



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Herramienta web de gestión de horarios en la ENM

Grado en Ingeniería Mecánica

ALUMNO: Raúl Urbano Burrel

DIRECTORES: Norberto Fernández García
Pedro Azorín Montesinos

CURSO ACADÉMICO: 2024-2025

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Herramienta web de gestión de horarios en la ENM

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

Universida_{de}Vigo

RESUMEN

Este Trabajo de Fin de Grado presenta el desarrollo de una herramienta web para la gestión y planificación del horario académico en la Escuela Naval Militar (ENM), permitiendo la edición de la tradicional “sábana” utilizada para organizar clases, aulas y recursos docentes. La planificación horaria en la ENM presenta una gran complejidad debido a la combinación de asignaturas universitarias y actividades militares, así como la necesidad de actualizar el horario cuando se producen situaciones que lo requieren. La herramienta desarrollada busca simplificar este proceso mediante una interfaz intuitiva y funcional que permita a los responsables de la planificación gestionar los horarios de manera eficiente y reducir errores.

El sistema está basado en Python y Flask para el *backend*, gestionando los datos mediante archivos Excel. En el *frontend*, se ha utilizado HTML, CSS y JavaScript (HTMX) para ofrecer una experiencia de usuario interactiva y sin necesidad de recargar la página. Además, la aplicación permite la exportación de horarios en formatos CSV y PDF, facilitando la distribución de la información.

Los resultados obtenidos muestran que la herramienta simplifica el proceso de edición de los horarios, ofreciendo una solución escalable y adaptable a futuras ampliaciones, como la incorporación de más cursos y titulaciones.

PALABRAS CLAVE

Sábana, Horario, *Backend*, *Frontend*, Herramienta web

AGRADECIMIENTOS

Primero de todo me gustaría agradecer a Dios por ayudarme una vez más a alcanzar un nuevo objetivo en mi vida. Por supuesto, también a mis tutores que son los que me han guiado de forma diligente a lo largo de todo el desarrollo del TFG con mucha paciencia. Y, por último, agradecer a todos mis familiares, amigos, compañeros y mandos que me han estado ayudando durante estos intensos cinco años de mi vida, aprendiendo de ellos día a día para la consecución de uno de mis sueños de pequeño. Gracias a todos de corazón.

Contenido

Contenido	1
Índice de Figuras	5
Índice de Tablas.....	8
1 Introducción y objetivos	9
1.1 Antecedentes	9
1.2 Motivación	9
1.3 Objetivos	10
1.4 Estructura y apartados de la memoria	10
2 Estado del arte	12
2.1 Formato de la sábana actual	12
2.1.1 Sábana general	12
2.1.2 Planificación horaria de cada asignatura	13
2.2 Lenguajes de programación	13
2.2.1 Python	14
2.2.2 C++	14
2.2.3 Java	15
2.2.4 C.....	15
2.2.5 JavaScript.....	15
2.3 Lenguajes de marcado.....	15
2.3.1 HTML5	16
2.3.2 XML	16
2.3.3 Markdown.....	16
2.4 Bibliotecas de Python para manejo de archivos de Microsoft Excel	16
2.4.1 Openpyxl.....	17
2.4.2 Pandas	17
2.4.3 XlsxWriter	17
2.5 Otras herramientas útiles.....	18
2.5.1 Excel	18
2.5.2 Framework Flask	18
2.5.3 Lenguaje de diseño CSS	19
2.5.4 HTMX.....	19
2.5.5 Jinja2.....	19
2.5.6 Librería ReportLab	20
2.6 Opciones disponibles para la gestión de horarios en la web.....	20

2.6.1 Software de gestión de horarios.....	20
2.6.2 Algoritmos y metodologías utilizadas en la planificación de horarios.....	21
2.7 Trabajos de Fin de Grado previos	21
2.7.1 Desarrollo de una herramienta de apoyo a la planificación horaria de la ENM.....	21
2.7.2 Desarrollo automático de la sábana de la ENM-CUD.....	22
3 Desarrollo del TFG.....	23
3.1 Elección de tecnologías de trabajo.....	23
3.1.1 Backend y gestión de datos.....	23
3.1.2 Frontend y diseño de la interfaz.....	23
3.1.3 Instalación del software necesario para la herramienta	24
3.1.3.1 Instalación de Python	24
3.1.3.2 Instalación de Openpyxl	24
3.1.3.3 Instalación de Flask	24
3.1.3.4 Instalación de ReportLab.....	24
3.2 Diseño de la aplicación web.....	24
3.2.1 Base de datos en archivos Excel	25
3.2.1.1 Asignaturas	25
3.2.1.2 Profesores	27
3.2.1.3 POD (profesores – asignaturas).....	28
3.2.1.4 Aulas.....	29
3.2.1.5 Titulación.....	30
3.2.1.6 Horario.....	32
3.2.2 Frontend.....	33
3.2.2.1 Configuración y estructura del <head>	33
3.2.2.2 Configuración y estructura de los estilos CSS	33
3.2.2.2.1 Diseño general y estilización del cuerpo.....	33
3.2.2.2.2 Estilización de la tabla de horarios.....	34
3.2.2.2.3 Estilización de las celdas de encabezados y horas	34
3.2.2.2.4 Estilización de botones interactivos	35
3.2.2.2.5 Estilización de los botones copiar y pegar	36
3.2.2.2.6 Jerarquización del texto y diferenciación de elementos	37
3.2.2.2.7 Botones de exportación de datos	38
3.2.2.2.8 Estilos de imágenes y elementos gráficos	39
3.2.2.2.9 Estilización de las celdas Deporte/Comida	39
3.2.2.3 Script para el restablecimiento de datos al recargar la página web.....	40
3.2.2.4 Script para capturar en imagen el estado de la herramienta web	40
3.2.2.5 Configuración y estructura del <body>.....	41

3.2.2.5.1 Estructura y funcionalidad del <table>	43
3.2.3 Backend	45
3.2.3.1 Diccionarios.....	46
3.2.3.2 Ruta /menú	50
3.2.3.3 Ruta /desplegar_menú	51
3.2.3.4 Ruta /copiar	53
3.2.3.5 Ruta /pegar	53
3.2.3.6 Ruta /asignaturas	54
3.2.3.7 Ruta /selección_asignatura	55
3.2.3.8 Ruta /profesores.....	57
3.2.3.9 Ruta /selección_profesor	58
3.2.3.10 Ruta /aulas	59
3.2.3.11 Ruta /selección_aula.....	60
3.2.3.12 Ruta /sesión	61
3.2.3.13 Ruta /selección_sesión	62
3.2.3.14 Ruta /deporte_comida.....	62
3.2.3.15 Ruta /seleccionar_opción.....	63
3.2.3.16 Ruta /exportar_csv.....	63
3.2.3.17 Ruta /resetear_selecciones.....	65
3.2.3.18 Ruta /exportar_pdf.....	65
3.2.4 Comunicación entre el servidor y el navegador	66
3.2.4.1 Rutas GET	66
3.2.4.2 Rutas POST	67
3.2.4.3 Interacción entre JavaScript y Flask.....	67
4 Resultados	69
4.1 Inicializar la herramienta web	69
4.2 Interacción con la herramienta web para rellenar una celda	70
4.2.1 Cambiar estudio por una asignatura	70
4.2.1.1 Asignatura con profesor asociado	71
4.2.1.2 Asignatura sin profesor asociado	72
4.2.2 Escoger profesor	72
4.2.3 Escoger aula.....	73
4.2.4 Escoger sesión	74
4.2.5 Pegar información en otra celda	75
4.3 Interacción con la herramienta web para exportar la sábana	75
4.3.1 Exportar sábana en archivo Excel formato CSV	76
4.3.2 Exportar sábana en archivo PDF	77

