En estas matrices, cada fila corresponde a un píxel de la imagen y cada columna representa un instante temporal. Estos resultados permiten analizar con mayor precisión las características dinámicas de la señal térmica y constituyen una base fundamental para la posterior identificación de defectos en el material estudiado.

- El **Bloque 7_5** tiene como objetivo calcular diferentes parámetros estadísticos asociados a la evolución temporal de las derivadas obtenidas mediante el algoritmo TSR. Este análisis se realiza de forma independiente para la primera, segunda y tercera derivada calculadas en el bloque anterior.

Para cada píxel de la imagen termográfica se analizan los valores de la derivada a lo largo del tiempo y se calculan varios indicadores estadísticos representativos. Entre ellos se incluyen la diferencia absoluta entre el valor máximo y mínimo, la media, el valor máximo, el valor mínimo, la desviación estándar, así como dos parámetros de forma de la distribución: la asimetría (*skewness*) y la curtosis (*kurtosis*)

Estos parámetros permiten describir de forma cuantitativa el comportamiento temporal de las derivadas de la señal térmica, proporcionando información útil sobre la variabilidad y la distribución de los valores obtenidos. Este tipo de análisis resulta especialmente relevante para resaltar patrones térmicos asociados a posibles defectos internos en el material.

Finalmente, los resultados calculados para cada orden de derivada se almacenan en archivos CSV independientes, organizados dentro de una carpeta específica destinada a las estadísticas del análisis TSR. Estos datos permiten realizar posteriormente comparaciones y análisis adicionales sobre el comportamiento térmico de las probetas estudiadas (**Bloque 9** hacia adelante).

- El **Bloque 7_6** tiene como finalidad visualizar los parámetros estadísticos calculados previamente para las derivadas de la señal térmica obtenidas mediante el algoritmo TSR. Para ello se generan representaciones gráficas que permiten analizar la distribución espacial de dichas características sobre la superficie de la probeta.

Para cada parámetro estadístico, los valores correspondientes a todos los píxeles se reorganizan en una matriz bidimensional que reproduce las dimensiones originales de la imagen termográfica. A partir de esta matriz se genera un mapa de calor (*heatmap*) que representa visualmente la distribución espacial del parámetro analizado.

Finalmente, las imágenes generadas se guardan en formato PNG dentro de subcarpetas específicas correspondientes a cada orden de derivada. Este procedimiento permite identificar de forma visual regiones donde las características estadísticas presentan comportamientos anómalos, lo que puede estar asociado a la presencia de defectos internos en el material.

- El **Bloque 7_7** tiene como objetivo visualizar el ajuste polinómico óptimo obtenido mediante el algoritmo TSR para un píxel representativo de la imagen termográfica. Esta representación permite comprobar de forma gráfica la calidad del modelo utilizado para describir la evolución térmica.

En primer lugar, se selecciona de forma aleatoria un píxel de entre todos los procesados en el análisis. A continuación, se recuperan tanto los valores originales de temperatura a lo largo del tiempo como los coeficientes del polinomio óptimo previamente calculados para dicho píxel.

Posteriormente, los datos se transforman al dominio logarítmico (logaritmo del tiempo y de la temperatura), siguiendo el mismo procedimiento utilizado durante el ajuste TSR. Con estos valores se calcula la curva predicha por el modelo polinómico, que representa la reconstrucción de la señal térmica.

Finalmente, se genera una representación gráfica donde se muestran simultáneamente los datos experimentales y la curva de ajuste obtenida mediante TSR. Esta visualización se guarda en formato PNG dentro de la carpeta de resultados del análisis TSR.

Este procedimiento permite realizar una verificación visual del ajuste polinómico, facilitando la evaluación de la calidad del modelo empleado para describir la evolución térmica de los píxeles analizados (la calidad siempre fue adecuada en este estudio).

- El **Bloque 8** se encarga de definir la geometría de referencia de los defectos utilizados en el análisis. En este apartado no se realiza ningún procesamiento de datos, sino que se establecen los parámetros espaciales que describen la posición y el tamaño de las regiones donde se han modelado los defectos dentro de la probeta.

En primer lugar, se define un cuadrado principal que representa el área total de la región de interés analizada en la imagen (en otras palabras, el área de la superficie frontal de la probeta). Este cuadrado se describe mediante sus coordenadas mínimas en los ejes x e y (origen de coordenadas esquina inferior izquierda), junto con sus dimensiones de ancho y altura.

Posteriormente, se define una lista de cuadrados más pequeños, donde cada uno de ellos corresponde a un defecto interno simulado en la pieza. Para cada defecto se especifican igualmente las coordenadas de su posición y sus dimensiones geométricas.

Estas definiciones permiten establecer un mapa de referencia de defectos, que será utilizado en etapas posteriores del análisis para comparar la localización de las anomalías térmicas detectadas por los distintos algoritmos de procesamiento. De esta forma, se facilita la evaluación del rendimiento de los métodos empleados para la detección de defectos en las probetas analizadas.

- El **Bloque 8_2** se encarga de definir los parámetros necesarios para generar la imagen binaria de referencia que representa la distribución espacial de los defectos definidos previamente. Este paso permite establecer las características de la imagen rasterizada que será utilizada en el análisis posterior.

En primer lugar, se solicita al usuario el número de filas y columnas de la imagen final, que serán las mismas dimensiones que el ROI correspondiente, lo que determina su resolución espacial y, por tanto, el tamaño de la matriz que representará la distribución de píxeles.

Posteriormente, se especifica la ruta de almacenamiento donde se guardará la imagen generada. Esta ruta permite almacenar el archivo resultante dentro del entorno de trabajo, facilitando su utilización en etapas posteriores del procesamiento.

Una vez introducidos estos parámetros, el sistema confirma las dimensiones definidas y la ubicación de guardado. Esta configuración constituye el paso previo necesario para generar la imagen binaria de referencia de los defectos, que será utilizada posteriormente para comparar los resultados obtenidos mediante los algoritmos de procesamiento de imágenes térmicas.

- El **Bloque 8_3** tiene como objetivo generar la imagen binaria de referencia que representa la distribución espacial de los defectos definidos previamente. Esta imagen se construye combinando la geometría de los defectos con las dimensiones de la imagen especificadas en el bloque anterior.

En primer lugar, se crea una matriz bidimensional inicializada con valores cero, cuyas dimensiones corresponden al número de filas y columnas definido para la imagen final. Esta matriz constituye la base de la representación rasterizada.

Posteriormente, se realiza un proceso de transformación de coordenadas, mediante el cual las posiciones geométricas definidas en el sistema de referencia del modelo se convierten en coordenadas de píxel dentro de la matriz de la imagen. Este paso permite relacionar las dimensiones físicas de los defectos con su representación digital.

A continuación, se representa la región principal de la probeta mediante un cuadrado grande al que se asigna el valor binario 1, indicando la zona de material sin defectos. Posteriormente, los defectos simulados, definidos como cuadrados de menor tamaño, se insertan dentro de esta región asignándoles el valor 0, lo que permite distinguir claramente las zonas defectuosas.

Finalmente, la matriz resultante se guarda como una imagen binaria en formato PNG, utilizando una escala de grises para representar los valores binarios. Esta imagen constituye

una referencia geométrica de los defectos, que será utilizada en etapas posteriores para comparar los resultados obtenidos mediante los diferentes algoritmos de procesamiento termográfico y evaluar su capacidad de detección (ver a partir del **Bloque 9**).

- El **Bloque 9** constituye el punto de inicio de la fase de comparación cuantitativa entre los algoritmos PCT, SPCT y TSR, utilizando como métrica principal la Relación Señal-Ruido. En este apartado se recopila la información de entrada necesaria para poder realizar el análisis comparativo.

En primer lugar, se solicita la ruta de la carpeta que contiene los archivos CSV procesados, los cuales incluyen los resultados obtenidos mediante los diferentes algoritmos de procesamiento de imágenes térmicas. Estos datos constituyen la base para el cálculo posterior de los indicadores de rendimiento.

Posteriormente, se introducen las dimensiones del ROI empleado, especificando el número de filas y columnas. Esta información resulta necesaria para reconstruir correctamente las matrices bidimensionales a partir de los datos almacenados en formato tabular.

Finalmente, se solicita la ruta de la imagen binaria de referencia, generada previamente, que define la localización de las regiones con defectos y las zonas sin defectos dentro de la probeta. Esta imagen actúa como referencia para separar las regiones de señal y ruido durante el cálculo del SNR.

En conjunto, este bloque permite reunir todos los parámetros y archivos necesarios para la fase posterior de evaluación del rendimiento de los algoritmos de procesamiento termográfico aplicados en este trabajo.

- El **Bloque 9_2** tiene como objetivo preparar las máscaras de referencia que permiten distinguir entre las regiones saludables y las regiones que contienen defectos dentro de la imagen termográfica. Estas máscaras son fundamentales para el cálculo posterior de la Relación Señal-Ruido (SNR) utilizada en la comparación entre los algoritmos PCT, SPCT y TSR.

En primer lugar, se carga la imagen binaria de referencia generada previamente. Esta imagen contiene valores binarios que indican la localización de las regiones sin defectos y de las zonas donde se han definido defectos. Para asegurar la coherencia del procesamiento, la imagen se convierte a un formato monocanal y se normaliza para que los valores correspondan estrictamente a 0 y 1.

A continuación, se genera una máscara de región saludable, identificando todos los píxeles cuyo valor en la imagen binaria es igual a 1. Estos píxeles representan las zonas del material consideradas libres de defectos.

Posteriormente, se crean máscaras individuales para cada defecto, utilizando las coordenadas geométricas definidas previamente. Para cada región defectuosa se calcula la correspondiente área en el espacio de píxeles, generando así una máscara que identifica únicamente los píxeles pertenecientes a dicho defecto.

Una vez generadas las máscaras individuales, estas se combinan para obtener una máscara global de defectos, que agrupa todos los píxeles correspondientes a regiones defectuosas. Finalmente, se realiza una verificación para asegurar que no exista solapamiento entre las regiones saludables y defectuosas.

Estas máscaras constituyen la base para separar las zonas de señal y ruido durante el cálculo del SNR, permitiendo evaluar cuantitativamente la capacidad de los distintos algoritmos para detectar los defectos presentes en las probetas analizadas.

- El **Bloque 9_3** constituye el núcleo del análisis comparativo entre los algoritmos PCT, SPCT y TSR, ya que en este apartado se calcula la Relación Señal-Ruido para los distintos resultados obtenidos en el procesamiento de las imágenes termográficas.

En primer lugar, se crea una carpeta específica donde se almacenarán los resultados del análisis SNR. A continuación, se define una función encargada de calcular el SNR a partir de los datos de la imagen procesada (uno de los resultados de los algoritmos de procesamiento utilizado) y de las máscaras de regiones saludables y defectuosas generadas previamente.

El cálculo del SNR se realiza comparando las características estadísticas de ambas regiones. En particular, se calcula la media de los valores de la región saludable, así como su desviación estándar, que se interpreta como el nivel de ruido de fondo. Posteriormente, se calcula la media de los valores de la región defectuosa. A partir de estos parámetros, el **SNR se obtiene como la diferencia absoluta entre ambas medias dividida por la desviación estándar de la región saludable.**

Una vez definida la función de cálculo, el bloque localiza y carga todos los archivos CSV que contienen los resultados generados por los diferentes algoritmos. Cada archivo se procesa de forma iterativa, analizando cada uno: componentes principales (EOFs del PCT), componentes dispersos (SCs, que es lo equivalente a los EOFs pero obtenidos con el SPCT) y parámetros derivados del análisis TSR (diferencia absoluta entre el valor máximo y mínimo, la media, el valor máximo, el valor mínimo, la desviación estándar, así como la asimetría (*skewness*) y la curtosis (*kurtosis*) de cada una de las tres derivadas).

Para cada resultado se calculan dos tipos de métricas:

- **SNR global, utilizando la máscara que agrupa todas las regiones defectuosas.**
- **SNR individual por defecto, empleando las máscaras correspondientes a cada región defectuosa definida en el experimento.**

De esta forma, se obtiene una evaluación cuantitativa tanto del rendimiento general del algoritmo como de su capacidad para detectar cada defecto de manera individual.

En conjunto, este bloque permite transformar los resultados obtenidos mediante los algoritmos de procesamiento termográfico en indicadores cuantitativos comparables, facilitando la evaluación objetiva de la capacidad de detección de defectos de cada método analizado.

- El **Bloque 9_4** se encarga de organizar, almacenar y resumir los resultados de SNR calculados previamente. Este bloque permite consolidar los valores obtenidos para facilitar su posterior análisis y comparación entre los distintos algoritmos de procesamiento.

En primer lugar, los valores de SNR global calculados para cada resultado algorítmico se organizan en una tabla y se almacenan en un archivo CSV, lo que permite conservar de forma estructurada los resultados obtenidos durante el análisis.

De manera similar, los valores de SNR calculados para cada defecto individual también se agrupan en otro archivo CSV independiente. Esta separación permite analizar con mayor detalle el comportamiento de cada algoritmo en la detección de defectos específicos.

Una vez guardados los resultados, el bloque realiza un resumen automático, identificando los valores máximos de SNR obtenidos. En particular, se determina:

- El máximo SNR global, indicando el archivo y el resultado algorítmico correspondiente.
- El máximo SNR por defecto, señalando además el identificador del defecto asociado.

Este resumen proporciona una primera referencia sobre qué representaciones o resultados algorítmicos ofrecen mayor contraste entre regiones defectuosas y saludables, lo cual resulta útil para evaluar la eficacia de los distintos métodos analizados.

En conjunto, este bloque finaliza la fase de análisis SNR, almacenando los resultados obtenidos y proporcionando una visión inicial del rendimiento relativo de los algoritmos estudiados.

- El **Bloque 9_5** tiene como objetivo recuperar los resultados del análisis de la SNR previamente almacenados, con el fin de utilizarlos en las etapas posteriores de visualización y análisis gráfico.

En primer lugar, se carga el archivo `Overall_SNR_Results.csv`, que contiene los valores de SNR global calculados para los distintos resultados algorítmicos. Estos datos se almacenan en un *DataFrame* de pandas, lo que permite su manipulación y análisis dentro del entorno de programación.

Posteriormente, se carga el archivo `Per_Defect_SNR_Results.csv`, que incluye el valor máximo SNR calculado para cada defecto individual junto con el algoritmo de procesamiento correspondiente además del resultado específico correspondiente (ejemplo, EOF2 si es PCT, SC3 si es SPCT o curtosis si es TSR). Este conjunto de datos permite evaluar con mayor detalle la capacidad de los algoritmos para detectar defectos específicos.

Tras la carga de cada archivo, el bloque muestra un breve resumen de los datos, incluyendo las primeras filas y las dimensiones del *DataFrame*. Este paso permite verificar que los archivos se han cargado correctamente y que los datos están disponibles para el análisis posterior.

En caso de que se produzca algún error durante la carga de los archivos, el bloque gestiona la excepción correspondiente e inicializa los *DataFrames* como vacíos, evitando así interrupciones en el flujo de ejecución.

En conjunto, este bloque permite recuperar los resultados previamente calculados de SNR, asegurando que estén disponibles para las representaciones gráficas y el análisis comparativo final entre los algoritmos estudiados.

- El **Bloque 9_6** tiene como objetivo reorganizar y consolidar los datos de SNR previamente cargados, con el fin de prepararlos para su posterior representación gráfica y análisis comparativo.

En primer lugar, se extrae el valor máximo de SNR general para cada resultado algorítmico. Para ello, los datos se agrupan según el identificador del resultado del algoritmo y se selecciona el valor máximo de SNR dentro de cada grupo. Este proceso permite identificar el mejor rendimiento global asociado a cada representación generada por los algoritmos.

Posteriormente, se realiza un procedimiento similar para los valores de SNR por defecto individual. En este caso, además de identificar el valor máximo de SNR para cada resultado algorítmico, también se determina el defecto específico que produce dicho valor máximo. Esto permite conocer qué defectos son detectados con mayor contraste por cada método.

Una vez obtenidos ambos conjuntos de datos, se realiza una fusión de los resultados en una única tabla, utilizando el identificador del resultado algorítmico como referencia común. El resultado es un *DataFrame* consolidado que contiene, para cada resultado algorítmico, tanto el máximo SNR global como el máximo SNR asociado a un defecto individual.

Esta reorganización de los datos permite disponer de una estructura más clara y compacta, facilitando la generación de gráficas comparativas y el análisis del rendimiento relativo de los distintos métodos de procesamiento termográfico.

- El **Bloque 9_7** tiene como objetivo visualizar gráficamente los resultados del análisis SNR previamente reorganizados. Para ello, se genera un **gráfico de barras** comparativo que permite analizar de forma clara el rendimiento relativo de los distintos resultados algorítmicos.

En primer lugar, se configura el estilo gráfico utilizando la biblioteca Seaborn, con el fin de mejorar la claridad visual de la representación. A continuación, se preparan dos conjuntos de barras para cada resultado algorítmico analizado.

El primer conjunto de barras representa el valor máximo de SNR global, mientras que el segundo conjunto muestra el valor máximo de SNR obtenido para un defecto

específico. Esta doble representación permite comparar simultáneamente el rendimiento general de cada resultado algorítmico y su capacidad para detectar defectos individuales.

Además, se incluyen anotaciones sobre las barras correspondientes al SNR por defecto, indicando el identificador del defecto que produce el valor máximo de SNR. Esta información facilita la interpretación del gráfico, permitiendo identificar qué defectos son detectados con mayor contraste por cada método.

Finalmente, el gráfico se personaliza mediante la inclusión de título, etiquetas de los ejes y leyenda, y se guarda como una imagen en la carpeta de resultados del análisis SNR.

En conjunto, este bloque proporciona una representación visual clara y comparativa del rendimiento de los diferentes algoritmos, facilitando la identificación de los métodos que ofrecen mayor capacidad de detección de defectos en el análisis termográfico.

- El **Bloque 9_8** tiene como finalidad realizar una visualización selectiva de los resultados de SNR, **centrando el análisis únicamente en aquellos valores que superan un umbral definido** por el usuario. Este procedimiento permite destacar los resultados algorítmicos que presentan una mayor capacidad de detección de defectos.

En primer lugar, el bloque solicita de forma interactiva un valor de umbral de SNR, que se utilizará como criterio para filtrar los resultados obtenidos anteriormente. A continuación, el conjunto de datos que contiene los valores de SNR por defecto se filtra para conservar únicamente aquellas entradas cuyo valor sea superior al umbral especificado.

Posteriormente, se genera un gráfico de barras agrupadas, donde el eje horizontal representa los identificadores de los defectos, mientras que el eje vertical muestra los valores de SNR correspondientes. Cada grupo de barras se distingue según el resultado algorítmico, lo que permite comparar el comportamiento de los diferentes métodos para cada defecto.

Además, cada barra incluye anotaciones con el valor exacto de SNR y el resultado algorítmico asociado, facilitando la interpretación de los resultados. También se ajustan los límites del eje vertical para comenzar en un valor próximo al umbral, de modo que la visualización enfatice las diferencias entre los resultados que superan dicho criterio.

Finalmente, el gráfico generado se guarda en la carpeta de resultados del análisis SNR, permitiendo su utilización en la interpretación final de los resultados.

En conjunto, este bloque proporciona una herramienta de análisis visual que permite identificar de forma clara los algoritmos y resultados que presentan un rendimiento superior, al centrarse exclusivamente en aquellos valores de SNR que superan un nivel mínimo de calidad definido por el usuario.

- El **Bloque 9_9** tiene como objetivo identificar, para cada defecto individual, el resultado algorítmico que produce el mayor valor de SNR. Este análisis permite determinar qué método o representación de imagen ofrece la mejor capacidad de detección para cada defecto específico.

En primer lugar, se verifica que el conjunto de datos que contiene los valores de SNR por defecto individual no esté vacío. Si los datos están disponibles, se procede a agrupar la información según el identificador del defecto.

Posteriormente, dentro de cada grupo correspondiente a un defecto específico, se identifica la fila que contiene el valor máximo de SNR. Este procedimiento permite seleccionar el resultado algorítmico que proporciona el mayor contraste entre la región defectuosa y la región saludable para dicho defecto.

Una vez identificados estos valores máximos, se crea un nuevo conjunto de datos que incluye únicamente la información relevante: el identificador del defecto, el resultado algorítmico asociado y el valor máximo de SNR obtenido. Para mejorar la claridad de la información, las columnas se renombran adecuadamente.

El resultado de este proceso es una tabla resumida que indica, para cada defecto analizado, qué resultado algorítmico proporciona la mejor capacidad de detección.

En conjunto, este bloque permite identificar de forma directa el método más eficaz para detectar cada defecto individual, proporcionando información clave para la evaluación comparativa final de los algoritmos estudiados.

- El **Bloque 9_10** tiene como objetivo generar una visualización final del rendimiento de los algoritmos en la detección de defectos individuales. Para ello, se crea un gráfico que muestra el valor máximo de SNR alcanzado para cada defecto, junto con el resultado algorítmico responsable de dicho valor.

En primer lugar, se configura el estilo gráfico utilizando la biblioteca Seaborn, lo que permite mejorar la claridad visual del gráfico. Posteriormente, se genera un gráfico de barras en el que cada barra representa un defecto específico.

En esta representación, el eje horizontal muestra los identificadores de los defectos analizados, mientras que el eje vertical representa el valor máximo de SNR obtenido para cada uno de ellos. De esta manera, es posible observar directamente qué defectos presentan mayor contraste en las imágenes procesadas.

Además, sobre cada barra se incluye una anotación que indica el resultado algorítmico asociado al valor máximo de SNR. Esto permite identificar qué representación generada por los algoritmos (por ejemplo, componentes principales, componentes dispersos o parámetros derivados del análisis TSR) proporciona la mejor detección para cada defecto.

Finalmente, el gráfico se completa con título, etiquetas de los ejes y una cuadrícula, y se guarda como un archivo de imagen en la carpeta de resultados del análisis SNR.

En conjunto, este bloque proporciona una visión clara y consolidada del rendimiento de los distintos métodos para cada defecto individual, permitiendo identificar qué técnica ofrece la mejor capacidad de detección en cada caso.

4 RESULTADOS Y DISCUSIÓN

4.1 Configuración de los resultados

En este capítulo se presentan e interpretan los resultados obtenidos a partir del procesamiento de las secuencias termográficas. Con el fin de facilitar la comprensión del comportamiento de los distintos algoritmos analizados, se incluyen diferentes gráficas y representaciones comparativas que permiten evaluar su capacidad para detectar defectos en los materiales estudiados.

Los resultados se analizan utilizando la métrica *Signal to Noise Ratio* (SNR), que permite cuantificar el contraste entre las regiones defectuosas y las regiones sanas del material. Esta métrica es ampliamente utilizada en el análisis de termografía infrarroja activa, ya que proporciona una medida objetiva del rendimiento de los algoritmos de procesamiento.

Cuando es necesario, también se incorporan tablas y representaciones gráficas que resumen los resultados obtenidos, así como comparaciones entre los distintos métodos analizados. Además, se realiza una validación de los resultados mediante el uso de la imagen de referencia con la localización conocida de los defectos.

La Figura 4-1 muestra la imagen de referencia utilizada para identificar la posición y el tamaño relativo de los defectos presentes en la placa analizada, visto desde la superficie frontal, la superficie grabada con la cámara, visto desde la superficie frontal, la superficie grabada con la cámara. Esta imagen se emplea como base para la evaluación de los resultados obtenidos mediante los diferentes algoritmos de procesamiento termográfico.

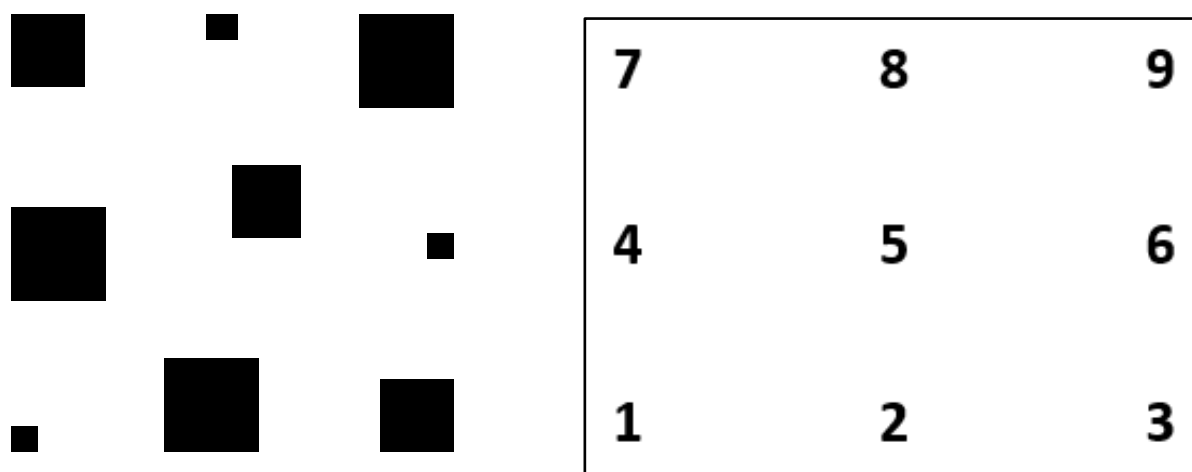


Figura 4-1 Referencia utilizada para identificar la posición y el tamaño relativo de los defectos presentes en la placa analizada, visto desde la superficie frontal, la superficie grabada con la cámara (Elaboración propia)

Los defectos se clasifican en tres grupos según su tamaño y profundidad, tal y como se ha descrito en el apartado 3, (espesor de la base se recuerda que es de 1,5 cm):

- Defectos pequeños y gran profundidad: 1, 6 y 8, con dimensiones 0,5 cm x 0,5 cm, profundidad con respecto a la superficie frontal de 1,1 cm, y longitud de los huecos de 0,4 cm.
- Defectos medianos y profundidad intermedia: 3, 5 y 7, con dimensiones 1,5 cm x 1,5 cm, profundidad con respecto a la superficie frontal de 0,7 cm y longitudes de los huecos de 0,8 cm.
- Defectos grandes y poca profundidad: 2, 4 y 9, con dimensiones 2 cm x 2 cm, profundidad con respecto a la superficie frontal de 0,3 cm y longitudes de los huecos de 1,2 cm.

Esta clasificación permite analizar cómo influye el tamaño y profundidad del defecto en la capacidad de detección de los algoritmos evaluados, de escenario menos exigente a más exigente: defectos grandes (2 cm x 2 cm) a profundidades de 0,3 cm, defectos medianos (1,5 cm x 1,5 cm) a profundidades de 0,7 cm y defectos pequeños (0,5 cm x 0,5 cm) a profundidades de 1,1 cm.

Durante el análisis de resultados, estas posiciones se utilizan para calcular SNR en cada región defectuosa. De este modo, es posible comparar cuantitativamente el rendimiento de los algoritmos PCT, SPCT y TSR en la detección de defectos de diferentes dimensiones y profundidades.

Se va a representar cuatro casos de estudio para cada polímero, tal y como se ha dictado en el apartado 3. Se va a tener casos con las probetas sin la tapa de espesor de 1 cm, quedando los huecos libres en contacto con el aire en la superficie posterior de la base y, por tanto, en contacto directo con la entrada del flujo de calor de la excitación térmica correspondiente. Como se va a emplear dos fuentes, la lámpara halógena de 500 W (modo foco) y la lámpara infrarroja de 2000 W (modo estufa), se obtiene aquí dos casos de estudio. Con la introducción de la tapa, y empleando de nuevo en diferentes ensayos cada una de las lámparas previa, se obtiene los otros dos casos de estudio restantes. Así se simula tanto defectos internos expuestos al entorno en su superficie posterior como totalmente herméticos con la introducción de dicha tapa.

Por último, comentar que en cada caso de estudio se representa **primero el gráfico del SNR general** (global, promediando el SNR teniendo en cuenta el SNR obtenido en cada defecto) para cada resultado

de cada algoritmo, indicando en ese mismo gráfico el **defecto con el SNR máximo detectado por cada resultado**. El **segundo gráfico** es la representación del **SNR máximo obtenido para cada defecto**, indicando **con qué resultado** se ha obtenido en cada una de dichas metas. Por **último**, un **grupo de gráficos** que representan **qué defectos superan un umbral de SNR previamente definido por el usuario**, junto con los **resultados que consiguen dichas hazañas**. Tras el **cierre de la representación de estos gráficos**, se **visualiza, como conclusión, los resultados con las mejores prestaciones** en cada caso de estudio. La nomenclatura utilizada en cada resultado es el siguiente:

PCT

- Primer EOF: EOF_0
- Segundo EOF: EOF_1
- Tercer EOF: EOF_2
- Cuarto EOF: EOF_3
- Quinto EOF: EOF_4
- Sexto EOF: EOF_5

SPCT

- Primer EOF dispersado: SC_0
- Segundo EOF dispersado: SC_1
- Tercer EOF dispersado: SC_2
- Cuarto EOF dispersado: SC_3
- Quinto EOF dispersado: SC_4
- Sexto EOF dispersado: SC_5

TSR (en todas las nomenclaturas terminan añadiendo como sufijo “_x”, siendo “x” la primera (1), segunda (2) o tercera derivada (3))

- Diferencia de temperaturas en términos absoluto: *abs_diff*
- Curtosis: *kurtosis*
- Temperatura máxima: *max*
- Temperatura media: *mean*
- Temperatura mínima: *min*
- Asimetría: *skew*
- Desviación típica: *std*

4.2 Resultados en PETG-CF

4.2.1 Caso de estudio 1: PETG-CF con lámpara halógena sin tapa

En primer lugar, la Figura 4-2 presenta la comparación entre el SNR general (no existe máximo SNR general, sólo hay SNR general, corrija eso en el resto del documento) y el máximo SNR por defecto para cada resultado algorítmico. A partir de este gráfico se observa que el valor más alto de SNR general corresponde al algoritmo SC_1, Sigue poniendo en varias partes como símbolo decimal el punto, en vez de la coma. Revise eso también en todo el documento. Este resultado indica que dicho componente disperso proporciona el mayor contraste global entre las regiones defectuosas y las regiones sanas.

Además, se ha analizado el número de defectos detectados por los diferentes algoritmos. En este caso, los defectos que aparecen con mayor frecuencia en los resultados corresponden a los defectos 8, 2 y 9, lo que sugiere que estos presentan una mayor respuesta térmica durante el experimento.

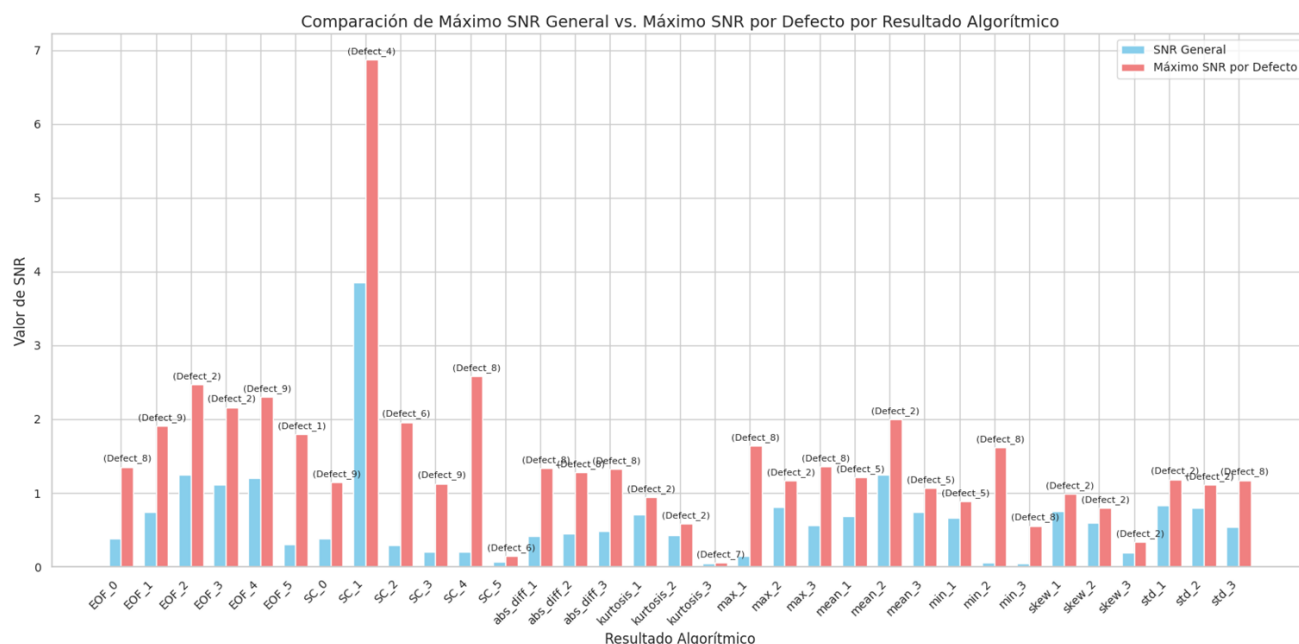


Figura 4-2 La comparación entre el SNR general y el máximo SNR por defecto para cada resultado algorítmico. Caso de estudio 1 (Elaboración propia)

La Figura 4-3 muestra el gráfico de Máximo SNR por defecto con el algoritmo correspondiente. En este caso se analizan los defectos que presentan un valor SNR superior a 1, considerado como un umbral mínimo para una detección fiable. Los defectos detectados bajo este criterio son 1, 2, 4, 5, 6, 8 y 9.

Entre ellos, los defectos 4, 9 y 2 presentan los valores de SNR más elevados, con aproximadamente 7,0, 6,5 y 6,1, respectivamente. Es importante destacar que estos tres defectos han sido detectados mediante el algoritmo SC_1, lo que confirma la elevada capacidad de este componente disperso para resaltar defectos de mayor tamaño y menos profundas. Adicionalmente, el defecto 8 presenta un valor de SNR cercano a 2,6, siendo detectado mediante el algoritmo SC_4.

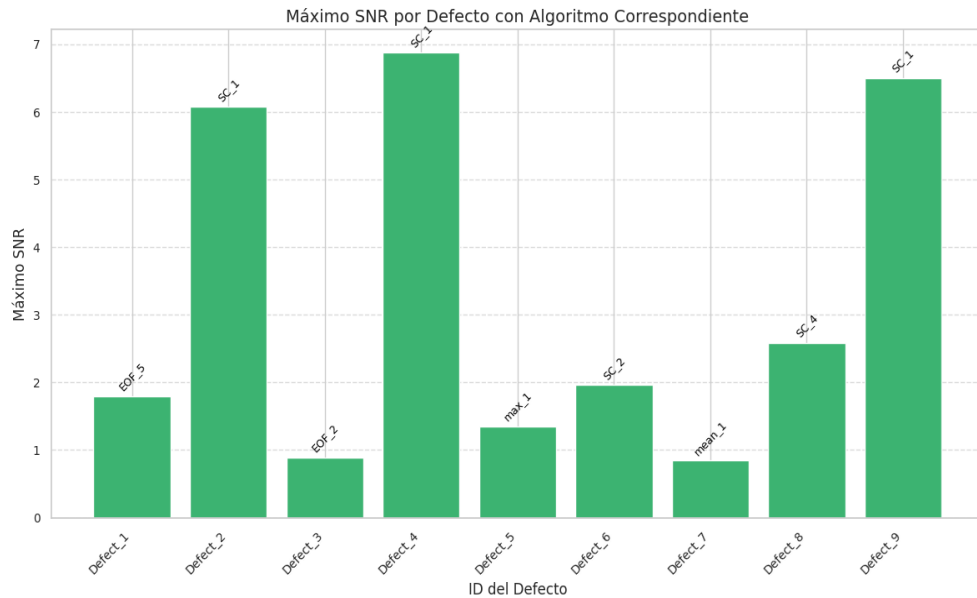


Figura 4-3 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

Posteriormente, la Figura 4-4 presenta el análisis de los defectos que alcanzan valores SNR > 1,5. Bajo este criterio se detectan los defectos 1, 2, 4, 6, 8 y 9, lo que indica una detección relativamente robusta para estas posiciones. Asimismo, se observa que el algoritmo que detecta un mayor número de defectos en este rango es SC_1.

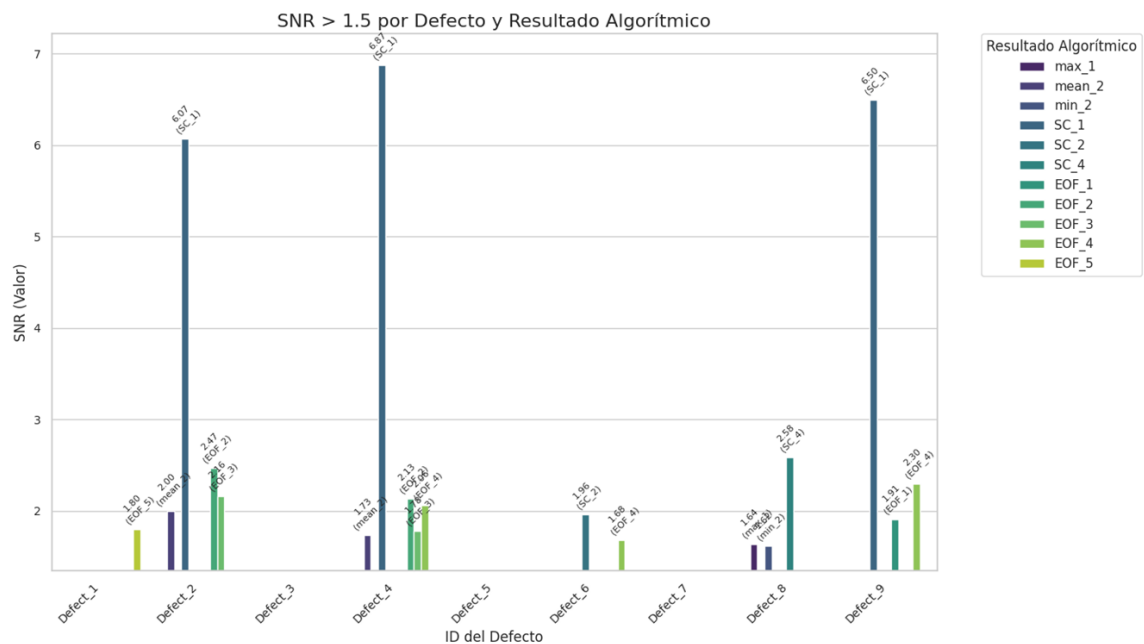


Figura 4-4 El análisis de los defectos que alcanzan valores SNR > 1,5 (Elaboración propia)

En la Figura 4-5 se muestra el análisis con un criterio más restrictivo, considerando únicamente valores SNR > 2,0. En este caso, los defectos detectados son 2, 4, 8 y 9, lo que indica que los defectos de mayor tamaño y menor profundidad presentan una mayor detectabilidad térmica. De nuevo, el algoritmo SC_1 destaca como el método más eficaz, seguido por SC_4, que también logra detectar algunos defectos relevantes (como el defecto 8).

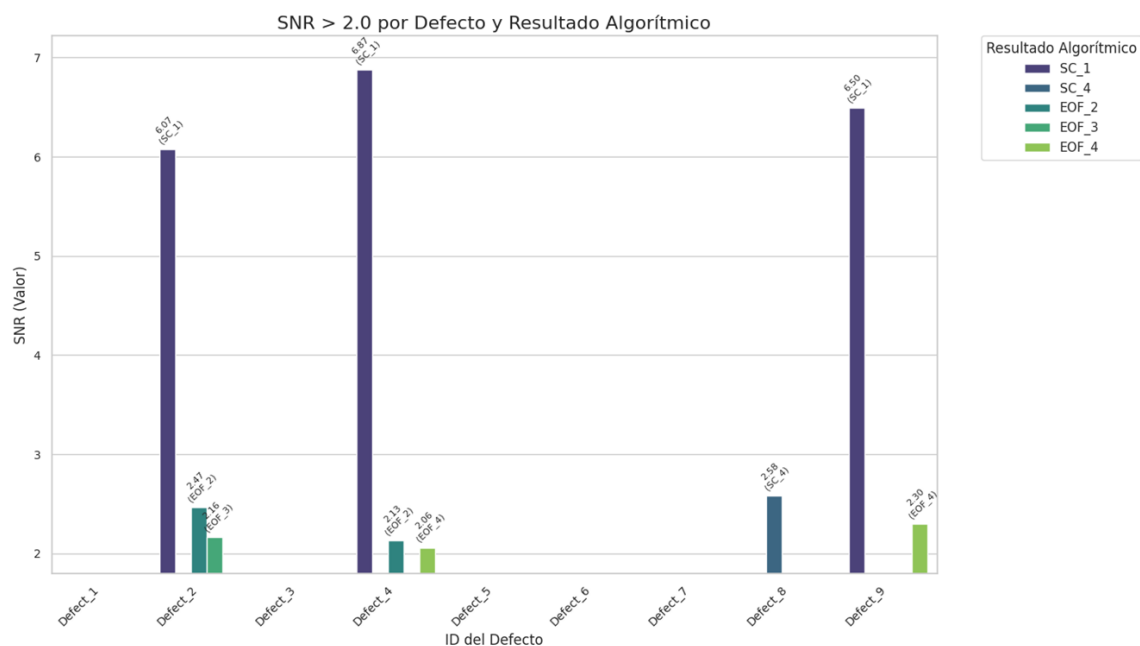


Figura 4-5 El análisis de los defectos que alcanzan valores SNR > 2 (Elaboración propia)

Para complementar el análisis cuantitativo, la Figura 4-6 presenta una comparación visual entre los resultados obtenidos con mejor rendimiento SNR, es decir, mediante los algoritmos SC_1 y SC_4. En estas imágenes se puede observar claramente cómo los defectos aparecen con mayor contraste respecto al fondo térmico, las grandes y pocas profundas (2, 4, y 9) con SC_1, la pequeña y poca profunda (defecto 8) asemeja más a ruido detectado por la SC_4, aunque con la SC_4 se aprecia ligeramente los defectos medios y con profundidad intermedia (3, 7) a pesar del bajo SNR.

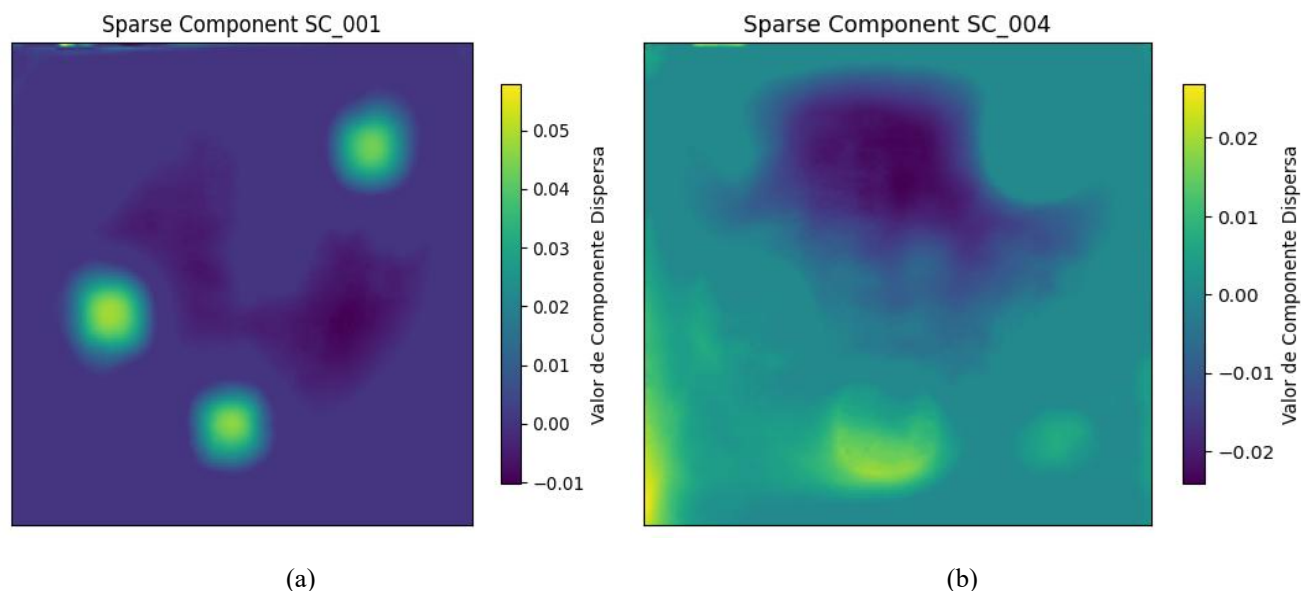


Figura 4-6 Comparación visual entre los resultados obtenidos a) SC_1 y b) SC_4 (Elaboración propia)

Por otro lado, la Figura 4-7 muestra la varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT. Se observa que las componentes SC_1 y SC_4 presentan valores relativamente elevados de varianza explicada, lo que indica que contienen una parte significativa de la información relevante del experimento.

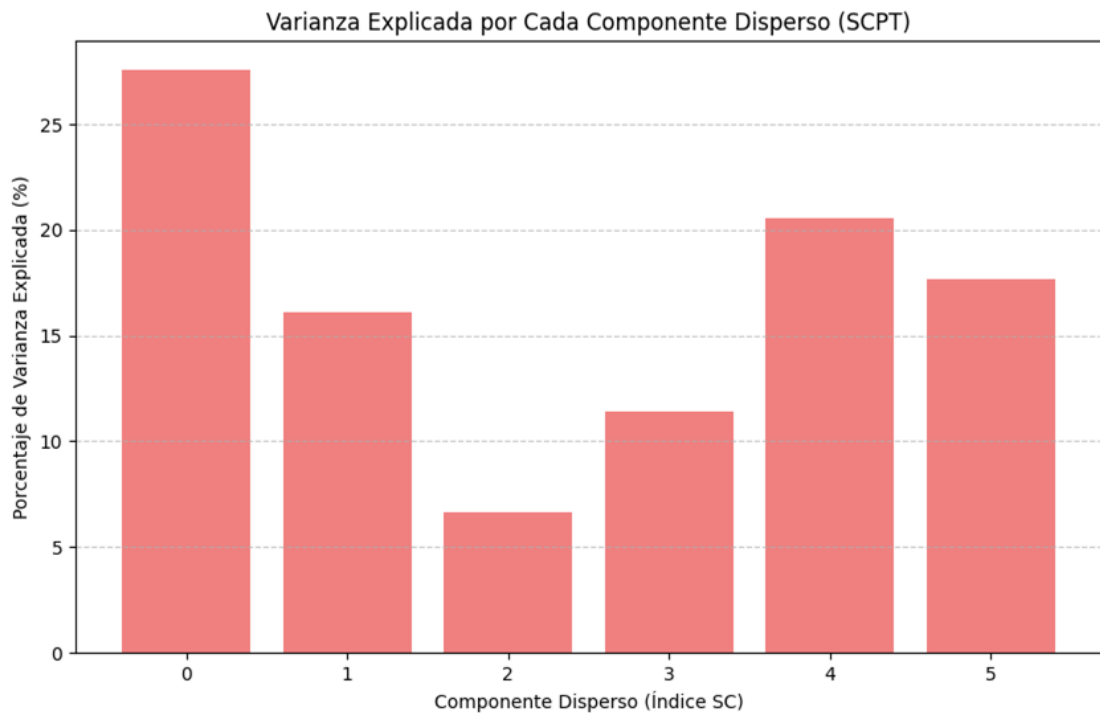


Figura 4-7 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)

Finalmente, la Figura 4-8 presenta la evolución temporal de los coeficientes dispersos (SCPT). En este gráfico se puede observar que las curvas correspondientes a SC_1 y SC_4 presentan tendencias similares a lo largo del tiempo, lo que sugiere que ambas componentes capturan patrones térmicos relacionados con la aparición de los defectos durante el proceso el ensayo, y distintas con respecto a las tendencias de los otros coeficientes.

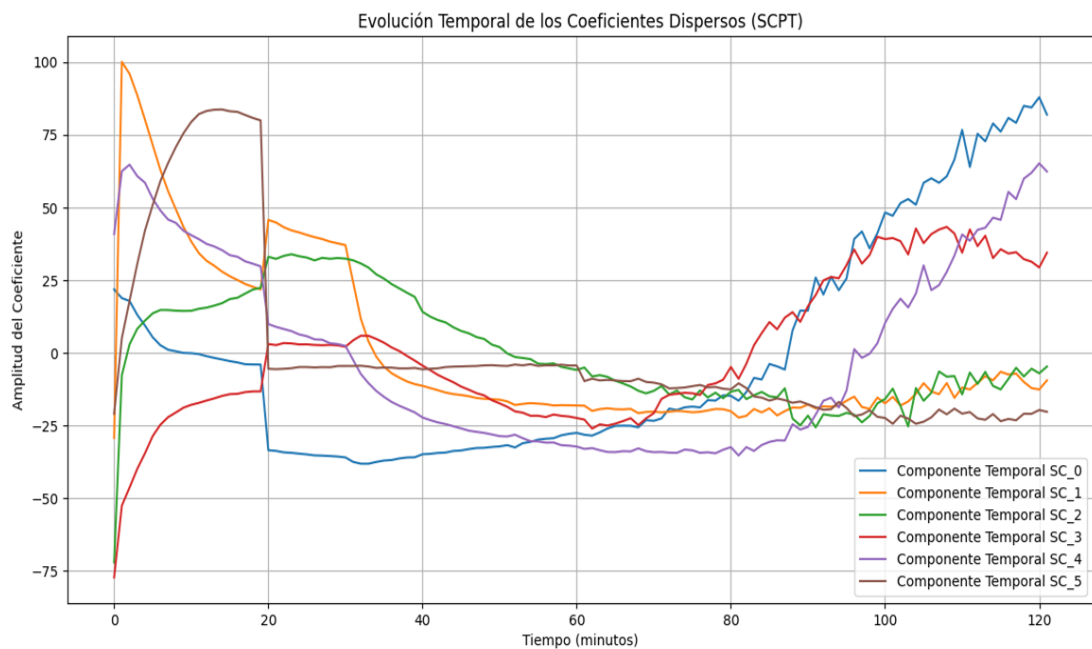


Figura 4-8 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia)

En conjunto, estos resultados indican que el método SPCT, especialmente a través de las componentes SC_1 y SC_4, ofrece una capacidad elevada para la detección de defectos en la muestra de PETG-CF bajo excitación térmica con lámpara halógena sin tapa.

4.2.2 Caso de estudio 2: PETG-CF con lámpara halógena con tapa

En primer lugar, la Figura 4-9 muestra la comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico. A partir de este gráfico se observa que el valor más alto de SNR general corresponde al algoritmo SC_3, obtenido mediante el método SPCT, con un valor aproximado de 1.1.

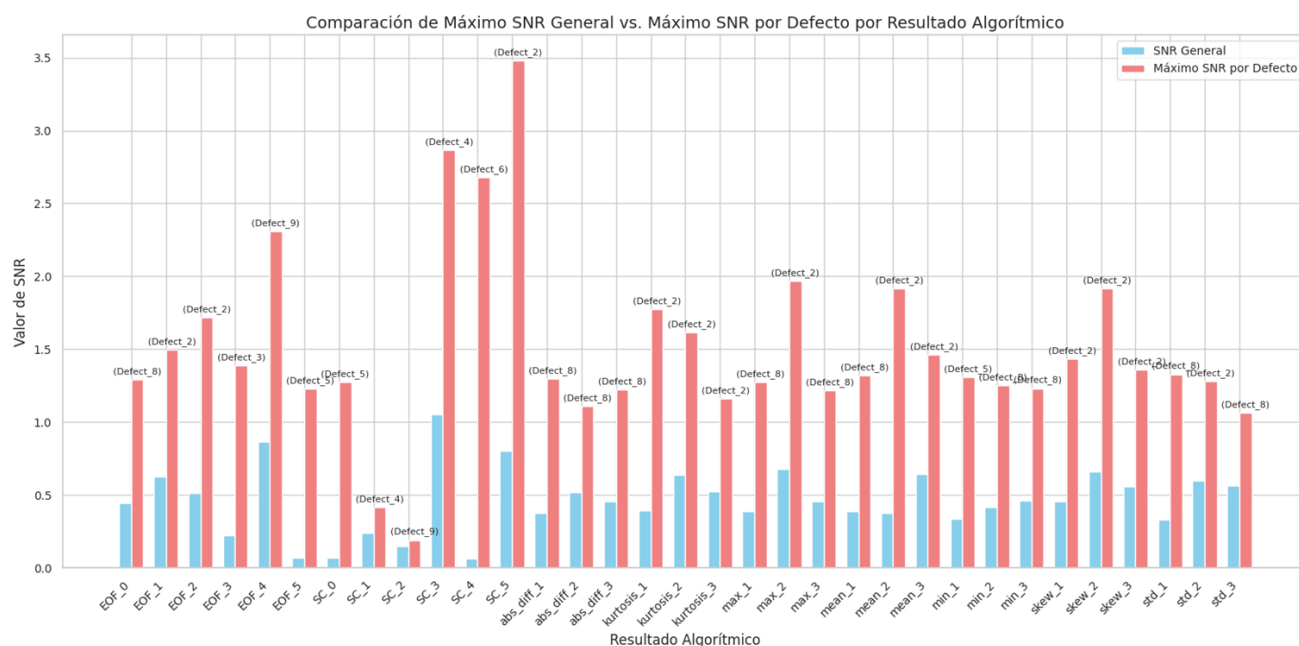


Figura 4-9 La comparación entre el SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)

Además, se ha analizado la frecuencia de detección de defectos en los distintos resultados algorítmicos. En este caso, los defectos que aparecen con mayor frecuencia corresponden a los defectos 2, 8 y 5, lo que sugiere que estas regiones presentan una respuesta térmica más evidente bajo estas condiciones experimentales.

La Figura 4-10 presenta el gráfico de Máximo SNR por defecto con el algoritmo correspondiente. En este caso se analizan los defectos que presentan valores de SNR superiores a 1, considerados como indicativos de una detección fiable. Los defectos identificados bajo este criterio son 2, 3, 4, 5, 6, 7, 8 y 9.

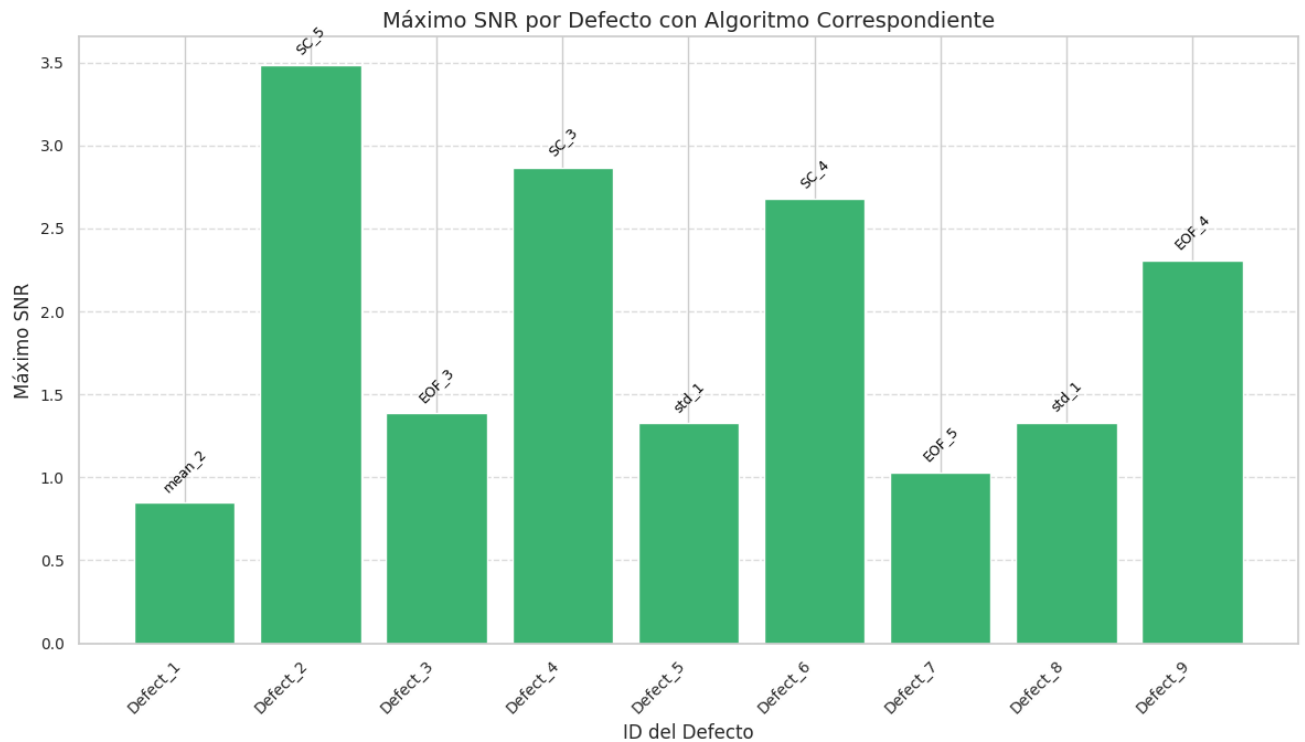


Figura 4-10 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

Entre ellos, el defecto 2 presenta el valor de SNR más elevado, aproximadamente 3,5, detectado mediante el algoritmo SC_5. Asimismo, el defecto 4 alcanza un valor de SNR cercano a 2,8, detectado mediante SC_3, mientras que el defecto 6 presenta un valor de SNR aproximado de 2,6, detectado mediante SC_4. Finalmente, el defecto 9 presenta un valor de SNR cercano a 2,4, identificado mediante el algoritmo Eof_4, correspondiente al método PCT.

Posteriormente, la Figura 4-11 muestra el análisis considerando únicamente los resultados con SNR > 1,5. En este caso se detectan los defectos 2, 4, 6 y 9, lo que indica que estos presentan un contraste térmico relativamente alto respecto al fondo. En este rango de detección destacan principalmente los algoritmos SC_3 y Eof_4, que logran identificar varios de estos defectos.

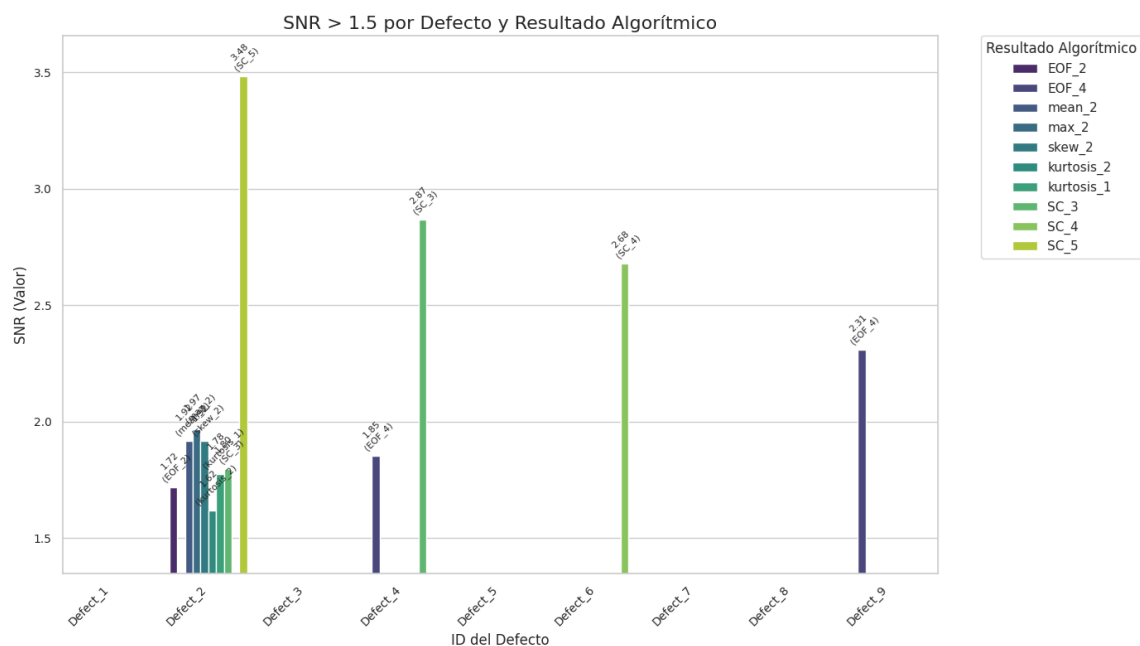


Figura 4-11 El análisis de los defectos que alcanzan valores SNR > 1,5 (Elaboración propia)

En la Figura 4-12 se presenta un análisis más restrictivo considerando SNR > 2,0. Bajo este criterio se mantienen los defectos 2, 4, 6 y 9, confirmando que estos defectos son los más detectables en estas condiciones experimentales. Los algoritmos que alcanzan los valores más elevados de SNR en este rango son SC_5, SC_3, SC_4 y EOF_4, en orden decreciente de rendimiento.

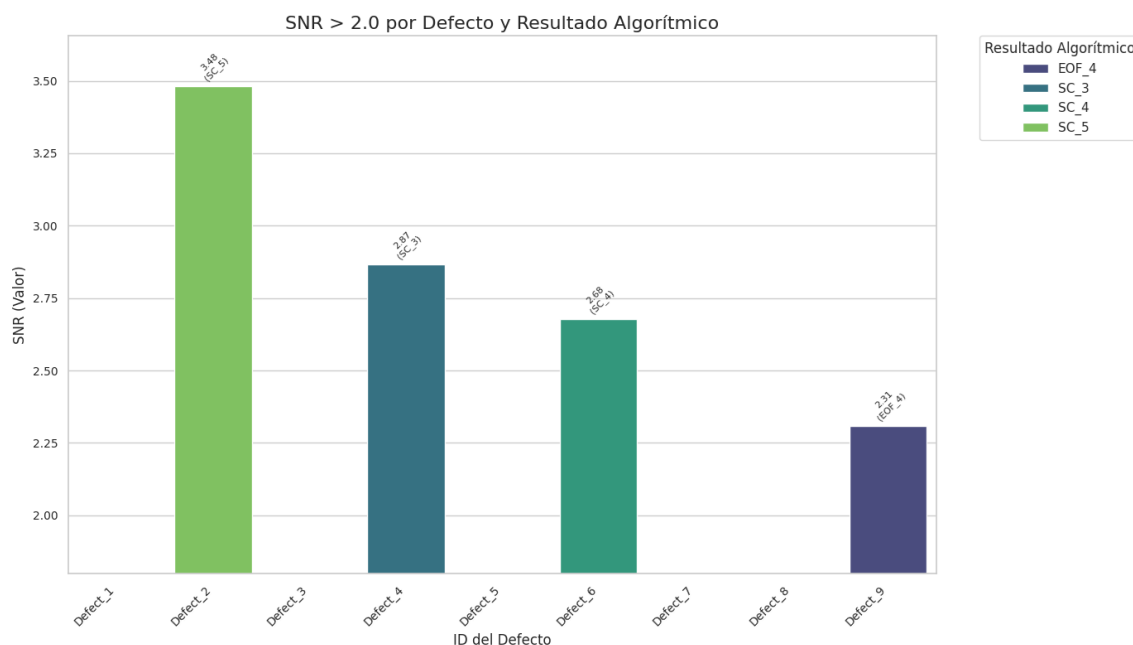


Figura 4-12 El análisis de los defectos que alcanzan valores SNR > 2 (Elaboración propia)

Para complementar el análisis cuantitativo, la Figura 4-13 presenta una comparación visual entre los resultados que mejores prestaciones arrojaron en cuanto a términos de SNR, es decir, mediante los algoritmos SC_3 y EOF_4. Estos dos algoritmos se han seleccionado para el análisis visual debido a que, aunque algunos algoritmos presentan valores de SNR ligeramente superiores, las imágenes resultantes muestran patrones que podrían estar asociados a ruido térmico. En cambio, los resultados obtenidos con SC_3 y EOF_4 permiten identificar los defectos de manera más clara y consistente,

mostrando una representación visual más fiable de las regiones defectuosas. A pesar de la introducción de la tapa, con SC_3 se sigue visualizando los defectos grandes y poco profundos (2, 4, 9), y lo mismo ocurre con EOF_4 aunque dichos defectos no se aprecien tan nítidos.

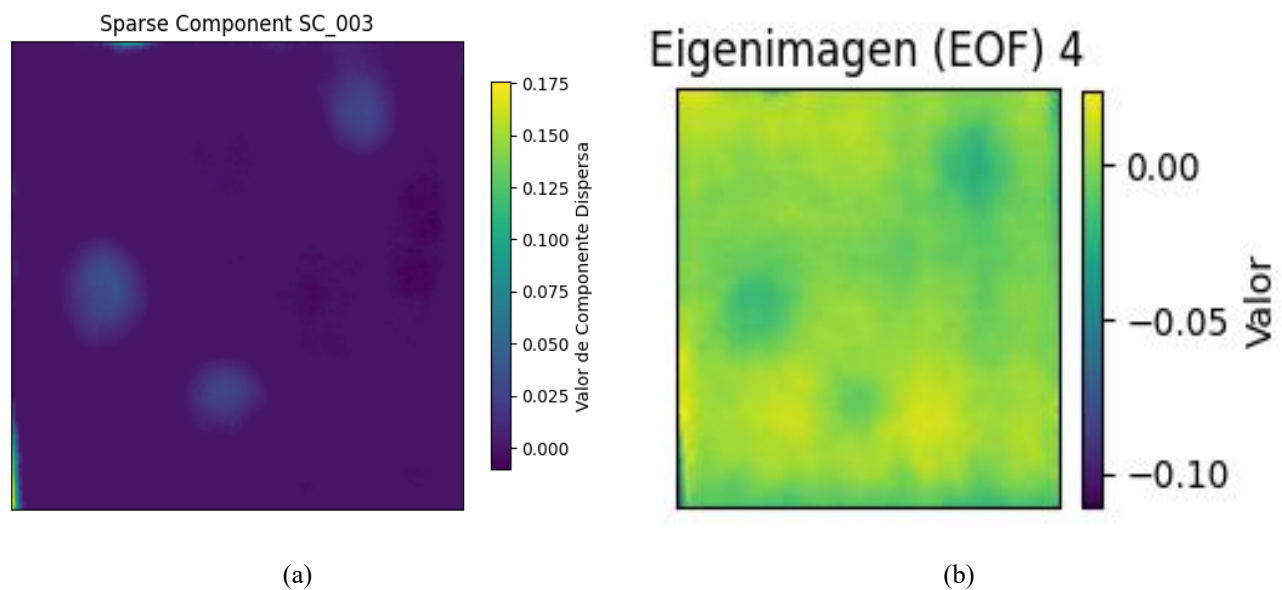


Figura 4-13 Comparación visual entre los resultados obtenidos a) SC_3 y b) EOF_4 (Elaboración propia)

En conjunto, estos resultados indican que, bajo condiciones de calentamiento con lámpara halógena con tapa, los algoritmos basados en SPCT continúan mostrando un buen rendimiento en la detección de defectos, especialmente mediante las componentes SC_3, SC_4 y SC_5. Sin embargo, también se observa que el algoritmo EOF_4 derivado de PCT puede proporcionar representaciones visuales útiles para la identificación de defectos, particularmente en aquellos casos donde la interpretación visual de las imágenes resulta crítica.

4.2.3 Caso de estudio 3: PETG-CF con lámpara infrarroja sin tapa

En primer lugar, la Figura 4-14 muestra la comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico. A partir de este análisis se observa que el valor más alto de SNR general corresponde al algoritmo SC_1, obtenido mediante el método SPCT, con un valor aproximado de 3,9.

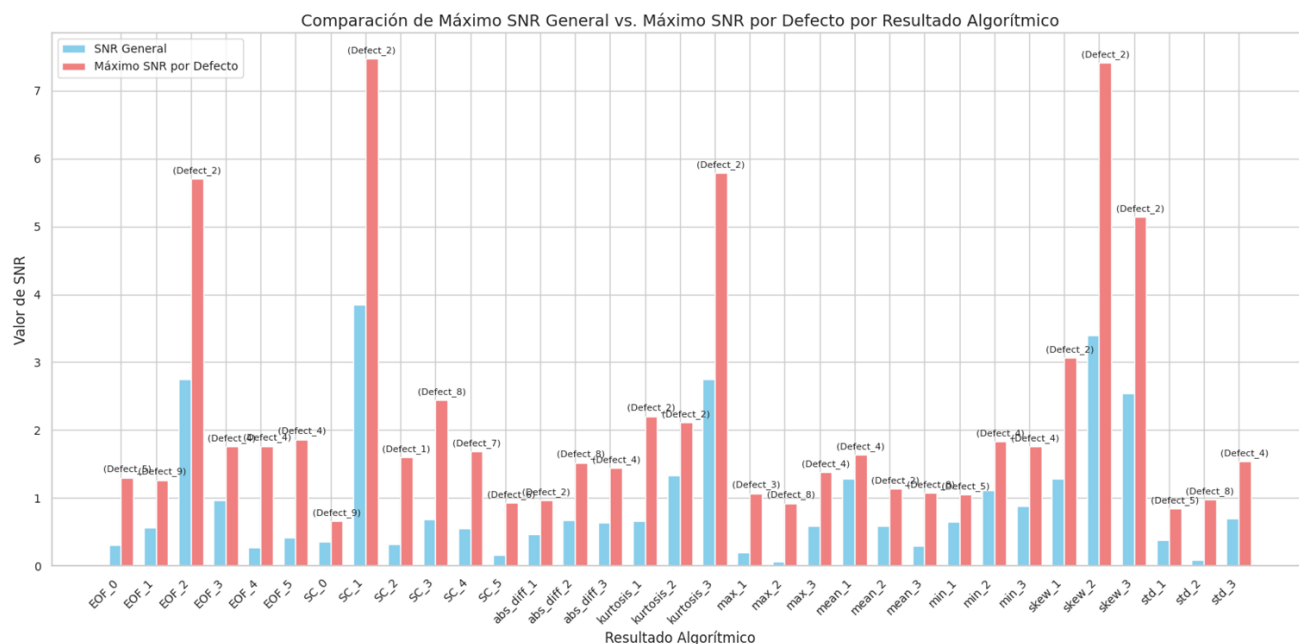


Figura 4-14 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)

Además, se ha evaluado la frecuencia de detección de los defectos a partir de los distintos resultados algorítmicos. Los defectos que aparecen con mayor frecuencia en las detecciones corresponden a los defectos 2, 4 y 8, lo que indica que estas regiones presentan un contraste térmico más significativo bajo las condiciones de calentamiento con lámpara infrarroja.

La Figura 4-15 presenta el gráfico de Máximo SNR por defecto con el algoritmo correspondiente. En este caso se analizan los defectos que presentan valores de SNR superiores a 1, considerados como detecciones fiables. En los resultados obtenidos se observa que todos los defectos pueden ser detectados bajo este criterio.

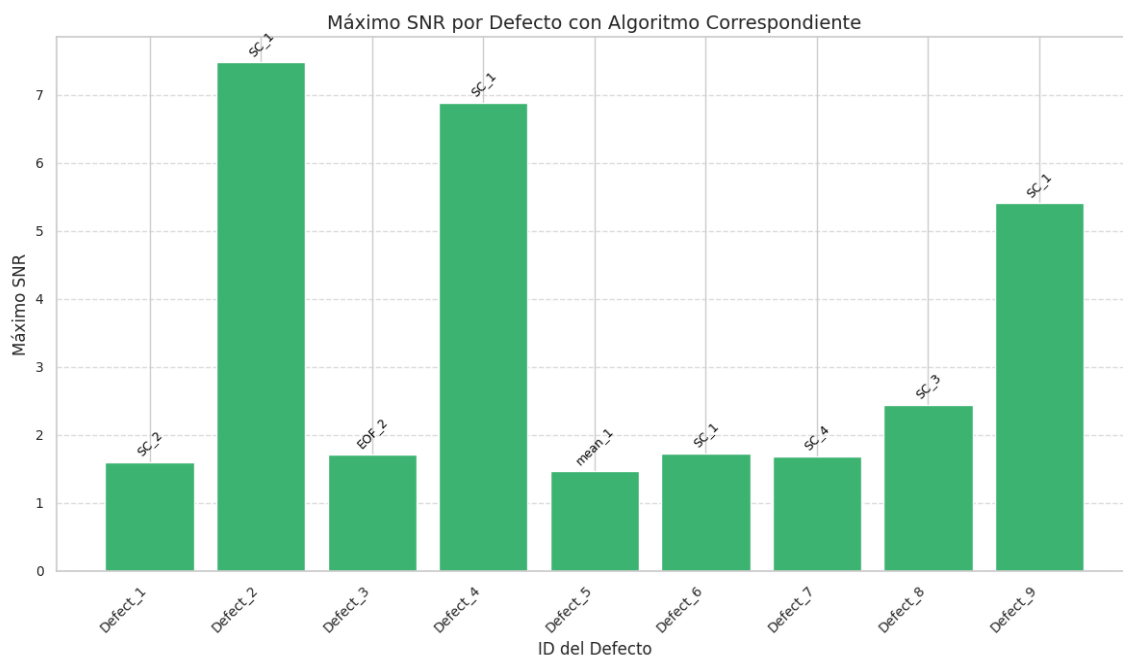


Figura 4-15 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

Entre ellos, los defectos 2, 4 y 9 presentan los valores de SNR más elevados, aproximadamente 7,5, 6,9 y 5,4, respectivamente. En los tres casos, estos valores se obtienen mediante el algoritmo SC_1, lo que confirma el alto rendimiento de este algoritmo en la identificación de defectos bajo estas condiciones experimentales.

Adicionalmente, el defecto 8 presenta un valor de SNR cercano a 2,3, detectado mediante el algoritmo SC_3, lo que también indica una detección clara de este defecto.

Posteriormente, la Figura 4-16 muestra el análisis considerando únicamente los resultados con $SNR > 1,5$. Bajo este criterio se detectan los defectos 2, 4, 8 y 9, los cuales presentan un contraste térmico suficientemente alto respecto al fondo. Asimismo, se ha observado que los algoritmos que detectan estos defectos con mayor frecuencia son SC_1 y skew_2, siendo este último derivado del método TSR.

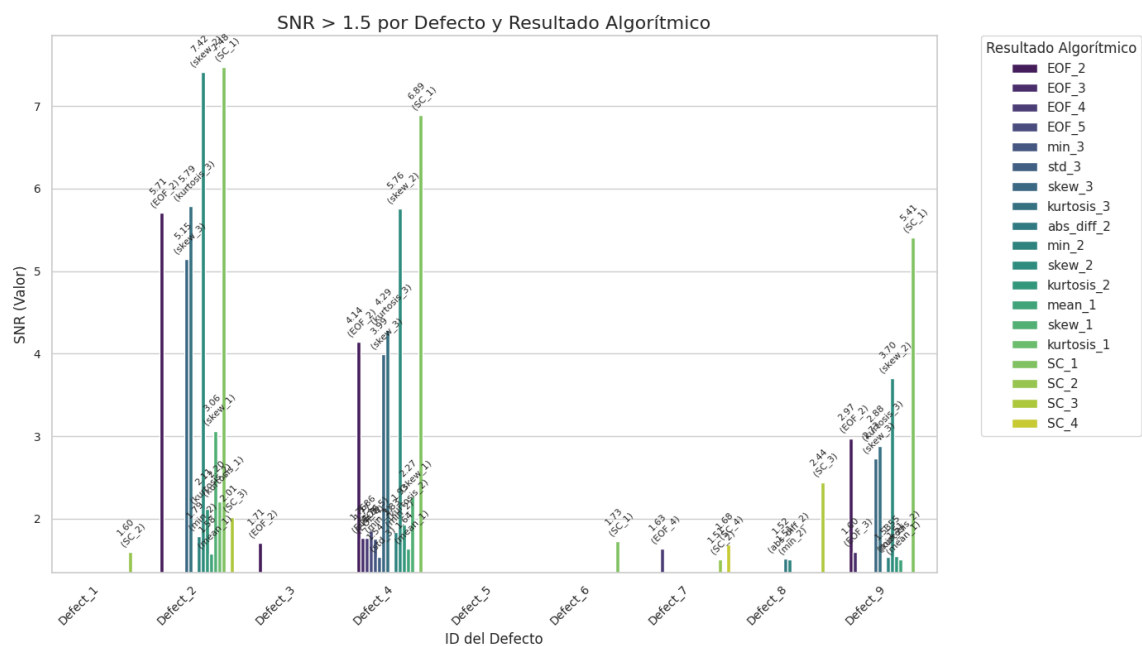


Figura 4-16 El análisis de los defectos que alcanzan valores $SNR > 1,5$ (Elaboración propia)

En la Figura 4-17 se presenta un análisis más restrictivo considerando $SNR > 2,0$. En este caso se mantienen los mismos defectos detectados (2, 4, 8 y 9), lo que confirma que estos defectos corresponden a las discontinuidades más claramente identificables en el experimento. De nuevo, los algoritmos que presentan un mayor número de detecciones son SC_1 y skew_2, destacando especialmente el primero por alcanzar los valores de SNR más elevados.

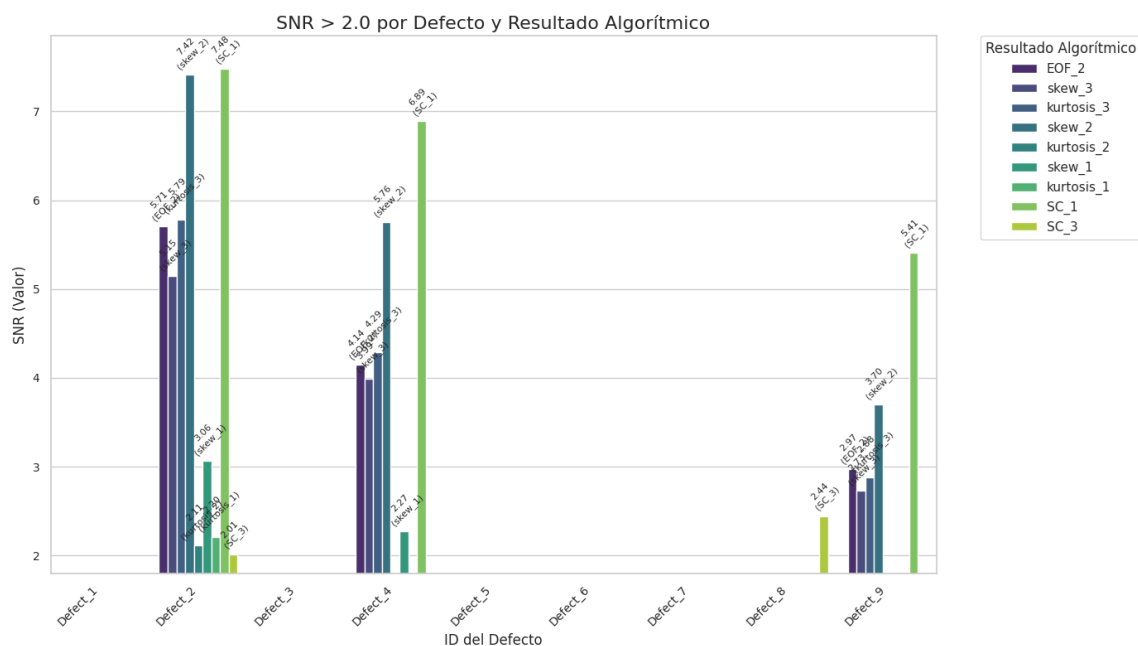


Figura 4-17 El análisis de los defectos que alcanzan valores SNR > 2 (Elaboración propia)

Finalmente, la Figura 4-18 muestra una comparación visual entre los resultados obtenidos mediante los algoritmos SC_1 y skew_2, por ser los algoritmos con mejores prestaciones en cuanto al SNR. En estas imágenes se puede observar que ambos algoritmos permiten identificar los defectos grandes y poco profundos de manera clara (2, 4 y 9), mostrando un contraste térmico bien definido en las regiones correspondientes a las discontinuidades. Además, los defectos medianos con profundidad intermedia (3, 5, 7) son ligeramente apreciables con skew_2. Esta comparación visual confirma los resultados obtenidos en el análisis cuantitativo basado en SNR.

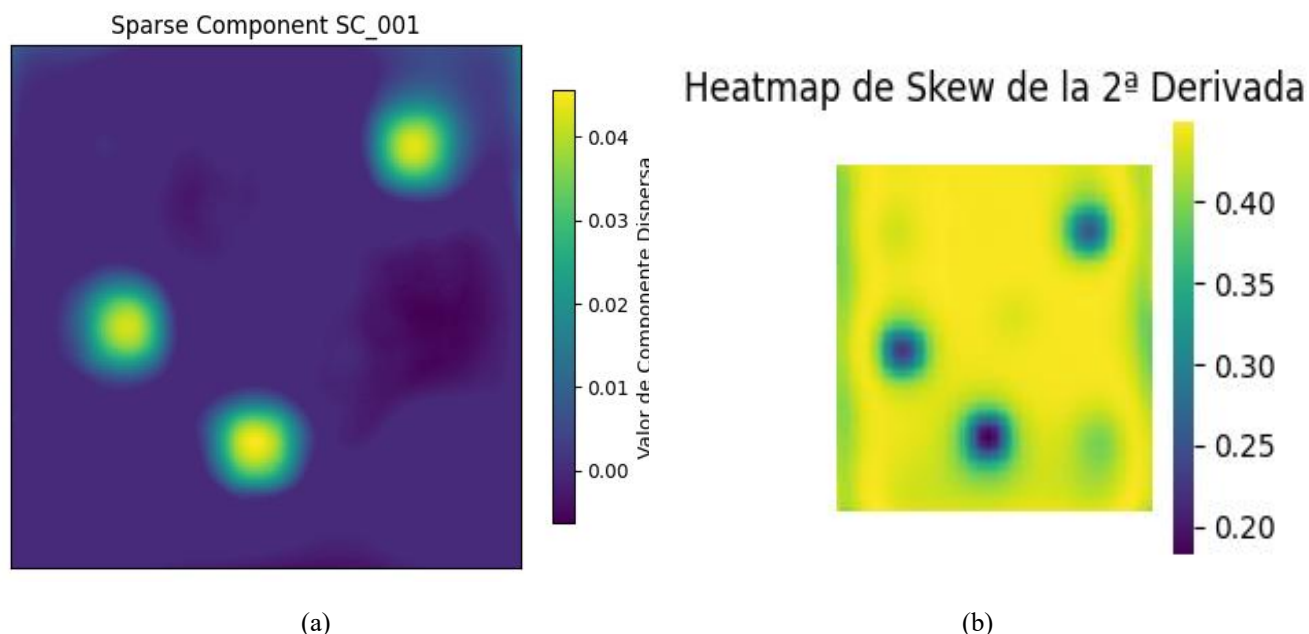


Figura 4-18 Comparación visual entre los resultados obtenidos a) SC_1 y b) skew_2 (Elaboración propia)

En conjunto, los resultados indican que el algoritmo SC_1, basado en el método SPCT, presenta el mejor rendimiento en términos de detección de defectos y contraste térmico para el caso de calentamiento mediante lámpara infrarroja sin tapa. Asimismo, el algoritmo skew_2, derivado del

método TSR, también muestra un comportamiento destacado como técnica complementaria para la identificación de defectos.

4.2.4 Caso de estudio 4: PETG-CF con lámpara infrarroja con tapa

En primer lugar, la Figura 4-19 muestra la comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico. A partir de este gráfico se observa que el valor más alto de SNR general corresponde al algoritmo SC_1, obtenido mediante el método SPCT, con un valor aproximado de 1.8.

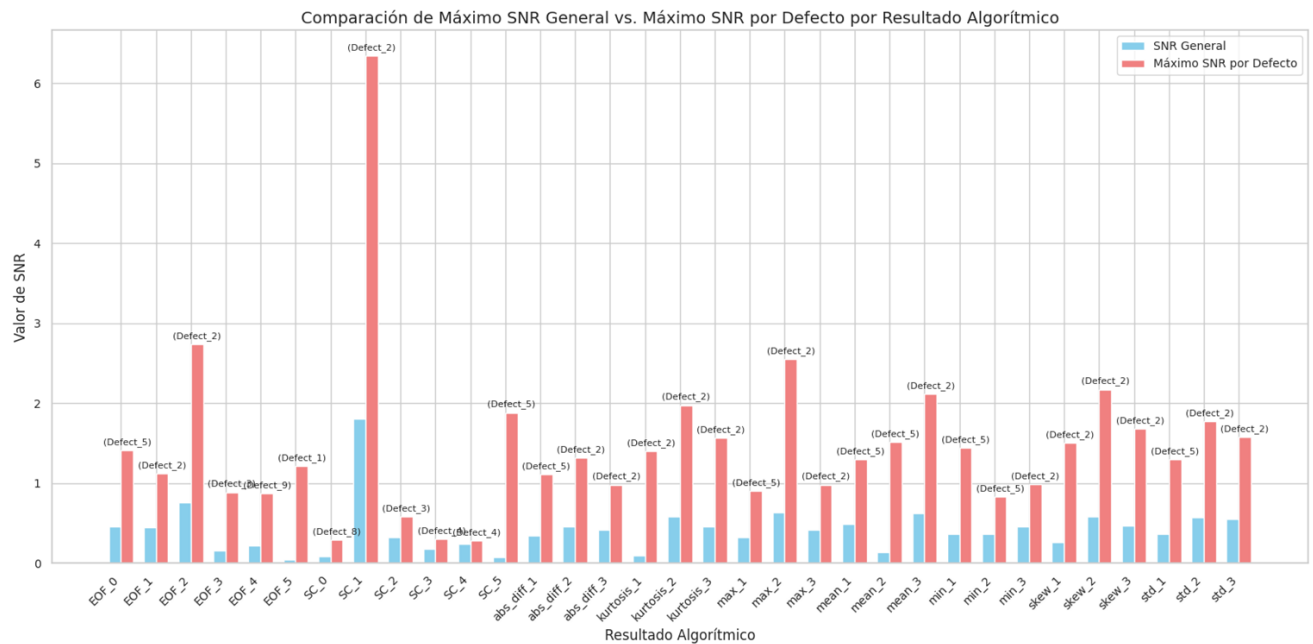


Figura 4-19 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)

Además, se ha evaluado la frecuencia de detección de defectos en los distintos resultados algorítmicos. En este caso, los defectos detectados con mayor frecuencia corresponden a los defectos 2 y 5, lo que indica que estas zonas presentan una respuesta térmica más clara bajo las condiciones experimentales consideradas.

La Figura 4-20 presenta el gráfico de Máximo SNR por defecto con el algoritmo correspondiente. En este análisis se identifican los defectos que presentan valores de SNR superiores a 1, considerados como indicativos de una detección fiable. Bajo este criterio se detectan los defectos 1, 2, 3, 4, 5, 6 y 8.

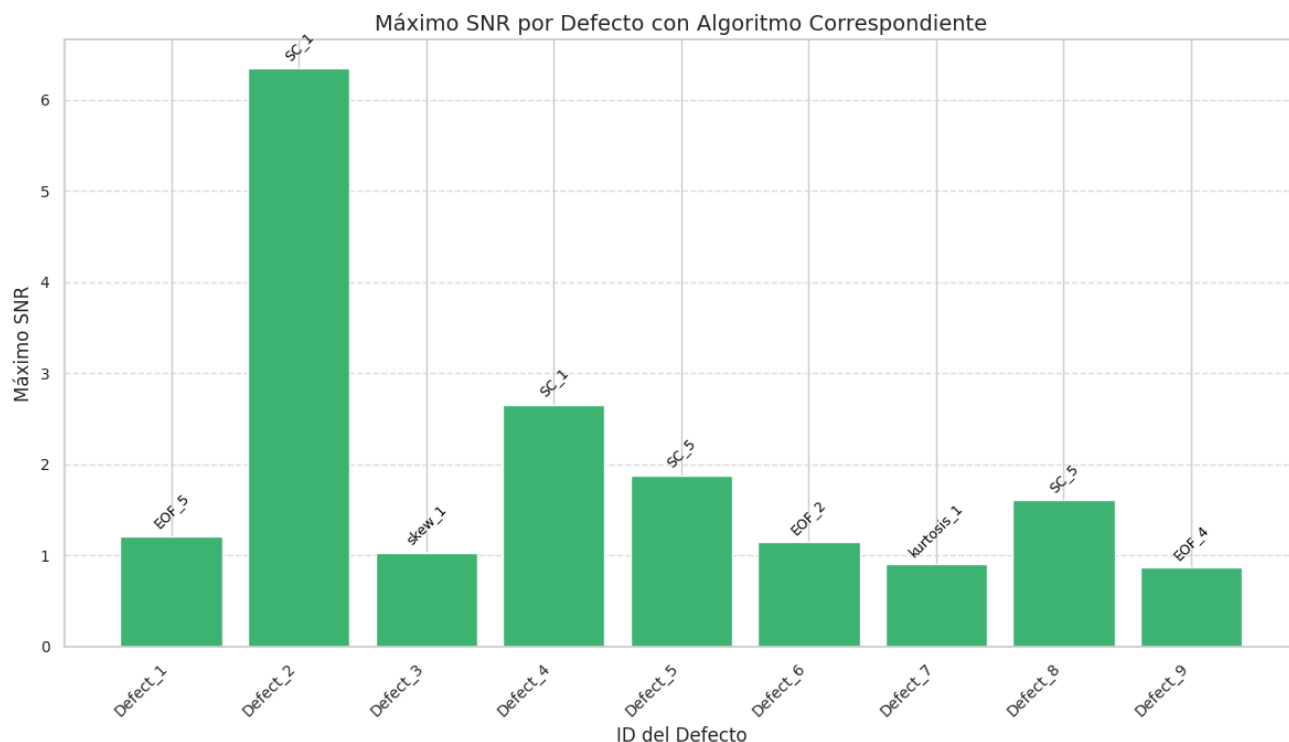


Figura 4-20 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

Entre ellos, el defecto 2 presenta el valor más alto de SNR, aproximadamente 6,3, seguido por el defecto 4, con un valor cercano a 2,8. Ambos defectos son detectados mediante el algoritmo SC_1, lo que confirma su buen rendimiento en este experimento. Por otra parte, el defecto 5 presenta un valor de SNR cercano a 1,9, mientras que el defecto 8 alcanza aproximadamente 1,6, ambos detectados mediante el algoritmo SC_5.

Posteriormente, la Figura 4-21 muestra el análisis considerando únicamente los resultados con $SNR > 1,5$. En este caso se detectan los defectos 2, 4, 5 y 8, lo que indica que estas regiones presentan un contraste térmico suficientemente elevado respecto al fondo. Asimismo, se observa que los algoritmos que detectan estos defectos con mayor frecuencia son SC_1 y SC_5.

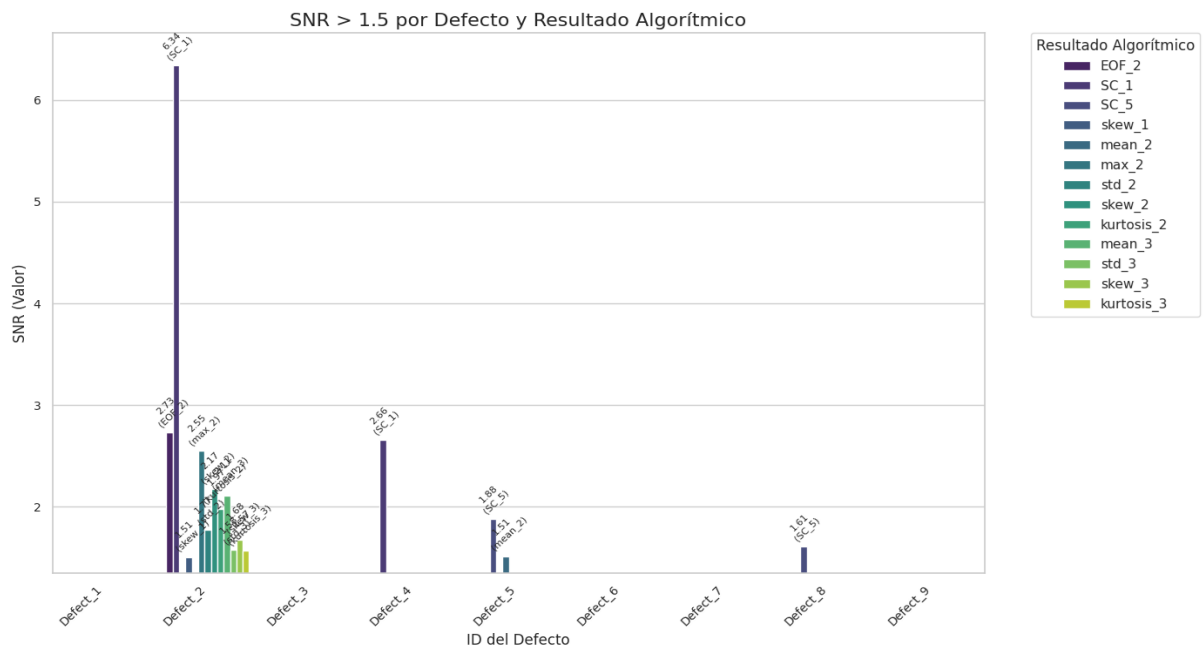


Figura 4-21 El análisis de los defectos que alcanzan valores SNR > 1,5 (Elaboración propia)

En la Figura 4-22 se presenta un análisis más restrictivo considerando $SNR > 2,0$. Bajo este criterio únicamente se detectan los defectos 2 y 4, lo que indica que estos defectos presentan el mayor contraste térmico en las condiciones experimentales analizadas. En este caso, el algoritmo que alcanza el valor más alto de SNR corresponde nuevamente a SC_1.