5 Conclusiones y líneas futuras	78
5.1 Conclusiones	78
5.2 Líneas futuras	78
6 Bibliografía.....	80
Anexo I: Implicaciones Sociales, y/o Económicas, y/o Ambientales	82
Anexo II: Reflexiones Éticas y Sociales	83
Anexo III: Código backend – Flask (__init__.py)	84
Anexo IV: Código frontend – HTML (index.html).....	100

ÍNDICE DE FIGURAS

Figura 2-1 Recorte de la sábana general de la ENM	13
Figura 2-2 Tabla reparto de horas Redes y Ordenadores (curso 2023/2024).....	13
Figura 2-3 Índice de la comunidad de programadores TIOBE [1].....	14
Figura 2-4 Ejemplo sencillo de lenguaje HTML [9]	16
Figura 2-5 Ejemplo de elemento gráfico contenido en un documento PDF creado con ReportLab	20
Figura 2-6 Ejemplo interfaz web Untis [24]	21
Figura 3-1 Resumen gráfico del desarrollo	25
Figura 3-2 <head> HTML.....	33
Figura 3-3 Estilos CSS generales	34
Figura 3-4 Estilos CSS de la tabla de horarios	34
Figura 3-5 Estilos CSS de las franjas horarias y grupos de clase.....	35
Figura 3-6 Estilos CSS de los botones de selección.....	36
Figura 3-7 Estilos CSS de botones copiar y pegar	37
Figura 3-8 Estilos CSS de los elementos <div> de las celdas.....	38
Figura 3-9 Estilos CSS de los botones de exportación de datos en CSV y PDF.....	38
Figura 3-10 Estilos CSS de imágenes del encabezado	39
Figura 3-11 Estilo CSS de marca de agua	39
Figura 3-12 Estilos CSS celdas Deporte/Comida.....	40
Figura 3-13 Código JavaScript que elimina los datos al refrescar la página	40
Figura 3-14 Código JavaScript que hace captura de la herramienta web	41
Figura 3-15 <h1> del <body>	42
Figura 3-16 <div> de imágenes corporativas	42
Figura 3-17 <div> con botones para exportar datos en CSV y PDF e imagen de marca de agua ...	43
Figura 3-18 Estructura del <table> para el lunes a 1ª hora	44
Figura 3-19 Estructura HTML de las celdas de Deporte/Comida.....	45
Figura 3-20 Inicialización del <i>backend</i>	45
Figura 3-21 Cierre del <i>backend</i>	46
Figura 3-22 Diccionario de asignaturas.....	47
Figura 3-23 Diccionario que relaciona profesores con asignaturas.....	48
Figura 3-24 Diccionario de aulas ENM.....	49
Figura 3-25 Diccionario de tipos de sesión lectiva	49
Figura 3-26 Diccionario de días de la semana.....	50
Figura 3-27 Diccionario de franjas horarias	50
Figura 3-28 Ruta /menú.....	51

Figura 3-29 Ruta /desplegar_menú	52
Figura 3-30 Ruta /copiar.....	53
Figura 3-31 Ruta /pegar.....	54
Figura 3-32 Ruta /asignaturas.....	55
Figura 3-33 Ruta /selección_asignatura	56
Figura 3-34 Ruta /profesores	57
Figura 3-35 Ruta /selección_profesor	58
Figura 3-36 Ruta /aulas	59
Figura 3-37 Ruta /selección_aula	60
Figura 3-38 Ruta /sesión.....	61
Figura 3-39 Ruta /selección_sesión.....	62
Figura 3-40 Ruta /deporte_comida.....	63
Figura 3-41 Ruta /seleccionar_opción.....	63
Figura 3-42 Ruta /exportar_csv	64
Figura 3-43 Ruta /resetear_selecciones	65
Figura 3-44 Ruta /exportar_pdf.....	66
Figura 3-45 Solicitud GET de HTMX al servidor Flask.....	67
Figura 3-46 Respuesta del servidor Flask al navegador.....	67
Figura 3-47 Solicitud POST del navegador al servidor Flask.....	67
Figura 3-48 Respuesta del navegador.....	67
Figura 3-49 Solicitud POST del navegador al servidor para exportar en PDF	68
Figura 3-50 Respuesta del servidor con archivo PDF	68
Figura 4-1 Archivos necesarios para correr la herramienta.....	69
Figura 4-2 Ruta del servidor web local en el terminal	70
Figura 4-3 Apariencia de la herramienta web al ejecutar la ruta local.....	70
Figura 4-4 Menú en la celda.....	70
Figura 4-5 Listado de asignaturas dentro de la celda	71
Figura 4-6 Asignatura con profesores asociados.....	71
Figura 4-7 Diferencia entre asignaturas seleccionadas	72
Figura 4-8 Listado de profesores asociados a la asignatura	72
Figura 4-9 Celda con asignatura y profesor seleccionado.....	73
Figura 4-10 Listado de aulas ENM	73
Figura 4-11 Celda con asignatura, profesor y aula seleccionada	74
Figura 4-12 Listado de sesiones	74
Figura 4-13 Celda rellena al completo	75
Figura 4-14 Interfaz antes de pegar información	75

Figura 4-15 Celda al hacer clic sobre Pegar	75
Figura 4-16 Ejemplo de sábana rellena a exportar	76
Figura 4-17 Archivo CSV con la sábana devuelto por el servidor.....	76
Figura 4-18 Bloc de notas exportado con la sábana	76
Figura 4-19 PDF generado	77

ÍNDICE DE TABLAS

Tabla 2-1 Comparación de bibliotecas Python para lectura de archivos Excel	18
Tabla 3-1 Asignaturas 5° CGA - IM ST	26
Tabla 3-2 Profesores 5° CGA-IM ST	27
Tabla 3-3 POD (profesores - asignaturas).....	28
Tabla 3-4 Aulas ENM	30
Tabla 3-5 Titulación y grupos de clase.....	31
Tabla 3-6 Horario de la sábana para el lunes	32

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Antecedentes

La Escuela Naval Militar (ENM) es el centro docente donde se imparte la formación y los planes de estudio requeridos para los futuros oficiales de las distintas escalas y cuerpos de la Armada. Estos planes de estudios, para los alumnos sin titulación previa (ST) en específico, incluye un Título de Grado Universitario impartido por el Centro Universitario de la Defensa (CUD - ENM). Pero la ENM no solo cuenta con estudiantes ST del Cuerpo General de la Armada y del Cuerpo de Infantería de Marina, sino que también cuenta con alumnos con diferentes planes de estudio (Cuerpo de Intendencia de la Armada, Cuerpo de Ingenieros de la Armada, alumnos extranjeros, etc.).

Para impartir los planes de estudio en tiempo y forma, la ENM cuenta con un amplio número de aulas y de profesores. Para una organización más detallada, los cuerpos que cuentan con mayor número de alumnos cuentan a su vez con una subdivisión en grupos de clase. Todas estas variables de planes de estudio, aulas y profesores como en cualquier universidad u otro tipo de centro docente necesitan organizarse en un horario. En la ENM contamos con un horario semanal conocido en el ambiente de la escuela familiarmente como “La sábana”, en ella se reflejan todos los días de la semana con sus materias, tanto del Grado Universitario como las actividades y asignaturas militares, aulas, alumnos y profesores implicados en cada momento del día. Ésta es publicada en la aplicación Minerva al inicio de cada semana lectiva. Existen dos tipos de “sábana” que se repiten de manera rotativa: “sábana tipo A” y “sábana tipo B”. Aunque hay que tener en cuenta también las variaciones especiales que pueda haber en la sábana, tales como actos militares debido a la visita de una autoridad, semanas previas o posteriores a un permiso, semanas de exámenes finales, etc. Para estas semanas se crean sábanas únicas.

Tanto alumnos como profesores deben consultar diariamente la sábana para saber cuándo, dónde y con quién tendrán clase. Añadir que, además, deberán consultar también otro documento de obligatoria lectura diaria, denominado: “Orden Diaria de la ENM”, ya que en este documento se reflejarán los posibles cambios de clase que puedan suceder debido a imprevistos del día a día o errores de la sábana.

1.2 Motivación

En el ámbito de la ENM es conocido por todo el mundo que la realización de la sábana es francamente complejo y tedioso ya que hay que cuadrar las asignaturas del Grado Universitario, las materias militares y las actividades de instrucción y adiestramiento, añadiendo también los posibles imprevistos que puedan surgir debido a actos, conferencias, visitas, etc. Esto provoca que el creador de la sábana tenga que realizar numerosas modificaciones, llegando en ocasiones al extremo de tener que crear una sábana única/especial. Esto conlleva que en ciertas ocasiones se produzcan errores.

La complejidad de la sábana horaria en la ENM, derivada de la multitud de variables que intervienen en su planificación, incrementa la probabilidad de errores en la representación y coordinación de los datos. Estos fallos pueden generar confusiones en la distribución de asignaturas, aulas y profesores, afectando directamente tanto a los alumnos como al personal docente, quienes dependen de una planificación clara y precisa para el desarrollo de sus actividades. Además, la constante necesidad de realizar modificaciones y ajustes manuales convierte la gestión del horario en un proceso cada vez más laborioso y propenso a inconsistencias. Ante esta problemática, se hace evidente la necesidad de una herramienta digital, capaz de mejorar la organización de la sábana, reducir errores y facilitar su actualización de manera ágil y estructurada.

1.3 Objetivos

El objetivo principal de este proyecto es la creación de una herramienta web capaz de permitir la edición de la sábana de la Escuela Naval Militar de una forma menos arcaica y más simple, primero empezando por quinto curso sin titulación y posteriormente añadiendo el resto de los planes de estudios. La web por realizar debe ser intuitiva y sencilla tanto para los responsables de la gestión de la sábana, como para alumnos y profesores de la ENM.

Para lograr este objetivo principal se prevén los siguientes objetivos parciales:

- Elaborar estado del arte en el que se desarrolle y estudie sobre el estado actual de la sábana y las tecnologías existentes para tener en cuenta en el ámbito de desarrollo web.
- Creación de una base de datos que recoja asignaturas, profesores, horario, titulación y aulas.
- Confección de un sistema de gestión, modelado de datos y diseño de la interfaz de la herramienta.
- Desarrollar una aplicación web funcional que permita editar, guardar y exportar la sábana.
- Pruebas y evaluación de la herramienta web.

1.4 Estructura y apartados de la memoria

A continuación, se relaciona brevemente la organización de la información por apartados de este TFG:

- Capítulo 1: En el primer apartado de este TFG se introduce de manera breve el contexto de la sábana, la motivación de cambiar la forma de hacer la sábana y unos objetivos.
- Capítulo 2: En el segundo capítulo se hace una investigación, primero, de cómo se elabora la sábana actualmente, para entender los parámetros a tener en cuenta, y posteriormente se hace un estudio de las distintas tecnologías que podrían ser de ayuda para la realización de la herramienta web.
- Capítulo 3: En el tercer capítulo se desarrolla de forma detallada cuáles han sido las acciones tomadas para realizar la herramienta paso a paso, primero creando los archivos Excel que servirán como base de datos, el *frontend* y por último el *backend*.
- Capítulo 4: En el cuarto capítulo se exponen los resultados obtenidos tras toda la etapa de desarrollo de la plataforma.
- Capítulo 5: En el quinto capítulo se hace crítica con la herramienta desarrollada y se sacan tanto conclusiones como líneas futuras.
- Anexo I: En este anexo se exponen las implicaciones sociales, económicas y medioambientales que se han tenido en cuenta.
- Anexo II: En este anexo se incluye la reflexión moral ética y social realizada durante el proyecto.
- Anexo III y IV: En estos anexos se encuentra el código que ha sido necesario crear para el funcionamiento de la herramienta web.

2 ESTADO DEL ARTE

2.1 Formato de la sábana actual

En este apartado se explicará la forma en que se realiza la sábana de horarios actualmente.

2.1.1 *Sábana general*

El archivo que contiene los horarios de cada alumno y profesor se publica al comienzo de cada cuatrimestre. Se crea en formato de tabla mediante el programa Microsoft Excel. Cuenta con dos hojas, una hoja corresponde con la tabla de la semana tipo A y otra hoja con la tabla de la semana tipo B.

La composición de las tablas se divide en dos partes: la parte superior (eje de abscisas) y la parte lateral izquierda (eje de ordenadas). En la parte superior se representan las variables de las promociones y grupos de clase, y en la parte lateral de la izquierda se observan los días de la semana y las horas lectivas. Las promociones se dividen en columnas, estas mismas contienen una subdivisión en grupos de clase y están ordenadas de izquierda a derecha de más antigua a más moderna. Los días se dividen en filas, estas mismas contienen una subdivisión en franjas horarias y están ordenadas de arriba a abajo en orden cronológico. Las franjas horarias representadas son las reflejadas en el Horario General de Alumnos publicado cada cuatrimestre por Jefatura de Estudios de la ENM.

En cada celda de la tabla Excel observamos el nombre de la asignatura en la mitad superior, el número de aula en la esquina inferior derecha y en algunas asignaturas militares aparece el nombre del profesor en el espacio restante. El color de las celdas es específico de cada asignatura, de manera que se pueda diferenciar a simple vista el horario a seguir durante el día gracias al código de colores. Aunque no todas las asignaturas tienen color propio. Todo lo descrito se puede observar en la Figura 2-1

Figura 2-1 Recorte de la sábana general de la ENM

2.1.2 Planificación horaria de cada asignatura

Para organizarse dentro de cada asignatura los profesores tienen que crear un documento Excel en el que se reparten las horas de clase de cada semana de la asignatura. Este método solo se utiliza para asignaturas del Grado Universitario puesto que las asignaturas militares se gestionan de forma interna.

Esta tabla Excel contiene el número de horas de teoría, laboratorios y seminarios impartidas por cada profesor de materia. En la siguiente Figura 2-2 se observa un ejemplo.

PROFESOR/ES	SEMANA 13 (20-24 nov)				SEMANA 14 (27 nov - 1 dic)			
	T	P	S	TOTAL	T	P	S	TOTAL
Norberto Fernández García				0				0
Milagros Fernández Gavilanes	4			4		8		8
CC E. Delgado Franco				0	4			4
Martin Liz			8	8		8		8
				0				0
				0				0
TOTAL	4	0	8	12	4	8	8	20

Figura 2-2 Tabla reparto de horas Redes y Ordenadores (curso 2023/2024)

2.2 Lenguajes de programación

Un lenguaje de programación es un conjunto de instrucciones que los programadores utilizan para comunicarse con una computadora y darle órdenes que ejecuten tareas específicas. A través de estos lenguajes, se pueden escribir programas informáticos que realizan operaciones lógicas y matemáticas, entre otras. Los lenguajes de programación pueden clasificarse en diferentes tipos, como de bajo nivel (más cercanos al hardware) o de alto nivel (más cercanos al lenguaje humano), y son fundamentales en el desarrollo de software.

Para la elección de los lenguajes de programación hay que fijarse en cuales son los más acertados para el cumplimiento del objetivo del TFG, el desarrollo de una página web. Para ello primero es preciso hacer un estudio fijándose en los más usados por los programadores a nivel mundial, ya que estos

lenguajes son los que van a contar con más documentación, soporte y normalmente suelen ser los más sencillos de aprender en un corto período de tiempo.