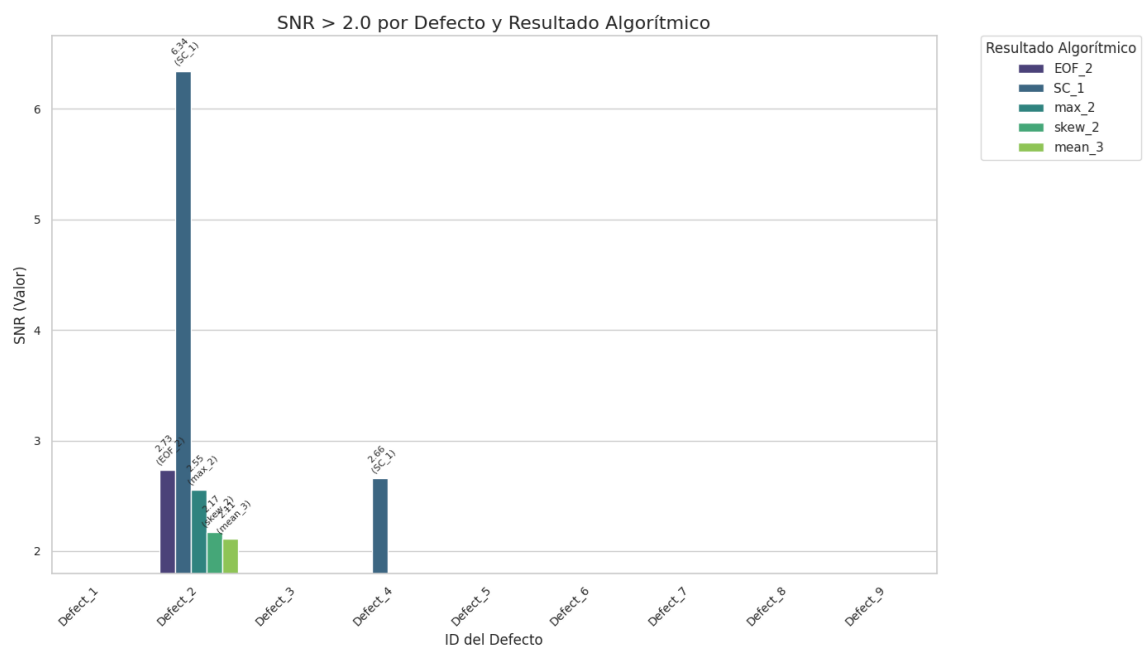


Figura 4-22 El análisis de los defectos que alcanzan valores SNR > 2 (Elaboración propia)

Para complementar el análisis cuantitativo, la Figura 4-23 muestra una comparación visual entre los resultados obtenidos mediante los algoritmos SC_1 y SC_5, los que mejores datos arrojaron en términos de SNR. En estas imágenes se observa que ambos algoritmos permiten identificar los defectos grandes y poco profundos (2, 4 y 9), a pesar del bajo SNR en la 9. El alto SNR en el defecto 8 se asocia a ruido, de ahí que además del análisis cuantitativo con el SNR, hay que apoyarse a un análisis cualitativo visualizando dichas imágenes.

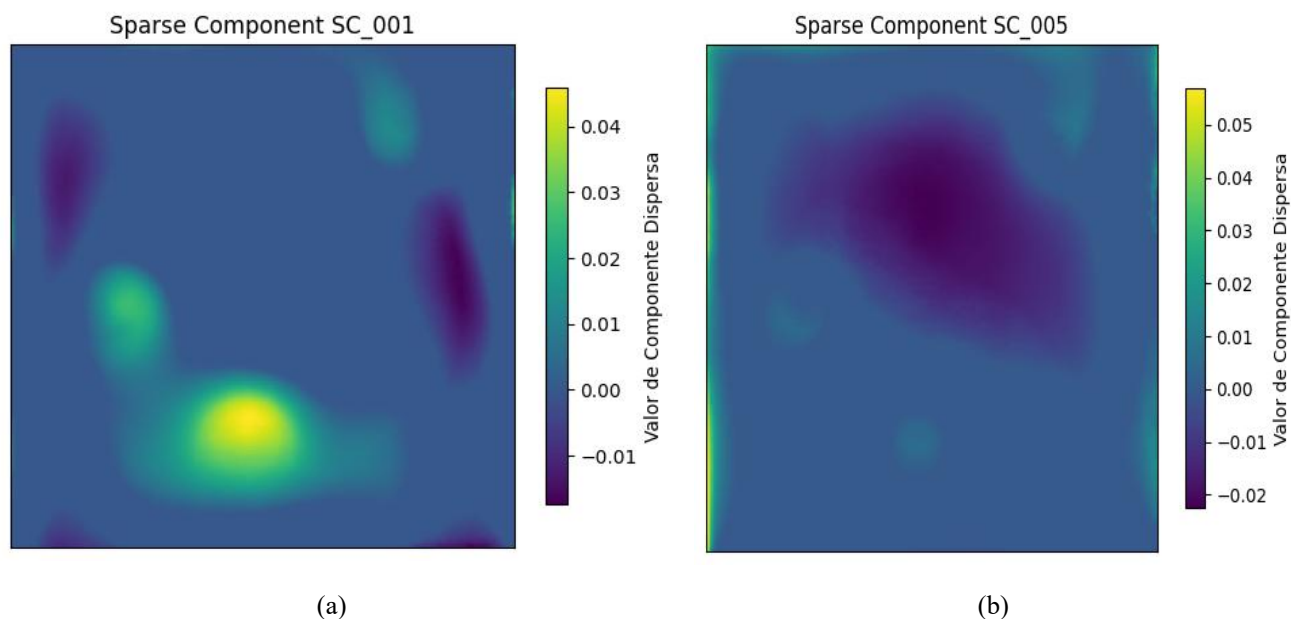


Figura 4-23 Comparación visual entre los resultados obtenidos a) SC_1 y b) SC_5 (Elaboración propia)

Adicionalmente, la Figura 4-24 presenta el análisis de varianza explicada por cada componente disperso (SPCT). En este caso se observa que las componentes SC_1 y SC_5 presentan valores aceptables de varianza explicada, lo que indica que contienen una parte significativa de la información relevante del proceso térmico. Por otro lado, la componente SC_0 podría estar asociada principalmente a la huella térmica dejada por la propia lámpara.

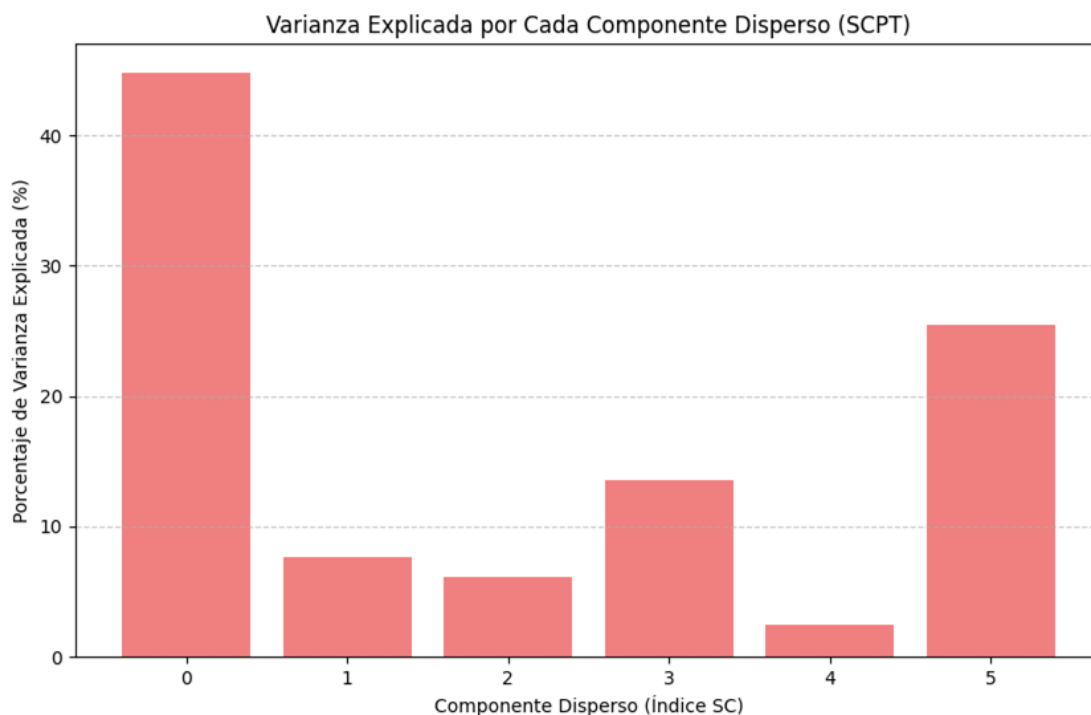


Figura 4-24 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)

Finalmente, la Figura 4-25 muestra la evolución temporal de los coeficientes dispersos (SCPT). En este gráfico se puede observar que las curvas correspondientes a SC_1 y SC_5 presentan una tendencia similar a lo largo del tiempo, lo que sugiere que ambas componentes están relacionadas con la dinámica

térmica generada durante el proceso del ensayo, es decir, relacionadas con la dinámica térmica de los defectos.

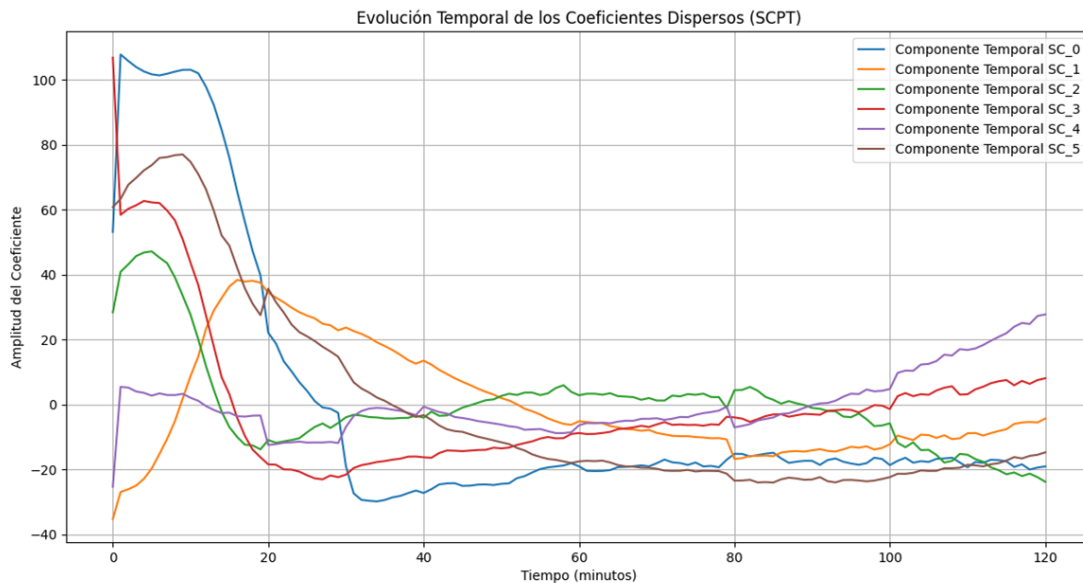


Figura 4-25 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia)

En conjunto, estos resultados indican que el algoritmo SC_1, basado en el método SPCT, ofrece el mejor rendimiento para la detección de defectos en el caso de calentamiento mediante lámpara infrarroja con tapa, mientras que SC_5 también proporciona información útil para la identificación de ciertas discontinuidades.

4.2.5 Comparación de los resultados en el PETG-CF

En esta sección se realiza una comparación global de los resultados obtenidos en los cuatro experimentos realizados con la muestra de PETG-CF, considerando las distintas configuraciones de calentamiento: lámpara halógena e infrarroja, así como las condiciones con tapa y sin tapa. El objetivo de este análisis es identificar qué condiciones experimentales y qué algoritmos ofrecen un mejor rendimiento en la detección de defectos mediante el análisis del SNR y la inspección cualitativa.

En primer lugar, se observa que los algoritmos basados en SPCT, particularmente SC_1, presentan el mejor rendimiento global en la mayoría de los experimentos. Este algoritmo proporciona sistemáticamente los valores más elevados de SNR, lo que indica una mayor capacidad para resaltar el contraste térmico entre los defectos y el fondo. Además, otras componentes como SC_3, SC_4 y SC_5 también muestran resultados relevantes en algunos casos específicos, contribuyendo a la detección de determinados defectos, junto al EOF_4 del PCT y skew_2 del TSR.

En cuanto a las condiciones de calentamiento, los resultados indican que los experimentos realizados sin tapa tienden a presentar valores de SNR más elevados, especialmente en el caso del calentamiento mediante lámpara infrarroja sin tapa, donde se obtienen algunos de los valores más altos de SNR observados en todo el estudio. Esto sugiere que la ausencia de tapa, y con el uso de lámparas de mayor potencia, puede favorecer un contraste térmico más marcado en la superficie de la muestra, facilitando así la detección de defectos.

Por otro lado, en las configuraciones con tapa, aunque los valores de SNR general tienden a ser menores, los defectos siguen siendo detectables mediante determinados algoritmos. En estos casos, las componentes derivadas de SPCT, como SC_1 y SC_5, continúan mostrando un comportamiento robusto en la identificación de defectos, aunque con un contraste térmico ligeramente inferior.

Asimismo, se observa que algunos defectos aparecen de forma recurrente en todos los experimentos, particularmente los defectos 2, 4 y 9 (los de tamaño grande y poco profundos). Esto indica que estos

defectos generan una perturbación térmica más significativa, lo que facilita su identificación mediante diferentes técnicas de procesamiento. Incluso en los defectos de tamaño mediano y profundidad intermedia (3, 5 y 7) se detectan sus huellas térmicas con la base sin tapa y con ambas lámparas.

Finalmente, el análisis visual de las imágenes resultantes confirma los resultados obtenidos mediante la métrica SNR, mostrando que los algoritmos basados en SPCT, especialmente SC_1, permiten identificar los defectos de manera más clara y consistente en comparación con otros métodos.

En conjunto, los resultados obtenidos para la muestra de PETG-CF indican que el método SPCT, y en particular la componente SC_1, proporciona el mejor rendimiento para la detección de defectos en los distintos escenarios experimentales analizados. Además, el calentamiento mediante lámpara infrarroja sin tapa parece ofrecer las condiciones más favorables para maximizar el contraste térmico y mejorar la detectabilidad de los defectos. La Tabla 4-1 presenta un resumen de los resultados más relevantes obtenidos en cada configuración experimental. En particular, se indican los algoritmos que han proporcionado los valores más elevados de SNR, así como los defectos detectados con mayor claridad en cada caso.

Configuración experimental	Algoritmo más efectivo	Defectos detectados	SNR máximo aproximado
Halógena sin tapa	SC_1	2, 4, 9	~7
Halógena con tapa	SC_3 / SC_5	2, 4, 6	~3,5
Infrarroja sin tapa	SC_1	2, 4, 9	~7,5
Infrarroja con tapa	SC_1	2, 4	~6

Tabla 4-1 Resumen comparativo de los resultados obtenidos para PETG-CF en las distintas configuraciones experimentales (Elaboración propia)

4.3 Resultados en PLA

4.3.1 Caso de estudio 5: PLA con lámpara halógena sin tapa

En la comparación global del SNR, se observa que el valor máximo de SNR general corresponde al algoritmo SC_0, derivado del método SPCT, con un valor aproximado de 3,7. Asimismo, los defectos detectados con mayor frecuencia son los defectos 2 y 8, lo que indica una mayor sensibilidad del sistema en estas localizaciones. La Figura 4-26 muestra la comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico.

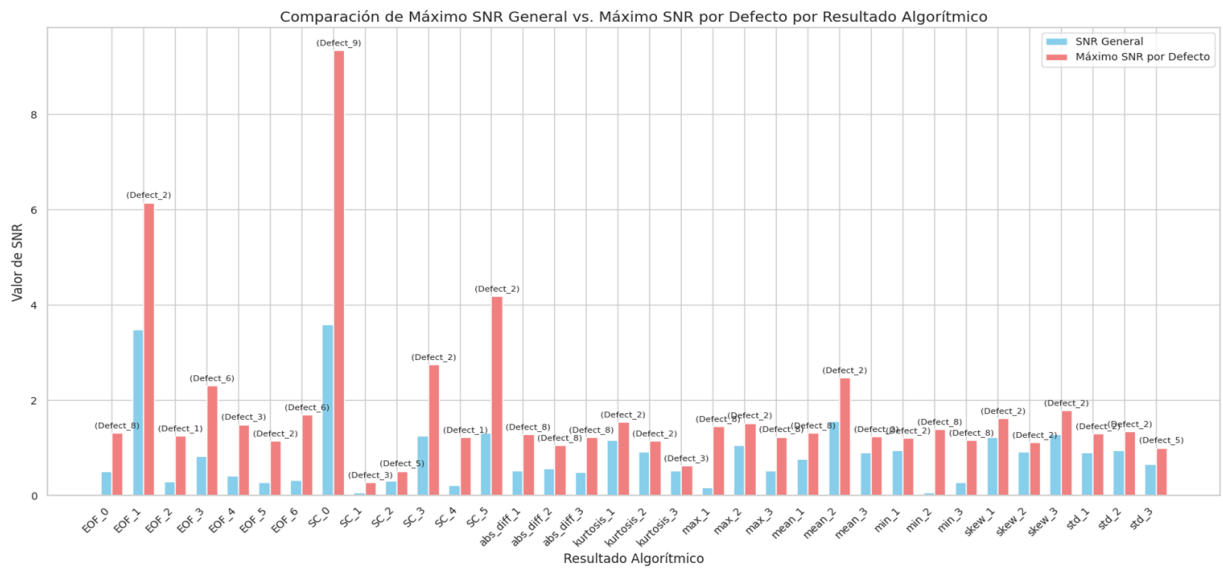


Figura 4-26 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)

En el análisis del máximo SNR por defecto, se observa que todos los defectos son detectados con valores de SNR superiores a 1. Destaca especialmente el defecto 9, que alcanza el valor máximo de SNR $\approx 8,6$, detectado mediante el algoritmo SC_0. Además, los defectos 2 y 4 presentan valores elevados ($\approx 6,1$ y $5,7$, respectivamente), siendo detectados mediante EOF_1, mientras que el defecto 6 alcanza un valor de SNR $\approx 4,7$, detectado mediante SC_5. La Figura 4-27 presenta el gráfico de Máximo SNR por defecto con el algoritmo correspondiente.

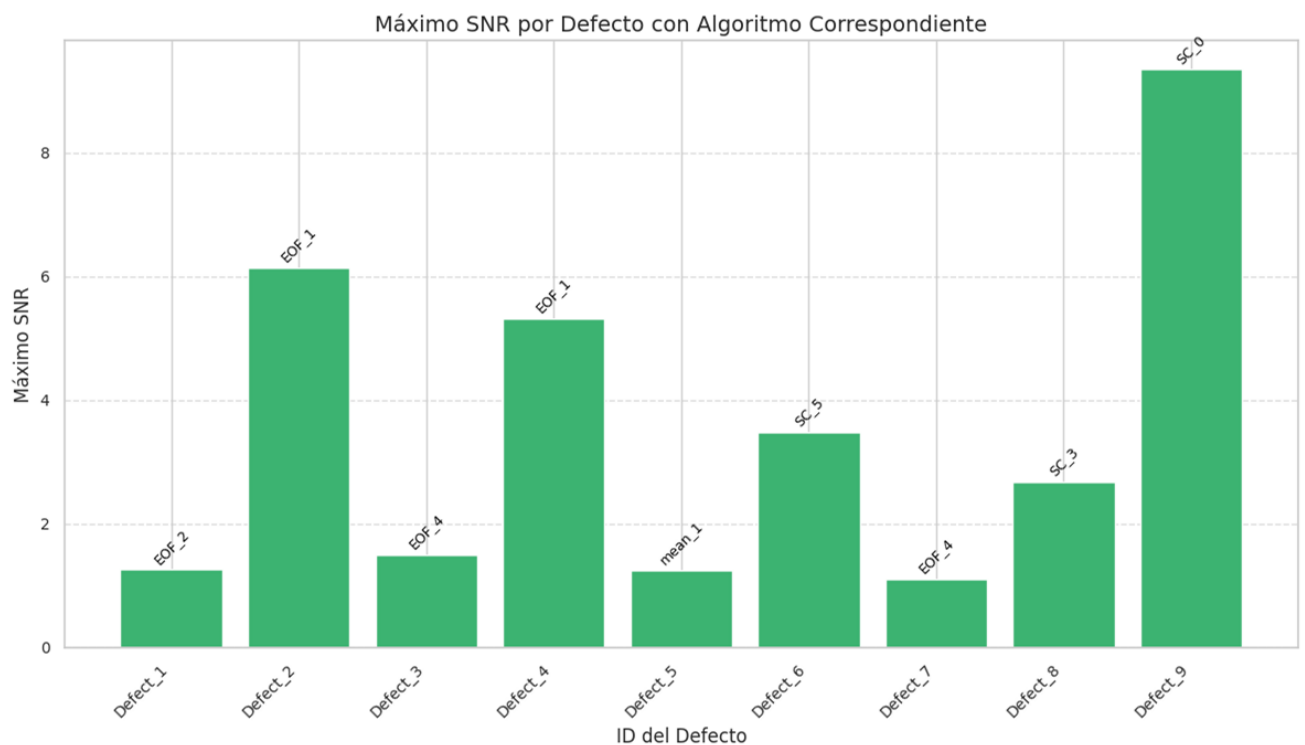


Figura 4-27 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

La Figura 4-28 muestra el análisis considerando únicamente los resultados con SNR $> 1,5$, los defectos detectados son los 2, 4, 6, 8 y 9. En este caso, los algoritmos con mejor rendimiento son SC_0, SC_5 y EOF_1, destacando especialmente los métodos basados en SPCT.

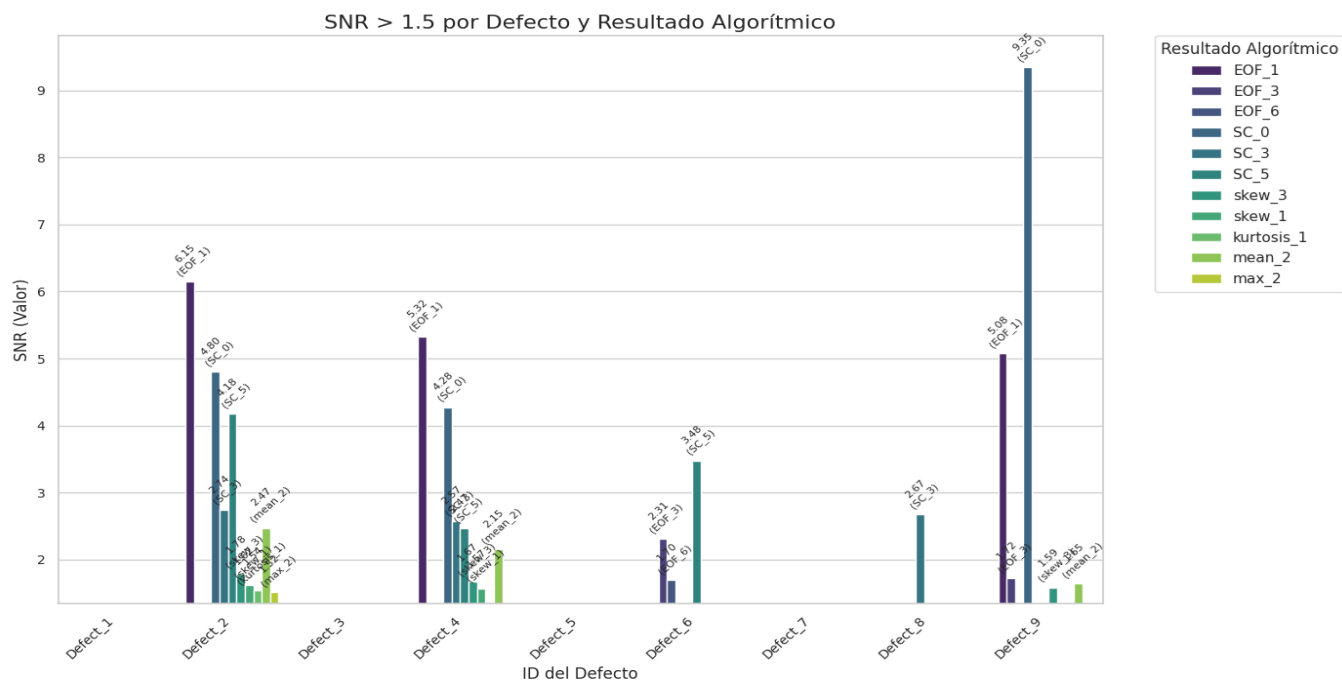


Figura 4-28 El análisis de los defectos que alcanzan valores SNR > 1,5 (Elaboración propia)

Al aumentar el umbral a SNR > 2,0, se mantienen los mismos defectos detectados 2, 4, 6, 8 y 9, observándose que los algoritmos SC_0 y EOF_1 presentan un rendimiento equivalente en términos de detección. En la Figura 4-29 se presenta un análisis más restrictivo considerando SNR > 2,0.

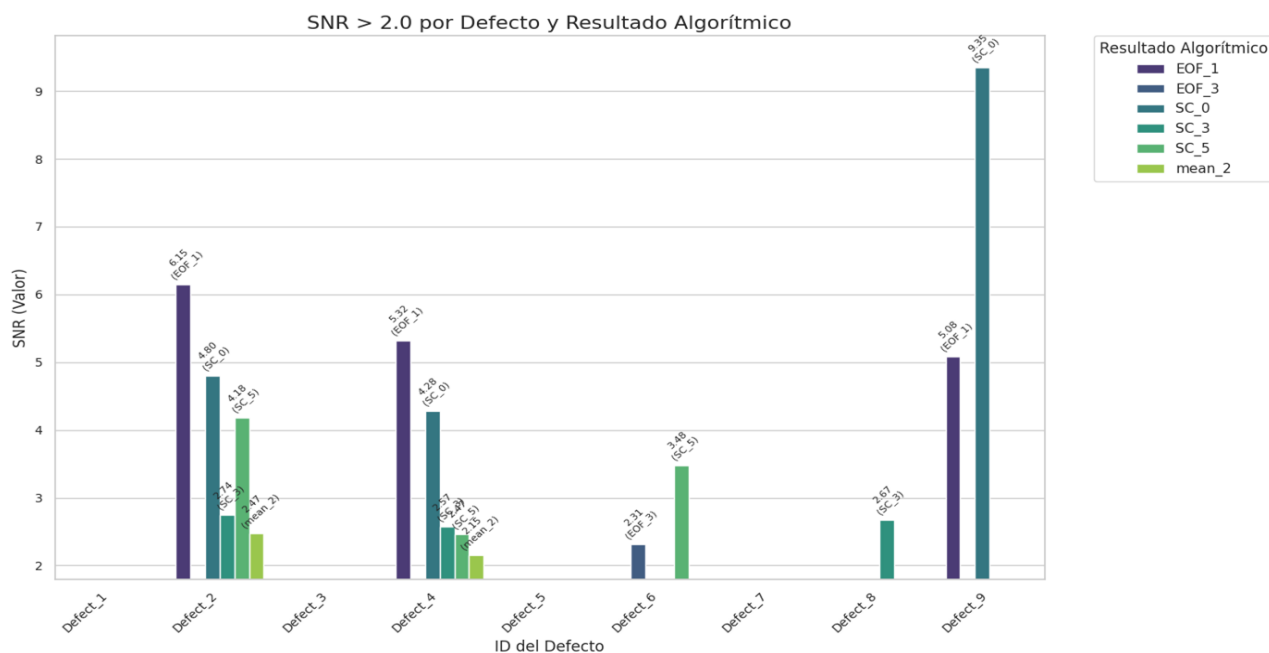


Figura 4-29 El análisis de los defectos que alcanzan valores SNR > 2 (Elaboración propia)

Desde el punto de vista cualitativo, las imágenes procesadas muestran que los defectos son claramente identificables mediante SC_0 (defectos grandes y poco profundos: 2, 4 y 9) y EOF_1 (defectos medianos y profundidad media: 3, 5 y 7; defectos grandes y poco profundos: 2, 4 y 9), mientras que SC_5, aunque presenta menor SNR en algunos casos, permite igualmente una identificación visual aceptable. La SNR de los defectos 6 y 8 se vinculan a ruido. la Figura 4-30 muestra una comparación visual entre los resultados obtenidos mediante los distintos algoritmos.

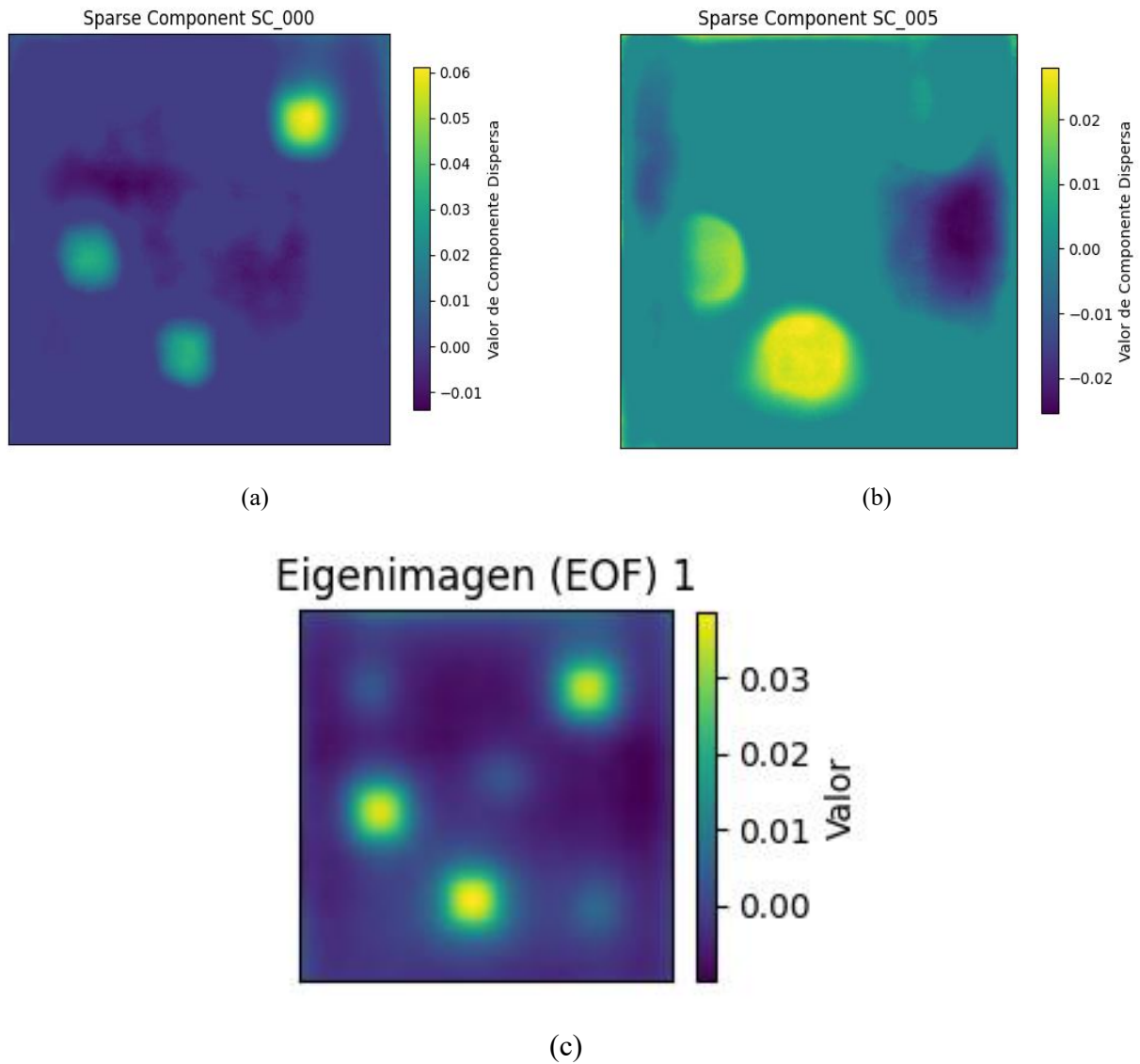


Figura 4-30 Comparación visual entre los resultados obtenidos a) SC_1, b) SC_5 y c) EOF_1 (Elaboración propia)

El análisis de la varianza explicada indica que las componentes SC_0 y SC_5 concentran una parte significativa de la información relevante del sistema. La Figura 4-31 presenta el análisis de varianza explicada por cada componente disperso (SPCT).

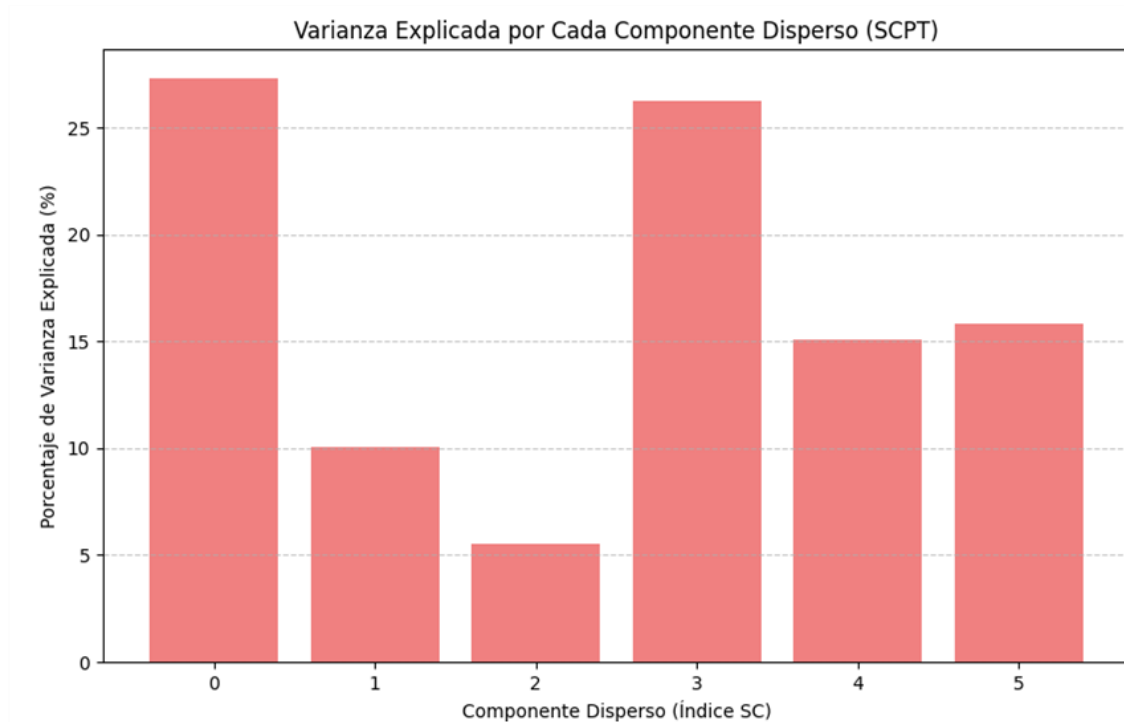


Figura 4-31 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)

Finalmente, la evolución temporal de los coeficientes dispersos muestra que SC_0 y SC_5 presentan tendencias similares, lo que sugiere una respuesta térmica coherente a la aparición y aumento de las huellas térmicas de los defectos identificados. La Figura 4-32 muestra la evolución temporal de los coeficientes dispersos (SCPT).

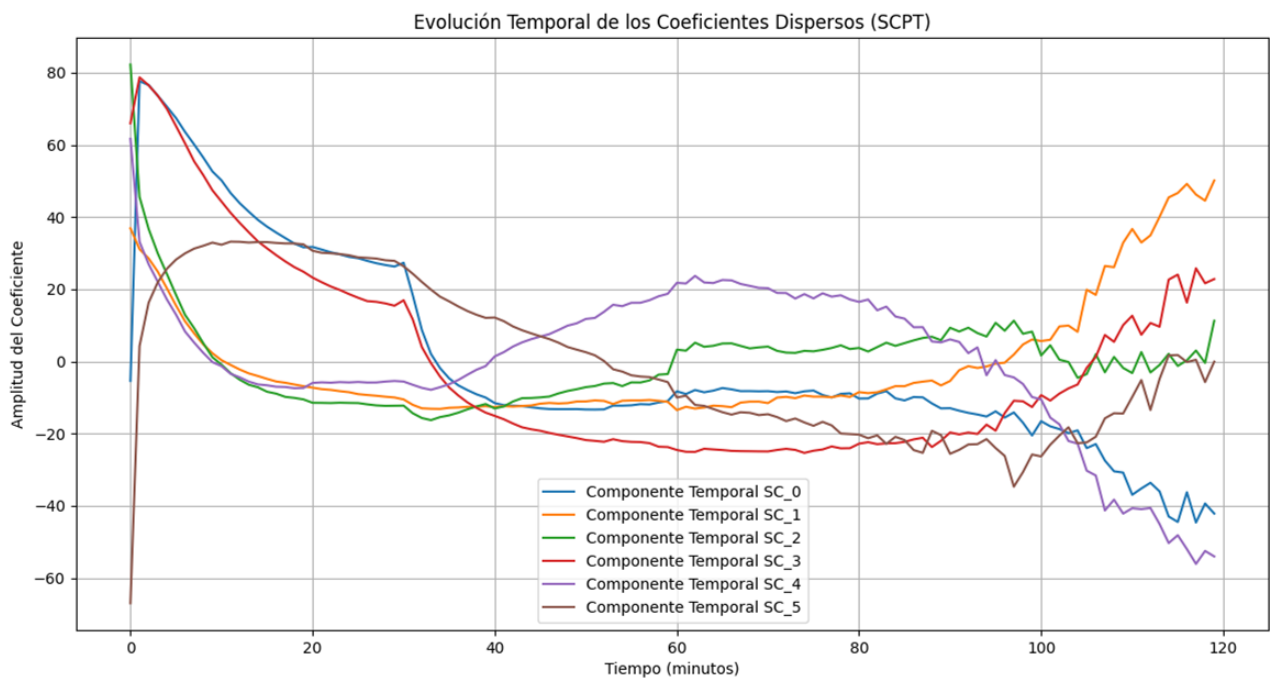


Figura 4-32 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia)

En resumen, la configuración de lámpara halógena sin tapa proporciona un alto rendimiento en la detección de defectos en PLA, destacando el algoritmo SC_0 (SPCT) como el más eficaz a nivel global, aunque para defectos tamaño y profundidad media el EOF_1 se sitúa como el más aconsejable.

4.3.2 Caso de estudio 6: PLA con lámpara halógena con tapa

En primer lugar, la Figura 4-33 muestra la comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico. A partir de este gráfico se observa que el valor más alto de SNR general corresponde al algoritmo SC_3, obtenido mediante el método SPCT, con un valor aproximado de 0,8.

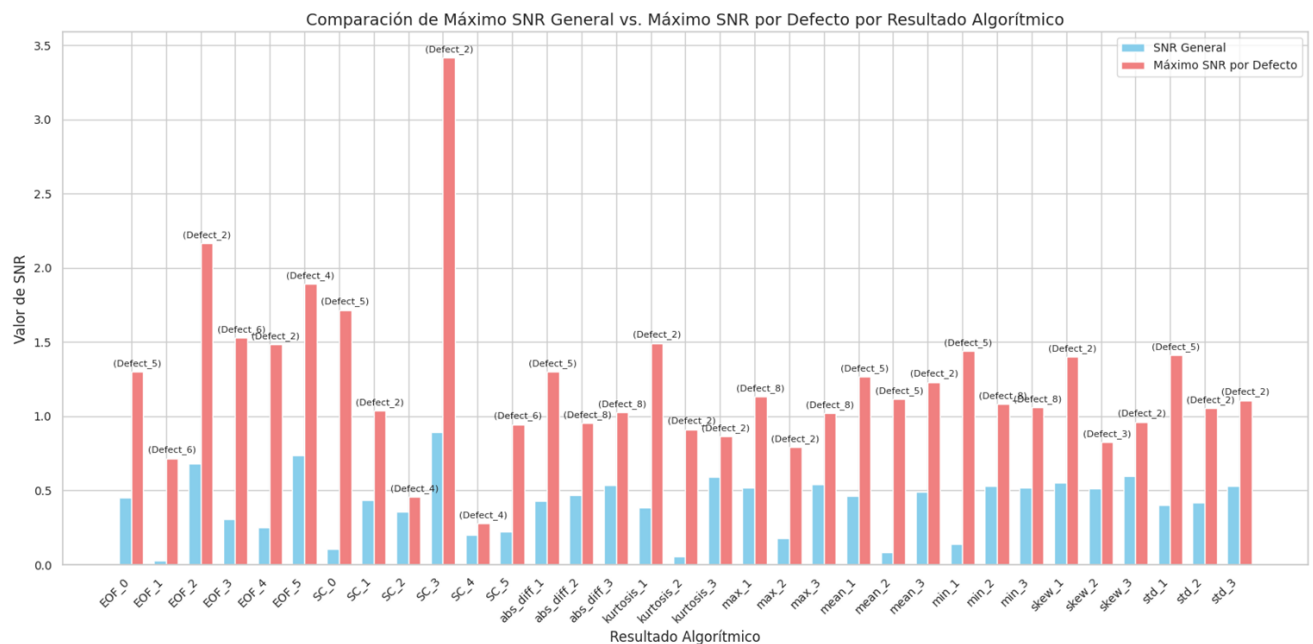


Figura 4-33 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)

Además, se ha evaluado la frecuencia de detección de defectos en los distintos resultados algorítmicos. En este caso, los defectos detectados con mayor frecuencia corresponden a los defectos 2 y 5, lo que indica que estas zonas presentan una respuesta térmica más clara bajo las condiciones experimentales consideradas.

La Figura 4-34 presenta el gráfico de Máximo SNR por defecto con el algoritmo correspondiente. En este análisis se identifican los defectos que presentan valores de SNR superiores a 1, considerados como indicativos de una detección fiable. Bajo este criterio se detectan los defectos 1, 2, 4, 5, 6, 7, 8 y 9.

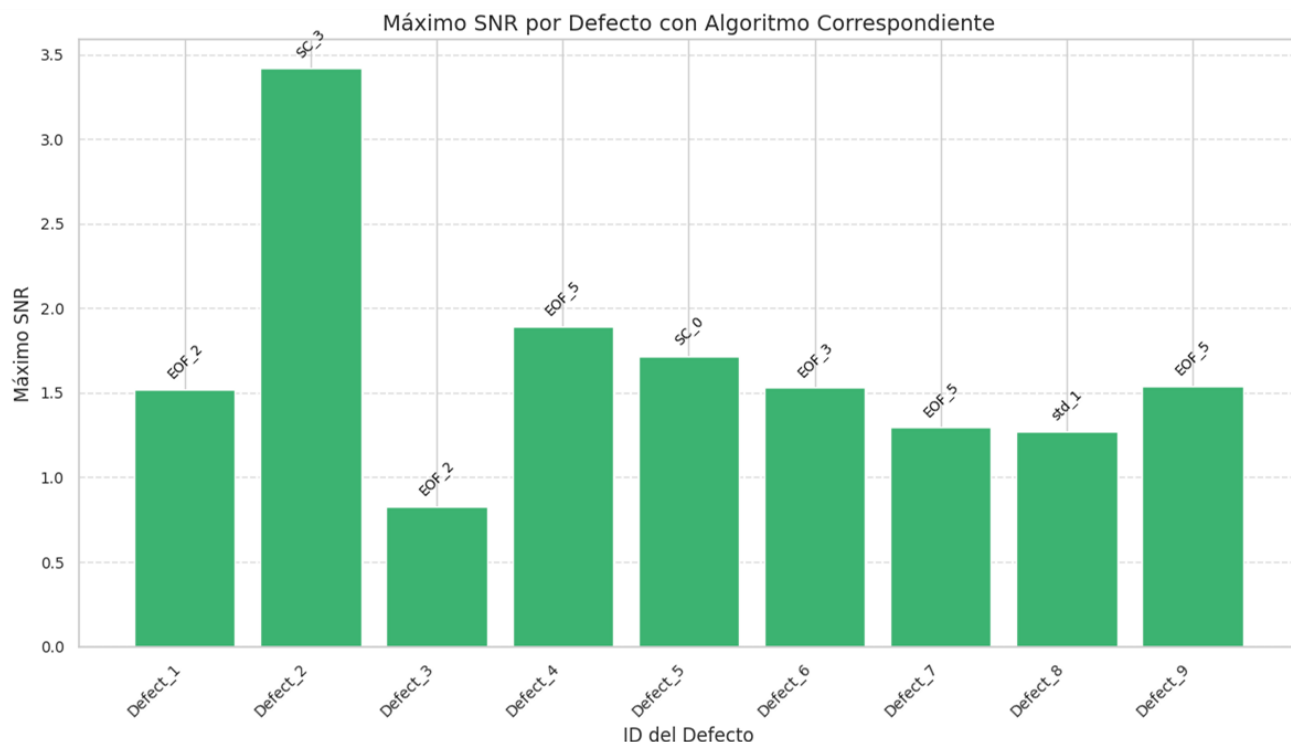


Figura 4-34 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

Entre ellos, el defecto 2 presenta el valor más alto de SNR, aproximadamente 3,4, detectado mediante el algoritmo SC_3. Por otro lado, el defecto 4 presenta un valor cercano a 1,8, detectado mediante EOF_5, mientras que el defecto 5 alcanza un valor aproximado de 1,7, detectado mediante el algoritmo SC_0.

Posteriormente, la Figura 4-35 muestra el análisis considerando únicamente los resultados con SNR > 1,5. En este caso se detectan los defectos 1, 2, 4, 5, 6 y 9, lo que indica que estas regiones presentan un contraste térmico suficiente para su identificación. Asimismo, se observa que los algoritmos que detectan estos defectos con mayor frecuencia son EOF_5 y EOF_2. No obstante, el algoritmo SC_3, aunque detecta un menor número de defectos, destaca por alcanzar el valor más elevado de SNR en el defecto 2.

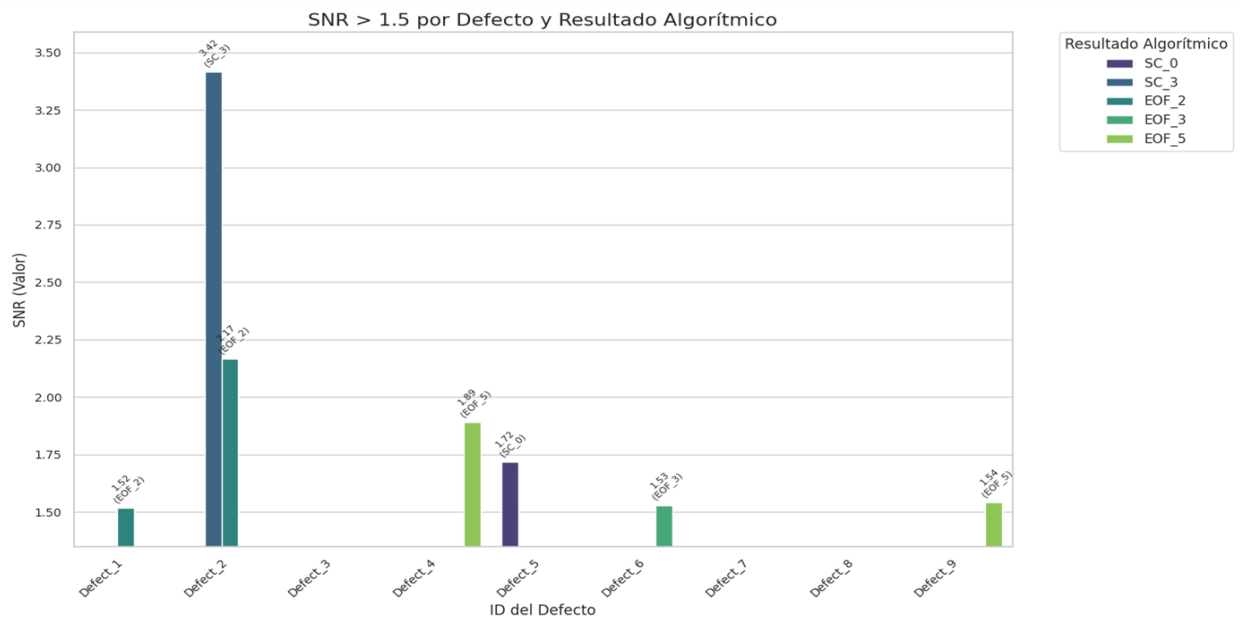


Figura 4-35 El análisis de los defectos que alcanzan valores SNR > 1,5 (Elaboración propia)

En la Figura 4-36 se presenta un análisis más restrictivo considerando $SNR > 2,0$. Bajo este criterio únicamente se detecta el defecto 2, lo que indica que es el defecto con mayor contraste térmico en esta configuración experimental. En este caso, los algoritmos que alcanzan valores en este rango son SC_3 y EOF_2.

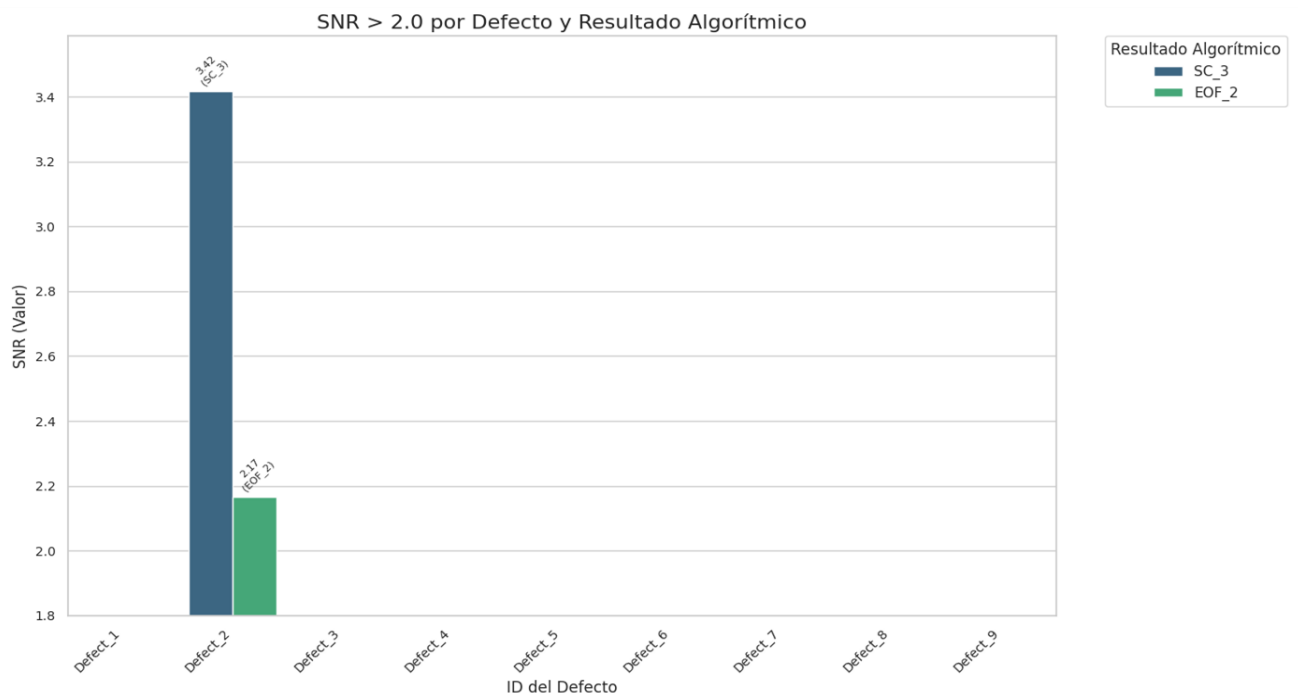


Figura 4-36 El análisis de los defectos que alcanzan valores SNR > 2 (Elaboración propia)

Para complementar el análisis cuantitativo, la Figura 4-37 muestra una comparación visual entre los resultados que mejores SNR presentaron, es decir, mediante los algoritmos SC_3, EOF_2, EOF_3 y EOF_5. En estas imágenes se observa que los defectos pueden identificarse visualmente, aunque con menor contraste en comparación con configuraciones sin tapa. SC_3 para identificar el defecto 2, EOF_2 para el defecto 9,

EOF_3 para el defecto 2 y EOF_5 para los defectos 4 y 9. El SNR del defecto 1 y 6 se vincula a ruido.

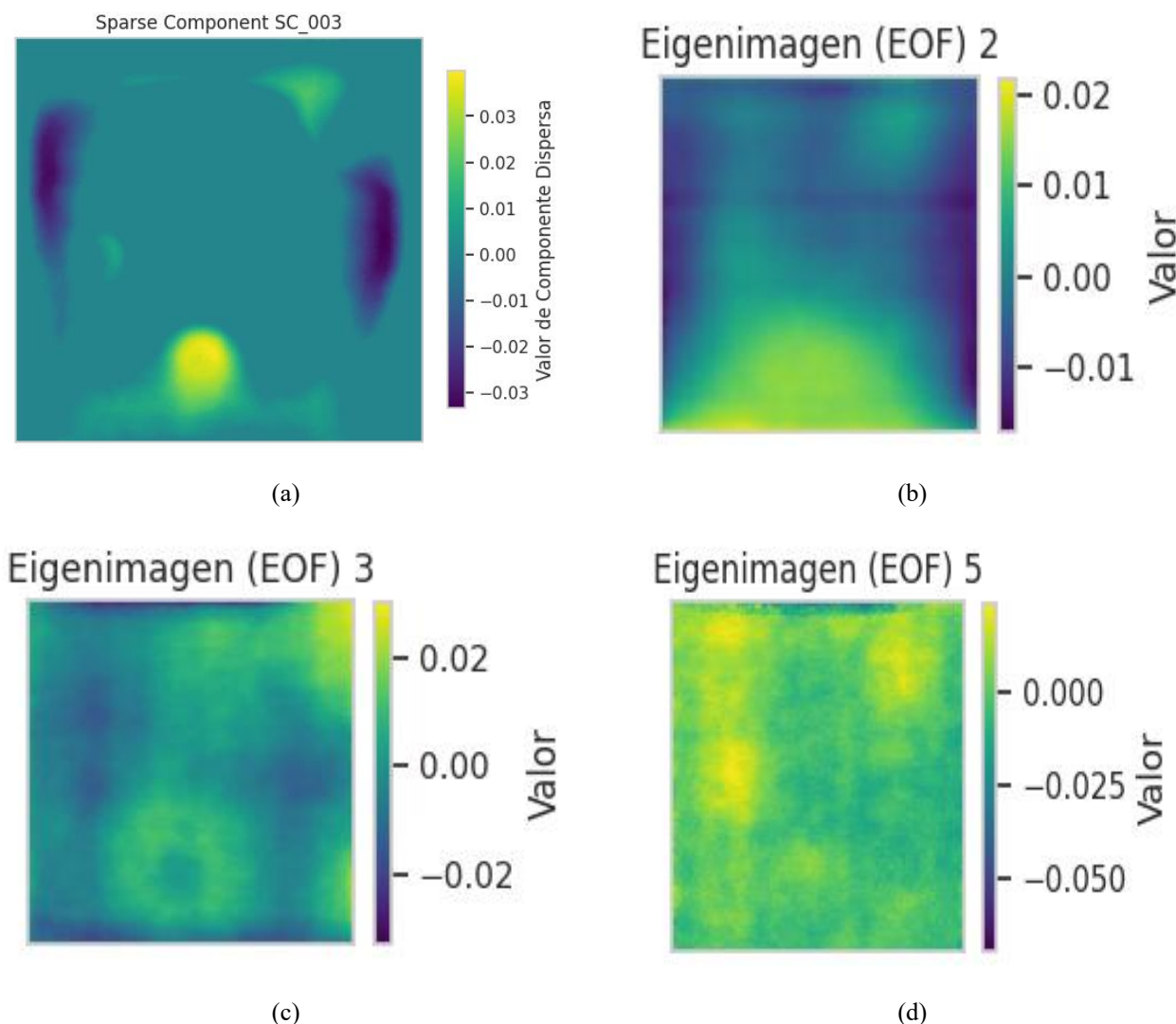


Figura 4-37 Comparación visual entre los resultados obtenidos a) SC_3, b) EOF_2, c) EOF_3 y d) EOF_5 (Elaboración propia)

En conjunto, estos resultados indican que la presencia de la tapa reduce significativamente los valores de SNR, afectando a la capacidad de detección global. No obstante, el algoritmo SC_3, basado en SPCT, destaca por su capacidad para detectar defectos con mayor intensidad, mientras que los métodos basados en EOF proporcionan una detección más consistente en varios defectos.

4.3.3 Caso de estudio 7: PLA con lámpara infrarroja sin tapa

En primer lugar, la Figura 4-38 muestra la comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico. A partir de este gráfico se observa que el valor más alto de SNR general corresponde al algoritmo SC_3, obtenido mediante el método SPCT, con un valor aproximado de 4,6.

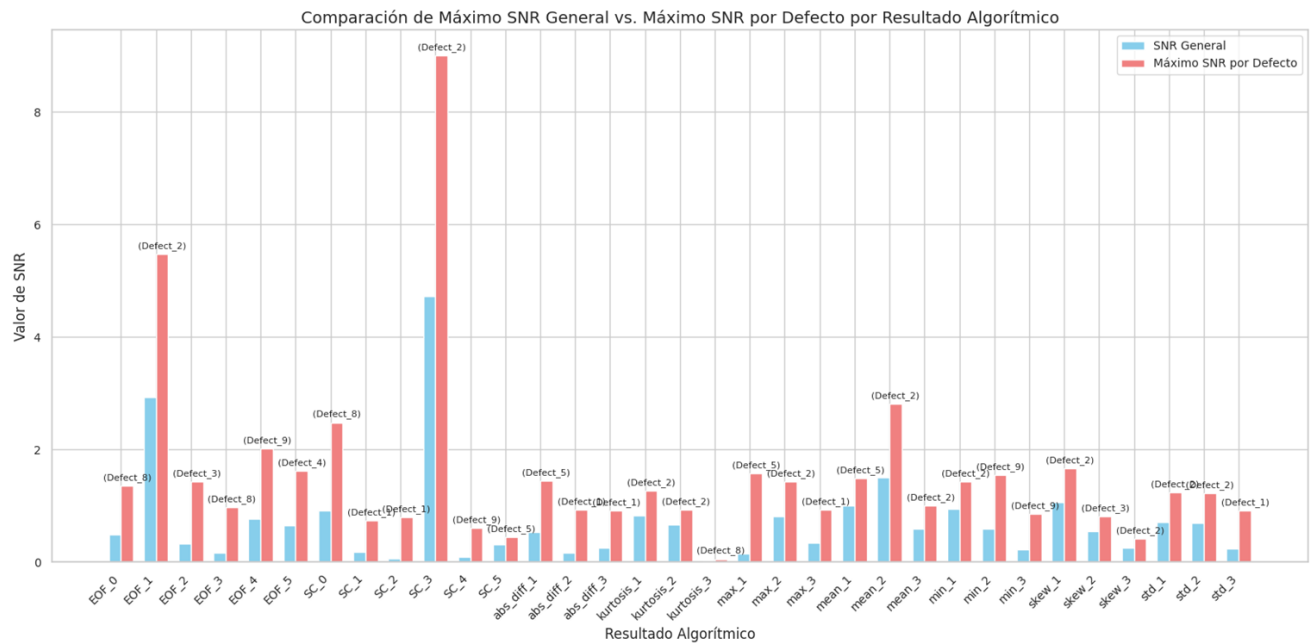


Figura 4-38 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)

Además, se ha evaluado la frecuencia de detección de defectos en los distintos resultados algorítmicos. En este caso, los defectos detectados con mayor frecuencia corresponden a los defectos 2 y 1, lo que indica que estas zonas presentan una respuesta térmica más clara bajo las condiciones experimentales consideradas.

La Figura 4-39 presenta el gráfico de Máximo SNR por defecto con el algoritmo correspondiente. En este análisis se identifican los defectos que presentan valores de SNR superiores a 1, considerados como indicativos de una detección fiable. Bajo este criterio se detectan los defectos 1, 2, 3, 4, 5, 6, 8 y 9.

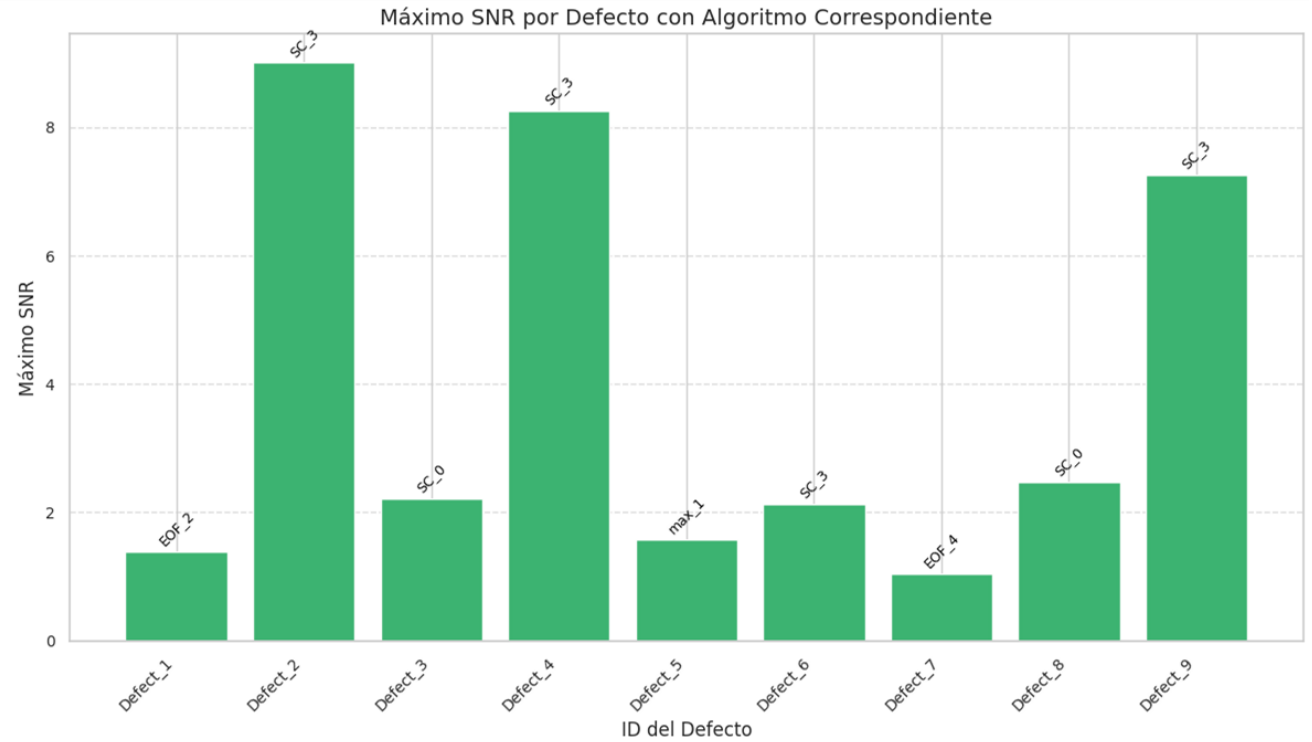


Figura 4-39 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

Entre ellos, el defecto 2 presenta el valor más alto de SNR, aproximadamente 8,6, seguido por el defecto 4 con un valor cercano a 8,1, y el defecto 9 con un valor aproximado de 7,3. Todos ellos son detectados mediante el algoritmo SC_3, lo que confirma su elevado rendimiento en este experimento. Por otra parte, el defecto 8 presenta un valor de SNR cercano a 2,3, siendo detectado mediante el algoritmo SC_0.

Posteriormente, la Figura 4-40 muestra el análisis considerando únicamente los resultados con SNR > 1,5. En este caso se detectan los defectos 2, 3, 4, 5, 6, 8 y 9, lo que indica un buen nivel de contraste térmico en la mayoría de las regiones analizadas. Asimismo, se observa que los algoritmos que detectan estos defectos con mayor frecuencia son SC_3, SC_0 y EOF_1.

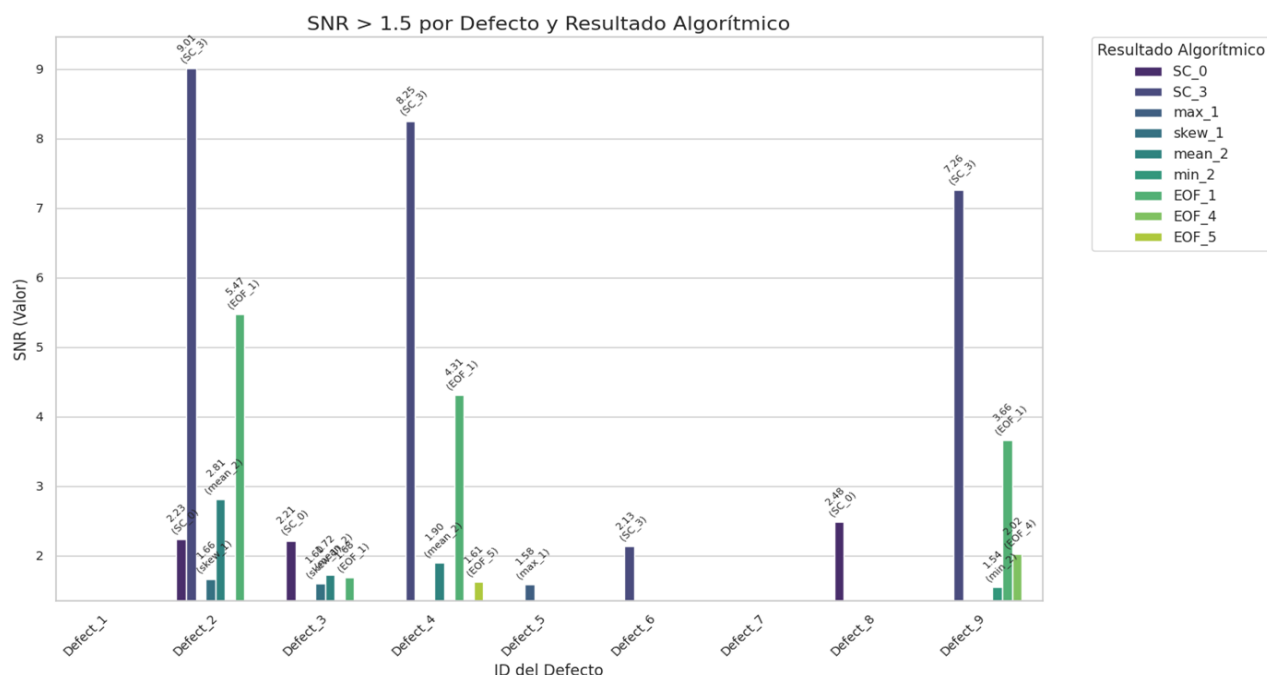


Figura 4-40 El análisis de los defectos que alcanzan valores SNR > 1,5 (Elaboración propia)

En la Figura 4-41 se presenta un análisis más restrictivo considerando SNR > 2,0. Bajo este criterio se detectan los defectos 2, 3, 4, 6, 8 y 9, lo que confirma que estas regiones presentan el mayor contraste térmico en esta configuración experimental. En este caso, el algoritmo que destaca por detectar un mayor número de defectos es SC_3.

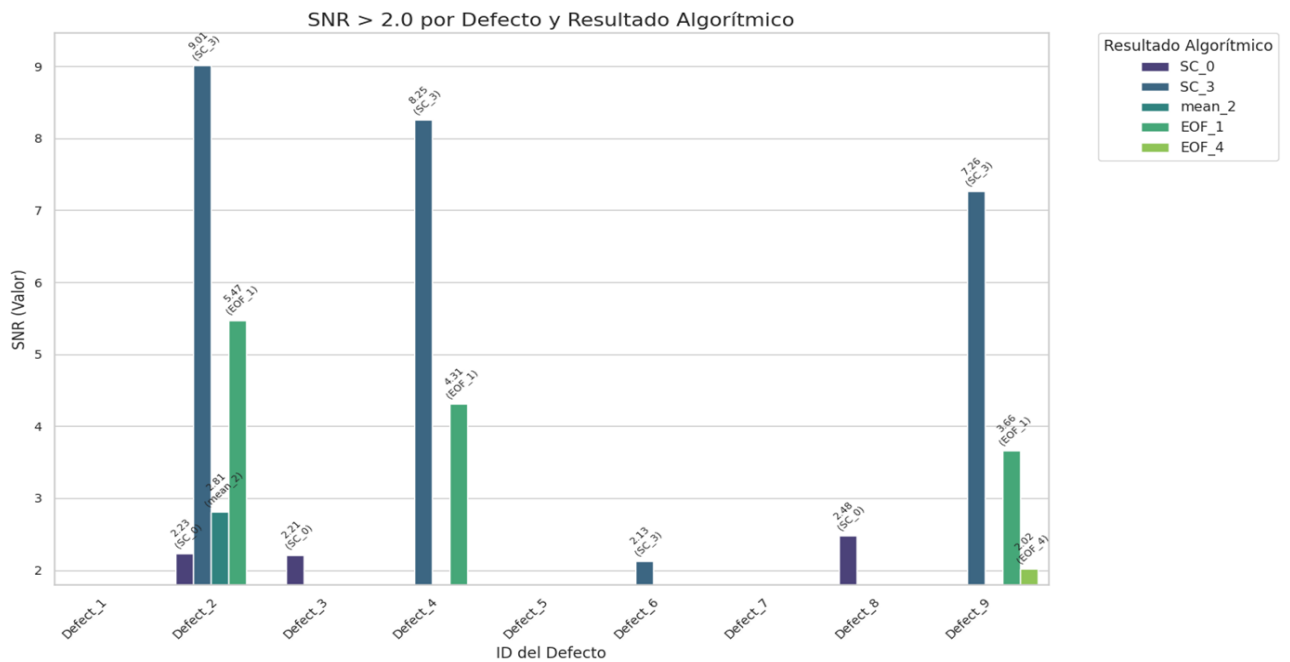


Figura 4-41 El análisis de los defectos que alcanzan valores SNR > 2 (Elaboración propia)

Para complementar el análisis cuantitativo, la Figura 4-42 muestra una comparación visual entre los resultados obtenidos con mejor rendimiento SNR, es decir, mediante los algoritmos SC_0, SC_3 y EOF_1. En estas imágenes se observa que los defectos pueden identificarse claramente, presentando un contraste térmico elevado, especialmente en los resultados asociados al algoritmo SC_3. En SC_3 los defectos grandes y de poca profundidad (2, 4 y 9) son perfectamente detectados, mientras que las medianas tanto en tamaño como en profundidad (3, 5, y 7) son identificadas tanto con SC_0 y EOF_1. Los SNR relacionados con los defectos pequeños y grandes profundidades (1, 6 y 8) de nuevo, como está ocurriendo con todos los casos de estudio, son más bien asociados a ruido, siendo demasiado pequeños y muy profundos para alcanzar sus huellas térmicas a la superficie frontal con los ensayos realizados.

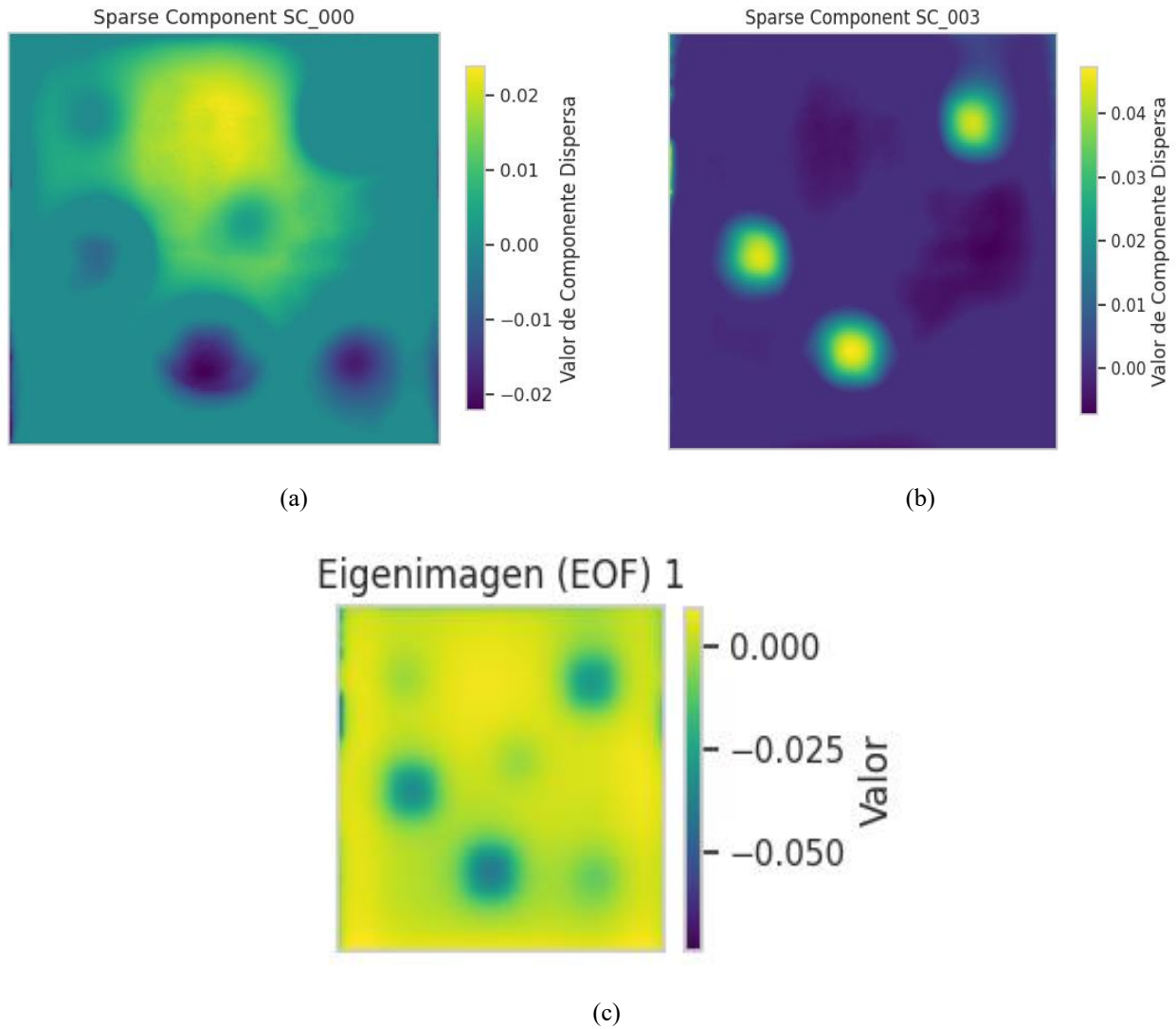


Figura 4-42 Comparación visual entre los resultados obtenidos a) SC_0, b) SC_3 y c) EOF_1 (Elaboración propia)

Adicionalmente, la Figura 4-43 presenta el análisis de varianza explicada por cada componente disperso (SPCT). En este caso se observa que las componentes SC_0 y SC_3 presentan valores elevados de varianza explicada, lo que indica que contienen una parte significativa de la información relevante del proceso térmico.

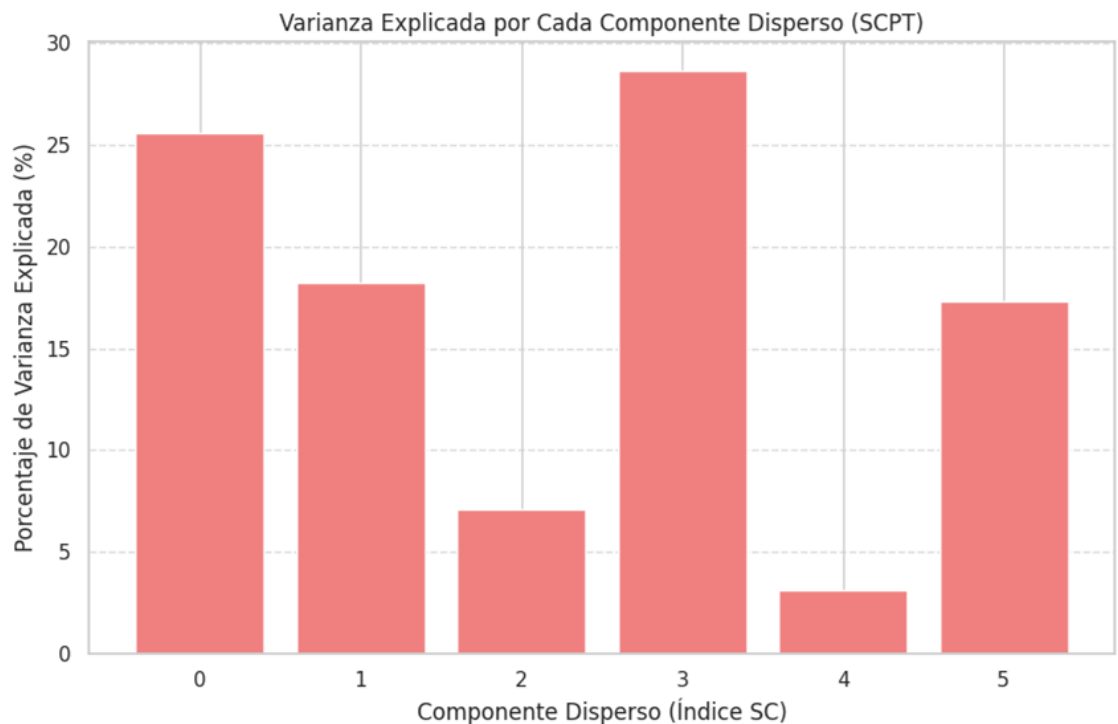


Figura 4-43 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)

Finalmente, la Figura 4-44 muestra la evolución temporal de los coeficientes dispersos (SCPT). En este gráfico se puede observar que las curvas correspondientes a SC_0 y SC_3, en este caso, no presentan una tendencia similar a lo largo del tiempo, pero se aprecia que ninguna de las componentes hay una tendencia común a pares, lo que sugiere las huellas térmicas están repartidas entre los diferentes componentes.

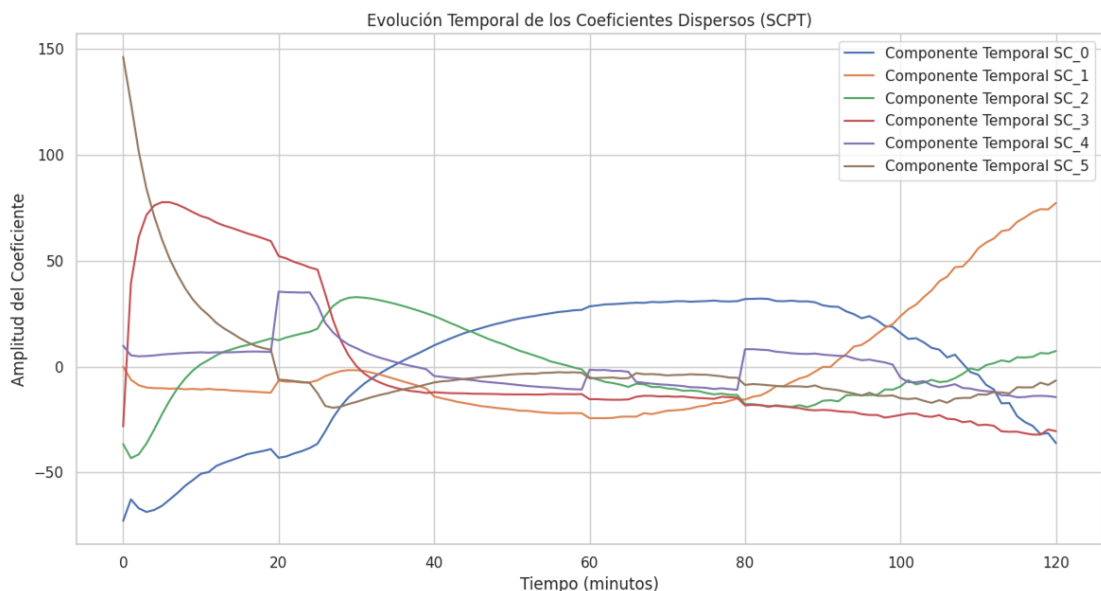


Figura 4-44 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)

En conjunto, estos resultados indican que el algoritmo SC_3, basado en el método SPCT, ofrece el mejor rendimiento para la detección de defectos en el caso de calentamiento mediante lámpara infrarroja sin tapa, proporcionando valores de SNR significativamente superiores y una detección más robusta en

comparación con el resto de algoritmos, aunque de nuevo, como está ocurriendo en todos los casos de estudio, resultados de otros algoritmos, PCT o TSR, en este caso PCT, ayuda a complementar en la detección de los defectos medianos tanto en tamaño como en profundidad.

4.3.4 Caso de estudio 8: PLA con lámpara infrarroja con tapa

En primer lugar, la Figura 4-45 muestra la comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico. A partir de este gráfico se observa que el valor más alto de SNR general corresponde al algoritmo SC_1, obtenido mediante el método SPCT, con un valor aproximado de 1,8.

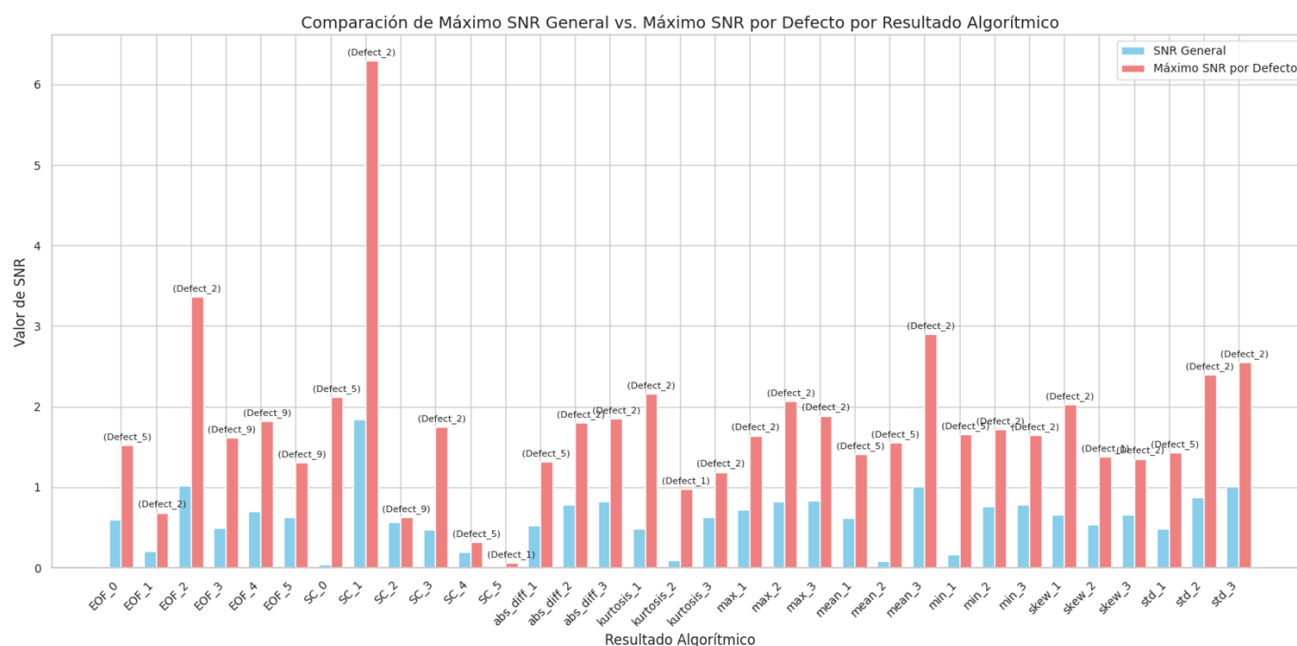


Figura 4-45 La comparación entre el máximo SNR general y el máximo SNR por defecto para cada resultado algorítmico (Elaboración propia)

Además, se ha evaluado la frecuencia de detección de defectos en los distintos resultados algorítmicos. En este caso, los defectos detectados con mayor frecuencia corresponden a los defectos 2 y 5, lo que indica que estas zonas presentan una respuesta térmica más clara bajo las condiciones experimentales consideradas.

La Figura 4-46 presenta el gráfico de Máximo SNR por defecto con el algoritmo correspondiente. En este análisis se identifican los defectos que presentan valores de SNR superiores a 1, considerados como indicativos de una detección fiable. Bajo este criterio se detectan los defectos 1, 2, 3, 4, 5, 6, 8 y 9.

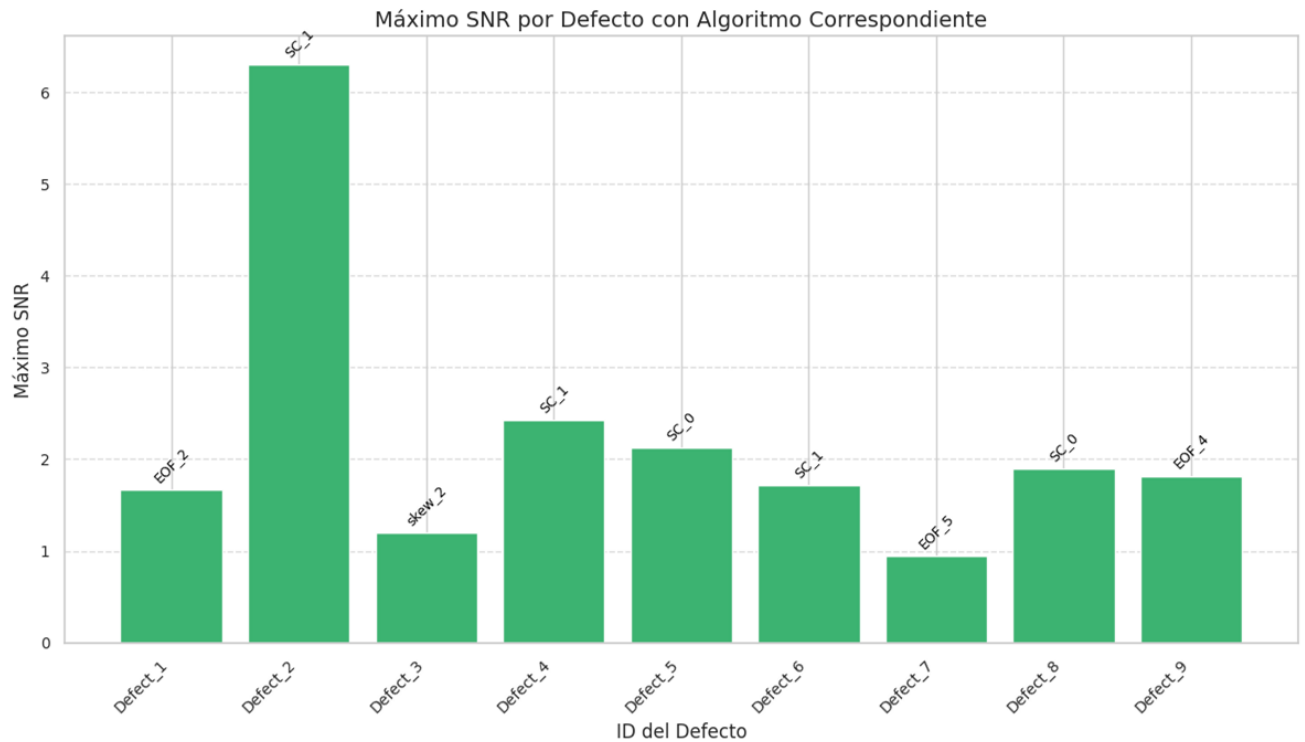


Figura 4-46 El gráfico de Máximo SNR por defecto con el algoritmo correspondiente (Elaboración propia)

Entre ellos, el defecto 2 presenta el valor más alto de SNR, aproximadamente 6,3, seguido por el defecto 4 con un valor cercano a 2,3, ambos detectados mediante el algoritmo SC_1, lo que confirma su buen rendimiento en este experimento. Por otra parte, el defecto 5 presenta un valor de SNR cercano a 2.1, siendo detectado mediante el algoritmo SC_0.

Posteriormente, la Figura 4-47 muestra el análisis considerando únicamente los resultados con SNR > 1,5. En este caso se detectan los defectos 1, 2, 4, 5, 6, 8 y 9, lo que indica un nivel adecuado de contraste térmico en la mayoría de las regiones analizadas. Asimismo, se observa que los algoritmos que detectan estos defectos con mayor frecuencia son SC_0, SC_1 y EOF_2.

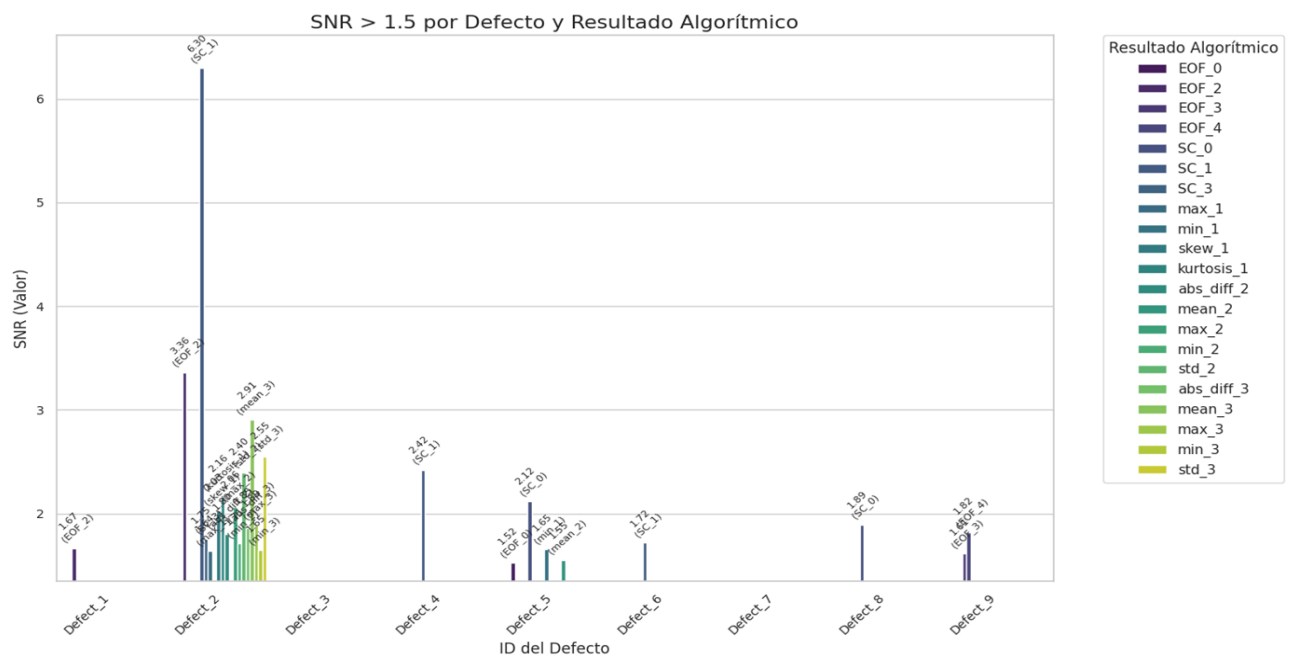


Figura 4-47 El análisis de los defectos que alcanzan valores SNR > 1,5 (Elaboración propia)

En la Figura 4-48 se presenta un análisis más restrictivo considerando $SNR > 2,0$. Bajo este criterio se detectan los defectos 2, 4 y 5, lo que indica que estos presentan el mayor contraste térmico en esta configuración experimental. En este caso, el algoritmo que destaca por detectar un mayor número de defectos es SC_1.