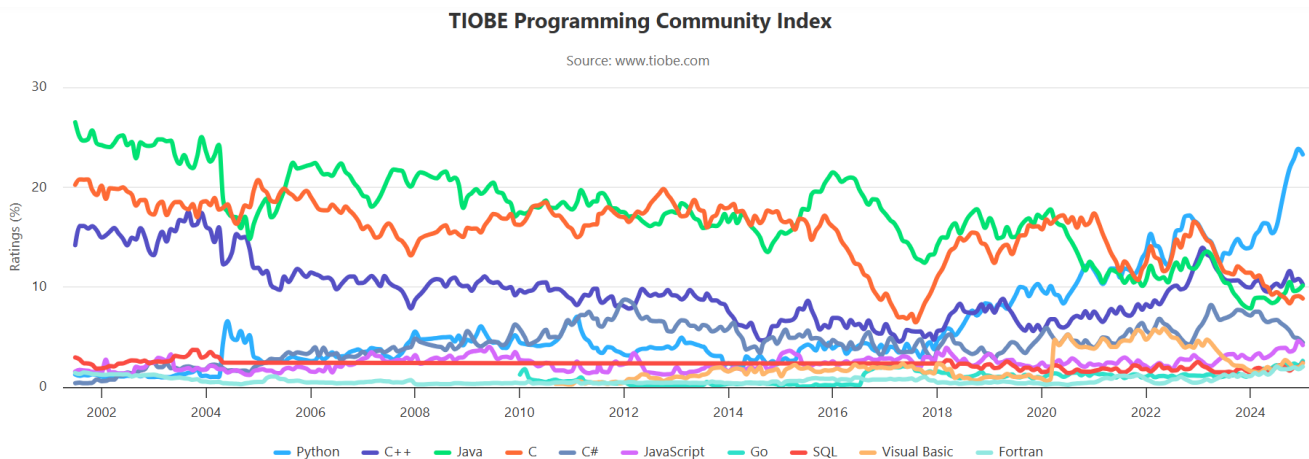


Figura 2-3 Índice de la comunidad de programadores TIOBE [1]

Atendiendo a la Figura 2-3, en la cual se aprecia el estudio del porcentaje de popularidad y uso de los lenguajes de programación a lo largo del tiempo por la comunidad TIOBE, podemos ver que hay un gran número de posibilidades de elección. A continuación, se explicarán los posibles lenguajes de programación que podrían ser elegidos para el desarrollo del trabajo.

2.2.1 Python

Desde su lanzamiento oficial en 1991 por Guido van Rossum, Python ha evolucionado drásticamente, convirtiéndose en uno de los lenguajes más populares y versátiles, utilizado en una amplia gama de aplicaciones, desde el desarrollo web hasta la inteligencia artificial. Python es conocido por su sintaxis sencilla y legible, lo que permite desarrollar código de manera eficiente y mantenerlo de forma clara y comprensible. Esta facilidad de uso es especialmente valiosa en proyectos académicos, donde la comprensión del código por parte de otros investigadores o colaboradores es crucial. Además, Python cuenta con una amplia variedad de bibliotecas y *frameworks* que facilitan el desarrollo de soluciones en diversas áreas, desde el análisis de datos hasta la inteligencia artificial, lo que lo convierte en una opción ideal para proyectos multidisciplinarios. La gran comunidad de desarrolladores que respalda Python también garantiza un constante flujo de recursos, documentación y soporte, lo que permite resolver problemas de manera ágil y optimizar el desarrollo. [2]

2.2.2 C++

C++ es un lenguaje de programación de propósito general desarrollado por Bjarne Stroustrup en 1979. Originalmente, Stroustrup lo creó como una extensión de C, incorporando características orientadas a objetos, lo que permitió a los programadores manejar la complejidad de los sistemas grandes con un enfoque modular. A lo largo de las décadas, C++ ha evolucionado y se ha consolidado como uno de los lenguajes más poderosos y eficientes, ampliamente utilizado en el desarrollo de software de alto rendimiento, videojuegos y aplicaciones de tiempo real. La principal ventaja de C++ radica en su capacidad para ofrecer un control detallado sobre los recursos del sistema, como la memoria, lo que lo hace especialmente útil en aplicaciones que requieren un uso intensivo de recursos. Además, C++ soporta tanto programación orientada a objetos (OOP) como programación de bajo nivel, lo que lo hace extremadamente versátil. A pesar de la aparición de lenguajes más modernos, C++ sigue siendo ampliamente utilizado en la industria debido a su alta eficiencia, su amplia comunidad y la extensa cantidad de bibliotecas disponibles para resolver tareas complejas. [3]

2.2.3 Java

Java es un lenguaje de programación de propósito general creado en 1991 por James Gosling y Mike Sheridan con el nombre de "Oak". Fue diseñado con el objetivo de ser un lenguaje simple, orientado a objetos y multiplataforma, ideal para aplicaciones en dispositivos electrónicos. En 1995, Java fue renombrado y lanzado al público, destacando por su característica de "escribir una vez, ejecutar en cualquier lugar", lo que significaba que el código compilado en Java podía ejecutarse en cualquier sistema operativo que tuviera la máquina virtual de Java (JVM). Esta característica hizo que Java se convirtiera en uno de los lenguajes más utilizados para el desarrollo de aplicaciones web, móviles y de software empresarial. A lo largo de los años, Java ha mantenido su relevancia debido a su robustez, seguridad y escalabilidad, siendo uno de los pilares en la industria del desarrollo de software. Su amplia comunidad de desarrolladores, la constante evolución del lenguaje y su ecosistema de bibliotecas y *frameworks* lo siguen posicionando como una de las opciones más elegidas por empresas y desarrolladores. [4]

2.2.4 C

C es un lenguaje de programación de alto nivel desarrollado en 1972 por Dennis Ritchie. C se destacó por su eficiencia y capacidad de manipular directamente la memoria, lo que lo hizo especialmente adecuado para escribir sistemas operativos y software de bajo nivel. A lo largo de los años, C ha evolucionado y sigue siendo uno de los lenguajes más influyentes en la programación. A pesar de la aparición de lenguajes más modernos, C mantiene su relevancia debido a su simplicidad, portabilidad y control sobre los recursos del sistema. Además, su influencia se extiende a otros lenguajes de programación, como C++, Java y Python, que incorporan elementos de C en sus propias sintaxis. El lenguaje C sigue siendo ampliamente utilizado en el desarrollo de sistemas operativos, controladores de hardware y aplicaciones donde se requiere un rendimiento máximo y un acceso eficiente a los recursos del sistema. [5]

2.2.5 JavaScript

JavaScript fue creado en 1995 por Brendan Eich. Inicialmente llamado Mocha y luego LiveScript, el lenguaje fue renombrado a JavaScript para aprovechar la popularidad de Java en ese momento, aunque ambos lenguajes son conceptualmente diferentes. JavaScript nació con la finalidad de agregar interactividad y dinamismo a las páginas web, funcionando directamente en el navegador del usuario. Desde su creación, ha evolucionado significativamente, convirtiéndose en uno de los lenguajes más utilizados en el desarrollo web. A través de los años, JavaScript ha ampliado sus capacidades más allá del simple manejo de eventos en el navegador, permitiendo la creación de aplicaciones web completas gracias a herramientas como Node.js. Esto ha permitido que JavaScript también se use en el desarrollo del lado del servidor. Su constante evolución, junto con el soporte de *frameworks* como React, Angular y Vue.js, ha consolidado a JavaScript como un pilar fundamental en la construcción de aplicaciones web modernas, lo que lo convierte en uno de los lenguajes más versátiles y populares del ecosistema tecnológico actual. [6]

2.3 Lenguajes de marcado

Un lenguaje de marcado es un tipo de lenguaje utilizado para estructurar, describir o dar formato a un documento de manera que los datos puedan ser fácilmente interpretados por diferentes aplicaciones. A diferencia de los lenguajes de programación, que ejecutan procesos y algoritmos, los lenguajes de marcado se enfocan en describir el contenido y la organización de los datos sin involucrar lógica de programación. Algunos ejemplos comunes son HTML, XML, y Markdown. Estos lenguajes permiten que los datos sean fácilmente accesibles y presentados de manera consistente. A continuación, se expondrán los posibles lenguajes de marcado seleccionables para el trabajo, en función de popularidad, sencillez y propósito.