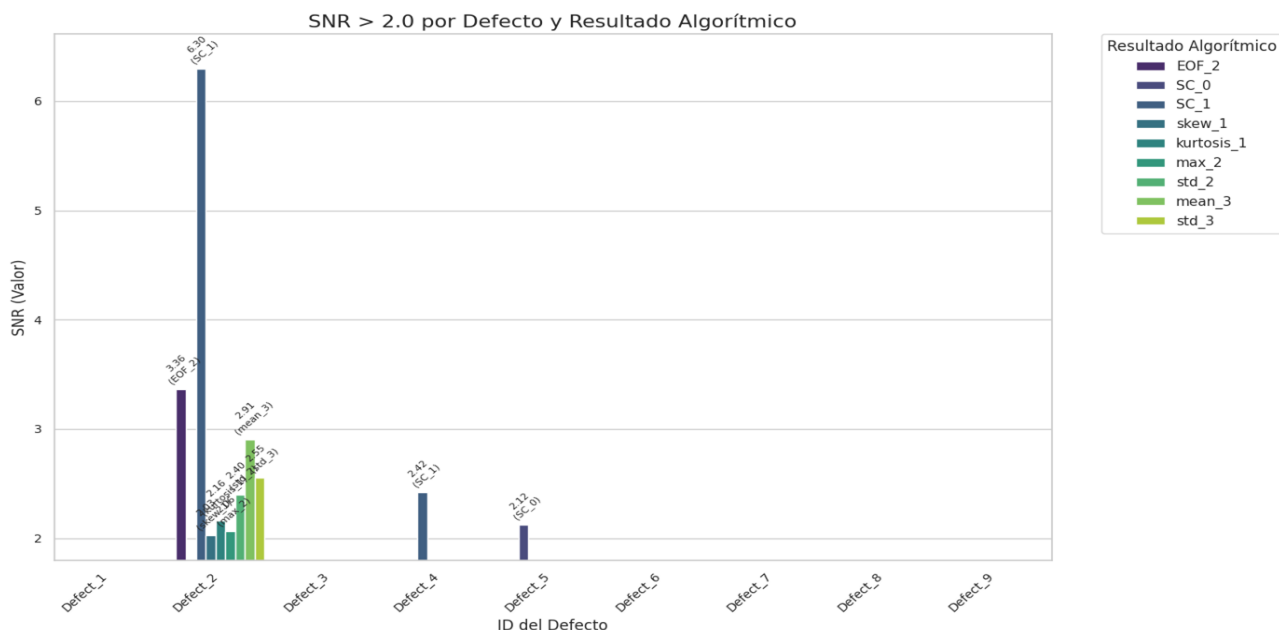


Figura 4-48 El análisis de los defectos que alcanzan valores $SNR > 2$ (Elaboración propia)

Para complementar el análisis cuantitativo, la Figura 4-49 muestra una comparación visual entre los resultados que mejores datos SNR han obtenidos, es decir, mediante los algoritmos SC_0, SC_1 y EOF_2. En estas imágenes se observa que los defectos pueden identificarse de forma clara, aunque con menor intensidad en comparación con la configuración sin tapa. En SC_0, el defecto 5 detectado (tamaño y profundidad media, situado en la ubicación central de la probeta) es más bien vinculado a ruido, de ahí la importancia de comparar los datos cuantitativos del SNR con los datos cualitativos mediante la inspección visual, y el SC_1 y EOF_2 identifican de forma aceptable los defectos grandes y de poca profundidad (2, 4 y 9).

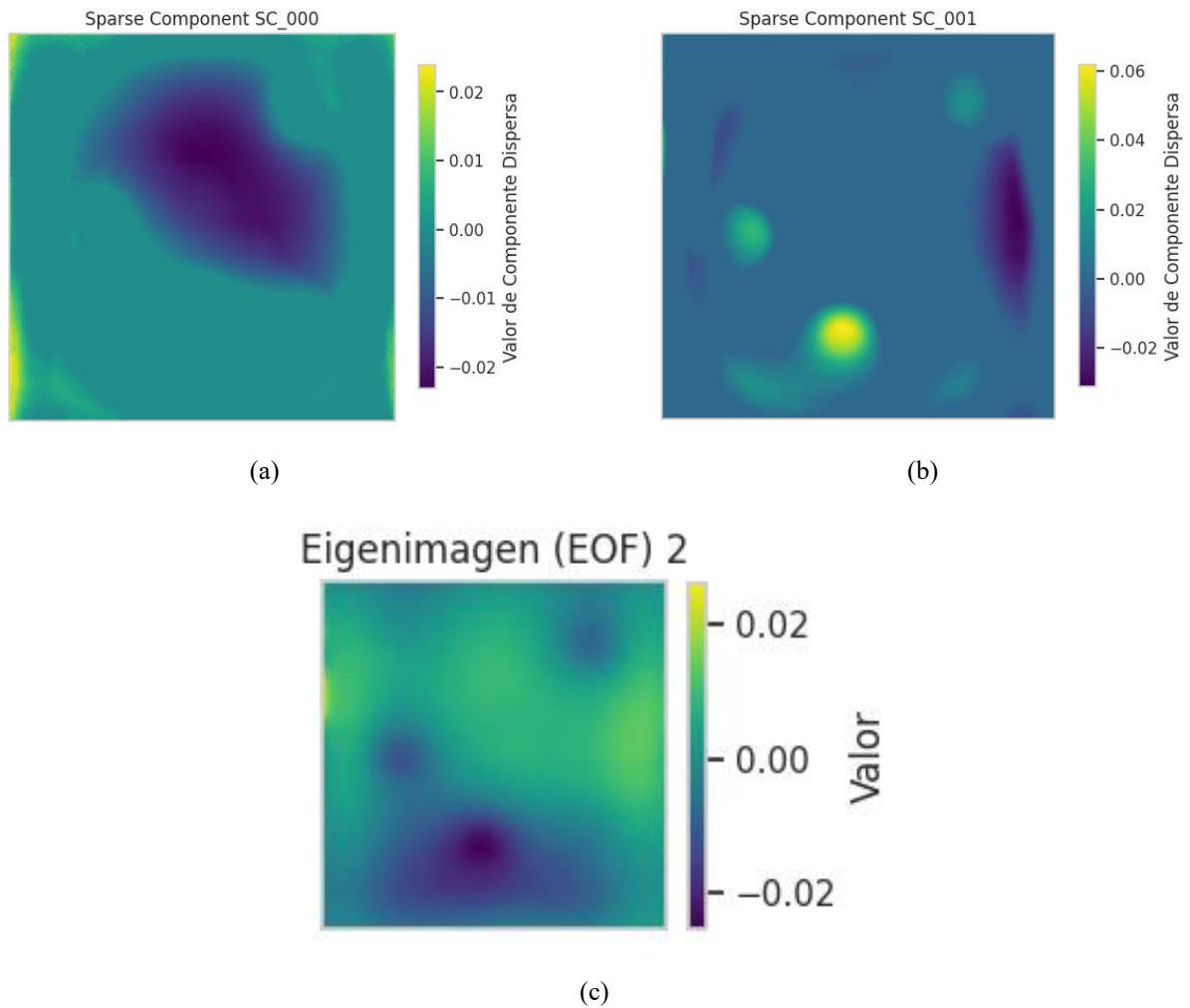


Figura 4-49 Comparación visual entre los resultados obtenidos a) SC_0, b) SC_1 y c) EOF_2 (Elaboración propia)

Adicionalmente, la Figura 4-50 presenta el análisis de varianza explicada por cada componente disperso (SPCT). En este caso se observa que las componentes SC_0 y SC_1 presentan valores relativamente bajos de varianza explicada, pero no insignificantes al superar con creces el 1%, lo que indica que contienen una parte pequeña pero igualmente de importante de la información relevante del proceso térmico.

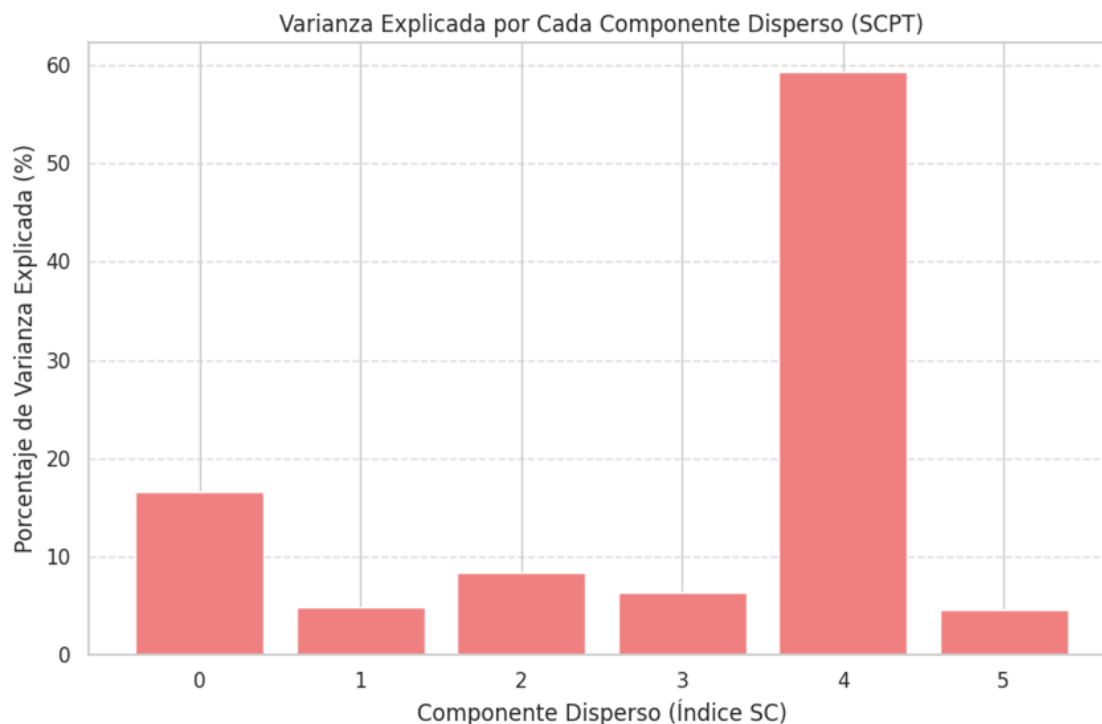


Figura 4-50 La varianza explicada por cada componente disperso obtenida mediante el algoritmo SPCT (Elaboración propia)

Finalmente, la Figura 4-51 muestra la evolución temporal de los coeficientes dispersos (SCPT). En este gráfico se puede observar que las curvas correspondientes a SC_0 y SC_1 presentan una cierta tendencia, sobre todo a partir del minuto 20, a lo largo del tiempo, lo que sugiere que ambas componentes están relacionadas con la dinámica térmica generada por los defectos durante el proceso del ensayo.

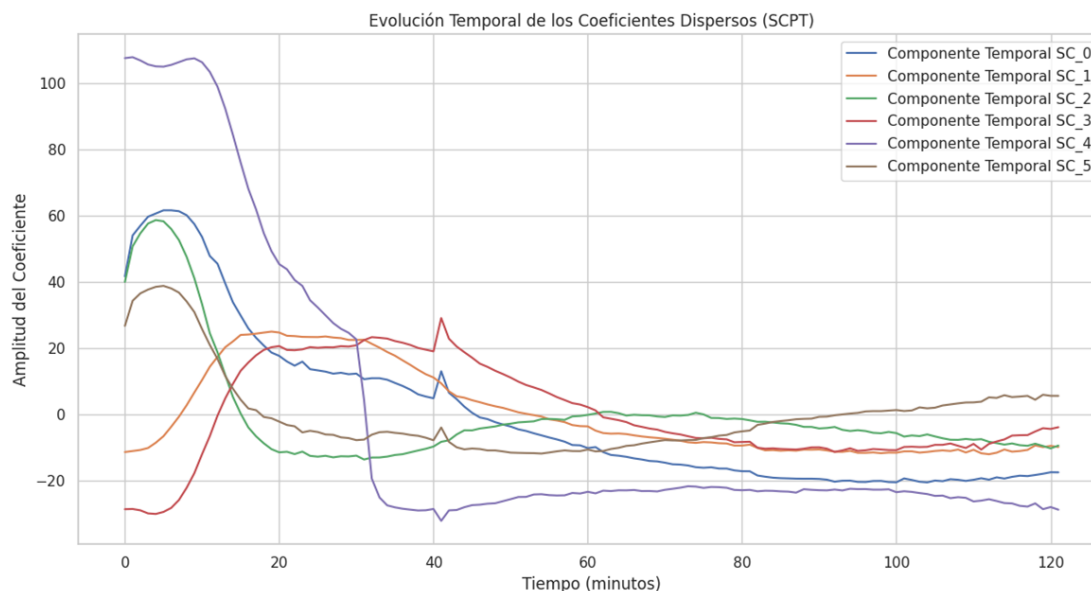


Figura 4-51 La evolución temporal de los coeficientes dispersos (SCPT) (Elaboración propia)

En conjunto, estos resultados indican que el algoritmo SC_1, basado en el método SPCT, ofrece el mejor rendimiento para la detección de defectos en el caso de calentamiento mediante lámpara infrarroja con tapa, aunque con una reducción general del nivel de SNR debido a la presencia de la tapa.

4.3.5 Comparación de los resultados en el PLA

En este apartado se realiza una comparación global de los resultados obtenidos para la muestra de PLA bajo las cuatro configuraciones experimentales analizadas: lámpara halógena sin tapa, lámpara halógena con tapa, lámpara infrarroja sin tapa y lámpara infrarroja con tapa. El objetivo es evaluar la influencia de las condiciones de calentamiento, y la inserción de la tapa en la base, en la detectabilidad de defectos mediante la métrica SNR y el rendimiento visual de los distintos algoritmos.

En primer lugar, se observa que la configuración sin tapa proporciona, en general, valores de SNR más elevados en comparación con las configuraciones con tapa. Esto se debe a una menor atenuación térmica y a una mejor transmisión del calor hacia la superficie de la muestra, lo que favorece el contraste térmico entre los defectos y el material base.

En particular, la configuración de lámpara infrarroja sin tapa presenta los valores más altos de SNR, alcanzando máximos superiores a 8 en varios defectos. Esto indica que esta configuración es la más favorable para la detección de defectos en PLA, permitiendo identificar un mayor número de discontinuidades con alta fiabilidad, dado que también la lámpara infrarroja presenta mayor potencia con respecto a la lámpara halógena.

Por otro lado, las configuraciones con tapa, tanto para lámpara halógena como infrarroja, muestran una reducción significativa en los valores de SNR. Esta disminución afecta directamente a la capacidad de detección, especialmente en defectos de menor tamaño, que requieren un mayor contraste térmico para ser identificados.

En cuanto al tipo de fuente de calor, se observa que la lámpara infrarroja tiende a proporcionar un mejor rendimiento que la halógena, especialmente en condiciones sin tapa. Esto puede atribuirse a lo comentado previamente, a la mayor potencia de la lámpara infrarroja a la hora de transmitir calor.

Desde el punto de vista de los algoritmos, los métodos basados en SPCT, en particular las componentes SC_1 y SC_3, muestran el mejor rendimiento en la mayoría de los casos, alcanzando los valores más altos de SNR y detectando un mayor número de defectos. Por otro lado, los métodos basados en EOF presentan un comportamiento más estable, aunque con valores de SNR generalmente inferiores, pero que ayudan a detectar los defectos de tamaño y profundidad media, incluso mejor que el SPCT.

Asimismo, el análisis con umbrales de SNR ($SNR > 1,5$ y $SNR > 2,0$) confirma estas tendencias, mostrando que las configuraciones sin tapa, con lámpara infrarroja y los algoritmos SPCT permiten mantener un mayor número de defectos detectables incluso bajo criterios más restrictivos.

En términos de interpretación visual, las imágenes procesadas mediante los algoritmos más relevantes muestran que los defectos son más claramente identificables en las configuraciones sin tapa, mientras que en presencia de tapa el contraste disminuye, dificultando su localización.

En conjunto, los resultados indican que la combinación más eficaz para la detección de defectos en PLA corresponde a la utilización de lámpara infrarroja sin tapa junto con algoritmos basados en SPCT, especialmente la componente SC_3. Esta configuración ofrece el mejor compromiso entre sensibilidad, contraste térmico y capacidad de detección. La Tabla 4-2 sintetiza los resultados correspondientes a las cuatro configuraciones estudiadas: lámpara halógena sin tapa, lámpara halógena con tapa, lámpara infrarroja sin tapa y lámpara infrarroja con tapa.

Configuración experimental	Algoritmo más efectivo	Defectos detectados	SNR máximo aproximado
Halógena sin tapa	SC_1	2, 4, 9	~8,6
Halógena con tapa	SC_3 / EOF_5	2, 4, 5, 6	~3,4
Infrarroja sin tapa	SC_3	2, 4, 6, 8, 9	~8,6
Infrarroja con tapa	SC_1	2, 4, 5	~6,3

Tabla 4-2 Resumen comparativo de resultados para PLA en diferentes configuraciones experimentales
(Elaboración propia)

4.4 Comparación resultados entre PETG-CF y PLA

En este apartado se realiza una comparación global entre los resultados obtenidos para las muestras de PETG-CF y PLA, con el objetivo de evaluar la influencia del material en la detectabilidad de defectos mediante termografía activa y el rendimiento de los algoritmos de procesamiento definidos.

En primer lugar, se observa que el material PETG-CF presenta, en general, una respuesta térmica menos definida en comparación con el PLA. Esto se traduce en valores de SNR menos elevados y estables, lo que empeora la detección de defectos en distintas configuraciones experimentales. A pesar de la presencia de fibras de carbono, que modifican las propiedades térmicas del material, aumentando la conductividad térmica y mejorando la difusión del calor, entre el **PETG.CF** y el **PLA utilizado en este trabajo**, para la identificación de **defectos**, sus huellas térmicas son **mejor detectadas en PLA frente al PETG-CF** con los algoritmos aquí procesados. Esto se ve con el SNR máximo entre la Tabla 4-1 y 4-2, donde es ligeramente mayor en PLA frente al PETG-CF. Además, se tiene las siguientes situaciones comparando los mejores resultados en los casos de estudio análogos.

- **Caso de estudio 1 vs Caso de estudio 5:** El SC-1 en PETG-CF (Figura 4-6) es **menos nítido** con respecto a SC_0 y SC_5 (Figura 4-30) en PLA referido a los **defectos grandes y poca profundidad**, y sobre todo en los **defectos medianos con profundidad intermedia**, el SC_4 en PETG-CF es **bastante menos nítido** con respecto a EOF_1 en PLA.

- **Caso de estudio 2 vs Caso de estudio 6:** El defecto 2 (defecto grande con poca profundidad) se aprecia **menos nítido** en SC_3 (Figura 4-13) en PETG-CF con respecto a SC_3 en PLA (Figura 4-37), aunque los **defectos 4 y 9 (defectos grandes con poca profundidad)** se visualiza ligeramente **mejor** en SC_3 en PETG-CF con respecto a EOF_5 en PLA.

- **Caso de estudio 3 vs Caso de estudio 7:** El SC_1 (detección **defectos grandes y poco profundos**) y skew_2 (detección **defectos medianos con profundidad intermedia**) en PETG-CF (Figura 4-18) es ligeramente **menos nítido** con respecto a SC_3 (detección **defectos grandes y poco profundos**) y EOF_1 (detección **defectos medianos con profundidad intermedia**) en PLA (Figura 4-42), sobre todo el skew_2 frente a EOF_1 (a los **defectos medianos con profundidad intermedia**).

- **Caso de estudio 4 vs Caso de estudio 8:** El SC_1 en PETG-CF (Figura 4-23) es ligeramente **menos nítido** con respecto al SC_1 en PLA (Figura 4-49) en cuanto a la detección de los **defectos grandes con poca profundidad**.

En la Tabla 4-3 se muestra un resumen de lo comentado.

Comparativa	Tipo de Defecto	PETG-CF	PLA	Observaciones de la Comparativa
Caso 1 vs Caso 5	Grandes y poca profundidad	SC_1 (Figura 4-6)	SC_0 y SC_5 (Figura 4-30)	PETG-CF es ligeramente MENOS nítido que PLA
	Medianos con profundidad intermedia	SC_4 (Figura 4-6)	EOF_1 (Figura 4-30)	PETG-CF es bastante MENOS nítido que PLA
Caso 2 vs Caso 6	Defecto 2 (Grande, poca prof.)	SC_3 (Figura 4-13)	SC_3 (Figura 4-37)	PETG-CF se ligeramente MENOS nítido que PLA
	Defectos 4 y 9 (Grandes, poca profundidad)	SC_3 (Figura 4-13)	EOF_5 (Figura 4-37)	PETG-CF es ligeramente MÁS nítido que PLA
Caso 3 vs Caso 7	Grandes y poca profundidad	SC_1 (Figura 4-18)	SC_3 (Figura 4-42)	PETG-CF es ligeramente menos nítido que PLA
	Medianos con profundidad intermedia	skew_2 (Figura 4-18)	EOF_1 (Figura 4-42)	PETG-CF es bastante MENOS nítido que PLA
Caso 4 vs Caso 8	Grandes y poca profundidad	SC_1 (Figura 4-23)	SC_1 (Figura 4-49)	PETG-CF es ligeramente MENOS nítido que PLA

Tabla 4-3 Resumen la comparación resultados entre PETG-CF y PLA (Elaboración propia)

En cuanto a las configuraciones de calentamiento, ambos materiales muestran un comportamiento similar en términos cualitativos: las condiciones sin tapa proporcionan mejores resultados que las configuraciones con tapa, y la lámpara infrarroja ofrece un mejor rendimiento que la halógena. Sin tapa se detecta tanto los defectos grandes con poca profundidad como también los defectos de tamaño y profundidad intermedia. Con tapa, solo los defectos grandes con poca profundidad son identificados, en donde en ningún ensayo se puede afirmar que los defectos pequeños con grande profundidad son detectables, vinculados sus SNR en algunas ocasiones algo elevados a simplemente ruido.

Desde el punto de vista de los algoritmos, los métodos basados en SPCT destacan claramente en ambos materiales. El componente SC_1 muestra un rendimiento óptimo, con valores elevados de SNR y una detección consistente de defectos, sobre todo en los defectos grandes con poca profundidad, seguido del SC_3 y, en la cola y al mismo nivel, SC_0, SC_4 y SC_5. Para los defectos medianos con profundidad intermedia, tanto el PCT como el TSR, arrojan mejores resultados, liderando el EOF_1 seguido del skew_2. Esto indica que los tres algoritmos de procesamiento aplicados son complementarios entre ellos, no siendo ninguno descartable en ninguna de las situaciones analizadas. En el PLA, aunque también se observa un buen comportamiento de las componentes SPCT (especialmente SC_3 y SC_1), su rendimiento depende en mayor medida de la configuración experimental.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusión

La relevancia de este trabajo se enmarca en el creciente uso de polímeros y materiales compuestos en sectores industriales estratégicos, como el naval y el de Defensa, donde la optimización del peso, la resistencia a la corrosión y la fiabilidad estructural son factores críticos. En este contexto, la fabricación aditiva mediante Modelado por Deposición Fundida (*Fused Deposition Modeling*, FDM) se ha consolidado como una tecnología clave para la producción de componentes complejos y personalizados. No obstante, el propio proceso de deposición capa a capa introduce heterogeneidades microestructurales que pueden dar lugar a defectos internos, tales como porosidad, vacíos o delaminaciones, comprometiendo el comportamiento mecánico de las piezas, además del inevitable deterioro del material a lo largo de su vida útil.

En este sentido, el presente Trabajo Fin de Grado surge como respuesta a la necesidad de implementar técnicas de Ensayo No Destructivo (END) eficaces, sin contacto y aplicables en entornos reales, que permitan evaluar la integridad estructural de componentes fabricados mediante FDM, superando las limitaciones de los métodos convencionales.

El desarrollo del trabajo se ha estructurado en varias fases. En primer lugar, se diseñaron probetas mediante modelado paramétrico en Autodesk Inventor, incorporando defectos internos controlados simulando vacíos con el objetivo de evaluar la capacidad de detección. Posteriormente, se fabricaron las muestras empleando una impresora Bambu Lab, seleccionando dos materiales con propiedades termofísicas diferenciadas: PETG-CF y PLA. La fase experimental se llevó a cabo mediante termografía infrarroja activa, utilizando una cámara FLIR E85 junto con distintas fuentes de excitación térmica (una lámpara halógena de 500 W y una lámpara infrarroja de 2000 W). Finalmente, los datos obtenidos fueron procesados en Google Colab mediante algoritmos avanzados, siendo *Principal Component Thermography* (PCT), *Sparse Component Thermography* (SPCT) y *Thermography Reconstruction Signal* (TSR), con el fin de mejorar la relación señal-ruido y facilitar la identificación de defectos.

Los resultados obtenidos permiten concluir que el material usado PLA presenta un comportamiento ligeramente superior frente al PETG-CF en términos de detectabilidad de defectos internos. Además, al tener mayor potencia, la lámpara infrarroja ha arrojado mejores resultados con respecto a la lámpara halógena. Asimismo, si se tiene al descubierto la superficie posterior de las probetas, donde los defectos están en contacto directo con el flujo de calor de la lámpara, los resultados también son más nítidos a diferencia de taparlos y convertirlos en defectos totalmente interiores.

Desde el punto de vista algorítmico, el método SPCT, y en particular la componente SC_1, ha demostrado ser el más eficaz, alcanzando los mayores valores de *Signal to Noise Ratio* (SNR) y

proporcionando una representación clara de las zonas defectuosas. Sobre todo, los defectos clasificados como grandes y poca profundidad fueron bien identificados con SPCT. No obstante, los defectos clasificados como tamaño y profundidad media fueron mejor detectados con PCT y TSR, demostrando que estos algoritmos no son excluyentes ni uno mejor que el otro, sino que son complementarios aportando información nueva cada uno de ellos. Referido a los defectos clasificados como pequeños y gran profundidad, en ningún momento se ha identificado claramente ninguno de ellos, confirmando que efectivamente era el escenario más exigente y proporcionando una representación clara de las zonas defectuosas.

En conjunto, estos resultados confirman el cumplimiento de los objetivos planteados al inicio del trabajo, validando la termografía infrarroja activa como una técnica viable, eficiente y prometedora para la evaluación no destructiva de componentes poliméricos fabricados mediante impresión 3D.

5.2 Líneas futuras

A pesar de los resultados satisfactorios obtenidos, este trabajo pone de manifiesto diversas líneas de mejora y desarrollo futuro que permitirán ampliar su alcance y robustez. En primer lugar, resulta fundamental aumentar el tamaño y la diversidad del conjunto de datos experimental, incorporando un mayor número de muestras, geometrías diferentes tanto de la muestra como de los defectos internos (además de distintas profundidades y rellenos que permita simular otros tipos de defectos además de vacíos) y materiales poliméricos, tales como ABS, PA12 o composites con diferentes porcentajes de refuerzo. Esto permitirá mejorar la generalización de los resultados y analizar con mayor precisión la influencia de las propiedades térmicas en la detectabilidad de defectos.

Una de las líneas más prometedoras consiste en la integración de técnicas de Inteligencia Artificial y *Deep Learning* para la automatización del proceso de detección. En este sentido, los resultados obtenidos mediante PCT, SPCT y TSR podrían emplearse como datos de entrada para el aprendizaje de modelos de segmentación avanzados, como Mask R-CNN. Este enfoque permitirá reducir la dependencia del análisis manual, mejorar la precisión en la identificación de defectos y avanzar hacia sistemas de diagnóstico en tiempo real.

Asimismo, se propone la exploración de enfoques multimodales que combinen la información térmica con otras técnicas de inspección, como georradar o la radiografía digital. Esta integración permitirá obtener una caracterización más completa de los defectos y mejorar la fiabilidad del diagnóstico estructural.

Finalmente, la extensión de esta metodología a aplicaciones reales en sectores industriales, como la inspección de infraestructuras navales, aeronáuticas o civiles, así como en la conservación del patrimonio, permitirá validar su aplicabilidad práctica y consolidar la termografía activa como una herramienta clave dentro del marco de la Industria 4.0.

6 BIBLIOGRAFÍA

- [1] «Polymers Market Size, Share & Growth Trends Report 2035», Research Nester. Accedido: 17 de marzo de 2026. [En línea]. Disponible en: <https://www.researchnester.com/reports/polymers-market/1387>
- [2] I. Garrido, S. Lagüela, R. Otero, y P. Arias, «Thermographic methodologies used in infrastructure inspection: A review—Post-processing procedures», *Applied Energy*, vol. 266, p. 114857, may 2020, doi: 10.1016/j.apenergy.2020.114857.
- [3] A. Alex Alberto González, «Cuantificación de impactos en materiales composites con termografía infrarroja activa».
- [4] W. D. Callister, *Materials science and engineering: an introduction*, 6th ed. New York, NY: John Wiley & Sons, 2003.
- [5] «Termoestable - definición y ejemplos de termoestables». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://losadhesivos.com/termoestable.html>
- [6] I. Garcia Vargas y H. Fernandes, «Spatial and temporal deep learning algorithms for defect segmentation in infrared thermographic imaging of carbon fibre-reinforced polymers», *Nondestructive Testing and Evaluation*, vol. 41, n.º 1, pp. 45-65, ene. 2026, doi: 10.1080/10589759.2025.2457593.
- [7] «Introducción completa al polímero reforzado con fibra de carbono (CFRP)». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://tymagnets.com/es/Introducci%C3%B3n-de-cfrp-de-pol%C3%ADmero-reforzado-con-fibra-de-carbono/>
- [8] «BARRAS DE REFUERZO GFRP». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.panawebssc.com/barras-de-refuerzo-gfrp>
- [9] M. Jiménez, L. Romero, I. A. Domínguez, M. D. M. Espinosa, y M. Domínguez, «Additive Manufacturing Technologies: An Overview about 3D Printing Methods and Future Prospects», *Complexity*, vol. 2019, n.º 1, p. 9656938, ene. 2019, doi: 10.1155/2019/9656938.
- [10] J. C. R. Granados, «Detección y Reconstrucción de Defectos Subsuperficiales en 3-D mediante el Análisis de Datos de Termografía Pulsada».
- [11] S. Guessasma, N. Stephant, S. Durand, y S. Belhabib, «Digital Light Processing Route for 3D Printing of Acrylate-Modified PLA/Lignin Blends: Microstructure and Mechanical Performance», *Polymers*, vol. 16, n.º 10, may 2024, doi: 10.3390/polym16101342.
- [12] «Fig. 1. Photographs of GFRP composite vessels, mats, laminates, and...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Photographs-of-GFRP-composite-vessels-mats-laminates-and-illustration-of-fabrication_fig1_361477127

- [13] A. P. Mouritz, E. Gellert, P. Burchill, y K. Challis, «Review of advanced composite structures for naval ships and submarines», *Composite Structures*, vol. 53, n.º 1, pp. 21-42, jul. 2001, doi: 10.1016/S0263-8223(00)00175-6.
- [14] «X. P. V. Maldague, “Theory and Practice of Infrared Technology for Nondestructive Testing,” John Wiley & Son, New York, 2001. - References - Scientific Research Publishing». Accedido: 17 de marzo de 2026. [En línea]. Disponible en: <https://www.scirp.org/reference/referencespapers?referenceid=244328>
- [15] «TABLE 1 -Summary of differences between steel and composites construction», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Summary-of-differences-between-steel-and-composites-construction_tbl1_265101727
- [16] I. Garrido, S. Lagüela, P. Arias, y J. Balado, «Thermal-based analysis for the automatic detection and characterization of thermal bridges in buildings», *Energy and Buildings*, vol. 158, pp. 1358-1367, ene. 2018, doi: 10.1016/j.enbuild.2017.11.031.
- [17] «Visby-class corvette | Saab», Start. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.saab.com/products/visby-class-corvette>
- [18] R. Usamentiaga, P. Venegas, J. Guerediaga, L. Vega, J. Molleda, y F. Bulnes, «Infrared Thermography for Temperature Measurement and Non-Destructive Testing», *Sensors*, vol. 14, n.º 7, pp. 12305-12348, jul. 2014, doi: 10.3390/s140712305.
- [19] «Saab Lightweight Integrated Mast | Saab», Start. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.saab.com/products/saab-lightweight-integrated-mast>
- [20] V. Crupi, E. Guglielmino, M. Maestro, y A. Marinò, «Fatigue analysis of butt welded AH36 steel joints: Thermographic Method and design $S-N$ curve», *Marine Structures*, vol. 22, n.º 3, pp. 373-386, jul. 2009, doi: 10.1016/j.marstruc.2009.03.001.
- [21] R. Hsissou, R. Seghiri, Z. Benzekri, M. Hilali, M. Rafik, y A. Elharfi, «Polymer composite materials: A comprehensive review», *Composite Structures*, vol. 262, p. 113640, abr. 2021, doi: 10.1016/j.compstruct.2021.113640.
- [22] «NEW Guidance Note for propellers made of composite materials | Marine & Offshore». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://marine-offshore.bureauveritas.com/magazine/new-guidance-note-propellers-made-composite-materials>
- [23] C. I. Castanedo, «QUANTITATIVE SUBSURFACE DEFECT EVALUATION BY PULSED PHASE THERMOGRAPHY: DEPTH RETRIEVAL WITH THE PHASE».
- [24] G. A. Martínez, I. G. González, y L. P. Rial, «Utilización de cámaras termográficas low cost para la búsqueda automatizada de defectos constructivos mediante técnicas de deep learning».
- [25] S. S. Samareh-Mousavi *et al.*, «Monitoring fatigue delamination growth in a wind turbine blade using passive thermography and acoustic emission», *Structural Health Monitoring*, vol. 23, n.º 5, pp. 2906-2921, sep. 2024, doi: 10.1177/14759217231217179.
- [26] «Typical structure irregularities (defects) in polymers, which can...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Typical-structure-irregularities-defects-in-polymers-which-can-reduce-the-strength-of_fig4_270986401
- [27] J. Peng, K. Tian, L. Kou, Q. Zhang, F. Zhang, y N. Yan, «Physics-Guided LSTM for Defect Detection in Newly Rolled Rails Using Induction Thermography».
- [28] B. Unlu, F. Wechsler, Y. Pu, y C. Moser, «A unified multiscale 3D printer combining single-photon Tomographic Volumetric Additive Manufacturing and Two- Photon Polymerization».
- [29] L. V. Alvarez Marin, Z. Naghizadeh, T. Raman, y M. S. Peresin, «Optimization of material and extrusion parameters for wood flour–epoxy composites with potential additive-manufacturing applications in construction», *Eur. J. Wood Prod.*, vol. 84, n.º 1, p. 12, ene. 2026, doi: 10.1007/s00107-025-02362-5.
- [30] Y. Tao *et al.*, «A review on voids of 3D printed parts by fused filament fabrication», *Journal of Materials Research and Technology*, vol. 15, pp. 4860-4879, nov. 2021, doi: 10.1016/j.jmrt.2021.10.108.

- [31] Overo K. E y Damini Righteous Gilbert, «Directional Dependence in Non-Destructive Concrete Strength Prediction: Multivariate Modeling Using Oriented Rebound Hammer and Triaxial Ultrasonic Pulse Velocity Measurements», ene. 2026, doi: 10.5281/ZENODO.18146089.
- [32] H. Towsyfyan, A. Biguri, R. Boardman, y T. Blumensath, «Successes and challenges in non-destructive testing of aircraft composite structures», *Chinese Journal of Aeronautics*, vol. 33, n.º 3, pp. 771-791, mar. 2020, doi: 10.1016/j.cja.2019.09.017.
- [33] «Figure 1.4 An optical micrograph of delamination between 0° and 90° ply...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/An-optical-micrograph-of-delamination-between-0-and-90-ply-groups-in-a-damaged-laminate_fig3_282667680
- [34] «A scheme of barely visible impact damage (BVID) in a composite structure.», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/A-scheme-of-barely-visible-impact-damage-BVID-in-a-composite-structure_fig1_336775733
- [35] J. Lian, B. Cheng, S. Tao, Y. Zhang, S. Hou, y H. Yang, «Detection and Quantitative Evaluation of Internal Bubble Defects in Basin Insulators Using Infrared Thermography», *IEEE Access*, vol. 14, pp. 6698-6711, 2026, doi: 10.1109/ACCESS.2026.3651832.
- [36] G. Caprino, A. Langella, y V. Lopresto, «Indentation and penetration of carbon fibre reinforced plastic laminates», *Composites Part B: Engineering*, vol. 34, pp. 319-325, jun. 2003, doi: 10.1016/S1359-8368(03)00002-7.
- [37] L. Afroz, S. B. Inverarity, M. Qian, M. Easton, y R. Das, «Analysing the effect of defects on stress concentration and fatigue life of L-PBF AlSi10Mg alloy using finite element modelling», *Prog Addit Manuf*, vol. 9, n.º 2, pp. 341-359, abr. 2024, doi: 10.1007/s40964-023-00457-0.
- [38] Z. Shao, Z. Liu, Y. Zhang, y X. Zhu, «Fatigue Life Prediction and Reliability Assessment of CFRP Adhesively Bonded Joints in Offshore Wind Turbine Blade Applications: A Physics-Informed Data-Driven Approach», *Quality and Reliability Engineering International*, vol. 41, n.º 3, pp. 943-956, 2025, doi: 10.1002/qre.3715.
- [39] R. Mordia y A. K. Verma, «Nondestructive testing methods for rail defects detection», *High-speed Railway*, vol. 3, n.º 2, pp. 163-173, jun. 2025, doi: 10.1016/j.hspr.2025.03.001.
- [40] «Fig. 1. Patterns of component fractures at fatigue failure [4].», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Patterns-of-component-fractures-at-fatigue-failure-4_fig1_358870674
- [41] I. Garrido, S. Lagüela, S. Sfarra, F. J. Madruga, y P. Arias, «Automatic detection of moistures in different construction materials from thermographic images», *J Therm Anal Calorim*, vol. 138, n.º 2, pp. 1649-1668, oct. 2019, doi: 10.1007/s10973-019-08264-y.
- [42] C. Lerma, E. Barreira, y R. M. S. F. Almeida, «A discussion concerning active infrared thermography in the evaluation of buildings air infiltration», *Energy and Buildings*, vol. 168, pp. 56-66, jun. 2018, doi: 10.1016/j.enbuild.2018.02.050.
- [43] I. Danielski y M. Fröling, «In Situ Measurements of Thermal Properties of Building Fabrics Using Thermography under Non-Steady State Heat Flow Conditions», *Infrastructures*, vol. 3, n.º 3, p. 20, jul. 2018, doi: 10.3390/infrastructures3030020.
- [44] J. ALEXANDRA DE ALMEIDA FERREIRA, «Técnicas de Diagnóstico de Patologias Em Edifícios».
- [45] R. Nicholls-Lee *et al.*, «Non-destructive examination (NDE) methods for dynamic subsea cables for offshore renewable energy», *Prog. Energy*, vol. 4, n.º 4, p. 042011, oct. 2022, doi: 10.1088/2516-1083/ac8ccb.
- [46] Admin, «Mastering Ship Drydocking: A Comprehensive Guide to Success», Virtue Marine. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.virtuemarine.nl/post/mastering-ship-drydocking-a-comprehensive-guide-to-success>
- [47] K. L. Rens, T. J. Wipf, y F. W. Klaiber, «REVIEW OF NONDESTRUCTIVE EVALUATION TECHNIQUES OF CIVIL INFRASTRUCTURE», *Journal of Performance of Constructed*

- Facilities*, vol. 11, n.º 4, nov. 1997, Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://trid.trb.org/View/474797>
- [48] M. Solla y B. Riveiro, *Non-Destructive Techniques for the Evaluation of Structures and Infrastructure*. 2016, p. 388.
- [49] «FIGURE 3. Infrared thermography defect detection system.», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/nfrared-thermography-defect-detection-system_fig1_338721854
- [50] D. Gertsvolf, M. Horvat, D. Aslam, A. Khademi, y U. Berardi, «A U-net convolutional neural network deep learning model application for identification of energy loss in infrared thermographic images», *Applied Energy*, vol. 360, p. 122696, abr. 2024, doi: 10.1016/j.apenergy.2024.122696.
- [51] J. Zhou, G. Liu, Y. Gu, J. Deng, Y. Wen, y S. Chen, «A Box-Supervised Instance Segmentation Method for Insulator Infrared Images Based on Shuffle Polarized Self-Attention», *IEEE Transactions on Instrumentation and Measurement*, vol. PP, pp. 1-1, ene. 2023, doi: 10.1109/TIM.2023.3311073.
- [52] Z. Tong, L. Cheng, S. Xie, y M. Kersemans, «A flexible deep learning framework for thermographic inspection of composites», *NDT & E International*, vol. 139, p. 102926, oct. 2023, doi: 10.1016/j.ndteint.2023.102926.
- [53] S.-A. Keo, D. Defer, F. Breaban, y F. Brachelet, «Comparison between Microwave Infrared Thermography and CO₂ Laser Infrared Thermography in Defect Detection in Applications with CFRP», *MSA*, vol. 04, n.º 10, pp. 600-605, 2013, doi: 10.4236/msa.2013.410074.
- [54] L. C. M. Dafico, E. Barreira, R. M. S. F. Almeida, y R. Vicente, «Machine learning models applied to moisture assessment in building materials», *Construction and Building Materials*, vol. 405, p. 133330, nov. 2023, doi: 10.1016/j.conbuildmat.2023.133330.
- [55] S. A. Haseeb y M. Krawczuk, «A State-of-the-Art Review of Structural Health Monitoring Techniques for Wind Turbine Blades», *J Nondestruct Eval*, vol. 45, n.º 1, p. 4, dic. 2025, doi: 10.1007/s10921-025-01296-5.
- [56] D. Balageas, «Termografía Infrarroja : una técnica multifacética para la Evaluación No Destructiva (END).», ene. 2007.
- [57] S. Lagüela, R. Otero, y P. Arias, «Thermographic methodologies used in infrastructure inspection: A review—data acquisition procedures», *Infrared Physics & Technology*, vol. 111, p. 103481, sep. 2020, doi: 10.1016/j.infrared.2020.103481.
- [58] arvindpdmn Cherukuri Jagannadha Rao, «Non-Destructive Testing», Devopedia. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://devopedia.org/non-destructive-testing>
- [59] M. Miskiewicz, J. Lachowicz, P. Tysiac, P. Jaskula, y K. Wilde, «The application of non-destructive methods in the diagnostics of the approach pavement at the bridges», *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 356, p. 012023, may 2018, doi: 10.1088/1757-899X/356/1/012023.
- [60] Q. Digest, «What Is Destructive Testing?», Quality Digest. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.qualitydigest.com/inside/what-destructive-testing-091521.html>
- [61] M. Hassan Vand, P. Nop, y J. Tippner, «Assessment of the Impact Strength Properties of Thermally Modified Wood by Non-Destructive Testing», *J Nondestruct Eval*, vol. 45, n.º 1, p. 17, mar. 2026, doi: 10.1007/s10921-025-01313-7.
- [62] «Non Destructive Testing on Ships & Boats – Marine NDT | Eddyfi». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.eddyfi.com/en/application/ships>
- [63] «World Economic Forum», *World Economic Forum*.
- [64] R. Analia, A. Forster, S.-Q. Xie, y Z. Zhang, «Adaptive Thermal Imaging Signal Analysis for Real-Time Non-Invasive Respiratory Rate Monitoring», *Sensors*, vol. 26, n.º 1, p. 278, ene. 2026, doi: 10.3390/s26010278.
- [65] H. Heather-Smith, A. Smith, N. Dixon, J. Flint, y J. Wordingham, «Monitoring buried infrastructure deformation using acoustic emissions», ene. 2018, Accedido: 18 de marzo de 2026. [En línea]. Disponible en:

- https://repository.lboro.ac.uk/articles/conference_contribution/Monitoring_buried_infrastructure_deformation_using_acoustic_emissions/9428741/1
- [66] «Categories of non-destructive testing and evaluation techniques.», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Categories-of-non-destructive-testing-and-evaluation-techniques_fig2_339550855
- [67] S. Bagavathiappan, B. B. Lahiri, T. Saravanan, J. Philip, y T. Jayakumar, «Infrared thermography for condition monitoring – A review», *Infrared Physics & Technology*, vol. 60, pp. 35-55, sep. 2013, doi: 10.1016/j.infrared.2013.03.006.
- [68] «What are the Different Inspection Methods of Non-destructive Testing for Composites?», Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.addcomposites.com/post/non-destructive-testing-for-composites-different-inspection-methods?utm_source=chatgpt.com
- [69] S. Sfarra *et al.*, «Evaluation of defects in panel paintings using infrared, optical and ultrasonic techniques», *Insight*, vol. 54, n.º 1, pp. 21-27, ene. 2012, doi: 10.1784/insi.2012.54.1.21.
- [70] «Fig. 4 The ultrasonic C-scan equipment for impact damage assessment», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/The-ultrasonic-C-scan-equipment-for-impact-damage-assessment_fig3_341286620
- [71] G. Kim, S. Hong, K.-Y. Jhang, y G. H. Kim, «NDE of low-velocity impact damages in composite laminates using ESPI, digital shearography and ultrasound C-scan techniques», *Int. J. Precis. Eng. Manuf.*, vol. 13, n.º 6, pp. 869-876, jun. 2012, doi: 10.1007/s12541-012-0113-4.
- [72] G. Busse, D. V. Hemelrijck, I. Solodov, y A. Anastasopoulos, *Emerging Technologies in NDT*. CRC Press, 2008.
- [73] K. Luo, L. Chen, y W. Liang, «Structural health monitoring of carbon fiber reinforced polymer composite laminates for offshore wind turbine blades based on dual maximum correlation coefficient method», *Renewable Energy*, vol. 201, pp. 1163-1175, dic. 2022, doi: 10.1016/j.renene.2022.11.063.
- [74] «Fig. 2. (a) Acoustic emission (AE) test set-up, (b) steel and composite...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/a-Acoustic-emission-AE-test-set-up-b-steel-and-composite-tensile-specimens-T-EG_fig2_371270216
- [75] G. Mook, R. Lange, y O. Koeser, «Non-destructive characterisation of carbon-fibre-reinforced plastics by means of eddy-currents», *Composites Science and Technology*, vol. 61, n.º 6, pp. 865-873, may 2001, doi: 10.1016/S0266-3538(00)00164-0.
- [76] J. Cheng, J. Qiu, H. Ji, E. Wang, T. Takagi, y T. Uchimoto, «Application of low frequency ECT method in noncontact detection and visualization of CFRP material», *Composites Part B: Engineering*, vol. 110, pp. 141-152, feb. 2017, doi: 10.1016/j.compositesb.2016.11.018.
- [77] C. Xu, W. Zhang, C. Wu, J. Xie, X. Yin, y G. Chen, «An improved method of eddy current pulsed thermography to detect subsurface defects in glass fiber reinforced polymer composites», *Composite Structures*, vol. 242, p. 112145, jun. 2020, doi: 10.1016/j.compstruct.2020.112145.
- [78] Y. He, G. Tian, M. Pan, y D. Chen, «Non-destructive testing of low-energy impact in CFRP laminates and interior defects in honeycomb sandwich using scanning pulsed eddy current», *Composites Part B: Engineering*, vol. 59, pp. 196-203, mar. 2014, doi: 10.1016/j.compositesb.2013.12.005.
- [79] «Eddy Current Inspection of In-Service Aircraft Structure | Quality Magazine». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.qualitymag.com/articles/96942-eddy-current-inspection-of-in-service-aircraft-structure>
- [80] J. Newman, «Aerospace NDT with Advanced Laser Shearography».
- [81] M. Hung, Y. Chen, S. P. Ng, S. Shepard, Y. Hou, y J. Lhota, «Review and comparison of shearography and pulsed thermography for adhesive bond evaluation», *Optical Engineering - OPT ENG*, vol. 46, may 2007, doi: 10.1117/1.2741277.

- [82] D. Francis, R. P. Tatam, y R. M. Groves, «Shearography technology and applications: a review», *Meas. Sci. Technol.*, vol. 21, n.º 10, p. 102001, oct. 2010, doi: 10.1088/0957-0233/21/10/102001.
- [83] S. Gholizadeh, «A review of non-destructive testing methods of composite materials», *Procedia Structural Integrity*, vol. 1, pp. 50-57, ene. 2016, doi: 10.1016/j.prostr.2016.02.008.
- [84] C. A. C. Leckey, M. D. Rogge, y F. Raymond Parker, «Guided waves in anisotropic and quasi-isotropic aerospace composites: Three-dimensional simulation and experiment», *Ultrasonics*, vol. 54, n.º 1, pp. 385-394, ene. 2014, doi: 10.1016/j.ultras.2013.05.007.
- [85] L. Ma, C. Liu, J. Han, A. Wen, B. Liu, y J. Zheng, «Delamination defects in composite hydrogen storage cylinders: CT scanning and shearography measurement», *International Journal of Hydrogen Energy*, vol. 97, pp. 140-149, ene. 2025, doi: 10.1016/j.ijhydene.2024.11.447.
- [86] S. Liu y Z. Huang, «Analysis of strength property and pore characteristics of Taihang limestone using X-ray computed tomography at high temperatures», *Sci Rep*, vol. 11, n.º 1, p. 13478, jun. 2021, doi: 10.1038/s41598-021-92928-z.
- [87] M. Jiang y G. Wang, «Convergence of the simultaneous algebraic reconstruction technique (SART)», *IEEE Transactions on Image Processing*, vol. 12, n.º 8, pp. 957-961, ago. 2003, doi: 10.1109/TIP.2003.815295.
- [88] K. Naresh, K. A. Khan, R. Umer, y W. J. Cantwell, «The use of X-ray computed tomography for design and process modeling of aerospace composites: A review», *Materials & Design*, vol. 190, p. 108553, may 2020, doi: 10.1016/j.matdes.2020.108553.
- [89] C. Ibarra-Castanedo, M. Genest, J.-M. Piau, S. Guibert, A. Bendada, y X. P. V. Maldague, «Active infrared thermography techniques for the nondestructive testing of materials», en *Ultrasonic and Advanced Methods for Nondestructive Testing and Material Characterization*, WORLD SCIENTIFIC, 2007, pp. 325-348. doi: 10.1142/9789812770943_0014.
- [90] J. Erazo-Aux, H. Loaiza-Correa, A. D. Restrepo-Giron, C. Ibarra-Castanedo, y X. Maldague, «Thermal imaging dataset from composite material academic samples inspected by pulsed thermography», *Data in Brief*, vol. 32, p. 106313, oct. 2020, doi: 10.1016/j.dib.2020.106313.
- [91] D. A. González Fernández, *Contribuciones a las técnicas no destructivas para evaluación y prueba de procesos y materiales basadas en radiaciones infrarrojas*. Universidad de Cantabria, 2007. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://repositorio.unican.es/xmlui/handle/10902/1130>
- [92] X. Maldague y S. Marinetti, «Pulse phase infrared thermography», *Journal of Applied Physics*, vol. 79, n.º 5, pp. 2694-2698, mar. 1996, doi: 10.1063/1.362662.
- [93] N. Rajic, «Principal component thermography for flaw contrast enhancement and flaw depth characterisation in composite structures», *Composite Structures*, vol. 58, n.º 4, pp. 521-528, dic. 2002, doi: 10.1016/S0263-8223(02)00161-7.
- [94] «General schematic diagram of active thermography testing.», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/General-schematic-diagram-of-active-thermography-testing_fig1_357212141
- [95] «Infrared Thermography Explained | Reliable Plant». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.reliableplant.com/infrared-thermography-31572?utm_source=chatgpt.com
- [96] Y. Yang *et al.*, «Study on thermo-hydraulic characteristics of porous sandstone using infrared non-destructive testing and neural network prediction model», *Construction and Building Materials*, vol. 511, p. 145336, feb. 2026, doi: 10.1016/j.conbuildmat.2026.145336.
- [97] X. Maldague, «NDT Techniques: Thermographic», en *Encyclopedia of Materials: Science and Technology*, 2001, pp. 6036-6039. doi: 10.1016/B0-08-043152-6/01065-2.
- [98] «The relationship between photon energy and wavelength. According to the...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/The-relationship-between-photon-energy-and-wavelength-According-to-the-energy-equation_fig1_360070112

- [99] «6.2: Blackbody Radiation», Physics LibreTexts. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: [https://phys.libretexts.org/Bookshelves/University_Physics/University_Physics_\(OpenStax\)/University_Physics_III_-_Optics_and_Modern_Physics_\(OpenStax\)/06%3A_Photons_and_Matter_Waves/6.02%3A_Blackbody_Radiation](https://phys.libretexts.org/Bookshelves/University_Physics/University_Physics_(OpenStax)/University_Physics_III_-_Optics_and_Modern_Physics_(OpenStax)/06%3A_Photons_and_Matter_Waves/6.02%3A_Blackbody_Radiation)
- [100] «Figure 3. Spectra of radiation for black, grey and dark bodies. The...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Spectra-of-radiation-for-black-grey-and-dark-bodies-The-grey-and-dark-bodies-have-the_fig1_215571874
- [101] T. Lafargue-Tallet *et al.*, «Active thermo-reflectometry for absolute temperature measurement by infrared thermography on specular materials», *Sci Rep*, vol. 12, n.º 1, p. 7814, may 2022, doi: 10.1038/s41598-022-11616-8.
- [102] K. Tomita y M. Y. L. Chew, «A Review of Infrared Thermography for Delamination Detection on Infrastructures and Buildings», *Sensors*, vol. 22, n.º 2, ene. 2022, doi: 10.3390/s22020423.
- [103] «FIGURE 1 Principle of infrared thermal wave imaging detection (Qu et...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Principle-of-infrared-thermal-wave-imaging-detection-Qu-et-al-2020_fig1_363107736
- [104] «Fig.1: Schematic diagram of Pulsed Thermography in a reflection...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Schematic-diagram-of-Pulsed-Thermography-in-a-reflection-configuration_fig1_252954002
- [105] «a Transmission setup: IR-camera and light source are on different sides...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/a-Transmission-setup-IR-camera-and-light-source-are-on-different-sides-of-the-sample-b_fig5_294289047
- [106] «Setup for pulsed thermography with four flash lamps.», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Setup-for-pulsed-thermography-with-four-flash-lamps_fig2_335976813
- [107] «Infrared Thermography: Fundamentals and Applications | Request PDF», en *ResearchGate*. doi: 10.1201/b19024-8.
- [108] «Schematic of the lock-in thermography (LIT) system for the measurements...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Schematic-of-the-lock-in-thermography-LIT-system-for-the-measurements-of-elastocaloric_fig2_360036449
- [109] C. Meola, G. M. Carlomagno, D. Annicchiarico, M. Giordano, y M. Zarrelli, «Detection of Delamination in Carbon-Fibre-Reinforced Polymers with Lock-In Thermography», *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 224, n.º 11, pp. 1219-1227, nov. 2010, doi: 10.1243/09544100JAERO743.
- [110] «Figure 3. Step-heating thermography test: a) schematic position of...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Step-heating-thermography-test-a-schematic-position-of-voids-and-reinforcing-bars-in_fig1_328908539
- [111] «Left: Setup for the calibration of transmissivity and emissivity....», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Left-Setup-for-the-calibration-of-transmissivity-and-emissivity-Infrared-thermography_fig4_398339142
- [112] «Comparison between RAW thermograms and reconstructed thermograms after...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Comparison-between-RAW-thermograms-and-reconstructed-thermograms-after-applying-the-PCT_fig5_400262743
- [113] «Fig. 3 Logarithmic decay function describing the time to reach basal...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en:

- https://www.researchgate.net/figure/Logarithmic-decay-function-describing-the-time-to-reach-basal-ventilation-rate-OB-R-bpm_fig1_236143135
- [114] I. Garrido *et al.*, «Introduction of Deep Learning in Thermographic Monitoring of Cultural Heritage and Improvement by Automatic Thermogram Pre-Processing Algorithms».
- [115] «Fig. 4 The Discrete Fourier Transform for thermogram sequence scheme», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/The-Discrete-Fourier-Transform-for-thermogram-sequence-scheme_fig1_268025808
- [116] «Finite element method simulation on heat conduction of the sensor in 3...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Finite-element-method-simulation-on-heat-conduction-of-the-sensor-in-3-h-a-Temperature_fig13_51873388
- [117] «Full article: Advancements in synthetic polymers for 3D bioprinting materials, applications, and future prospects». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.tandfonline.com/doi/full/10.1080/00914037.2025.2562081>
- [118] «Figure 1. (a) Process of slicing 3D models into layers and lines for...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/a-Process-of-slicing-3D-models-into-layers-and-lines-for-printing-b-fabrication_fig1_340939404
- [119] «From hobby to 3D printing producer: A multi-perspective analysis of entrepreneurial evolution in Romania», Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://mrforum.com/product/9781644903995-3/>
- [120] C. C. Mahlenbrey, F. Menzel, T. Ziegler, y J. M. Neumaier, «Electronic Supplementary Information (ESI)».
- [121] «(a) A schematic depiction of the filament extrusion process. The...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/a-A-schematic-depiction-of-the-filament-extrusion-process-The-kneaded-composite_fig3_325763159
- [122] «Fig.2. Schematic representation of Stereolithography (SLA) process...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Schematic-representation-of-Stereolithography-SLA-process-Image-courtesy-of-3D-Systems_fig2_331313275
- [123] A. İ. Akbaba, «3D PRINTERS AND AREAS OF USAGE».
- [124] «Layer Height in 3D Printing: How to Balance Quality, Strength, and Speed», Kingroon 3D. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://kingroon.com/blogs/3d-print-101/layer-height-in-3d-printing>
- [125] «BCN3D Filaments – ABS Technical Data Sheet».
- [126] «Voids in the FDM printed parts. (a) Scanning Electron Microscope (SEM)...», ResearchGate. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.researchgate.net/figure/Voids-in-the-FDM-printed-parts-a-Scanning-Electron-Microscope-SEM-image-showing_fig1_347943763
- [127] F. N. Ainin, M. D. Azaman, L. M. Ferreira, C. a. C. P. Coelho, M. S. Abdul Majid, y M. J. M. Ridzuan, «Influence of Water Absorption on the Impact Response and Failure Mechanisms of 3D-Printed PLA-Based Honeycomb Sandwich Composite Structures», *Polymer Composites*, vol. 47, n.º 2, pp. 1895-1911, 2026, doi: 10.1002/pc.70264.
- [128] V. I. Kubich, O. G. Cherneta, V. M. Yurov, y V. S. Oleshko, «X-ray spectral analysis of surface structures formed on copper alloys during film fasting», *J. Phys.: Conf. Ser.*, vol. 1901, n.º 1, p. 012091, may 2021, doi: 10.1088/1742-6596/1901/1/012091.
- [129] B. Milovanović, M. Gaši, y S. Gumbarević, «Principal Component Thermography for Defect Detection in Concrete», *Sensors*, vol. 20, n.º 14, jul. 2020, doi: 10.3390/s20143891.
- [130] Q. Fang, C. Ibarra-Castanedo, D. Yuxia, J. Erazo-Aux, I. Garrido, y X. Maldague, «Defect Enhancement and Image Noise Reduction Analysis Using Partial Least Square-Generative

- Adversarial Networks (PLS-GANs) in Thermographic Nondestructive Evaluation», *J Nondestruct Eval*, vol. 40, n.º 4, p. 92, nov. 2021, doi: 10.1007/s10921-021-00827-0.
- [131] R. Slayton y G. Spinardi, «Radical innovation in scaling up: Boeing's Dreamliner and the challenge of socio-technical transitions», *Technovation*, vol. 47, pp. 47-58, ene. 2016, doi: 10.1016/j.technovation.2015.08.004.
- [132] «AA Me, IN». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.aame.in//2012/12/use-of-composites-in-india-aircraft.html>.
- [133] «PLA Basic», Bambu Lab EU Store. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://eu.store.bambulab.com/products/pla-basic-filament>
- [134] «ISO 527-1:2019», ISO. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.iso.org/standard/527-1>
- [135] «Materials Science and Engineering - Ed. Rees D. Rawlings». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.eolss.net/outlinecomponents/materials-science-engineering.aspx>
- [136] «An Experimental Investigation into Additive Manufacturing-Induced Residual Stresses in 316L Stainless Steel | Request PDF», *ResearchGate*, doi: 10.1007/s11661-014-2549-x.
- [137] «PLA-CF», Bambu Lab EU Store. Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://eu.store.bambulab.com/products/pla-cf>
- [138] A. Agarwalla, W. Ahmed, A. H. Al-Marzouqi, E. Zanelidin, T. A. Rizvi, y M. Khan, «Advancements in synthetic polymers for 3D bioprinting materials, applications, and future prospects», *International Journal of Polymeric Materials and Polymeric Biomaterials*, vol. 75, n.º 4, pp. 404-442, mar. 2026, doi: 10.1080/00914037.2025.2562081.
- [139] «FLIR E85 Advanced Thermal Camera - Discontinued | Flir». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: https://www.flir.com/products/e85/?srsltid=AfmBOorHa8AEzbw_Q0n-oPPvN8ffteFCz0ugSYqQXskCU0Vhui2tBiZe
- [140] «FLIR Tools Thermal Analysis and Reporting (Desktop) | Flir». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://www.flir.com/en-eu/products/flir-tools/>
- [141] «Google Colab». Accedido: 18 de marzo de 2026. [En línea]. Disponible en: <https://colab.research.google.com/>

ANEXO I: IMPLICACIONES SOCIALES, Y/O ECONÓMICAS, Y/O AMBIENTALES

Aspectos ambientales

El desarrollo del presente Trabajo Fin de Grado contribuye de forma directa a la sostenibilidad ambiental mediante la optimización de los procesos de inspección y mantenimiento de componentes fabricados por fabricación aditiva. La aplicación de la termografía infrarroja activa como técnica de Ensayo No Destructivo (END) permite la detección temprana de defectos sin necesidad de destruir las piezas, lo que reduce significativamente la generación de residuos y el consumo innecesario de materiales.

Asimismo, el empleo de materiales poliméricos como el PLA, de origen renovable y potencialmente biodegradable, contribuye a disminuir el impacto ambiental en aplicaciones de prototipado. Por otra parte, el uso de materiales compuestos como el PETG-CF, junto con la capacidad de verificar su integridad estructural, favorece el desarrollo de componentes más ligeros. Esta reducción de peso tiene un impacto directo en la disminución del consumo energético en sectores como el naval o aeroespacial, contribuyendo a la reducción de emisiones de gases de efecto invernadero.

En conjunto, la implementación de estas técnicas favorece un modelo de economía circular, al prolongar la vida útil de los componentes y evitar su sustitución prematura.

Aspectos económicos e industriales

Desde el punto de vista económico e industrial, este trabajo se enmarca dentro de los principios de la Industria 4.0, promoviendo la digitalización y automatización de los procesos de inspección mediante el uso de algoritmos avanzados como TSR, PCT y SPCT.

La aplicación de estas herramientas permite evolucionar desde estrategias de mantenimiento correctivo hacia modelos de mantenimiento predictivo (*Structural Health Monitoring*, SHM), lo que conlleva importantes beneficios económicos, tales como:

- Reducción de fallos inesperados y averías críticas.
- Disminución de los tiempos de inactividad operativa.
- Optimización de los costes de mantenimiento y reparación.

Además, la integración de la fabricación aditiva facilita la producción descentralizada de componentes, reduciendo los costes asociados al transporte, almacenamiento y logística.

Por otro lado, la posibilidad de emplear sistemas de adquisición térmica de menor coste, combinados con técnicas avanzadas de procesamiento de señal, contribuye a democratizar el acceso a tecnologías de inspección avanzadas, permitiendo su implementación en pequeñas y medianas empresas.

Aspectos sociales, de salud y seguridad

En el ámbito social y de seguridad, la principal aportación de este trabajo radica en la mejora de la fiabilidad estructural de componentes fabricados mediante impresión 3D. La detección de defectos internos no visibles, como porosidades, delaminaciones o vacíos subsuperficiales, resulta fundamental para prevenir fallos estructurales que podrían derivar en situaciones de riesgo en sectores críticos como el naval, aeroespacial o de infraestructuras.

Asimismo, la termografía infrarroja activa es una técnica no invasiva y sin contacto, lo que incrementa la seguridad de los operarios durante las inspecciones, permitiendo evaluar los componentes en condiciones reales de operación y a distancia.

Finalmente, la mejora en los procesos de control de calidad y en la fiabilidad de los productos contribuye a aumentar la confianza de la sociedad en las tecnologías de fabricación aditiva, favoreciendo su adopción en aplicaciones industriales y estratégicas.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

Reflexiones éticas y sociales en la memoria del TFG

A lo largo del desarrollo del presente Trabajo Fin de Grado, se han incorporado reflexiones éticas y sociales en diferentes apartados de la memoria, evidenciando el compromiso del trabajo con la seguridad, la sostenibilidad y la responsabilidad profesional. Estos aspectos se reflejan principalmente en:

Apartados 1.1 y 1.2 (Introducción y Objetivos): donde se establece la importancia de mejorar la seguridad operativa en componentes poliméricos destinados a aplicaciones de elevada exigencia técnica.

Apartado 2.1.4 (Consecuencias de los defectos): donde se analizan los riesgos asociados a la presencia de defectos internos, destacando la necesidad ética de implementar técnicas de inspección fiables que permitan prevenir fallos estructurales y proteger tanto a las personas como a los activos.

Apartado 5.1 (Conclusiones): donde se pone de manifiesto la relevancia de la termografía infrarroja activa como herramienta para garantizar la seguridad, la sostenibilidad y la fiabilidad de los sistemas en el contexto de la Industria 4.0.

Reflexión ética y social en el desarrollo del trabajo

Desde una perspectiva ética, el uso de técnicas avanzadas de detección de defectos basadas en algoritmos como TSR, PCT y SPCT implica una responsabilidad significativa en cuanto a la veracidad de los resultados y la transparencia en el diagnóstico estructural. En este sentido, el ingeniero debe comunicar de forma rigurosa las capacidades y limitaciones de dichas técnicas, evitando interpretaciones erróneas que puedan comprometer la seguridad.

En el presente trabajo se ha mantenido un enfoque de honestidad técnica al evidenciar que el material PLA utilizado ofrece un comportamiento más favorable que el PETG-CF en términos de detectabilidad de defectos. La omisión o manipulación de este tipo de resultados podría derivar en la validación de componentes defectuosos, lo que supondría un riesgo inaceptable en sectores críticos como el naval o el de Defensa. Por ello, el rigor científico y la seguridad deben prevalecer siempre sobre criterios exclusivamente económicos.

Desde el punto de vista social, la implementación de tecnologías de inspección basadas en termografía infrarroja activa contribuye a mejorar la seguridad de las infraestructuras y sistemas, reduciendo la probabilidad de fallos inesperados. No obstante, esta evolución tecnológica también implica una transformación en el ámbito laboral, requiriendo la formación y adaptación de los profesionales hacia competencias digitales y el manejo de herramientas avanzadas de diagnóstico.

Por otro lado, el desarrollo y validación de soluciones basadas en sensores de menor coste favorece la democratización del acceso a tecnologías de inspección avanzadas. Esto permite que pequeñas y medianas empresas puedan beneficiarse de herramientas de control de calidad más sofisticadas, reduciendo la brecha tecnológica existente.

Finalmente, la mejora en la capacidad de detección de defectos internos contribuye a reforzar la confianza de la sociedad en la fabricación aditiva, facilitando su adopción en aplicaciones industriales. De este modo, este trabajo no solo aporta una solución técnica, sino que también promueve un enfoque de ingeniería responsable, orientado a la seguridad, la sostenibilidad y el bienestar social.

ANEXO III: CÓDIGO DE PROCESAMIENTO Y ALGORITMOS TÉCNICOS

Introducción

El presente anexo recoge la implementación técnica de los algoritmos empleados para el procesamiento de las secuencias termográficas obtenidas durante los ensayos experimentales con materiales PETG-CF y PLA.

El desarrollo se ha realizado en lenguaje *Python* dentro del entorno Google Colab, utilizando librerías especializadas en análisis de datos y cálculo numérico. El flujo de trabajo se estructura en distintos bloques funcionales, que abarcan desde la carga y organización de los datos térmicos hasta la evaluación cuantitativa de los resultados mediante la métrica *Signal-to-Noise Ratio* (SNR).

Implementación de algoritmos de procesamiento

1. Bloque de carga y organización de datos (Bloques 1–4)

Este bloque tiene como objetivo la importación, estructuración y organización de las secuencias termográficas. Los datos se transforman en una matriz tridimensional (espacio-tiempo), lo que permite analizar la evolución térmica de cada píxel a lo largo del tiempo.

```
# BLOQUE 1: Importación Google Drive
from google.colab import drive

# Montamos Google Drive
drive.mount('/content/drive')
print("✅ Drive montado correctamente.")
```

```
# BLOQUE 2: Procesar imágenes
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tqdm import tqdm

def procesar_csv_termico():
    print("--- ⚙️ CONFIGURACIÓN DEL PROCESO ---")

    # 1. Solicitar Ruta
    ruta_archivo = input("1. Pega la ruta completa de tu archivo .csv en Drive: ").strip()
    # Eliminar comillas si el usuario las pone por error
    ruta_archivo = ruta_archivo.replace("'", '').replace('"', '')

    if not os.path.exists(ruta_archivo):
        print(f"❌ Error: No encuentro el archivo en: {ruta_archivo}")
        return

    # 2. Solicitar Parámetros de la Cámara (CRUCIAL para separar las imágenes apiladas)
    print("\n--- 📷 DATOS DE LA IMAGEN ---")
    print("Necesito saber la resolución de la FLIR E85 (o la del recorte si lo has editado).")
    try:
        filas_img = int(input("Número de FILAS (alto) de una sola imagen (ej. 288) -- HAY QUE RESTAR UNA UNIDAD (1): "))
        cols_img = int(input("Número de COLUMNAS (ancho) de una sola imagen (ej. 384): "))
        fps = float(input("FPS de la grabación original (ej. 30): "))
        segundos_intervalo = float(input("Intervalo deseado en SEGUNDOS para guardar (ej. 1): "))
```

```

# Opcional: Saltar encabezados del CSV
# ¡IMPORTANTE! Vuelve a introducir aquí el número de filas de encabezado (texto/metadata) que tiene tu CSV.
# Si el error "Expected N fields" persiste, es muy probable que este valor sea incorrecto.
except ValueError:
    print("❌ Error: Debes introducir números válidos.")
    return

# --- PREPARACIÓN DE CARPETAS ---
directorio_base = os.path.dirname(ruta_archivo)
nombre_archivo = os.path.splitext(os.path.basename(ruta_archivo))[0]

folder_imagenes = os.path.join(directorio_base, f"{nombre_archivo}_IMAGENES")
folder_csv_final = os.path.join(directorio_base, f"{nombre_archivo}_RESULTADOS")

os.makedirs(folder_imagenes, exist_ok=True)
os.makedirs(folder_csv_final, exist_ok=True)
print(f"📁 Guardando resultados en:\n - {folder_imagenes}\n - {folder_csv_final}")

# --- CARGA DE DATOS ---
print(f"\n⌚ Leyendo archivo CSV de entrada (esto puede tardar si es muy pesado)...")
try:
    # Se cambia sep=None a sep=';' para forzar el separador de columnas a punto y coma.
    # El parámetro 'decimal=',' está configurado para interpretar la coma como separador decimal.
    df_raw = pd.read_csv(ruta_archivo, sep=';', skiprows = 4, on_bad_lines='warn', engine='python', header=None, encoding='utf-8-sig', decimal=',')
except Exception as e:
    print(f"❌ Error leyendo el CSV: {e}")
    return

```

```

# --- Modificaciones solicitadas: Eliminar la primera columna ---
# Asegúrate de que 'skip_rows' se haya configurado correctamente para las filas iniciales.
df_raw = df_raw.iloc[3:, :]
df_raw = df_raw.iloc[:, 1:]
print(f"✅ Se ha eliminado la primera columna del CSV (las filas iniciales se saltan con 'skip_rows').")
# -----

total_filas_csv = df_raw.shape[0]
total_cols_csv = df_raw.shape[1]

# Verificación de integridad
if total_cols_csv != cols_img:
    print(f"⚠️ ¡ADVERTENCIA!: El CSV tiene {total_cols_csv} columnas, pero dijiste que la imagen tiene {cols_img}.")
    print("El código intentará continuar, pero revisa si la resolución es correcta.")

# Calculamos cuántos frames (imágenes) hay en total apilados verticalmente
total_frames_encontrados = total_filas_csv // filas_img
print(f"🖼️ Filas totales CSV: {total_filas_csv}. Frames detectados: {total_frames_encontrados}")

# Calcular salto de frames según el intervalo pedido
frames_a_salto = int(fps * segundos_intervalo)
if frames_a_salto < 1: frames_a_salto = 1

print(f"📄 Se procesará 1 imagen cada {frames_a_salto} frames originales.")

# --- PROCESAMIENTO ---
matriz_final_columnas = []
encabezados_tiempo = []

```

```
# Iteramos sobre los bloques de filas
for i in tqdm(range(0, total_frames_encontrados, frames_a_saltar), desc="Procesando imágenes"):

    # Calcular tiempo actual
    tiempo_actual = i / fps

    # Definir índice de inicio y fin de filas para ESTA imagen
    start_row = i * filas_img
    end_row = start_row + filas_img

    # Extraer el bloque de datos correspondiente a una imagen (DataFrame -> Numpy Array)
    img_data = df_raw.iloc[start_row:end_row, :cols_img].to_numpy(dtype=float)

    # 1. GUARDAR COMO IMAGEN VISUAL (.jpg)
    # Necesitamos normalizar los datos de temperatura a 0-255 para que sea una imagen visible
    img_norm = (img_data - np.min(img_data)) / (np.max(img_data) - np.min(img_data))
    img_uint8 = (img_norm * 255).astype(np.uint8)

    # Se cambia el formato para que el tiempo sea un entero
    nombre_foto = f"img_{i:05d}_t_{int(tiempo_actual/60)}min.jpg"
    ruta_foto = os.path.join(folder_imagenes, nombre_foto)

    # Usamos colormap 'inferno' o 'gray' para termografía
    plt.imsave(ruta_foto, img_uint8, cmap='inferno')
```

```
# 2. PROCESAR PARA MATRIZ FINAL (Columnar)
# .flatten() en numpy lee de izquierda a derecha y baja fila a fila.
columna_datos = img_data.flatten()

matriz_final_columnas.append(columna_datos)
encabezados_tiempo.append(f"t_{tiempo_actual/60}min")

# --- GUARDAR CSV FINAL ---
print("\n📄 Generando matriz final transpuesta...")

if matriz_final_columnas:
    # column_stack convierte la lista de arrays (filas) en columnas verticales
    matriz_np = np.column_stack(matriz_final_columnas)

    df_resultado = pd.DataFrame(matriz_np, columns=encabezados_tiempo)

    path_resultado = os.path.join(folder_csv_final, f"{nombre_archivo}_PROCESADO_MATRIZ.csv")
    df_resultado.to_csv(path_resultado, index=False)

    print("\n✅ ¡ÉXITO!")
    print(f"📄 Matriz guardada: {path_resultado}")
    print(f"📏 Tamaño matriz: {df_resultado.shape} (Filas [píxeles] x Columnas [tiempo])")
    print(f"📁 Imágenes guardadas en carpeta: {folder_imagenes}")
```

```
else:
    print("⚠️ No se generaron datos. Revisa los parámetros de entrada.")

# EJECUTAR
procesar_csv_termico()
```

```

# BLOQUE 3: Unión matrices de cada grabación (cada columna es una imagen de un instante temporal diferente y justo el siguiente con respecto al anterior)
import os
import glob
import pandas as pd
from google.colab import drive
drive.mount('/content/drive')

csv_folder_path = input("1. Pega la ruta COMPLETA de la carpeta en Google Drive donde se encuentran tus archivos CSV: ").strip()
csv_folder_path = csv_folder_path.replace("'", '').replace('"', '')

if not os.path.exists(csv_folder_path):
    print(f"❌ Error: No encuentro la carpeta en: {csv_folder_path}")
elif not os.path.isdir(csv_folder_path):
    print(f"❌ Error: La ruta proporcionada no es un directorio: {csv_folder_path}")
else:
    print(f"✅ Carpeta de CSVs encontrada: {csv_folder_path}")

# List all files in the directory
all_files = os.listdir(csv_folder_path)

# Filter for CSV files (all of them) and sort alphabetically
csv_files_to_concatenate = sorted([
    os.path.join(csv_folder_path, f)
    for f in all_files
    if f.endswith('.csv')
])

```

```

if not csv_files_to_concatenate:
    print(f"❌ Error: No se encontraron archivos CSV en la carpeta: {csv_folder_path}")
else:
    print(f"✅ Se encontraron {len(csv_files_to_concatenate)} archivos CSV para concatenar:")
    for f in csv_files_to_concatenate:
        print(f" - {os.path.basename(f)}")

# Initialize an empty list to hold dataframes
dataframes_to_concat = []

# Read each CSV file and append its DataFrame to the list
if csv_files_to_concatenate:
    print("⌚ Leyendo y combinando archivos CSV...")
    for file_path in csv_files_to_concatenate:
        try:
            # Assuming CSVs are comma-separated and have a header
            df = pd.read_csv(file_path)
            dataframes_to_concat.append(df)
            print(f" - Leído: {os.path.basename(file_path)}")
        except Exception as e:
            print(f"❌ Error al leer {os.path.basename(file_path)}: {e}")
            continue

```

```

if dataframes_to_concat:
    # Concatenate all dataframes horizontally (axis=1)
    combined_df = pd.concat(dataframes_to_concat, axis=1)
    print("✅ Archivos CSV combinados exitosamente.")
    print(f"📏 Dimensiones del DataFrame combinado: {combined_df.shape}")
    print("Primeras 5 filas del DataFrame combinado:")
    print(combined_df.head())
else:
    print("❌ No se pudo combinar ningún archivo CSV válido.")
else:
    print("⚠️ No hay archivos CSV para combinar. Por favor, asegúrate de que la ruta y los prefijos sean correctos.")

output_filename = input("3. Introduce un nombre para el archivo CSV combinado (ej. 'Archivos_Combinados.csv'): ").strip()

# Construir la ruta completa para el archivo de salida
output_filepath = os.path.join(csv_folder_path, output_filename)

# Guardar el DataFrame combinado en un nuevo archivo CSV
try:
    combined_df.to_csv(output_filepath, index=False)
    print(f"✅ DataFrame combinado guardado exitosamente en: {output_filepath}")
    print(f"📏 Dimensiones del archivo guardado: {combined_df.shape}")
except Exception as e:
    print(f"❌ Error al guardar el archivo CSV combinado: {e}")

```

```

#BLOQUE 4: Llamar y cargar la matriz combinada
import os
import pandas as pd
from google.colab import drive
drive.mount('/content/drive')

ruta_csv_combinado = input("1. Pega la ruta COMPLETA en Google Drive de tu archivo CSV combinado para el análisis PCT: ").strip()
ruta_csv_combinado = ruta_csv_combinado.replace("'", '').replace('"', '')

if not os.path.exists(ruta_csv_combinado):
    print(f"❌ Error: No encuentro el archivo en la ruta especificada: {ruta_csv_combinado}")
elif not os.path.isfile(ruta_csv_combinado):
    print(f"❌ Error: La ruta proporcionada no es un archivo válido: {ruta_csv_combinado}")
else:
    print(f"✅ Archivo CSV combinado encontrado: {ruta_csv_combinado}")

try:
    # Load the combined CSV into a DataFrame
    combined_df = pd.read_csv(ruta_csv_combinado)
    print("✅ CSV combinado cargado exitosamente en un DataFrame.")
    print("Primeras 5 filas del DataFrame combinado:")
    print(combined_df.head())
except Exception as e:
    print(f"❌ Error al cargar el archivo CSV combinado: {e}")

```

```

print("\n--- 📄 DATOS DE LA IMAGEN PARA EL ANÁLISIS TÉRMICO ---")
print("Para el análisis térmico, necesito las dimensiones originales de las imágenes que formaron este CSV combinado.")
try:
    rows_img_pct = int(input("Número de FILAS (alto) de una sola imagen: "))
    cols_img_pct = int(input("Número de COLUMNAS (ancho) de una sola imagen: "))
    print(f"✅ Dimensiones de imagen para el análisis térmico: {rows_img_pct} filas x {cols_img_pct} columnas.")
except ValueError:
    print("❌ Error: Debes introducir números enteros válidos para las dimensiones de la imagen.")
    rows_img_pct = None
    cols_img_pct = None

```

2. Análisis multivariante: PCT y SPCT (Bloques 5–6)

En este bloque se aplican técnicas de reducción de dimensionalidad mediante Análisis de Componentes Principales (PCT) y su variante dispersa (SPCT). Estas técnicas permiten extraer las componentes más relevantes del campo térmico, generando las denominadas eigenimágenes (EOF y SC), que facilitan la identificación de defectos.

```

#BLOQUE 5: Aplicación Principal Component Thermopgraphy/Analysis
import os
import numpy as np
from sklearn.preprocessing import StandardScaler

# Convert DataFrame to NumPy array
# Transpose combined_df so that each column represents an image (time instant)
# and each row represents a pixel.
# This matches the typical input format for PCA/SVD where observations are rows and features are columns.
# However, the previous `matrizT_to_PCT_2` in BLOQUE 4 had pixels as rows and time instants as columns.
# To maintain consistency with the original code's SVD application (where 'u' gives EOFs per pixel),
# we need the input matrix for SVD to have pixels as rows and time instants as columns.
# combined_df already has pixels as rows and time instants as columns, so no transpose needed here yet.
# We will convert it to numpy array directly.

matrizT_to_PCT = combined_df.to_numpy(dtype=float)

print(f"✅ DataFrame convertido a array NumPy. Forma: {matrizT_to_PCT.shape}")

# Standardize the data
# StandardScaler expects samples as rows and features as columns.
# In our case, 'pixels' are features and 'time instants' are samples if we consider a pixel's temperature over time.
# Or, if we consider each image (time instant) as a sample, then pixels are features.
# The original PCT code (BLOQUE 4) standardizes across time (columns), meaning each pixel's temperature profile is standardized.
# Let's align with that: each column (time instant) is considered a 'sample' for standardization of pixel values.

scaler = StandardScaler()
matrizT_to_PCT_standard_procedure = scaler.fit_transform(matrizT_to_PCT)

print("✅ Datos estandarizados (media 0, desviación estándar 1).")

# Apply SVD
u, s, vh = np.linalg.svd(matrizT_to_PCT_standard_procedure, full_matrices=False)

print("✅ SVD aplicado exitosamente.")
print(f" - Dimensiones de U (Eigenimágenes/EOFs): {u.shape}")
print(f" - Dimensiones de S (valores singulares): {s.shape}")
print(f" - Dimensiones de Vh (componentes principales temporales): {vh.shape}")

# Get the base directory from the combined CSV path
directorio_base_pct = os.path.dirname(ruta_csv_combinado)
nombre_archivo_pct = os.path.splitext(os.path.basename(ruta_csv_combinado))[0]

# Create a new subfolder for PCT results
folder_pct_results = os.path.join(directorio_base_pct, f"{nombre_archivo_pct}_PCT_RESULTADOS")
folder_eigenimages = os.path.join(folder_pct_results, "Eigenimages")

os.makedirs(folder_eigenimages, exist_ok=True)
print(f"📁 Carpeta para resultados PCT creada: {folder_eigenimages}")

# Ask the user which EOFs to save
print(f"\n📄 Se han calculado {s.shape[0]} Eigenimágenes (EOFs).")
eofs_to_save_str = input("2. Introduce los índices de las Eigenimágenes (EOFs) que deseas guardar (ej. '0,1,2,3' o '0-5'): ").strip()

```

```
eofs_indices = []
if '-' in eofs_to_save_str:
    start, end = map(int, eofs_to_save_str.split('-'))
    eofs_indices = list(range(start, end + 1))
elif ',' in eofs_to_save_str:
    eofs_indices = [int(i.strip()) for i in eofs_to_save_str.split(',')]
else:
    try:
        eofs_indices.append(int(eofs_to_save_str))
    except ValueError:
        print("❌ Formato de entrada no válido. Por favor, usa '0,1,2' o '0-5'.")

# Filter out invalid indices
eofs_indices = [idx for idx in eofs_indices if 0 <= idx < s.shape[0]]

if not eofs_indices:
    print("⚠️ No se seleccionaron Eigenimágenes válidas para guardar.")
else:
    print(f"✅ Se guardarán las Eigenimágenes con índices: {eofs_indices}")
```

#Bloque 5_2: Guardado de las autoimágenes (EOFs), valores singulares (s) y componentes temporales (vh)

```
import matplotlib.pyplot as plt
import numpy as np # Asegurarse de que numpy esté disponible
import pandas as pd # Importar pandas para guardar CSV de vh

if eofs_indices and rows_img_pct is not None and cols_img_pct is not None:
    print("\n📁 Guardando Eigenimágenes seleccionadas...")
    # Prepare a list to store all Eigenimage data (flattened) for CSV export
    all_eigenimages_data = []
    eigenimage_headers = []

    for i in eofs_indices:
        if i < u.shape[1]: # Ensure index is within bounds
            # Extract the i-th Eigenimage (column from U matrix)
            eigenimage_flat = u[:, i]

            # Reshape the flattened Eigenimage back to original image dimensions
            eigenimage_2d = eigenimage_flat.reshape(rows_img_pct, cols_img_pct)
```

```

# --- Modificación para añadir título y colorbar ---
fig, ax = plt.subplots(figsize=(cols_img_pct / 50, rows_img_pct / 50)) # Ajustar tamaño dinámicamente
img = ax.imshow(eigenimage_2d, cmap='viridis', origin='upper') # Mostrar datos no normalizados para la escala
cbar = fig.colorbar(img, ax=ax, orientation='vertical', shrink=0.8)
cbar.set_label('Valor')
ax.set_title(f'Eigenimagen (EOF) {i}')
ax.set_xticks([])
ax.set_yticks([])
ax.set_xticklabels([])
ax.set_yticklabels([])
# -----

# Save as PNG image
nombre_foto = f"EOF_{i:03d}.png"
ruta_foto = os.path.join(folder_eigenimages, nombre_foto)
plt.savefig(ruta_foto, bbox_inches='tight', pad_inches=0.1) # Guardar la figura
plt.close(fig) # Cerrar la figura para liberar memoria
print(f" - Guardado: {os.path.basename(ruta_foto)}")

# Store flattened Eigenimage data for CSV
all_eigenimages_data.append(eigenimage_flat)
eigenimage_headers.append(f"EOF_{i}")

```

```

# Save all selected Eigenimage raw data to a single CSV
if all_eigenimages_data:
    # Stack the flattened Eigenimages as columns
    df_eigenimages_raw = pd.DataFrame(np.column_stack(all_eigenimages_data), columns=eigenimage_headers)
    csv_output_path = os.path.join(folder_pct_results, f"{nombre_archivo_pct}_Eigenimages_RawData.csv")
    df_eigenimages_raw.to_csv(csv_output_path, index=False)
    print(f"✅ Datos crudos de Eigenimágenes guardados en: {csv_output_path}")
else:
    print("⚠️ No hay datos de Eigenimágenes válidos para guardar en CSV.")

# --- GUARDAR VALORES SINGULARES (s) ---
folder_singular_values = os.path.join(folder_pct_results, "valores_singulares")
os.makedirs(folder_singular_values, exist_ok=True)

df_singular_values = pd.DataFrame(s/np.sum(s[0:len(s)])*100, columns=['Valor_Singular'])
singular_values_path = os.path.join(folder_singular_values, f"{nombre_archivo_pct}_Valores_Singulares.csv")
df_singular_values.to_csv(singular_values_path, index=False)
print(f"✅ Valores singulares guardados en: {singular_values_path}")

```

```

# --- GUARDAR VALORES SINGULARES (s) ---
folder_singular_values = os.path.join(folder_pct_results, "valores_singulares")
os.makedirs(folder_singular_values, exist_ok=True)

df_singular_values = pd.DataFrame(s/np.sum(s[0:len(s)])*100, columns=['Valor_Singular'])
singular_values_path = os.path.join(folder_singular_values, f"{nombre_archivo_pct}_Valores_Singulares.csv")
df_singular_values.to_csv(singular_values_path, index=False)
print(f"✅ Valores singulares guardados en: {singular_values_path}")

# Plot y guardar Valores Singulares
if eof_indices: # Asegurarse de que haya EOFs seleccionados
    max_eof_index = max(eof_indices) # Obtener el índice más alto elegido por el usuario
    # Graficar hasta este índice, más 1 para incluirlo si es 0-basado
    num_singular_values_to_plot = max_eof_index + 1 if max_eof_index < len(s) else len(s)

    plt.figure(figsize=(10, 6))
    plt.bar(range(num_singular_values_to_plot), df_singular_values['Valor_Singular'].iloc[0:num_singular_values_to_plot], color='skyblue')
    plt.title('Valores Singulares (Importancia de cada Eigenimagen)')
    plt.xlabel('Eigenimagen (Índice EOF)')
    plt.ylabel('Porcentaje de Varianza Explicada (%)')
    plt.xticks(range(num_singular_values_to_plot))
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    singular_values_plot_path = os.path.join(folder_singular_values, f"{nombre_archivo_pct}_Valores_Singulares_Plot.png")
    plt.savefig(singular_values_plot_path)
    plt.close() # Cerrar el plot para liberar memoria
    print(f"✅ Gráfica de Valores Singulares guardada en: {singular_values_plot_path}")

```

```
# --- GUARDAR COMPONENTES TEMPORALES (vh) EN CSV Y GRÁFICA ---
folder_temporal_components = os.path.join(folder_pct_results, "componentes_principales_temporales")
os.makedirs(folder_temporal_components, exist_ok=True)

# Obtener los instantes de tiempo del DataFrame combinado original
# Asumiendo que las columnas de combined_df son los instantes de tiempo
time_labels = combined_df.columns

# MODIFICACIÓN: Extraer los valores numéricos del tiempo, manejando sufijos como '.1', '.2', etc.
time_in_minutes_parsed = []
for label in time_labels:
    stripped_label = label.replace('t_', '').replace('min', '')
    parts = stripped_label.split('.')
    if len(parts) > 2:
        # If it has a '.N' suffix, take only the time part before it
        time_str = f"{parts[0]}.{parts[1]}"
    else:
        # Otherwise, it's just the normal time string
        time_str = stripped_label
    time_in_minutes_parsed.append(float(time_str))

# --- INICIO DE MODIFICACIÓN PARA TIEMPO ACUMULATIVO ---
cumulative_time_in_minutes = []
current_offset = 0.0
last_time_value = -1.0 # Initialize to ensure the first segment starts at 0
```

```
for i, current_time_val in enumerate(time_in_minutes_parsed):
    if current_time_val < last_time_value: # Detects when a new segment starts (time resets or decreases)
        current_offset += (last_time_value + 1.0) # Add duration of previous segment (last time point + 1 minute)

    cumulative_time_in_minutes.append(current_time_val + current_offset)
    last_time_value = current_time_val

time_in_minutes = cumulative_time_in_minutes # Overwrite the non-cumulative list with the cumulative one
# --- FIN DE MODIFICACIÓN PARA TIEMPO ACUMULATIVO ---

# Collect selected vh data for CSV
temporal_components_data = []
temporal_components_headers = []

# Sort eofs_indices to ensure consistent column order in CSV
sorted_eofs_indices = sorted(eofs_indices)

for i in sorted_eofs_indices:
    if i < vh.shape[0]: # Ensure index is within bounds
        temporal_components_data.append(vh[i, :])
        temporal_components_headers.append(f"EOF_{i}")
```

```
if temporal_components_data:
    # Transpose the collected data so each component becomes a column
    df_temporal_components_raw = pd.DataFrame(np.array(temporal_components_data).T, columns=temporal_components_headers)
    # Insert time_in_minutes as the first column
    df_temporal_components_raw.insert(0, 'Tiempo_min', time_in_minutes)

    csv_temporal_output_path = os.path.join(folder_temporal_components, f"{nombre_archivo_pct}_Componentes_Temporales_RawData.csv")
    df_temporal_components_raw.to_csv(csv_temporal_output_path, index=False)
    print(f"✅ Datos crudos de Componentes Temporales (vh) guardados en: {csv_temporal_output_path}")
else:
    print("⚠ No hay datos de Componentes Temporales válidos para guardar en CSV.")

# Plotting vh components
plt.figure(figsize=(14, 7))
for i in eofs_indices: # Iterar solo sobre los EOFs seleccionados
    if i < vh.shape[0]: # Asegurarse de que el índice esté dentro de los límites de las filas de vh
        plt.plot(time_in_minutes, vh[i, :], label=f'Componente Temporal EOF_{i}')
```

```

    if eofs_indices: # Solo mostrar el plot si hay EOFs para graficar
        plt.title('Evolución Temporal de los Coeficientes (vh) de las Eigenimágenes Seleccionadas')
        plt.xlabel('Tiempo (minutos)')
        plt.ylabel('Amplitud del Coeficiente (Unidades estandarizadas)')
        plt.legend()
        plt.grid(True)
        temporal_components_plot_path = os.path.join(folder_temporal_components, f"{nombre_archivo_pct}_Componentes_Temporales_Plot.png")
        plt.savefig(temporal_components_plot_path)
        plt.close() # Cerrar el plot para liberar memoria
        print(f"✅ Gráfica de Componentes Temporales (vh) guardada en: {temporal_components_plot_path}")
    else:
        print("⚠️ No se seleccionaron Eigenimágenes válidas para graficar componentes temporales.")

    print("\n✅ Proceso de PCT completado exitosamente.")
else:
    print("❌ No se pudieron guardar las Eigenimágenes. Asegúrate de que las dimensiones de la imagen y los índices de EOF sean válidos.")