2.3.1 HTML5

HTML es el lenguaje de marcado estándar utilizado para estructurar y presentar contenido en la web. Fue creado en 1991 por Tim Berners-Lee como parte del desarrollo de la World Wide Web. HTML permite a los desarrolladores definir la estructura de un documento web mediante el uso de etiquetas (Figura 2-4) que indican cómo debe organizarse y presentarse el contenido, como encabezados, párrafos, enlaces, imágenes y tablas. A lo largo de los años, HTML ha evolucionado significativamente con actualizaciones que introdujeron nuevas funcionalidades y mejoras para soportar multimedia, formularios interactivos y un diseño más accesible. Hoy en día, HTML sigue siendo el pilar fundamental para el desarrollo web y es utilizado junto con lenguajes como CSS y JavaScript para crear sitios web modernos, dinámicos y responsivos. [7] [8]

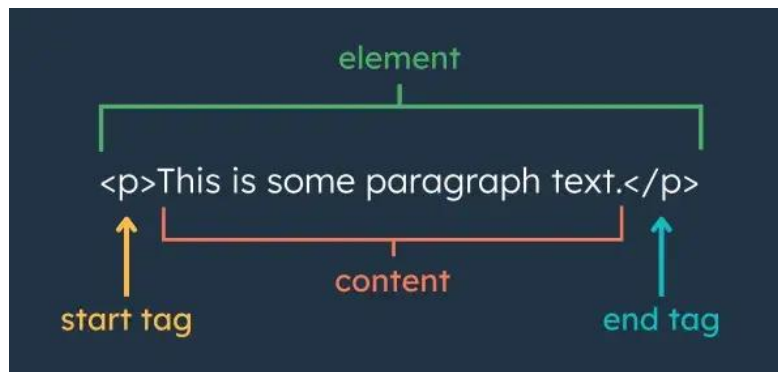


Figura 2-4 Ejemplo sencillo de lenguaje HTML [9]

2.3.2 XML

XML es un lenguaje de marcado extensible diseñado para almacenar y transportar datos de manera estructurada y legible tanto para humanos como para máquinas. Fue desarrollado por el World Wide Web Consortium (W3C) en 1998. A diferencia de HTML, que se utiliza para mostrar información en la web, XML se enfoca en describir y almacenar datos sin predefinir una presentación visual. Su principal característica es que permite a los usuarios definir sus propias etiquetas, lo que lo hace altamente flexible y adecuado para una variedad de aplicaciones, desde la transferencia de datos entre sistemas hasta la creación de documentos complejos. Hoy en día, XML sigue siendo utilizado en áreas como la integración de sistemas, el intercambio de datos en aplicaciones web. [10]

2.3.3 Markdown

Markdown es un lenguaje de marcado ligero creado en 2004 por John Gruber con la colaboración de Aaron Swartz. Su principal objetivo es facilitar la escritura de texto que se pueda convertir fácilmente en HTML, utilizando una sintaxis simple y legible. A diferencia de otros lenguajes de marcado más complejos como HTML o XML, Markdown permite crear documentos estructurados con una sintaxis muy sencilla que usa caracteres como asteriscos, guiones y almohadillas para definir elementos como negritas, listas y encabezados. Debido a su simplicidad y versatilidad, Markdown se ha convertido en un estándar popular en plataformas como GitHub, Reddit y en la creación de blogs y documentación técnica. Hoy en día, es ampliamente utilizado para generar contenido de manera rápida, accesible y sin necesidad de conocimientos técnicos avanzados. [11]

2.4 Bibliotecas de Python para manejo de archivos de Microsoft Excel

En el desarrollo de aplicaciones que requieren el procesamiento de datos estructurados, Python ofrece múltiples bibliotecas especializadas en la lectura, escritura y manipulación de archivos Excel. Estas herramientas permiten automatizar tareas que normalmente se realizarían de forma manual en Excel, como la creación de informes, la extracción de datos y la modificación de hojas de cálculo.

El uso de estas bibliotecas es fundamental en ámbitos como análisis de datos, automatización de procesos, generación de informes y gestión de bases de datos almacenadas en hojas de cálculo. A diferencia del uso tradicional de Excel, estas herramientas permiten trabajar con grandes volúmenes de datos de manera programática, sin necesidad de abrir la aplicación de Microsoft Excel.

Las principales bibliotecas para trabajar con archivos Excel en Python incluyen Openpyxl, Pandas y XlsxWriter, cada una con funcionalidades específicas que se adaptan a distintos escenarios de uso. En los siguientes apartados se describen las características más relevantes de cada una de ellas.

2.4.1 Openpyxl

La biblioteca Openpyxl fue creada en 2010 como una alternativa moderna para manejar archivos en el formato .xlsx de Microsoft Excel. Su desarrollo se centró en proporcionar una herramienta flexible para trabajar con hojas de cálculo sin necesidad de depender de software externo. Con el tiempo, se ha convertido en una de las bibliotecas más utilizadas para manipular archivos Excel en Python. Algunas de sus funcionalidades principales son: lectura y escritura de archivos Excel, modificación de celdas y hojas de trabajo, aplicación de formatos y estilos personalizados, inserción de imágenes y gráficos dinámicos y gestión de archivos Excel sin necesidad de instalar Microsoft Excel. [12]

2.4.2 Pandas

Pandas fue desarrollado inicialmente en 2008 por Wes McKinney con el objetivo de proporcionar herramientas avanzadas para el análisis y manipulación de datos estructurados. Aunque no fue creada específicamente para trabajar con archivos Excel, su módulo `pandas.read_excel()` y `DataFrame.to_excel()` la han convertido en una de las opciones más populares para analizar datos almacenados en hojas de cálculo. Como características principales cuenta con: lectura de archivos Excel, manipulación eficiente de datos tabulares, conversión de archivos Excel a otros formatos (CSV y JSON) y exportación de datos desde DataFrames a archivos Excel. [13]

2.4.3 XlsxWriter

Desarrollada en 2013 por John McNamara, XlsxWriter se diseñó con el propósito de generar archivos Excel en formato .xlsx de forma rápida y optimizada. A diferencia de Openpyxl, esta biblioteca no permite leer ni modificar archivos existentes, sino que está enfocada exclusivamente en la creación de nuevos archivos Excel con características avanzadas. Algunas de sus aportaciones son: creación de archivos Excel desde cero con múltiples hojas, aplicación de formatos avanzados y mejor rendimiento en la generación de archivos grandes en comparación con otras bibliotecas. [14]

A continuación, en la Tabla 2-1, se van a exponer algunas de las características interesantes y necesarias para el desarrollo del trabajo para diferenciar rápidamente entre estas librerías.

Característica	Openpyxl	Pandas	XlsxWriter
Capacidad de lectura de archivos Excel	✓	✓	✗
Capacidad de escritura de archivos Excel	✓	✓	✓
Edición de archivos Excel	✓	✗	✗
Manipulación y análisis de datos	✗	✓	✗

Soporte para la inserción de fórmulas de Excel	✓	✗	✓
Formateo avanzado (colores, estilos y fuentes)	✓	✗	✓
Soporte para gráficos e imágenes	✓	✗	✓
Enfoque principal	Lectura y modificación de archivos Excel	Análisis y manipulación de datos tabulares	Generación optimizada de archivos Excel con diseño avanzado

Tabla 2-1 Comparación de bibliotecas Python para lectura de archivos Excel

2.5 Otras herramientas útiles

En este apartado del estado del arte se expondrán sobre posibles herramientas que pueden ser de gran utilidad a la hora de gestionar los datos y crear la aplicación web. Las herramientas por estudiar son Excel, *framework* Flask, lenguaje de diseño CSS, HTMX, Jinja2 y la librería de Python ReportLab para generar documentos PDF.