```

```

#Bloque 5_3: Mapa de calor correlación entre las distintas EOFs
import seaborn as sns
import os
import matplotlib.pyplot as plt

# 1. Calcular Matriz de Correlación
# Seleccionar solo las columnas que corresponden a las Componentes Temporales (EOF_X)
# Excluir la columna 'Tiempo_min' si está presente
temporal_component_columns = [col for col in df_temporal_components_raw.columns if col.startswith('EOF_')]
correlation_matrix = df_temporal_components_raw[temporal_component_columns].corr(method='pearson')

print("✅ Matriz de correlación de las Componentes Temporales calculada.")
print("Primeras 5x5 de la matriz de correlación:")
print(correlation_matrix.head().iloc[:, :5]) # Display first 5x5 for brevity

# Asegúrate de que la carpeta de componentes temporales exista
os.makedirs(folder_temporal_components, exist_ok=True)

# 3. Guardar Matriz de Correlación en CSV
csv_correlation_path = os.path.join(folder_temporal_components, f"{nombre_archivo_pct}_Correlation_Matrix.csv")
correlation_matrix.to_csv(csv_correlation_path, index=True) # index=True para guardar los nombres de las EOFs como índice
print(f"✅ Matriz de correlación guardada en: {csv_correlation_path}")

```

```

# 2. Visualizar Correlación (Heatmap)
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Matriz de Correlación de las Componentes Temporales (EOF_X)')
plt.xlabel('Componentes Temporales (EOF_X)')
plt.ylabel('Componentes Temporales (EOF_X)')
plt.tight_layout() # Ajustar el diseño para evitar recortes

# 4. Guardar Heatmap de Correlación
heatmap_correlation_path = os.path.join(folder_temporal_components, f"{nombre_archivo_pct}_Correlation_Heatmap.png")
plt.savefig(heatmap_correlation_path)
plt.close() # Cerrar el plot para liberar memoria
print(f"✅ Heatmap de correlación guardado en: {heatmap_correlation_path}")

# 5. Final Task
print("\n--- Análisis de Correlación de Componentes Temporales Completado ---")
print("Se ha calculado la matriz de correlación de Pearson entre las Componentes Temporales.")
print("La matriz de correlación ha sido guardada en:", csv_correlation_path)
print("La visualización del heatmap de correlación ha sido guardada en:", heatmap_correlation_path)

```

```
temporal_components_only = df_temporal_components_raw.filter(like='EOF_')

correlation_matrix = temporal_components_only.corr(method='pearson')

print("✅ Matriz de correlación de Pearson calculada para las Componentes Temporales.")
print("Primeras 5 filas y 5 columnas de la matriz de correlación:")
print(correlation_matrix.iloc[:5, :5])

# Ensure the temporal components folder exists
os.makedirs(folder_temporal_components, exist_ok=True)

# Save Correlation Matrix to CSV
csv_correlation_path = os.path.join(folder_temporal_components, f"{nombre_archivo_pct}_Correlation_Matrix.csv")
correlation_matrix.to_csv(csv_correlation_path, index=True) # index=True to save EOF names as index
print(f"✅ Matriz de correlación guardada en: {csv_correlation_path}")

# Visualize Correlation (Heatmap)
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Matriz de Correlación de las Componentes Temporales (EOF_X)')
plt.xlabel('Componentes Temporales (EOF_X)')
plt.ylabel('Componentes Temporales (EOF_X)')
plt.tight_layout() # Adjust layout to prevent labels from being cut off
```

```
# Save Heatmap of Correlation
heatmap_correlation_path = os.path.join(folder_temporal_components, f"{nombre_archivo_pct}_Correlation_Heatmap.png")
plt.savefig(heatmap_correlation_path)
plt.close() # Close the plot to free up memory
print(f"✅ Heatmap de correlación guardado en: {heatmap_correlation_path}")
```

```
#Bloque 6: Aplicación del algoritmo Sparse Principal Component Thermopgraphy/Analysis (SPCT)

from sklearn.decomposition import SparsePCA

print("\n--- CONFIGURACIÓN SCPT ---")
try:
    n_components_scpt = int(input("1. Introduce el número de componentes dispersos (Sparse Components) a extraer para SCPT (ej. 5): "))
except ValueError:
    print("❌ Error: Debes introducir un número entero válido.")
    n_components_scpt = None

if n_components_scpt is not None:
    print(f"✅ Se intentarán extraer {n_components_scpt} componentes dispersos.")

    # 3. Instanciar SparsePCA
    # random_state es para reproducibilidad
    spca = SparsePCA(n_components=n_components_scpt, random_state=42)
    print("✅ Objeto SparsePCA instanciado.")
```

```
# 4. Aplicar fit_transform para obtener las componentes temporales escasas
# Los datos deben tener filas = muestras, columnas = características.
# matrizT_to_PCT_standard_procedure tiene píxeles como filas y tiempo como columnas.
# Para SparsePCA, queremos que cada pixel sea una 'muestra' y cada instante de tiempo una 'característica'.
# O si consideramos cada instante de tiempo una 'muestra' y cada pixel una 'característica'
# Para los componentes temporales, `spca.fit_transform` en la transpuesta de los datos estandarizados
# Si `matrizT_to_PCT_standard_procedure` es (num_pixels, num_time_instants),
# y queremos componentes temporales, necesitamos transponer para que SparsePCA vea el tiempo como 'muestras'.
# Sin embargo, la estructura original del notebook para SVD (`u, s, vh`) sugiere que `matrizT_to_PCT` (pixels as rows, time as columns)
# es la entrada principal y `u` son las eigenimágenes (espacial) y `vh` los componentes temporales.
# Si queremos que `spca.components_` sean los componentes espaciales (como las eigenimágenes),
# entonces la entrada a `fit_transform` debe tener los píxeles como características y el tiempo como muestras.
# Es decir, (num_time_instants, num_pixels).

# En el contexto de PCA/SVD, usualmente se tiene (observaciones, características).
# Si queremos componentes espaciales (eigenimages) como `components_`, entonces las 'características' deben ser los píxeles.
# Y las 'observaciones' deben ser los instantes de tiempo.
# Por lo tanto, necesitamos transponer la matriz antes de aplicarla a SparsePCA para obtener los componentes espaciales en `components_`
# y los componentes temporales en la salida de `fit_transform`.
```

```

# Transponemos para que las columnas sean los píxeles (características) y las filas los instantes de tiempo (observaciones)
matrizT_for_spca = matrizT_to_PCT_standard_procedure.T
print(f"📄 Matriz de entrada para SparsePCA transpuesta a forma: {matrizT_for_spca.shape} (tiempo, píxeles).")

scpt_temporal_components = spca.fit_transform(matrizT_for_spca)
print("✅ SparsePCA aplicado. Componentes temporales dispersos calculados.")
print(f" - Dimensiones de Componentes Temporales Dispersos (scpt_temporal_components): {scpt_temporal_components.shape}")

# 5. Acceder a los componentes espaciales dispersos (los 'loadings')
# Estos son los 'patrones' espaciales que SparsePCA ha identificado.
scpt_spatial_components = spca.components_
print("✅ Componentes espaciales dispersos (patrones) accedidos.")
print(f" - Dimensiones de Componentes Espaciales Dispersos (scpt_spatial_components): {scpt_spatial_components.shape}")

```

```

#Bloque 6_2: Almacenamiento de los resultados del algoritmo Sparse Principal Component Thermopgraphy/Analysis

print("\n--- EXTRACCIÓN Y REESCALADO DE COMPONENTES ESPACIALES DISPERSOS ---")

# 1. Acceder a las componentes espaciales dispersos
# scpt_spatial_components ya está disponible desde el BLOQUE 7

# 2. Inicializar una lista vacía para almacenar las componentes espaciales reescaladas
scpt_eigenimages_resaped = []

# 3. Iterar sobre cada componente espacial dispersa en scpt_spatial_components
# 4. Para cada componente, reescalarla a las dimensiones de imagen originales
# 5. Almacenar cada componente reescalada en la lista

```

```

if 'scpt_spatial_components' in locals() and rows_img_pct is not None and cols_img_pct is not None:
    print(f"📄 Reescalando {scpt_spatial_components.shape[0]} componentes espaciales a {rows_img_pct}x{cols_img_pct}.")
    for i, component_flat in enumerate(scpt_spatial_components):
        try:
            component_2d = component_flat.reshape(rows_img_pct, cols_img_pct)
            scpt_eigenimages_resaped.append(component_2d)
            print(f" - Componente espacial {i} reescalada exitosamente.")
        except ValueError as e:
            print(f"❌ Error al reescalar la componente espacial {i}: {e}. Asegúrate de que las dimensiones de la imagen sean correctas y que el tamaño de la componente plana ({len(component_flat)}) sea un múltiplo de {rows_img_pct}x{cols_img_pct} ({rows_img_pct*cols_img_pct}).")

    print(f"✅ {len(scpt_eigenimages_resaped)} Componentes espaciales dispersos reescalados y almacenados.")
else:
    print("❌ Las variables 'scpt_spatial_components', 'rows_img_pct' o 'cols_img_pct' no están definidas o son inválidas. Asegúrate de ejecutar los bloques anteriores.")

```

```

#Bloque 6_3: Guardado los resultados del algoritmo Sparse Principal Component Thermopgraphy/Analysis. Los componentes espaciales dispersos

import matplotlib.pyplot as plt

print("\n--- GUARDANDO COMPONENTES ESPACIALES DISPERSOS ---")

# 1. Obtener el directorio base y el nombre del archivo
# (Variables 'directorio_base_pct' y 'nombre_archivo_pct' ya definidas en BLOQUE 5)

# 2. Crea una carpeta para los resultados de SCPT
folder_scpt_results = os.path.join(directorio_base_pct, f"{nombre_archivo_pct}_SCPT_RESULTADOS")
os.makedirs(folder_scpt_results, exist_ok=True)
print(f"📁 Carpeta principal de resultados SCPT creada: {folder_scpt_results}")

# 3. Dentro de esta, crea una subcarpeta para los componentes dispersos
folder_sparse_components = os.path.join(folder_scpt_results, "Sparse-Components")
os.makedirs(folder_sparse_components, exist_ok=True)
print(f"📁 Subcarpeta para Componentes Dispersos creada: {folder_sparse_components}")

# Lista para almacenar los datos planos de los componentes dispersos para CSV
all_sparse_components_flat_data = []
sparse_component_headers = []

```

```
if scpt_eigenimages_resaped:
    print("\n🖼 Guardando imágenes de Componentes Espaciales Dispersos...")
    for i, component_2d in enumerate(scpt_eigenimages_resaped):
        # Create a new figure and axes
        fig, ax = plt.subplots(figsize=(6, 5))

        # Display the component_2d data directly (non-normalized)
        img = ax.imshow(component_2d, cmap='viridis', origin='upper')

        # Add a colorbar
        cbar = fig.colorbar(img, ax=ax, orientation='vertical', shrink=0.8)
        cbar.set_label('Valor de Componente Dispersa')

        # Set a dynamic title
        ax.set_title(f'Sparse Component SC_{i:03d}')

        # Hide axis ticks and labels for a cleaner image
        ax.set_xticks([])
        ax.set_yticks([])
        ax.set_xticklabels([])
        ax.set_yticklabels([])

        # Save the figure to a PNG file
        filename_png = os.path.join(folder_sparse_components, f"SC_{i:03d}.png")
        plt.savefig(filename_png, bbox_inches='tight', pad_inches=0.1)
        print(f" - Guardado: {os.path.basename(filename_png)}")
```

```
# Close the plot to free up memory
plt.close(fig)

# 5. Recolecta los datos planos (flattened) de cada componente espacial disperso
# scpt_spatial_components ya contiene estos datos en formato plano
all_sparse_components_flat_data.append(scpt_spatial_components[i])
sparse_component_headers.append(f"SC_{i}")

# 6. Crea un DataFrame de pandas con estos datos planos
if all_sparse_components_flat_data:
    df_sparse_components_raw = pd.DataFrame(np.column_stack(all_sparse_components_flat_data), columns=sparse_component_headers)
    # 7. Guarda este DataFrame en un archivo CSV
    csv_output_path_scpt = os.path.join(folder_scpt_results, f"{nombre_archivo_pct}_Sparse_Components_RawData.csv")
    df_sparse_components_raw.to_csv(csv_output_path_scpt, index=False)
    print(f"✅ Datos crudos de Componentes Espaciales Dispersos guardados en: {csv_output_path_scpt}")
else:
    print("⚠ No hay datos de Componentes Espaciales Dispersos válidos para guardar en CSV.")
else:
    print("❌ No se encontraron componentes espaciales dispersos reescalados para guardar.")
```

```

#Bloque 6_4: Guardado los resultados del algoritmo Sparse Principal Component Thermopgrahy/Analysis. La varianza

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os

print("\n--- ANÁLISIS DE VARIANZA EXPLICADA POR SCPT ---")

if 'spca' in locals() and spca is not None:
    # SparsePCA does not directly have the 'explained_variance_ratio_' attribute
    # like standard PCA. We will calculate a proxy based on the variance of the
    # transformed (temporal) components.
    if 'scpt_temporal_components' in locals() and scpt_temporal_components is not None:
        # Calculate the variance of each sparse component's scores
        variance_of_sparse_component_scores = np.var(scpt_temporal_components, axis=0)

        # Calculate the ratio of these variances to their sum
        # This gives a sense of how much variance each component accounts for in the transformed space
        total_variance_of_scores = np.sum(variance_of_sparse_component_scores)
        if total_variance_of_scores > 0:
            explained_variance_ratio = variance_of_sparse_component_scores / total_variance_of_scores
        else:
            explained_variance_ratio = np.zeros_like(variance_of_sparse_component_scores)
        print("⚠ Advertencia: La suma total de la varianza de los scores es cero, la varianza explicada será cero.")

    print("✅ Varianza explicada por cada componente disperso obtenido.")

    # 2. Crea una nueva subcarpeta llamada 'Explained Variance' dentro de folder_scpt_results
    folder_explained_variance = os.path.join(folder_scpt_results, "Explained_Variance")
    os.makedirs(folder_explained_variance, exist_ok=True)
    print(f"📁 Carpeta para Varianza Explicada creada: {folder_explained_variance}")

    # 3. Crea un DataFrame de pandas con la varianza explicada, convirtiendo los valores a porcentaje.
    # Asigna el nombre de columna 'Porcentaje de Varianza Explicada'.
    df_explained_variance = pd.DataFrame(explained_variance_ratio * 100, columns=['Porcentaje de Varianza Explicada'])
    print("✅ DataFrame de varianza explicada creado.")

    # 4. Guarda este DataFrame en un archivo CSV
    csv_explained_variance_path = os.path.join(folder_explained_variance, f"{nombre_archivo_pct}_Explained_Variance_Ratio.csv")
    df_explained_variance.to_csv(csv_explained_variance_path, index=False)
    print(f"✅ Varianza explicada guardada en: {csv_explained_variance_path}")

    # 5. Genera un gráfico de barras
    plt.figure(figsize=(10, 6))
    plt.bar(range(len(explained_variance_ratio)), explained_variance_ratio * 100, color='lightcoral')

    # 6. Añade un título descriptivo y etiquetas a los ejes
    plt.title('Varianza Explicada por Cada Componente Disperso (SCPT)')
    plt.xlabel('Componente Disperso (Índice SC)')
    plt.ylabel('Porcentaje de Varianza Explicada (%)')
    plt.xticks(range(len(explained_variance_ratio))) # Asegura que todos los índices se muestren en el eje x
    plt.grid(axis='y', linestyle='--', alpha=0.7)

    # 7. Guarda el gráfico como una imagen PNG
    plot_explained_variance_path = os.path.join(folder_explained_variance, f"{nombre_archivo_pct}_Explained_Variance_Ratio_Plot.png")
    plt.savefig(plot_explained_variance_path)
    plt.close() # Cierra el plot para liberar memoria
    print(f"✅ Gráfico de varianza explicada guardado en: {plot_explained_variance_path}")

    print("\n✅ Proceso de análisis de varianza explicada de SCPT completado exitosamente.")
else:
    print("❌ 'scpt_temporal_components' no está definido. Asegúrate de ejecutar los bloques anteriores de SCPT.")
else:
    print("❌ El objeto 'spca' no está definido. Asegúrate de ejecutar los bloques anteriores de SCPT.")

```

```
#Bloque 6_5: Guardado los resultados del algoritmo Sparse Principal Component Thermopgraphy/Analysis. Coeficientes temporales dispersos (valores)

import pandas as pd
import os

print("\n--- EXTRACCIÓN Y GUARDADO DE COEFICIENTES TEMPORALES DISPERSOS ---")

if 'scpt_temporal_components' in locals() and scpt_temporal_components is not None and \
    'time_in_minutes' in locals() and time_in_minutes is not None:

    # 1. Crea una nueva subcarpeta llamada 'Temporal_Coefficients' dentro de folder_scpt_results
    folder_temporal_coefficients = os.path.join(folder_scpt_results, "Temporal_Coefficients")
    os.makedirs(folder_temporal_coefficients, exist_ok=True)
    print(f"📁 Carpeta para Coeficientes Temporales creada: {folder_temporal_coefficients}")

    # 2. Convierte el array NumPy scpt_temporal_components a un DataFrame de pandas
    # Asegúrate de que las columnas estén etiquetadas de forma descriptiva, por ejemplo, 'SC_0', 'SC_1', etc.
    scpt_temporal_components_headers = [f"SC_{i}" for i in range(scpt_temporal_components.shape[1])]
    df_scpt_temporal_components = pd.DataFrame(scpt_temporal_components, columns=scpt_temporal_components_headers)
    print("✅ DataFrame de coeficientes temporales dispersos creado.")

    # 3. Inserta la lista time_in_minutes como la primera columna de este DataFrame.
    df_scpt_temporal_components.insert(0, 'Tiempo_min', time_in_minutes)
    print("✅ Columna 'Tiempo_min' añadida al DataFrame.")

    # 4. Guarda el DataFrame resultante en un archivo CSV
    csv_temporal_coefficients_path = os.path.join(folder_temporal_coefficients, f"{nombre_archivo_pct}_Temporal_Coefficients_RawData.csv")
    df_scpt_temporal_components.to_csv(csv_temporal_coefficients_path, index=False)
    print(f"✅ Coeficientes temporales dispersos guardados en: {csv_temporal_coefficients_path}")
    print("\n✅ Proceso de extracción y guardado de coeficientes temporales dispersos completado exitosamente.")

else:
    print("❌ Las variables 'scpt_temporal_components' o 'time_in_minutes' no están definidas o son inválidas. Asegúrate de ejecutar los bloques anteriores de SCPT.)")

#Bloque 6_6: Guardado los resultados del algoritmo Sparse Principal Component Thermopgraphy/Analysis. Coeficientes temporales dispersos (gráfica)

import matplotlib.pyplot as plt
import os

print("\n--- VISUALIZANDO COEFICIENTES TEMPORALES DISPERSOS ---")

if 'df_scpt_temporal_components' in locals() and df_scpt_temporal_components is not None:
    plt.figure(figsize=(14, 7))

    # Iterate over columns that represent sparse components (excluding 'Tiempo_min')
    for col in df_scpt_temporal_components.columns:
        if col != 'Tiempo_min':
            plt.plot(df_scpt_temporal_components['Tiempo_min'], df_scpt_temporal_components[col], label=f'Componente Temporal {col}')

    # Add descriptive title and labels
    plt.title('Evolución Temporal de los Coeficientes Dispersos (SCPT)')
    plt.xlabel('Tiempo (minutos)')
    plt.ylabel('Amplitud del Coeficiente')
    plt.legend()
    plt.grid(True)

    # Construct the output path for the plot
    plot_temporal_coefficients_path = os.path.join(folder_temporal_coefficients, f"{nombre_archivo_pct}_Temporal_Coefficients_Plot.png")

    # Save the plot as a PNG image
    plt.savefig(plot_temporal_coefficients_path)
    plt.close() # Close the plot to free up memory
    print(f"✅ Gráfico de Componentes Temporales Dispersos guardado en: {plot_temporal_coefficients_path}")

    print("\n✅ Visualización de coeficientes temporales dispersos completada exitosamente.")
else:
    print("❌ El DataFrame 'df_scpt_temporal_components' no está definido. Asegúrate de ejecutar los bloques anteriores.")
```

```

# BLOQUE 6_7: Mapa de calor de correlación entre las distintas Componentes Temporales Dispersas (SCPT)
import seaborn as sns
import os
import matplotlib.pyplot as plt

print("\n--- ANÁLISIS DE CORRELACIÓN DE COMPONENTES TEMPORALES DISPERSAS (SCPT) ---")

if 'df_scpt_temporal_components' in locals() and df_scpt_temporal_components is not None:
    # 1. Seleccionar solo las columnas que corresponden a las Componentes Temporales Dispersas (SC_X)
    temporal_scpt_columns = [col for col in df_scpt_temporal_components.columns if col.startswith('SC_')]
    correlation_matrix_scpt = df_scpt_temporal_components[temporal_scpt_columns].corr(method='pearson')

    print("✅ Matriz de correlación de Pearson calculada para las Componentes Temporales Dispersas.")
    print("Primeras 5x5 de la matriz de correlación:")
    print(correlation_matrix_scpt.head().iloc[:, :5]) # Mostrar 5x5 para brevedad

    # Asegúrate de que la carpeta de componentes temporales exista (creada en 6_5)
    # folder_temporal_coefficients ya debe estar definido.
    os.makedirs(folder_temporal_coefficients, exist_ok=True)

    # 2. Guardar Matriz de Correlación en CSV
    csv_correlation_scpt_path = os.path.join(folder_temporal_coefficients, f"{nombre_archivo_pct}_SCPT_Correlation_Matrix.csv")
    correlation_matrix_scpt.to_csv(csv_correlation_scpt_path, index=True) # index=True para guardar los nombres de las SCs como índice
    print(f"✅ Matriz de correlación guardada en: {csv_correlation_scpt_path}")

    # 3. Visualizar Correlación (Heatmap)
    plt.figure(figsize=(10, 8))
    sns.heatmap(correlation_matrix_scpt, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
    plt.title('Matriz de Correlación de las Componentes Temporales Dispersas (SCPT)')
    plt.xlabel('Componentes Temporales Dispersas (SC_X)')
    plt.ylabel('Componentes Temporales Dispersas (SC_X)')
    plt.tight_layout() # Ajustar el diseño para evitar recortes

    # 4. Guardar Heatmap de Correlación
    heatmap_correlation_scpt_path = os.path.join(folder_temporal_coefficients, f"{nombre_archivo_pct}_SCPT_Correlation_Heatmap.png")
    plt.savefig(heatmap_correlation_scpt_path)
    plt.close() # Cerrar el plot para liberar memoria
    print(f"✅ Heatmap de correlación guardado en: {heatmap_correlation_scpt_path}")

    print("\n--- Análisis de Correlación de Componentes Temporales Dispersas (SCPT) Completado ---")
    print("Se ha calculado la matriz de correlación de Pearson entre las Componentes Temporales Dispersas.")
    print("La matriz de correlación ha sido guardada en:", csv_correlation_scpt_path)
    print("La visualización del heatmap de correlación ha sido guardada en:", heatmap_correlation_scpt_path)
else:
    print("❌ El DataFrame 'df_scpt_temporal_components' no está definido. Asegúrate de ejecutar los bloques anteriores de SCPT.")

```

3. Reconstrucción de la señal termográfica (TSR – Bloque 7)

Este bloque implementa la técnica TSR (*Thermographic Signal Reconstruction*), basada en el ajuste polinómico de la evolución térmica en el dominio logarítmico. Su objetivo principal es reducir el ruido y resaltar variaciones térmicas asociadas a defectos subsuperficiales.

```
# BLOQUE 7: Aplicación del algoritmo Thermography Signal Reconstruction (TSR)
# (ajuste polinómico a la evolución temporal de cada pixel)

import numpy as np
import pandas as pd
import os

print("--- INICIANDO PROCESO DE TSR AUTOMATIZADO ---")

# 1. Convertir la lista `time_in_minutes` a un array NumPy
time_in_minutes_array = np.array(time_in_minutes)
print(f"✅ `time_in_minutes_array` creado. Forma: {time_in_minutes_array.shape}")

# 2. Crear `time_seconds` y manejar ceros para logaritmo
time_seconds = time_in_minutes_array * 60
epsilon_time = 1e-6 # Pequeño valor para evitar log(0)
time_seconds[time_seconds == 0] = epsilon_time
print(f"✅ `time_seconds` creado y ceros reemplazados. Forma: {time_seconds.shape}")

# 3. Definir max_poly_degree
max_poly_degree = 5
print(f"✅ Grado polinomial máximo definido: {max_poly_degree}")

# 4. Crear la carpeta para resultados TSR
folder_tsr_results = os.path.join(directorio_base_pct, f"{nombre_archivo_pct}_TSR_RESULTADOS")
os.makedirs(folder_tsr_results, exist_ok=True)
print(f"📁 Carpeta para resultados TSR creada: {folder_tsr_results}")

# 5. Inicializar listas para almacenar resultados
optimal_coefficients_list = []
optimal_r_squared_values = []
print("✅ Listas para almacenar coeficientes y R^2 inicializadas.")

# Epsilon para evitar log(0) en temperaturas
epsilon_temp = 1e-6

# 6. Iterar a través de cada pixel (fila) del DataFrame `combined_df`
print("⌚ Iniciando iteración pixel por pixel para ajuste TSR...")
for pixel_id in range(combined_df.shape[0]):
    pixel_series = combined_df.iloc[pixel_id]
    pixel_temp_values = pixel_series.to_numpy(dtype=float)

    # 7b. Identificar y contar NaNs iniciales
    num_leading_nans = 0
    for val in pixel_temp_values:
        if np.isnan(val):
            num_leading_nans += 1
        else:
            break
```

```

# 7c. Slicing para excluir NaNs iniciales
current_time_seconds_raw = time_seconds[num_leading_nans:]
current_pixel_temp_values_raw = pixel_temp_values[num_leading_nans:]

# Filtrar NaNs que no son iniciales (pueden ocurrir en medio o al final)
valid_temp_indices = ~np.isnan(current_pixel_temp_values_raw)
current_time_seconds_filtered = current_time_seconds_raw[valid_temp_indices]
current_pixel_temp_values_filtered = current_pixel_temp_values_raw[valid_temp_indices]

# NUEVO PASO: Filtrar para asegurar que log_time no sea negativo (tiempo >= 1 segundo)
non_negative_log_time_indices = current_time_seconds_filtered >= 1.0
current_time_seconds = current_time_seconds_filtered[non_negative_log_time_indices]
current_pixel_temp_values = current_pixel_temp_values_filtered[non_negative_log_time_indices]

# 7d. Saltar píxeles con datos insuficientes
if len(current_pixel_temp_values) < (max_poly_degree + 1):
    # Si no hay suficientes puntos válidos, rellenar con NaN y 0 R^2
    optimal_coefficients_list.append(np.full(max_poly_degree + 1, np.nan))
    optimal_r_squared_values.append(np.nan)
    continue

# 7e. Aplicar transformación logarítmica al tiempo
log_time = np.log(current_time_seconds)

```

```

# 7f. Aplicar transformación logarítmica a la temperatura, manejando ceros
# Reemplazar valores <= 0 con epsilon antes de log
processed_pixel_temp_values = np.where(current_pixel_temp_values <= 0, epsilon_temp, current_pixel_temp_values)
log_temperature = np.log(processed_pixel_temp_values)

# 7g. Inicializar variables para optimización
best_r_squared = -1.0
best_coefficients = None
pixel_optimal_degree = None

# 7h. Iterar a través de grados polinomiales
for degree in range(1, max_poly_degree + 1):
    # 7h.i. Ajustar el polinomio
    coefficients = np.polyfit(log_time, log_temperature, degree)

    # 7h.ii. Calcular log_temperature predicho
    predicted_log_temperature = np.polyval(coefficients, log_time)

```

```
# 7h.iii. Calcular R-squared
ss_total = np.sum((log_temperature - np.mean(log_temperature)) ** 2)
ss_residual = np.sum((log_temperature - predicted_log_temperature) ** 2)

if ss_total == 0:
    r_squared = 0.0 # Evitar división por cero si todos los log_temperature son iguales
else:
    r_squared = 1 - (ss_residual / ss_total)

# 7h.iv. Actualizar el mejor ajuste
if r_squared > best_r_squared:
    best_r_squared = r_squared
    best_coefficients = coefficients
    pixel_optimal_degree = degree

# 7i. Rellenar coeficientes con ceros para tener una longitud fija
target_coefs_len = max_poly_degree + 1
if best_coefficients is not None:
    padded_coefficients = np.pad(best_coefficients, (target_coefs_len - len(best_coefficients), 0), 'constant')
else:
    # Si no se pudo ajustar ningún polinomio (e.g., por datos insuficientes a pesar de la comprobación previa)
    padded_coefficients = np.full(target_coefs_len, np.nan)

# 7j. y 7k. Almacenar los resultados óptimos
optimal_coefficients_list.append(padded_coefficients)
optimal_r_squared_values.append(best_r_squared)

print("✅ Proceso de ajuste TSR completado para todos los píxeles.")
```

```
# BLOQUE 7_2: Guardar resultados de los valores de los coeficientes y del R2

print("--- GUARDANDO RESULTADOS DE TSR ---")

# 8. Convertir la lista de coeficientes a un DataFrame
# Las columnas serán 'coeff_0', 'coeff_1', ..., 'coeff_max_poly_degree'
coeff_columns = [f"coeff_{i}" for i in range(max_poly_degree + 1)]
df_optimal_coefficients = pd.DataFrame(optimal_coefficients_list, columns=coeff_columns)
print(f"✅ DataFrame de coeficientes óptimos creado. Forma: {df_optimal_coefficients.shape}")

# 9. Guardar el DataFrame de coeficientes en un CSV
csv_coeff_path = os.path.join(folder_tsr_results, "optimal_tsr_coefficients.csv")
df_optimal_coefficients.to_csv(csv_coeff_path, index=False)
print(f"✅ Coeficientes TSR óptimos guardados en: {csv_coeff_path}")

# 10. Convertir la lista de R-squared a un DataFrame
df_r_squared_values = pd.DataFrame(optimal_r_squared_values, columns=["R_squared"])
print(f"✅ DataFrame de valores R-squared creado. Forma: {df_r_squared_values.shape}")

# 11. Guardar el DataFrame de R-squared en un CSV
csv_r2_path = os.path.join(folder_tsr_results, "tsr_r_squared_values.csv")
df_r_squared_values.to_csv(csv_r2_path, index=False)
print(f"✅ Valores R-squared de TSR guardados en: {csv_r2_path}")

print("✅ Resultados de TSR guardados exitosamente.")
```

```

# BLOQUE 7_3: Guardar resultados mapeo del R2

import matplotlib.pyplot as plt
import seaborn as sns
import os
import numpy as np

print("--- GENERANDO MAPA DE CALOR PARA VALORES R² DE TSR ---")

# 1. Create a subfolder named 'R_squared_Visualization'
folder_r2_vis = os.path.join(folder_tsr_results, "R_squared_Visualization")
os.makedirs(folder_r2_vis, exist_ok=True)
print(f"📁 Carpeta para visualizaciones de R² creada: {folder_r2_vis}")

# Ensure rows_img_pct and cols_img_pct are defined
if 'rows_img_pct' not in locals() or 'cols_img_pct' not in locals() or rows_img_pct is None or cols_img_pct is None:
    print("❌ Error: 'rows_img_pct' o 'cols_img_pct' no están definidos. Asegúrate de ejecutar los bloques anteriores.")
else:
    # 2. Extract the 'R_squared' column from the df_r_squared_values DataFrame.
    if 'df_r_squared_values' in locals() and df_r_squared_values is not None:
        r_squared_values = df_r_squared_values['R_squared'].to_numpy()
        print(f"✅ Valores R² extraídos. Forma: {r_squared_values.shape}")

# 3. Reshape the 1D array of R² values into a 2D array
try:
    if len(r_squared_values) != (rows_img_pct * cols_img_pct):
        print(f"❌ Error: El tamaño de los valores R² ({len(r_squared_values)}) no coincide con las dimensiones de la imagen ({rows_img_pct}x{cols_img_pct}). Saltando.")
    else:
        r_squared_map = r_squared_values.reshape(rows_img_pct, cols_img_pct)
        print(f"✅ Valores R² reestructurados a {r_squared_map.shape} (alto x ancho).")

        # 4. Create a heatmap of the reshaped R² values
        plt.figure(figsize=(cols_img_pct / 50, rows_img_pct / 50)) # Adjust figure size dynamically
        sns.heatmap(r_squared_map, cmap='viridis', cbar=True, square=True)

        # Set the title of the heatmap
        plt.title('Heatmap de Valores R² del Ajuste TSR Óptimo')
        plt.axis('off') # Hide axes for cleaner image representation

        # 5. Save the heatmap as a PNG file
        filename_png = os.path.join(folder_r2_vis, f"{nombre_archivo_pct}_R_squared_Heatmap.png")
        plt.savefig(filename_png, bbox_inches='tight', pad_inches=0.1)
        print(f"📁 Guardado: {os.path.basename(filename_png)}")

        # 6. Close the plot to free up memory
        plt.close()
        print("✅ Generación del mapa de calor para valores R² completada.")

except Exception as e:
    print(f"❌ Error al generar o guardar el heatmap para los valores R²: {e}")
else:
    print("⚠️ No se encontró 'df_r_squared_values' para generar el heatmap.")

```

```
# BLOQUE 7_4: Cálculo de la primera, segunda y tercera derivada

import numpy as np
import pandas as pd
import os

print("--- CALCULANDO DERIVADAS TSR ---")

# 1. Reconstruct `log_time` for the full duration
# `time_seconds` was already adjusted for epsilon_time in a previous block.
# NUEVO PASO: Filtrar time_seconds para asegurar que log_time no sea negativo (tiempo >= 1 segundo)
filtered_time_seconds_indices = time_seconds >= 1.0
time_seconds_for_log = time_seconds[filtered_time_seconds_indices]
log_time = np.log(time_seconds_for_log)
print(f"✅ `log_time` reconstruido y filtrado. Forma: {log_time.shape}")

# 2. Inicializar listas para almacenar las derivadas
first_derivatives_list = []
second_derivatives_list = []
third_derivatives_list = []
print(f"✅ Listas para almacenar derivadas inicializadas.")

# 3. Iterar a través de cada píxel (fila) en `df_optimal_coefficients`
print("⌚ Calculando derivadas para cada píxel...")
for pixel_id in range(df_optimal_coefficients.shape[0]):
    optimal_coeffs = df_optimal_coefficients.iloc[pixel_id].dropna().to_numpy()

    # Check if coefficients are valid (not all NaNs)
    if len(optimal_coeffs) == 0 or np.all(np.isnan(optimal_coeffs)):
        # If no valid coefficients, append NaNs for derivatives
        first_derivatives_list.append(np.full_like(log_time, np.nan))
        second_derivatives_list.append(np.full_like(log_time, np.nan))
        third_derivatives_list.append(np.full_like(log_time, np.nan))
        continue

    # 4. Calcular la primera derivada
    d1_coeffs = np.polyder(optimal_coeffs, m=1)
    d1_values = np.polyval(d1_coeffs, log_time)
    first_derivatives_list.append(d1_values)

    # 5. Calcular la segunda derivada
    d2_coeffs = np.polyder(optimal_coeffs, m=2)
    d2_values = np.polyval(d2_coeffs, log_time)
    second_derivatives_list.append(d2_values)

    # 6. Calcular la tercera derivada
    d3_coeffs = np.polyder(optimal_coeffs, m=3)
    d3_values = np.polyval(d3_coeffs, log_time)
    third_derivatives_list.append(d3_values)
```

```
# Convertir las listas de derivadas en DataFrames
df_first_derivatives = pd.DataFrame(first_derivatives_list)
df_second_derivatives = pd.DataFrame(second_derivatives_list)
df_third_derivatives = pd.DataFrame(third_derivatives_list)

print("✅ Cálculo de derivadas completado para todos los píxeles.")
```

```
# BLOQUE 7.5: Cálculo de parámetros estadísticos a la evolución temporal de cada pixel (diferencia en términos absolutos entre max y min, max, min, media,
desv típica, simetría (skew) e curtosis (kurtosis))
# según primera, segunda y tercera derivada

import numpy as np
import pandas as pd
import os
import scipy.stats as stats

print("--- CALCULANDO ESTADÍSTICAS DE LAS DERIVADAS TSR ---")

# Función auxiliar para calcular estadísticas
def calculate_derivative_stats(df_deriv, deriv_order):
    stats_list = []
    for pixel_id in range(df_deriv.shape[0]):
        deriv_values = df_deriv.iloc[pixel_id].to_numpy()

        # Handle cases where all derivative values are NaN
        if np.all(np.isnan(deriv_values)):
            stats_list.append({"abs_diff": np.nan, "mean": np.nan, "max": np.nan, "min": np.nan, "std": np.nan, "skew": np.nan, "kurtosis": np.nan})
            continue

        # Calculate stats, handling NaNs where appropriate
        mean_val = np.nanmean(deriv_values)
        max_val = np.nanmax(deriv_values)
        min_val = np.nanmin(deriv_values)
        std_dev_val = np.nanstd(deriv_values)

        # Absolute difference between max and min
        abs_diff_val = max_val - min_val

        # Skewness and Kurtosis need at least 4 non-NaN values for unbiased estimation, or 3 for biased.
        # Let's use 'stats.skew' and 'stats.kurtosis' which handle NaNs.
        skew_val = stats.skew(deriv_values[~np.isnan(deriv_values)]) if len(deriv_values[~np.isnan(deriv_values)]) >= 3 else np.nan
        kurt_val = stats.kurtosis(deriv_values[~np.isnan(deriv_values)]) if len(deriv_values[~np.isnan(deriv_values)]) >= 4 else np.nan

        stats_list.append({"abs_diff": abs_diff_val, "mean": mean_val, "max": max_val, "min": min_val, "std": std_dev_val, "skew": skew_val, "kurtosis": kurt_val})
    return pd.DataFrame(stats_list)

# Crear la carpeta para las estadísticas dentro de TSR_RESULTADOS
folder_stats_results = os.path.join(folder_tsr_results, "TSR_Statistics")
os.makedirs(folder_stats_results, exist_ok=True)
print(f"📁 Carpeta para estadísticas TSR creada: {folder_stats_results}")
```

```
# 1. Calcular estadísticas para la primera derivada
if 'df_first_derivatives' in locals() and df_first_derivatives is not None:
    print("⌚ Calculando estadísticas para la primera derivada...")
    df_stats_d1 = calculate_derivative_stats(df_first_derivatives, 1)
    csv_stats_d1_path = os.path.join(folder_stats_results, "tsr_stats_derivative1.csv")
    df_stats_d1.to_csv(csv_stats_d1_path, index=False)
    print(f"✅ Estadísticas de la primera derivada guardadas en: {csv_stats_d1_path}")
    print(f"Forma del DataFrame de estadísticas de d1: {df_stats_d1.shape}")
else:
    print("⚠️ No se encontró df_first_derivatives para calcular estadísticas.")

# 2. Calcular estadísticas para la segunda derivada
if 'df_second_derivatives' in locals() and df_second_derivatives is not None:
    print("⌚ Calculando estadísticas para la segunda derivada...")
    df_stats_d2 = calculate_derivative_stats(df_second_derivatives, 2)
    csv_stats_d2_path = os.path.join(folder_stats_results, "tsr_stats_derivative2.csv")
    df_stats_d2.to_csv(csv_stats_d2_path, index=False)
    print(f"✅ Estadísticas de la segunda derivada guardadas en: {csv_stats_d2_path}")
    print(f"Forma del DataFrame de estadísticas de d2: {df_stats_d2.shape}")
else:
    print("⚠️ No se encontró df_second_derivatives para calcular estadísticas.")
```

```
# 3. Calcular estadísticas para la tercera derivada
if 'df_third_derivatives' in locals() and df_third_derivatives is not None:
    print("⌚ Calculando estadísticas para la tercera derivada...")
    df_stats_d3 = calculate_derivative_stats(df_third_derivatives, 3)
    csv_stats_d3_path = os.path.join(folder_stats_results, "tsr_stats_derivative3.csv")
    df_stats_d3.to_csv(csv_stats_d3_path, index=False)
    print(f"✅ Estadísticas de la tercera derivada guardadas en: {csv_stats_d3_path}")
    print(f"Forma del DataFrame de estadísticas de d3: {df_stats_d3.shape}")
else:
    print("⚠️ No se encontró df_third_derivatives para calcular estadísticas.")

print("✅ Cálculo de estadísticas de las derivadas TSR completado.")
```

```
# BLOQUE 7_6: Guardar los resultados estadísticos

import matplotlib.pyplot as plt
import seaborn as sns
import os
import numpy as np

print("--- GENERANDO MAPAS DE CALOR PARA ESTADÍSTICAS DE DERIVADAS TSR ---")

# 1. Define a list of statistical features to visualize
statistical_features = ['abs_diff', 'mean', 'max', 'min', 'std', 'skew', 'kurtosis']

# Dictionary to map derivative order to its DataFrame
derivative_dataframes = {
    1: df_stats_d1,
    2: df_stats_d2,
    3: df_stats_d3
}
```

```

# Ensure rows_img_pct and cols_img_pct are defined
if 'rows_img_pct' not in locals() or 'cols_img_pct' not in locals() or rows_img_pct is None or cols_img_pct is None:
    print("❌ Error: 'rows_img_pct' o 'cols_img_pct' no están definidos. Asegúrate de ejecutar los bloques anteriores.")
else:
    # 2. For each derivative order (1, 2, 3):
    for deriv_order, df_stats in derivative_dataframes.items():
        if df_stats is None:
            print(f"⚠️ No se encontraron datos para la derivada {deriv_order}. Saltando visualizaciones.")
            continue

        # b. Create a new subfolder for the derivative order's visualizations
        folder_deriv_vis = os.path.join(folder_stats_results, f"Derivative{deriv_order}_Visualizations")
        os.makedirs(folder_deriv_vis, exist_ok=True)
        print(f"📁 Carpeta para visualizaciones de Derivada {deriv_order} creada: {folder_deriv_vis}")

        # c. For each statistical feature in the predefined list:
        for feature in statistical_features:
            if feature not in df_stats.columns:
                print(f"⚠️ La característica '{feature}' no se encontró en las estadísticas de la derivada {deriv_order}. Saltando.")
                continue

            # i. Extract the column corresponding to the current feature
            feature_values = df_stats[feature].to_numpy()

            # ii. Reshape this 1D array into a 2D array
            try:
                # Handle cases where feature_values might not perfectly reshape
                if len(feature_values) != (rows_img_pct * cols_img_pct):
                    print(f"❌ Error: Tamaño de los datos de la característica '{feature}' para la derivada {deriv_order} ({len(feature_values)}) no coincide con las dimensiones de la imagen ({rows_img_pct}x{cols_img_pct}). Saltando.")
                    continue

                feature_map = feature_values.reshape(rows_img_pct, cols_img_pct)

                # iii. Create a heatmap
                plt.figure(figsize=(cols_img_pct / 50, rows_img_pct / 50)) # Adjust figure size dynamically
                sns.heatmap(feature_map, cmap='viridis', cbar=True, square=True)

                # iv. Set the title of the heatmap
                plt.title(f'Heatmap de {feature.replace("_", " ").title()} de la {deriv_order}ª Derivada', pad=20)
                plt.axis('off') # Hide axes for cleaner image representation

                # v. Save the heatmap as a PNG file
                filename_png = os.path.join(folder_deriv_vis, f"Derivative{deriv_order}_{feature}_Heatmap.png")
                plt.savefig(filename_png, bbox_inches='tight', pad_inches=0.1)
                print(f" - Guardado: {os.path.basename(filename_png)}")

                # vi. Close the plot to free up memory
                plt.close()
            except Exception as e:
                print(f"❌ Error al generar o guardar el heatmap para la derivada {deriv_order}, característica '{feature}': {e}")

        print("✅ Generación de mapas de calor para estadísticas de derivadas completada.")