2.5.1 Excel

Microsoft Excel es una de las herramientas de hojas de cálculo más utilizadas a nivel mundial para la gestión y análisis de datos. Fue lanzado por primera vez en 1985 por Microsoft convirtiéndose en un estándar en entornos empresariales, académicos y administrativos. A lo largo de los años, Excel ha evolucionado incorporando funciones avanzadas como fórmulas, tablas dinámicas, macros con VBA (*Visual Basic for Applications*) y herramientas de análisis de datos, lo que lo hace una solución versátil para la organización y manipulación de información. Su compatibilidad con formatos como CSV y XML permite la interoperabilidad con otros sistemas y lenguajes de programación, facilitando su uso en diversas aplicaciones, incluida la gestión de horarios académicos. Gracias a su interfaz intuitiva y su capacidad de automatización, Excel sigue siendo una referencia en el manejo de datos estructurados y continúa siendo ampliamente empleado en distintos sectores. [15]

2.5.2 Framework Flask

Flask es un *micro-framework* para el desarrollo de aplicaciones web en Python que fue creado por Armin Ronacher en 2010. A diferencia de otros *frameworks* más completos como Django, Flask fue diseñado con un enfoque minimalista, lo que significa que proporciona las herramientas esenciales para desarrollar aplicaciones web, dejando al desarrollador la libertad de añadir funcionalidades adicionales según las necesidades del proyecto. Flask sigue el principio de "no imponer convenciones", lo que lo hace altamente flexible y adecuado tanto para aplicaciones pequeñas como para proyectos más grandes y complejos. A lo largo de los años, Flask ha ganado popularidad debido a su simplicidad, facilidad de uso y la gran comunidad que lo respalda. Ofrece soporte para RESTful APIs (Interfaz de Programación de Aplicaciones), plantillas HTML a través de Jinja2, y un sistema de rutas flexible, entre otras características. Si bien Flask no incluye muchas de las herramientas que otros *frameworks* proporcionan por defecto, su diseño modular permite integrar fácilmente bibliotecas adicionales, como bases de datos y autenticación. Por estas razones, Flask se ha convertido en una excelente opción para desarrollar aplicaciones ligeras, prototipos rápidos o aplicaciones que requieren un control total sobre las decisiones de implementación. [16]

2.5.3 Lenguaje de diseño CSS

CSS (*Cascading Style Sheets*) es un lenguaje utilizado para definir la apariencia y presentación de documentos HTML. Fue creado en 1996 por Håkon Wium Lie en el W3C con el propósito de separar el contenido de su diseño, facilitando la creación de páginas web más accesibles y flexibles. Desde entonces, ha evolucionado con CSS2 (1998), que introdujo mejoras en la estructura del diseño, y CSS3 (1999 en adelante), que añadió funcionalidades avanzadas como *media queries*, transiciones, animaciones y diseño responsivo.

Actualmente, CSS permite controlar el color, tipografía, márgenes, espaciado y distribución de elementos en una página. Gracias a sistemas como Flexbox y CSS Grid, los desarrolladores pueden crear diseños complejos de manera eficiente y adaptable a distintos dispositivos. Además, CSS admite personalización avanzada con variables, efectos visuales como sombras y degradados, y compatibilidad con temas oscuros o dinámicos mediante el uso de *custom properties* y selectores. [17]

2.5.4 HTMX

HTMX es una biblioteca de JavaScript creada por Carson Gross y lanzada en 2020 con el objetivo de simplificar el desarrollo de aplicaciones web dinámicas sin necesidad de utilizar *frameworks* complejos de JavaScript.

Su origen se basa en la filosofía "*Hypermedia as the Engine of Application State*" (HATEOAS), que busca aprovechar al máximo las capacidades de HTML y HTTP en lugar de delegar la interactividad al *frontend* mediante grandes cantidades de código JavaScript. Desde su lanzamiento, HTMX ha ganado popularidad en la comunidad de desarrollo web, especialmente entre quienes buscan un enfoque minimalista, declarativo y centrado en el *backend* para crear aplicaciones interactivas.

Específicamente, HTMX es especialmente útil en aplicaciones web que requieren interactividad sin sobrecargar el *frontend* con código JavaScript innecesario. Su facilidad de integración con *backends* permite mejorar la experiencia del usuario al actualizar dinámicamente secciones específicas de la interfaz sin recargar la página completa. Además, al mantener la lógica de la aplicación en el *backend*, se facilita el mantenimiento del código y se optimiza el rendimiento de la web. Por ejemplo, en una aplicación para la gestión de horarios, HTMX puede utilizarse para cargar asignaturas de forma dinámica en una tabla, evitando la necesidad de escribir JavaScript adicional y reduciendo la carga del navegador. [18]

2.5.5 Jinja2

Jinja2 es un motor de plantillas para Python desarrollado por Armin Ronacher, quien lo lanzó por primera vez en 2008. Este motor permite la creación de plantillas HTML dinámicas que se pueden renderizar con datos en tiempo de ejecución, facilitando el desarrollo de aplicaciones web interactivas.

Su diseño se inspira en Django Templates. Con una sintaxis similar a Python, lo que lo hace intuitivo para los desarrolladores familiarizados con este lenguaje. Entre sus características clave se incluyen la capacidad de herencia de plantillas, filtros personalizados, macros, bloques de código reutilizables y un sistema de escape automático para prevenir ataques de inyección de código.

Jinja2 es ampliamente utilizado en *frameworks* web como Flask, donde se encarga de la renderización de vistas y facilita la separación entre la lógica del *backend* y la presentación en el *frontend*. Su integración con Flask permite a los desarrolladores crear aplicaciones modulares y escalables sin necesidad de gestionar manualmente la generación de contenido HTML.

Gracias a su eficiencia y versatilidad, Jinja2 se ha convertido en una herramienta esencial en el desarrollo de aplicaciones web dinámicas y basadas en datos. [19]

2.5.6 Librería ReportLab

ReportLab es una biblioteca de Python que permite la creación de documentos en formato PDF de manera programática. Creada en el año 2000, se ha consolidado como una herramienta robusta y versátil para la generación de informes, gráficos y documentos complejos. Su funcionalidad abarca desde la incorporación de textos e imágenes hasta la creación de tablas y gráficos dinámicos, lo que la convierte en una opción adecuada para aplicaciones que requieren la producción automatizada de documentos PDF. Además, su capacidad para integrarse con otras bibliotecas de Python amplía aún más sus posibilidades en el desarrollo de soluciones personalizadas. [20]

A continuación, se observa un ejemplo de representación simple haciendo uso de ReportLab en la Figura 2-5.



Figura 2-5 Ejemplo de elemento gráfico contenido en un documento PDF creado con ReportLab

2.6 Opciones disponibles para la gestión de horarios en la web

Antes de seleccionar las tecnologías específicas para la solución propuesta, es preciso analizar las distintas opciones disponibles para gestionar horarios de manera eficiente. A continuación, se presentan algunas de las principales aproximaciones encontradas.

2.6.1 Software de gestión de horarios

Existen diversas herramientas en línea que permiten la planificación y gestión de horarios de manera automatizada. Algunos ejemplos incluyen:

- **Untis:** Sistema de generación de horarios utilizado en múltiples instituciones educativas. [21]
En la Figura 2-6, se observa la interfaz de la aplicación Untis.
- **aSc TimeTables:** Software comercial que permite una programación flexible y ajustable a múltiples parámetros. [22]
- **OpenTimetable:** Proyecto de código abierto que facilita la creación y edición de horarios personalizados. [23]

Si bien estas herramientas son eficaces, suelen estar orientadas a instituciones con estructuras más estándar y no permiten una personalización profunda para necesidades específicas como las de la ENM.

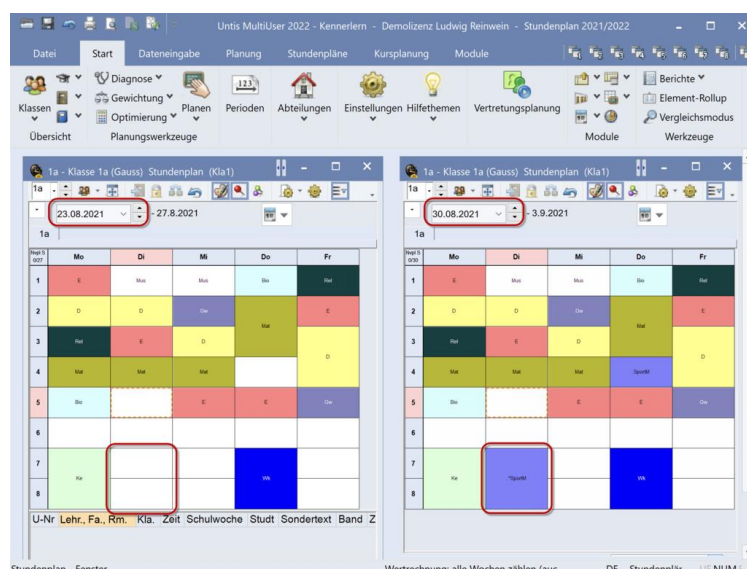


Figura 2-6 Ejemplo interfaz web Untis [24]

2.6.2 Algoritmos y metodologías utilizadas en la planificación de horarios

Los sistemas de gestión de horarios suelen emplear algoritmos de optimización para distribuir eficientemente clases, aulas y profesores. Por ejemplo, Untis utiliza algoritmos de programación lineal para generar horarios óptimos basados en restricciones predefinidas, mientras que aSc TimeTables incorpora heurísticas avanzadas y técnicas de búsqueda para minimizar conflictos en la planificación. Estas metodologías han demostrado ser efectivas en instituciones educativas al reducir errores y mejorar la asignación de recursos disponibles. Algunos enfoques comunes incluyen:

- Algoritmos genéticos: Modelos inspirados en la evolución que buscan optimizar el horario de forma progresiva.
- Programación lineal: Métodos matemáticos que buscan la mejor asignación de recursos.
- *Backtracking* y *Branch and Bound*: Estrategias utilizadas para encontrar la mejor combinación posible en sistemas con múltiples restricciones.

Estos métodos permiten reducir solapamientos y optimizar la asignación de clases. [25]

2.7 Trabajos de Fin de Grado previos

En el desarrollo del TFG, resulta relevante analizar otros estudios previos que han abordado temáticas relacionadas con la planificación y gestión de información en entornos educativos y militares.

En este sentido, se ha investigado sobre dos trabajos previos que presentan elementos de interés para la presente investigación, tanto en términos metodológicos como tecnológicos. A continuación, se analiza su contenido y su posible aportación al desarrollo de la herramienta web.

2.7.1 Desarrollo de una herramienta de apoyo a la planificación horaria de la ENM

Este TFG [26], plantea un enfoque estructurado para la gestión y organización de datos en un contexto académico dentro de la ENM. En particular, se destaca su uso de bases de datos y metodologías para la estructuración de la información. Uno de los aspectos más útiles de este estudio es su enfoque en la optimización de la administración de datos, lo cual puede ser aplicado a la lógica de almacenamiento y procesamiento de la sábana horaria. Además, la forma en que se presentan los datos en su interfaz puede servir de referencia para el diseño de la visualización de los horarios en la herramienta web.

La metodología utilizada en este TFG proporciona una base para comprender los desafíos que implica la integración de sistemas de información en un entorno académico-militar. Su aproximación a la recopilación y estructuración de datos puede ayudar a definir estrategias más útiles para la gestión de

horarios y la sincronización de asignaturas, profesores y aulas dentro del sistema que se está desarrollando.

2.7.2 Desarrollo automático de la sábana de la ENM-CUD

El TFG [27] se centra en el desarrollo de una solución tecnológica aplicada a la ENM, incorporando herramientas y lenguajes de programación que pueden ser útiles para la implementación de la herramienta web. Este trabajo ofrece un análisis detallado sobre la integración de sistemas en un entorno educativo, abordando cuestiones de escalabilidad y adaptabilidad, elementos clave para las mejoras de la herramienta de horarios. Aporta ideas valiosas sobre cómo estructurar el sistema para permitir su implementación en un servidor accesible a múltiples usuarios, lo que facilitaría su uso en la ENM más allá de una ejecución local.

3 DESARROLLO DEL TFG

3.1 Elección de tecnologías de trabajo

El presente apartado tiene como objetivo justificar las tecnologías utilizadas en el desarrollo de la herramienta web, explicando su propósito dentro del sistema y los criterios de selección.

3.1.1 Backend y gestión de datos

El *backend* es la parte que maneja la lógica y los datos. No es visible para los usuarios, pero es crucial porque gestiona la información que se muestra en el *frontend*.

- Python: será el lenguaje principal en el desarrollo del *backend*, gestionando la lógica de la aplicación y la interacción con los datos. Se selecciona por su sintaxis clara, facilidad de uso y amplia comunidad de desarrolladores, lo que facilita su mantenimiento.
- Flask: actuará como el *framework* web para gestionar las peticiones entre el *frontend* y el *backend*. Se ha optado por Flask debido a su ligereza, flexibilidad y facilidad de integración con otras tecnologías, lo que permite crear una API eficiente sin sobrecargar el sistema.
- Excel: se utilizará como base de datos ligera para almacenar y recuperar información sobre los horarios de clase. Se ha elegido este formato por su simplicidad, portabilidad y fácil integración con Python, evitando la complejidad de bases de datos más robustas innecesarias para el alcance del proyecto.

3.1.2 Frontend y diseño de la interfaz

El *frontend* es la parte de la aplicación con la que interactúan los usuarios. Incluye todo lo que se muestra en la pantalla, como botones, formularios, tablas, colores y estilos. En este caso será la tabla de horarios interactiva donde los usuarios podrán ver, modificar y gestionar las clases.

- HTML5: se empleará para estructurar el contenido de la aplicación web, definiendo los elementos de la interfaz de usuario. Su elección responde a ser el estándar actual de desarrollo web, permitiendo una estructura clara, semántica y adaptable a cualquier navegador.
- CSS: será utilizado para definir la apariencia visual de la herramienta, garantizando una experiencia de usuario atractiva y funcional. Se ha escogido por sus capacidades avanzadas en diseño responsivo, flexibilidad en personalización y compatibilidad con los estándares web.
- JavaScript (HTMX): JavaScript permitirá dotar de interactividad a la aplicación, facilitando la manipulación dinámica del contenido y mejorando la experiencia del usuario. Se ha elegido junto con HTMX debido a su capacidad para simplificar la interactividad en el *frontend* sin necesidad de JavaScript complejo ni *frameworks* adicionales. Permite realizar peticiones AJAX

(*Asynchronous JavaScript and XML*), actualizar elementos dinámicamente y mejorar la experiencia de usuario manteniendo una estructura ligera y eficiente en el *backend*.

3.1.3 Instalación del software necesario para la herramienta

Para poder desarrollar la herramienta se necesita cumplir con los siguientes requisitos y descargas:

- Tener un ordenador con el sistema operativo de Windows 7 o superior.
- Tener instalada la última versión de Python.
- Tener instalada la biblioteca Openpyxl para Python.
- Tener instalado el *framework* Flask para Python.
- Tener instalada la biblioteca ReportLab para Python.
- Disponer de los ficheros que componen la herramienta web.

3.1.3.1 Instalación de Python

Para instalar Python, es necesario acceder a la página web oficial [28] y descargar la versión compatible con el sistema operativo utilizado. Una vez descargado el instalador, se debe seguir el proceso de instalación correspondiente, asegurándose de incluir la opción que permite añadir Python al PATH del sistema para facilitar su uso desde la línea de comandos

3.1.3.2 Instalación de Openpyxl

Para instalar la biblioteca Openpyxl, que permite manipular archivos de Excel en formato .xlsx desde Python, se utiliza el gestor de paquetes *pip*. La instalación se realiza ejecutando el comando *pip install openpyxl* en la terminal o línea de comandos, lo que descargará e instalará automáticamente la biblioteca y sus dependencias. Una vez completada la instalación, se puede verificar su correcto funcionamiento importándola en un *script* de Python con *import openpyxl*.