```

```
# BLOQUE 7_7: Guardar evolución del ajuste polinómico de un pixel cualquiera
```

```
import random
import matplotlib.pyplot as plt
import numpy as np
```

```
print("--- VISUALIZANDO EL AJUSTE TSR ÓPTIMO PARA UN PÍXEL ALEATORIO ---")
```

```
# 1. Generar un random pixel_id
```

```
pixel_id = random.randint(0, combined_df.shape[0] - 1)
```

```
print(f"✅ Píxel de muestra seleccionado aleatoriamente: ID {pixel_id}")
```

```
# 2. y 3. Recuperar valores de temperatura originales y coeficientes óptimos
```

```
pixel_series = combined_df.iloc[pixel_id]
```

```
pixel_temp_values_original = pixel_series.to_numpy(dtype=float)
```

```
optimal_coeffs = df_optimal_coefficients.iloc[pixel_id].dropna().to_numpy()
```

```
# Verificar si hay coeficientes válidos para el píxel seleccionado
```

```
if len(optimal_coeffs) == 0 or np.all(np.isnan(optimal_coeffs)):
```

```
    print(f"❌ No se encontraron coeficientes TSR válidos para el píxel {pixel_id}. Saltando visualización.")
```

```
else:
```

```
    # 4. Preparar `log_time` y `log_temperature` para el píxel seleccionado
```

```
    # `time_seconds` ya ha sido ajustado con `epsilon_time` en un bloque anterior.
```

```
    # Identificar y contar NaNs iniciales (del proceso original de TSR)
```

```
    num_leading_nans = 0
```

```
    for val in pixel_temp_values_original:
```

```
        if np.isnan(val):
```

```
            num_leading_nans += 1
```

```
        else:
```

```
            break
```

```
    current_time_seconds_raw = time_seconds[num_leading_nans:]
```

```
    current_pixel_temp_values_raw = pixel_temp_values_original[num_leading_nans:]
```

```
    valid_indices = ~np.isnan(current_pixel_temp_values_raw)
```

```
    current_time_seconds_filtered = current_time_seconds_raw[valid_indices]
```

```
    current_pixel_temp_values_filtered = current_pixel_temp_values_raw[valid_indices]
```

```
    # Filtrar para asegurar que log_time no sea negativo (tiempo >= 1 segundo)
```

```
    non_negative_log_time_indices = current_time_seconds_filtered >= 1.0
```

```
    current_time_seconds_for_plot = current_time_seconds_filtered[non_negative_log_time_indices]
```

```
    current_pixel_temp_values_for_plot = current_pixel_temp_values_filtered[non_negative_log_time_indices]
```

```

# Filtrar datos si no hay suficientes puntos
if len(current_pixel_temp_values_for_plot) < (max_poly_degree + 1):
    print(f"✗ Píxel {pixel_id} tiene datos insuficientes para visualización tras filtrado. Saltando.")
else:
    log_time_plot = np.log(current_time_seconds_for_plot)

    processed_pixel_temp_values = np.where(current_pixel_temp_values_for_plot <= 0, epsilon_temp, current_pixel_temp_values_for_plot)
    log_temperature_plot = np.log(processed_pixel_temp_values)

    # 5. Calcular los valores de temperatura logarítmicos predichos
    predicted_log_temperature = np.polyval(optimal_coeffs, log_time_plot)

    # 6. Crear un plot de Matplotlib
    plt.figure(figsize=(12, 7))
    plt.scatter(log_time_plot, log_temperature_plot, label='Datos Originales (logT vs log_t)', color='blue', s=10)
    plt.plot(log_time_plot, predicted_log_temperature, label='Ajuste TSR Óptimo', color='red', linestyle='--')

    # 7. Añadir título, etiquetas y leyenda
    plt.title(f'Ajuste TSR Óptimo para el Píxel ID: {pixel_id}')
    plt.xlabel('Logaritmo del Tiempo (s)')
    plt.ylabel('Logaritmo de la Temperatura')
    plt.legend()
    plt.grid(True)

    # 8. Guardar el plot como un archivo PNG
    plot_filename = f"optimal_tsr_fit_pixel_{pixel_id}.png"
    plot_path = os.path.join(folder_tsr_results, plot_filename)
    plt.savefig(plot_path)
    plt.close() # Cerrar el plot para liberar memoria
    print(f"✅ Plot del ajuste TSR óptimo para el píxel {pixel_id} guardado en: {plot_path}")

print("--- VISUALIZACIÓN DE PÍXEL DE MUESTRA COMPLETADA ---")

```

4. Generación de imagen binaria de referencia (Bloque 8)

Este bloque tiene como objetivo la generación de una imagen binaria de referencia que representa la geometría idealizada de la probeta con los defectos definidos. Esta imagen se utiliza como base para la validación de los algoritmos de detección y para la definición de regiones de interés (ROIs).

En primer lugar, se define un dominio geométrico principal correspondiente al área total de la probeta, junto con un conjunto de subregiones que representan los defectos de distintos tamaños y posiciones. Estas regiones se describen en un sistema de coordenadas continuo.

Posteriormente, se realiza una transformación de coordenadas desde el sistema geométrico (mundo real) al sistema discreto de píxeles de la imagen, teniendo en cuenta la resolución espacial definida por el usuario. Este proceso incluye la normalización de coordenadas y la corrección de la orientación del eje vertical.

A continuación, se genera una matriz binaria inicializada a cero, donde:

- El valor 1 representa la región sólida (material sano).
- El valor 0 representa las zonas correspondientes a defectos.

El algoritmo rellena primero la región global y posteriormente sustrae las regiones correspondientes a los defectos, generando así una representación binaria final de la geometría.

Finalmente, la imagen resultante se guarda en formato rasterizado, permitiendo su uso posterior en el cálculo de métricas cuantitativas como la SNR y en la validación visual de los resultados.

```
# BLOQUE 8: Obtener la imagen con los defectos de referencia. Definición de los defectos
```

```
large_square = {'x_min': 0, 'y_min': 0, 'width': 15, 'height': 15}
small_squares = [
    {'x_min': 2.5, 'y_min': 2.5, 'width': 0.5, 'height': 0.5},
    {'x_min': 6, 'y_min': 2.5, 'width': 2, 'height': 2},
    {'x_min': 11, 'y_min': 2.5, 'width': 1.5, 'height': 1.5},
    {'x_min': 2.5, 'y_min': 6, 'width': 2, 'height': 2},
    {'x_min': 7.5, 'y_min': 7.5, 'width': 1.5, 'height': 1.5},
    {'x_min': 12, 'y_min': 7, 'width': 0.5, 'height': 0.5},
    {'x_min': 2.5, 'y_min': 11, 'width': 1.5, 'height': 1.5},
    {'x_min': 7, 'y_min': 12, 'width': 0.5, 'height': 0.5},
    {'x_min': 10.5, 'y_min': 10.5, 'width': 2, 'height': 2}
]
```

```
# BLOQUE 8_2: Obtener la imagen con los defectos de referencia. Definición de la imagen
```

```
print("\n--- CONFIGURACIÓN DE LA IMAGEN RASTERIZADA FINAL ---")
try:
    # 1. Solicitar el número de filas
    final_rows = int(input("1. Introduce el número de FILAS para la imagen rasterizada final: "))
    # 2. Solicitar el número de columnas
    final_cols = int(input("2. Introduce el número de COLUMNAS para la imagen rasterizada final: "))

    # 3. Solicitar la ruta completa de guardado
    output_image_path = input("3. Pega la RUTA COMPLETA en Google Drive donde deseas guardar la imagen binaria resultante (ej. '/content/drive/MyDrive/ruta/imagen_final.png'): ").strip()
    output_image_path = output_image_path.replace("'", '').replace('"', '')

    print(f"\n✅ Dimensiones de la imagen final: {final_rows} filas x {final_cols} columnas.")
    print(f"✅ Ruta de guardado para la imagen binaria: {output_image_path}")

except ValueError:
    print("❌ Error: Debes introducir números enteros válidos para las dimensiones de la imagen.")
    final_rows = None
    final_cols = None
    output_image_path = None
```

```
# BLOQUE 8_3: Obtener la imagen con los defectos de referencia. Combinación defectos con imagen y guardado
```

```
import numpy as np
import matplotlib.pyplot as plt
import os

print("\n--- GENERANDO IMAGEN BINARIA DE GEOMETRÍA ---")

if final_rows is None or final_cols is None or output_image_path is None:
    print("❌ Error: Las dimensiones de la imagen o la ruta de guardado no están definidas. Por favor, revisa el bloque anterior.")
elif final_rows <= 0 or final_cols <= 0:
    print("❌ Error: Las dimensiones de la imagen deben ser positivas. Por favor, revisa el bloque anterior.")
else:
    # 1. Crear una matriz vacía para la imagen binaria (inicialmente todo ceros)
    # Usaremos np.float32 para asegurar compatibilidad con imsave para valores 0 y 1
    binary_image = np.zeros((final_rows, final_cols), dtype=np.float32)
    print(f"✅ Matriz binaria inicializada con dimensiones: {binary_image.shape}")

    # 2. Obtener las dimensiones del espacio de coordenadas de los cuadrados
    # Asumimos que los x_min, y_min, width, height están en un sistema de coordenadas que empieza en 0,0
    # y queremos mapearlo al espacio de píxeles.
```

```

# Encontrar el rango total de las coordenadas para el mapeo
# Suponemos que el large_square define el espacio de trabajo
x_min_world = large_square['x_min']
y_min_world = large_square['y_min']
x_max_world = large_square['x_min'] + large_square['width']
y_max_world = large_square['y_min'] + large_square['height']

world_width = x_max_world - x_min_world
world_height = y_max_world - y_min_world

if world_width == 0 or world_height == 0:
    print("❌ Error: El ancho o alto del cuadrado grande es cero, no se puede mapear la geometría.")
else:
    # Función auxiliar para mapear coordenadas del mundo a píxeles
    # is_y_axis=True invertirá el mapeo para que y_min_world (abajo) -> final_rows-1 (fila inferior) y y_max_world (arriba) -> 0 (fila superior)
    def map_world_to_pixel(val_world, world_min, world_max, pixel_dim, is_y_axis=False):
        if world_max == world_min:
            return 0 # Evitar división por cero, aunque ya se maneja fuera

        # Normalizar val_world al rango [0, 1] dentro de las coordenadas del mundo
        normalized_val = (val_world - world_min) / (world_max - world_min)

        if is_y_axis:
            # Para el eje Y, las coordenadas del mundo aumentan hacia arriba (bottom-up).
            # Queremos mapear esto a coordenadas de píxeles que aumentan hacia abajo (top-down).
            # Por lo tanto, (1 - normalized_val) invierte la escala.
            mapped_val = int((1 - normalized_val) * pixel_dim)
        else:
            # Para el eje X, las coordenadas del mundo aumentan hacia la derecha (left-to-right).
            # Las coordenadas de píxeles también aumentan hacia la derecha.
            mapped_val = int(normalized_val * pixel_dim)

        return np.clip(mapped_val, 0, pixel_dim - 1) # Asegurarse de que el valor esté dentro de los límites de los píxeles

    # 3. Representar el cuadrado grande con '1'
    print("🔧 Rellenando cuadrado grande...")
    ls_x_start_pixel = map_world_to_pixel(large_square['x_min'], x_min_world, x_max_world, final_cols, is_y_axis=False)
    ls_x_end_pixel = map_world_to_pixel(large_square['x_min'] + large_square['width'], x_min_world, x_max_world, final_cols, is_y_axis=False)

    # Para el eje Y, 'y_min' del large_square es la coordenada MÁS BAJA en el mundo, 'y_min + height' es la MÁS ALTA.
    # La función map_world_to_pixel con is_y_axis=True mapeará la coordenada MÁS ALTA del mundo a la fila 0 del píxel
    # y la coordenada MÁS BAJA del mundo a la fila final_rows-1 del píxel.
    # Por lo tanto, necesitamos pasar la coordenada MÁS ALTA del cuadrado para obtener la fila SUPERIOR del píxel,
    # y la coordenada MÁS BAJA del cuadrado para obtener la fila INFERIOR del píxel.
    ls_y_top_pixel = map_world_to_pixel(large_square['y_min'] + large_square['height'], y_min_world, y_max_world, final_rows, is_y_axis=True)
    ls_y_bottom_pixel = map_world_to_pixel(large_square['y_min'], y_min_world, y_max_world, final_rows, is_y_axis=True)

    # Al indexar, 'start_index' es inclusivo y 'end_index' es exclusivo. Por eso se suma 1 a los píxeles finales.
    binary_image[ls_y_top_pixel : ls_y_bottom_pixel + 1, ls_x_start_pixel : ls_x_end_pixel + 1] = 1
    print("✅ Cuadrado grande rellenado.")

    # 4. Representar los cuadrados pequeños con '0'
    print("🔧 Rellenando cuadrados pequeños...")
    for sq in small_squares:
        ss_x_start_pixel = map_world_to_pixel(sq['x_min'], x_min_world, x_max_world, final_cols, is_y_axis=False)
        ss_x_end_pixel = map_world_to_pixel(sq['x_min'] + sq['width'], x_min_world, x_max_world, final_cols, is_y_axis=False)

        ss_y_top_pixel = map_world_to_pixel(sq['y_min'] + sq['height'], y_min_world, y_max_world, final_rows, is_y_axis=True)
        ss_y_bottom_pixel = map_world_to_pixel(sq['y_min'], y_min_world, y_max_world, final_rows, is_y_axis=True)

        binary_image[ss_y_top_pixel : ss_y_bottom_pixel + 1, ss_x_start_pixel : ss_x_end_pixel + 1] = 0
        print("✅ Cuadrados pequeños rellenados.")

    # 5. Guardar la imagen binaria resultante
    output_dir = os.path.dirname(output_image_path)
    os.makedirs(output_dir, exist_ok=True)

    plt.imsave(output_image_path, binary_image, cmap='gray') # Usar cmap='gray' para imágenes binarias
    print(f"✅ Imagen binaria guardada exitosamente en: {output_image_path}")

print("--- PROCESO DE GENERACIÓN DE IMAGEN BINARIA COMPLETADO ---")

```

Evaluación de resultados: cálculo de SNR (Bloque 9)

Este bloque permite cuantificar la calidad de detección de defectos mediante el cálculo de la relación señal-ruido (SNR). Para ello, se comparan regiones de interés (ROI) correspondientes a defectos con zonas de referencia libres de discontinuidades.

```
# BLOQUE 9: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Cargar todos los datos de entrada

print("### Solicitar rutas y dimensiones necesarias para el análisis ###")

# 1. Solicitar ruta de la carpeta de CSVs
try:
    folder_path_csrs = input("1. Introduce la RUTA COMPLETA en Google Drive de la carpeta con los CSVs procesados  
(ej. '/content/drive/MyDrive/Your_Folder/Processed_CSVs'): ").strip()
    folder_path_csrs = folder_path_csrs.replace("'", '').replace('"', "")
    print(f"✅ Carpeta de CSVs: {folder_path_csrs}")
except Exception as e:
    print(f"❌ Error al obtener la ruta de la carpeta de CSVs: {e}")
    folder_path_csrs = None

# 2. Solicitar número de filas de la imagen
try:
    img_rows = int(input("2. Introduce el número de FILAS de las imágenes térmicas originales: "))
    print(f"✅ Filas de imagen: {img_rows}")
except ValueError:
    print(f"❌ Error: Debes introducir un número entero válido para las filas de la imagen.")
    img_rows = None
```

```
# 3. Solicitar número de columnas de la imagen
try:
    img_cols = int(input("3. Introduce el número de COLUMNAS de las imágenes térmicas originales: "))
    print(f"✅ Columnas de imagen: {img_cols}")
except ValueError:
    print(f"❌ Error: Debes introducir un número entero válido para las columnas de la imagen.")
    img_cols = None

# 4. Solicitar ruta completa de la imagen binaria de referencia
try:
    binary_image_path = input("4. Introduce la RUTA COMPLETA en Google Drive de la imagen binaria de referencia  
(ej. '/content/drive/MyDrive/Your_Folder/imagen_referencia.png'): ").strip()
    binary_image_path = binary_image_path.replace("'", '').replace('"', "")
    print(f"✅ Ruta de imagen binaria: {binary_image_path}")
except Exception as e:
    print(f"❌ Error al obtener la ruta de la imagen binaria de referencia: {e}")
    binary_image_path = None
```

```
# BLOQUE 9_2: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Preparación de las máscaras

import numpy as np
import pandas as pd
import os
import glob
from PIL import Image # For loading the image

print("### Preparando máscaras de regiones saludables y defectuosas ###")

# 1. Cargar la imagen binaria de referencia
if binary_image_path and os.path.exists(binary_image_path):
    try:
        # Load the image using Pillow
        img_pil = Image.open(binary_image_path)

        # Convert to grayscale to ensure single channel, then to numpy array
        # This ensures it's truly a binary image (0s and 1s or 0s and 255s)
        loaded_binary_image_single_channel = np.array(img_pil.convert('L')).astype(np.float32)

        # Normalize if loaded as 0/255 (common for grayscale images)
        if loaded_binary_image_single_channel.max() > 1.0:
            loaded_binary_image_single_channel = loaded_binary_image_single_channel / 255.0
```

```

# Ensure it's strictly binary 0 or 1 after normalization
loaded_binary_image_single_channel = (loaded_binary_image_single_channel > 0.5).astype(np.float32)

print(f"✅ Imagen binaria de referencia cargada y convertida a monocanal desde: {binary_image_path}")
print(f" - Dimensiones de la imagen binaria monocanal: {loaded_binary_image_single_channel.shape}")
except Exception as e:
    print(f"❌ Error al cargar la imagen binaria: {e}")
    loaded_binary_image_single_channel = None
else:
    print(f"❌ Error: La ruta de la imagen binaria '{binary_image_path}' no es válida o el archivo no existe.")
    loaded_binary_image_single_channel = None

# Check against the single-channel image
if loaded_binary_image_single_channel is not None and img_rows == loaded_binary_image_single_channel.shape[0] and img_cols == loaded_binary_image_single_channel.shape[1]:

    # 2. Definir la máscara de región 'saludable'
    # Pixels with value 1 are considered healthy
    healthy_mask = (loaded_binary_image_single_channel == 1).flatten() # Flatten for easier indexing of flattened data
    print(f"✅ Máscara de región saludable creada. Total de píxeles saludables: {np.sum(healthy_mask)}")

    # 3. Generar máscaras individuales para cada región de defecto
    # Pixels with value 0 within each small_square are defects
    defect_masks = []
    for i, sq in enumerate(small_squares):
        # Create a temporary mask for the current defect region (initially all False)
        current_defect_mask_2d = np.zeros_like(loaded_binary_image_single_channel, dtype=bool)

```

```

# Re-map coordinates from world to pixel space for the current defect
x_min_world = large_square['x_min']
y_min_world = large_square['y_min']
x_max_world = large_square['x_min'] + large_square['width']
y_max_world = large_square['y_min'] + large_square['height']

def map_world_to_pixel_reloaded(val_world, world_min, world_max, pixel_dim, is_y_axis=False):
    normalized_val = (val_world - world_min) / (world_max - world_min)
    if is_y_axis:
        mapped_val = int((1 - normalized_val) * pixel_dim)
    else:
        mapped_val = int(normalized_val * pixel_dim)
    return np.clip(mapped_val, 0, pixel_dim - 1)

ss_x_start_pixel = map_world_to_pixel_reloaded(sq['x_min'], x_min_world, x_max_world, img_cols, is_y_axis=False)
ss_x_end_pixel = map_world_to_pixel_reloaded(sq['x_min'] + sq['width'], x_min_world, x_max_world, img_cols, is_y_axis=False)

ss_y_top_pixel = map_world_to_pixel_reloaded(sq['y_min'] + sq['height'], y_min_world, y_max_world, img_rows, is_y_axis=True)
ss_y_bottom_pixel = map_world_to_pixel_reloaded(sq['y_min'], y_min_world, y_max_world, img_rows, is_y_axis=True)

# Adjust end pixels to be inclusive for slicing (Python slices are exclusive at end)
current_defect_mask_2d[ss_y_top_pixel : ss_y_bottom_pixel + 1, ss_x_start_pixel : ss_x_end_pixel + 1] = True

```

```

# Flatten the defect mask for consistency with healthy_mask
defect_masks.append(current_defect_mask_2d.flatten())
print(f"✅ Máscara para Defecto {i+1} creada. Total de píxeles en el defecto: {np.sum(defect_masks[-1])}")
print(f"✅ {len(defect_masks)} máscaras de defectos individuales generadas.")

# 4. Combinar todas las máscaras de defectos para obtener una máscara general de 'defecto'
all_defects_mask = np.zeros_like(healthy_mask, dtype=bool)
for mask in defect_masks:
    all_defects_mask = np.logical_or(all_defects_mask, mask)
print(f"✅ Máscara combinada de todos los defectos creada. Total de píxeles defectuosos: {np.sum(all_defects_mask)}")

# Verify that healthy and defect regions do not overlap
# Healthy regions are 1, defect regions are 0 (within the large square)
# The masks are now created correctly. We should re-verify this logic.
# A pixel is healthy if loaded_binary_image_single_channel == 1
# A pixel is defective if loaded_binary_image_single_channel == 0 AND it's within a small_square definition

# The previous overlap warning was likely due to comparing the healthy_mask derived from a 4-channel image
# with defect_masks derived by mapping coordinates. Let's ensure the final masks are coherent.

# The healthy_mask should only contain pixels from loaded_binary_image_single_channel == 1
# The all_defects_mask should only contain pixels from loaded_binary_image_single_channel == 0 which are within small_squares

```

```
# Let's perform a more robust check:
# An overlap would mean a pixel is marked as both healthy AND part of a defect region.
# healthy_mask selects pixels where the image has a value of 1.
# all_defects_mask selects pixels where the mapped small squares are located.
# If the small squares are indeed '0' in the binary image, there should be no overlap.

# Let's check for overlap: Pixels that are part of the 'healthy' mask and also part of 'all_defects_mask'
# The actual healthy regions are where 'loaded_binary_image_single_channel' is 1.
# The actual defect regions are where 'loaded_binary_image_single_channel' is 0 AND a 'small_square' covers it.

# Re-evaluate healthy and defect pixels directly from the binary image.
# A pixel is truly healthy if its value in the binary image is 1.
# A pixel is truly defective if its value in the binary image is 0 AND it falls within a small square's coordinates.

# The 'healthy_mask' already takes pixels == 1.
# For 'all_defects_mask', we need to ensure it only covers pixels that are 0 in the binary image.

# The 'all_defects_mask' is constructed from mapped coordinates. We need to intersect it with the 'loaded_binary_image_single_channel == 0'
true_defect_pixels_mask = np.logical_and(all_defects_mask, (loaded_binary_image_single_channel == 0).flatten())
```

```
overlap = np.sum(np.logical_and(healthy_mask, true_defect_pixels_mask))
if overlap > 0:
    print(f"⚠ Advertencia: Hay {overlap} píxeles de solapamiento entre las regiones saludables y defectuosas.")
else:
    print("✅ No hay solapamiento entre las regiones saludables y defectuosas (según la imagen binaria).")

# Update 'all_defects_mask' to reflect only the actual '0' pixels within the specified defect regions
all_defects_mask = true_defect_pixels_mask
print(f"✅ Máscara combinada de todos los defectos actualizada para incluir solo píxeles '0'. Total: {np.sum(all_defects_mask)}")

else:
    print("❌ Error: La imagen binaria no pudo ser cargada o sus dimensiones no coinciden con las especificadas (img_rows, img_cols).")
    healthy_mask = None
    defect_masks = []
    all_defects_mask = None
```

```
# BLOQUE 9_3: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Cálculo del SNR

import re # Importar la librería re para expresiones regulares
import numpy as np
import pandas as pd
import os
import glob

print("### Iniciando el análisis SNR ###")

# Crear la carpeta para los resultados del análisis SNR
output_snr_folder = os.path.join(os.path.dirname(folder_path_csrs), "SNR_Analysis_Results")
os.makedirs(output_snr_folder, exist_ok=True)
print(f"✅ Carpeta para resultados SNR creada: {output_snr_folder}")

# Lista para almacenar los resultados del SNR general
overall_snr_results = []
# Lista para almacenar los resultados del SNR por defecto
per_defect_snr_results = []

# 5. Función para calcular SNR
def calculate_snr(image_data_flat, healthy_mask, defect_mask):
    # Asegurarse de que las máscaras sean booleanas
    healthy_mask = np.array(healthy_mask).astype(bool)
    defect_mask = np.array(defect_mask).astype(bool)
```

```

# Extraer temperaturas de las regiones saludables y defectuosas
healthy_temperatures = image_data_flat[healthy_mask]
defect_temperatures = image_data_flat[defect_mask]

# Calcular media de la región saludable (mu_h)
mu_h = np.mean(healthy_temperatures)

# Calcular desviación estándar de la región saludable (sigma_h) - "ruido"
sigma_h = np.std(healthy_temperatures)

# Calcular media de la región defectuosa (mu_d)
mu_d = np.mean(defect_temperatures)

# Prevenir división por cero para sigma_h
if sigma_h == 0:
    return np.nan # SNR indefinido si no hay ruido en la región saludable

# Calcular el SNR
snr = abs(mu_d - mu_h) / sigma_h
return snr

```

```

# Obtener la lista de archivos CSV en la carpeta
csv_files = glob.glob(os.path.join(folder_path_csrs, '*.csv'))

if not csv_files:
    print(f"❌ Error: No se encontraron archivos CSV en la carpeta: {folder_path_csrs}")
else:
    print(f"✅ Se encontraron {len(csv_files)} archivos CSV para el análisis SNR.")

    # Iterar a través de cada archivo CSV
    for csv_file_path in csv_files:
        file_name = os.path.basename(csv_file_path)
        print(f"\n⌚ Procesando archivo: {file_name}")

        try:
            df_processed_images = pd.read_csv(csv_file_path)

            # Determinar el sufijo de la derivada si el archivo es de TSR
            derivative_suffix = ''
            if 'tsr_stats_derivative' in file_name:
                match = re.search(r'tsr_stats_derivative(\d+)\.csv', file_name)
                if match:
                    derivative_suffix = f'_{match.group(1)}'

```

```
# Iterar a través de cada columna (instante de tiempo/imagen) en el DataFrame
for col_idx, column in enumerate(df_processed_images.columns):
    algorithmic_result = f"{column}{derivative_suffix}"
    flattened_image_data = df_processed_images[column].to_numpy()

    # Validar que los datos de la imagen tienen el tamaño esperado
    if len(flattened_image_data) != (img_rows * img_cols):
        print(f"⚠ Advertencia: El tamaño de los datos en la columna '{column}' del archivo '{file_name}' no coincide con (img_rows * img_cols). Saltando este resultado algorítmico.")
        continue

    # 6. Calcular el SNR general
    overall_snr = calculate_snr(flattened_image_data, healthy_mask, all_defects_mask)
    overall_snr_results.append({
        'file_name': file_name,
        'algorithmic_result': algorithmic_result, # Renamed key
        'overall_snr': overall_snr
    })

    # 7. Calcular el SNR por defecto
    for i, defect_mask in enumerate(defect_masks):
        defect_snr = calculate_snr(flattened_image_data, healthy_mask, defect_mask)
        per_defect_snr_results.append({
            'file_name': file_name,
            'algorithmic_result': algorithmic_result, # Renamed key
            'defect_id': f'Defect_{i+1}',
            'defect_snr': defect_snr
        })
```

```
print(f"✅ Archivo '{file_name}' procesado.")
```

```
except Exception as e:
```

```
    print(f"❌ Error al procesar el archivo '{file_name}': {e}")
```

```
# BLOQUE 9_4: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Guardar valores SNR

print("### Agregando y guardando resultados SNR ###")

# 8. Guardar los resultados del SNR general
if overall_snr_results:
    df_overall_snr = pd.DataFrame(overall_snr_results)
    overall_snr_output_path = os.path.join(output_snr_folder, "Overall_SNR_Results.csv")
    df_overall_snr.to_csv(overall_snr_output_path, index=False)
    print(f"✅ Resultados del SNR general guardados en: {overall_snr_output_path}")
    print(f" - Total de registros: {len(df_overall_snr)}")
else:
    print("⚠ No se generaron resultados de SNR general.")
    df_overall_snr = pd.DataFrame()

# 9. Guardar los resultados del SNR por defecto
if per_defect_snr_results:
    df_per_defect_snr = pd.DataFrame(per_defect_snr_results)
    per_defect_snr_output_path = os.path.join(output_snr_folder, "Per_Defect_SNR_Results.csv")
    df_per_defect_snr.to_csv(per_defect_snr_output_path, index=False)
    print(f"✅ Resultados del SNR por defecto guardados en: {per_defect_snr_output_path}")
    print(f" - Total de registros: {len(df_per_defect_snr)}")
else:
    print("⚠ No se generaron resultados de SNR por defecto.")
    df_per_defect_snr = pd.DataFrame()
```

```

# 10. Identificar y mostrar el máximo SNR general
print("\n### Resumen de SNR ###")
if not df_overall_snr.empty and 'overall_snr' in df_overall_snr.columns:
    max_overall_snr = df_overall_snr['overall_snr'].max()
    max_overall_snr_row = df_overall_snr.loc[df_overall_snr['overall_snr'].idxmax()]
    print(f"✅ Máximo SNR General: {max_overall_snr:.4f}")
    print(f" - Archivo: {max_overall_snr_row['file_name']}")
    print(f" - Resultado Algorítmico: {max_overall_snr_row['algorithmic_result']}") # Updated label and key
else:
    print("❌ No se pudo encontrar el máximo SNR general (DataFrame vacío o columna no encontrada).")

# 11. Identificar y mostrar el máximo SNR por defecto
if not df_per_defect_snr.empty and 'defect_snr' in df_per_defect_snr.columns:
    max_per_defect_snr = df_per_defect_snr['defect_snr'].max()
    max_per_defect_snr_row = df_per_defect_snr.loc[df_per_defect_snr['defect_snr'].idxmax()]
    print(f"✅ Máximo SNR por Defecto: {max_per_defect_snr:.4f}")
    print(f" - Archivo: {max_per_defect_snr_row['file_name']}")
    print(f" - Resultado Algorítmico: {max_per_defect_snr_row['algorithmic_result']}") # Updated label and key
    print(f" - Defecto ID: {max_per_defect_snr_row['defect_id']}")
else:
    print("❌ No se pudo encontrar el máximo SNR por defecto (DataFrame vacío o columna no encontrada).")

print("\n✅ Análisis SNR completado.")

```

```

# BLOQUE 9_5: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Representar gráficamente los valores SNR guardados, cargándolos primero de nuevo

import pandas as pd
import os

print("### Cargando resultados SNR ###")

# Construir la ruta completa para Overall_SNR_Results.csv
overall_snr_path = os.path.join(output_snr_folder, "Overall_SNR_Results.csv")

# Cargar el CSV en un DataFrame
try:
    df_overall_snr = pd.read_csv(overall_snr_path)
    print(f"✅ 'Overall_SNR_Results.csv' cargado exitosamente.")
    print("Primeras 5 filas de df_overall_snr:")
    print(df_overall_snr.head())
    print(f"Forma de df_overall_snr: {df_overall_snr.shape}")
except Exception as e:
    print(f"❌ Error al cargar 'Overall_SNR_Results.csv': {e}")
    df_overall_snr = pd.DataFrame() # Initialize as empty DataFrame on error

print("\n")

```

```

# Construir la ruta completa para Per_Defect_SNR_Results.csv
per_defect_snr_path = os.path.join(output_snr_folder, "Per_Defect_SNR_Results.csv")

# Cargar el CSV en un DataFrame
try:
    df_per_defect_snr = pd.read_csv(per_defect_snr_path)
    print(f"✅ 'Per_Defect_SNR_Results.csv' cargado exitosamente.")
    print("Primeras 5 filas de df_per_defect_snr:")
    print(df_per_defect_snr.head())
    print(f"Forma de df_per_defect_snr: {df_per_defect_snr.shape}")
except Exception as e:
    print(f"❌ Error al cargar 'Per_Defect_SNR_Results.csv': {e}")
    df_per_defect_snr = pd.DataFrame() # Initialize as empty DataFrame on error

```

```
# BLOQUE 9_6: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Representar gráficamente los valores SNR guardados, reestructurando ahora los datos

print("### Preparando datos para el gráfico combinado ###")

# 1. Extraer el máximo SNR general por 'algorithmic_result'
# Group by 'algorithmic_result' and get the maximum 'overall_snr'
if not df_overall_snr.empty:
    max_overall_snr_grouped = df_overall_snr.groupby('algorithmic_result')['overall_snr'].max().reset_index()
    max_overall_snr_grouped.rename(columns={'overall_snr': 'SNR General'}, inplace=True)
    print("✅ Máximo SNR general por resultado algorítmico extraído y renombrado.")
    print("Primeras 5 filas de max_overall_snr_grouped:")
    print(max_overall_snr_grouped.head())
else:
    print("⚠ df_overall_snr está vacío. No se pueden extraer máximos SNR generales.")
    max_overall_snr_grouped = pd.DataFrame(columns=['algorithmic_result', 'SNR General'])

print("\n")
```

```
# 2. Extraer el máximo SNR por defecto por 'algorithmic_result' y su 'defect_id'
# Group by 'algorithmic_result', find the row with max 'defect_snr', and select relevant columns
if not df_per_defect_snr.empty:
    # Find the index of the maximum 'defect_snr' within each 'algorithmic_result' group
    idx = df_per_defect_snr.groupby('algorithmic_result')['defect_snr'].idxmax()
    max_per_defect_snr_grouped = df_per_defect_snr.loc[idx, ['algorithmic_result', 'defect_id', 'defect_snr']].reset_index(drop=True)
    max_per_defect_snr_grouped.rename(columns={'defect_snr': 'Max Per-Defect SNR', 'defect_id': 'Max Per-Defect ID'}, inplace=True)
    print("✅ Máximo SNR por defecto (con ID) por resultado algorítmico extraído y renombrado.")
    print("Primeras 5 filas de max_per_defect_snr_grouped:")
    print(max_per_defect_snr_grouped.head())
else:
    print("⚠ df_per_defect_snr está vacío. No se pueden extraer máximos SNR por defecto.")
    max_per_defect_snr_grouped = pd.DataFrame(columns=['algorithmic_result', 'Max Per-Defect ID', 'Max Per-Defect SNR'])
```

```
# 3. Fusionar los dos DataFrames
combined_max_snr = pd.merge(
    max_overall_snr_grouped,
    max_per_defect_snr_grouped,
    on='algorithmic_result',
    how='outer'
)
combined_max_snr.rename(columns={'algorithmic_result': 'Algorithmic Result'}, inplace=True)

print("\n✅ DataFrames de máximos SNR combinados y columna renombrada.")
print("Primeras 5 filas del DataFrame combinado (combined_max_snr):")
print(combined_max_snr.head())
print(f"Forma del DataFrame combinado: {combined_max_snr.shape}")
```

```
# BLOQUE 9_7: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Representar gráficamente los valores SNR guardados, guarda los primeros gráficos

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np # Import numpy for array operations

print("### Generando y guardando gráfico de barras de máximos SNR ###")

# Establecer el estilo de Seaborn para una mejor visualización
sns.set_theme(style="whitegrid")

# Aumentar el tamaño de la figura para acomodar todas las etiquetas
plt.figure(figsize=(16, 8))

# Posición para las barras
bar_width = 0.35
index = np.arange(len(combined_max_snr['Algorithmic Result']))

# Crear las barras para 'SNR General'
plt.bar(index, combined_max_snr['SNR General'], bar_width, label='SNR General', color='skyblue')

# Crear las barras para 'Max Per-Defect SNR'
rects = plt.bar(index + bar_width, combined_max_snr['Max Per-Defect SNR'], bar_width, label='Máximo SNR por Defecto', color='lightcoral')
```

```

# Añadir anotaciones para 'Max Per-Defect ID' en las barras de 'Max Per-Defect SNR'
for i, rect in enumerate(rects):
    height = rect.get_height()
    plt.annotate(f"({combined_max_snr['Max Per-Defect ID'][i]})",
                 xy=(rect.get_x() + rect.get_width() / 2, height),
                 xytext=(0, 3), # 3 points vertical offset
                 textcoords="offset points",
                 ha='center', va='bottom', fontsize=8)

# Añadir etiquetas, título y leyenda
plt.xlabel('Resultado Algorítmico', fontsize=12)
plt.ylabel('Valor de SNR', fontsize=12)
plt.title('Comparación de Máximo SNR General vs. Máximo SNR por Defecto por Resultado Algorítmico', fontsize=14)
plt.xticks(index + bar_width / 2, combined_max_snr['Algorithmic Result'], rotation=45, ha='right', fontsize=10)
plt.yticks(fontsize=10)
plt.legend(fontsize=10)
plt.tight_layout() # Ajustar el diseño para evitar el solapamiento

# Construir la ruta de salida para el gráfico
snr_comparison_plot_path = os.path.join(output_snr_folder, "SNR_Comparison_Bar_Chart.png")

# Guardar el gráfico
plt.savefig(snr_comparison_plot_path)
plt.close() # Cerrar el plot para liberar memoria

print(f"✅ Gráfico de comparación de SNR guardado en: {snr_comparison_plot_path}")

```

BLOQUE 9.8: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Representar gráficamente los valores SNR guardados, los segundos gráficos es filtrando el valor del SNR. Petición

```

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

print("### Generando gráfico de barras para SNR > Umbral por Defecto y Algoritmo ###")

# Solicitar el valor del umbral de SNR al usuario
try:
    snr_threshold = float(input("1. Introduce el valor del umbral de SNR (ej. 1.0, 1.5, 2.0): "))
    print(f"✅ Umbral de SNR establecido en: {snr_threshold}")
except ValueError:
    print("❌ Error: Debes introducir un número válido para el umbral de SNR. Usando el valor por defecto 1.0.")
    snr_threshold = 1.0

# 1. Filtrar el DataFrame para incluir solo SNR > snr_threshold
df_snr_above_1 = df_per_defect_snr[df_per_defect_snr['defect_snr'] > snr_threshold].copy()
print(f"✅ {len(df_snr_above_1)} entradas con SNR > {snr_threshold} encontradas.")

# 2. Obtener todos los IDs de defecto únicos para asegurar que todos aparezcan en el eje x
all_defect_ids = sorted(df_per_defect_snr['defect_id'].unique())

```

```
# 3. Configurar el estilo del gráfico
sns.set_theme(style="whitegrid")
plt.figure(figsize=(16, 8))

# 4. Crear el gráfico de barras. Usaremos `x` para los IDs de defecto, `y` para el SNR y `hue` para el resultado algorítmico.
# Si un defecto no tiene entradas en df_snr_above_1, no se dibujará una barra para él, pero su etiqueta en el eje X sí aparecerá.
ax = sns.barplot(x='defect_id', y='defect_snr', hue='algorithmic_result', data=df_snr_above_1, palette='viridis', order=all_defect_ids)

# 5. Añadir anotaciones a las barras
handles, labels = ax.get_legend_handles_labels() # Get handles and labels for the legend
for i, container in enumerate(ax.containers):
    algorithmic_result_label = labels[i] # Use the label from the legend
    for bar in container:
        height = bar.get_height()
        if height > 0: # Solo anotar barras existentes
            ax.annotate(f'{height:.2f}\n({algorithmic_result_label})', # Show SNR value and Algorithmic Result
                        xy=(bar.get_x() + bar.get_width() / 2, height),
                        xytext=(0, 3), # 3 points vertical offset
                        textcoords="offset points",
                        ha='center', va='bottom', fontsize=8, rotation=45)

# 6. Añadir título y etiquetas
plt.title(f'SNR > {snr_threshold} por Defecto y Resultado Algorítmico', fontsize=16)
plt.xlabel('ID del Defecto', fontsize=12)
plt.ylabel('SNR (Valor)', fontsize=12)
```

```
# 7. Ajustar el eje Y para que comience en el umbral (o ligeramente por debajo si es necesario para visibilidad)
plt.ylim(bottom=snr_threshold * 0.9 if snr_threshold > 0 else 0.0)

# 8. Rotar las etiquetas del eje X para mayor legibilidad
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.yticks(fontsize=10)

# 9. Añadir una leyenda con título
ax.legend(handles=handles, labels=labels, title='Resultado Algorítmico', bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0.)

# 10. Ajustar el diseño y guardar el gráfico
plt.tight_layout(rect=[0, 0, 0.88, 1]) # Ajustar rect para que la leyenda no se superponga

snr_above_1_plot_path = os.path.join(output_snr_folder, f"SNR_Above_{str(snr_threshold).replace('.', '_')}Per_Defect_Plot.png")
plt.savefig(snr_above_1_plot_path)
plt.close() # Cerrar el plot para liberar memoria

print(f"✅ Gráfico 'SNR > {snr_threshold} por Defecto y Resultado Algorítmico' guardado en: {snr_above_1_plot_path}")
```

```
# BLOQUE 9.9: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Representar gráficamente los valores SNR guardados,
los segundos gráficos es filtrando el valor del SNR. Procesamiento

print("### Procesando máximos SNR por cada defecto individual ###")

# 1. Verificar si df_per_defect_snr está vacío
if not df_per_defect_snr.empty:
    # 2. Agrupar por 'defect_id' y encontrar la fila con el máximo 'defect_snr'
    idx = df_per_defect_snr.groupby('defect_id')['defect_snr'].idxmax()

    # 3. Seleccionar las columnas relevantes de estas filas
    max_snr_per_defect = df_per_defect_snr.loc[idx, ['defect_id', 'algorithmic_result', 'defect_snr']].reset_index(drop=True)

    # 4. Renombrar las columnas para claridad
    max_snr_per_defect.rename(columns={'defect_snr': 'Max SNR', 'algorithmic_result': 'Algorithmic Result'}, inplace=True)
    print("✅ Máximo SNR por cada defecto individual extraído.")
    print("Primeras 5 filas de max_snr_per_defect:")
    print(max_snr_per_defect.head())
    print(f"Forma de max_snr_per_defect: {max_snr_per_defect.shape}")
else:
    print("⚠️ df_per_defect_snr está vacío. No se pueden extraer máximos SNR por defecto individual.")
# 5. Inicializar como DataFrame vacío con las columnas esperadas
max_snr_per_defect = pd.DataFrame(columns=['defect_id', 'Algorithmic Result', 'Max SNR'])
```

```

# BLOQUE 9_10: Comparación entre los resultados PCT, SPCT y TSR. Signal to Noise Ratio. Representar gráficamente los valores SNR guardados,
# los segundos gráficos es filtrando el valor del SNR. Guardado

import matplotlib.pyplot as plt
import seaborn as sns

print("### Generando gráfico de barras de Máximo SNR por Defecto ###")

# 1. Establecer el estilo de Seaborn para una mejor visualización
sns.set_theme(style="whitegrid")

# 2. Crear una Matplotlib figure con un tamaño apropiado
plt.figure(figsize=(12, 7))

# 3. Crear un gráfico de barras
bars = plt.bar(max_snr_per_defect['defect_id'], max_snr_per_defect['Max SNR'], color='mediumseagreen')

# 4. Añadir anotaciones para el 'Algorithmic Result' en cada barra
for bar in bars:
    yval = bar.get_height()
    # Adjust annotation position to be just above the bar
    plt.text(bar.get_x() + bar.get_width()/2, yval + 0.05,
             max_snr_per_defect.loc[max_snr_per_defect['Max SNR'] == yval, 'Algorithmic Result'].iloc[0],
             ha='center', va='bottom', fontsize=9, color='black', rotation=45)

# 5. Añadir un título descriptivo
plt.title('Máximo SNR por Defecto con Algoritmo Correspondiente', fontsize=14)

# 6. Etiquetar los ejes
plt.xlabel('ID del Defecto', fontsize=12)
plt.ylabel('Máximo SNR', fontsize=12)

# 7. Asegurar que las etiquetas del eje x son legibles
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.yticks(fontsize=10)

# 8. Añadir una cuadrícula
plt.grid(axis='y', linestyle='--', alpha=0.7)

# 9. Usar plt.tight_layout() para evitar solapamiento
plt.tight_layout()

# 10. Construir la ruta de salida para el gráfico
per_defect_snr_plot_path = os.path.join(output_snr_folder, "Max_SNR_Per_Defect_Plot.png")

# 11. Guardar el gráfico como un archivo PNG
plt.savefig(per_defect_snr_plot_path)

# 12. Cerrar el plot para liberar memoria
plt.close()

print(f"✅ Gráfico de Máximo SNR por Defecto guardado en: {per_defect_snr_plot_path}")

```

Conclusión del anexo

La implementación de los algoritmos descritos permite el procesamiento avanzado de las secuencias termográficas y la obtención de resultados clave, como las eigenimágenes (SC_0, SC_1, EOF_1), analizadas en el Capítulo 4 de la memoria.

Estos procedimientos, basados en metodologías ampliamente validadas en la literatura científica, han permitido mejorar la detectabilidad de defectos subsuperficiales y comparar el comportamiento térmico de los materiales estudiados.

El código completo desarrollado, junto con las máscaras de referencia utilizadas para la definición de las regiones de interés, se encuentra disponible en el repositorio asociado a este Trabajo Fin de Grado.

ANEXO IV: DECLARACIÓN SOBRE LA UTILIZACIÓN DE HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL

En la elaboración de este Trabajo Fin de Grado se ha recurrido a herramientas de inteligencia artificial únicamente como instrumento de apoyo para la redacción y revisión estilística del documento. La aplicación de estas tecnologías se ha restringido a la corrección gramatical y la mejora de la claridad expositiva, con el fin de adecuar el texto a las exigencias del lenguaje académico. En consecuencia, no se ha utilizado la IA para generar contenido técnico, diseñar la metodología, analizar resultados ni formular conclusiones, siendo todas estas fases fruto exclusivo del trabajo personal del alumno. El contenido final ha sido supervisado de manera crítica para garantizar su rigor y coherencia, manteniendo el alumno el control absoluto y la responsabilidad sobre la obra.