3.1.3.3 Instalación de Flask

Por último, para instalar Flask es necesario utilizar el gestor de paquetes *pip*, incluido por defecto en las versiones más recientes de Python. La instalación se realiza ejecutando el siguiente comando en la terminal o línea de comandos: *pip install flask*. Este proceso descargará e instalará automáticamente Flask junto con sus dependencias. Para verificar que la instalación se ha realizado correctamente, se puede ejecutar *python -m flask --version*, lo que mostrará la versión instalada del *framework*.

3.1.3.4 Instalación de ReportLab

Para instalar ReportLab, una de las bibliotecas más utilizadas en Python para la generación de documentos PDF, es necesario ejecutar el siguiente comando en la terminal o en el entorno de desarrollo donde se ejecutará el proyecto: *pip install reportlab*. Este comando descargará e instalará automáticamente la biblioteca y sus dependencias. Permitiendo a los desarrolladores integrar la creación de archivos PDF de manera programática en sus aplicaciones.

3.2 Diseño de la aplicación web

Para empezar, se decide dividir la aplicación web a crear en tres grandes partes: creación de archivos Excel en formato CSV que servirán como base de datos ligera, diseño del *frontend* y diseño del *backend*.

Esta división proporciona diversas ventajas que optimizan el desarrollo y funcionamiento de la herramienta web. Al delegar el almacenamiento de datos a los archivos Excel, la lógica en el *backend* y la presentación en el *frontend*, se evita sobrecargar el servidor con procesos innecesarios. Esto permite una experiencia de usuario más fluida y un consumo de recursos optimizado.

Además, si surge un error o es necesario implementar una mejora, solo es preciso modificar una de las partes sin afectar al resto de la aplicación. Esta separación de responsabilidades hace que el desarrollo sea más ordenado.

Por último, la estructura en tres capas permite que cada componente de la aplicación se desarrolle de manera independiente, favoreciendo su mantenimiento y actualización. Esta organización modular facilita la depuración del código y posibilita futuras ampliaciones sin necesidad de realizar cambios en toda la arquitectura de la aplicación.

A continuación, se muestra en la Figura 3-1 un resumen gráfico de las partes del desarrollo

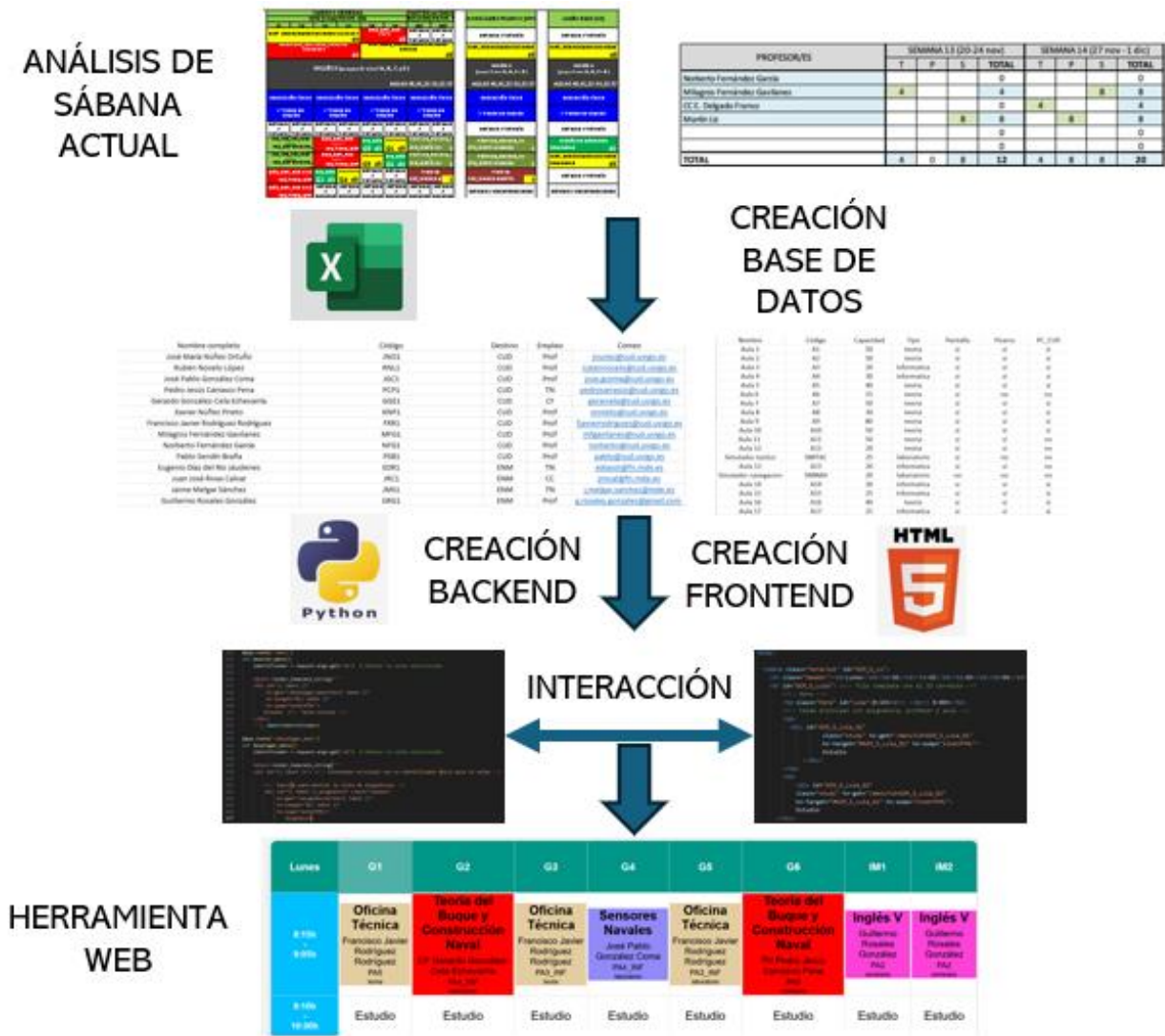


Figura 3-1 Resumen gráfico del desarrollo

3.2.1 Base de datos en archivos Excel

En este apartado se explica cuáles son los archivos que ha sido necesario crear para la correcta recolección de datos útiles en la realización de la herramienta web de la sábana. Estos son: asignaturas, aulas, horario, profesores, POD (relaciona profesores y asignaturas) y titulación.

3.2.1.1 Asignaturas

La Tabla 3-1 contiene información de forma detallada de las asignaturas que se imparten a los alumnos de quinto curso sin titulación de la Escuela Naval Militar.

- **Nombre completo:** Nombre descriptivo de la asignatura, según viene definido por el plan de estudios, facilitando la identificación por parte del usuario.
- **Código:** Identificador único asignado a cada asignatura, útil para la gestión en el *backend* y la vinculación con otros archivos como el de profesores.
- **Curso:** Curso académico en el que se imparte la asignatura, permitiendo en un futuro incluir asignaturas de distintos cursos.
- **Créditos:** Número de créditos ECTS asociados a la asignatura, determinando su carga lectiva dentro del plan formativo.
- **Titulación:** Identificación del plan de estudios al que pertenece la asignatura, en este caso, Sin Titulación (ST).
- **Código de fondo:** Color en formato RGB hexadecimal utilizado para la representación visual de la asignatura en la interfaz gráfica, lo que permite una rápida diferenciación dentro del horario a la hora de realizar la sábana.

Nombre Completo	Código	Curso	Créditos	Plan estudios	Color fondo
Sensores Navales	GIM_5_SN	5	6	ST	#382FA7
Teoría del Buque y Construcción Naval	GIM_5_TBCN	5	6	ST	#FF0101
Oficina Técnica	GIM_5_OT	5	6	ST	#E4CDA0
Fundamentos de Redes de Ordenadores	GIM_5_FRO	5	6	ST	#DF559
Táctica Naval III	GIM_5_TAC3	5	9	ST	#11FF05
Tiro Naval II	GIM_5_TN2	5	6	ST	#84F1FC
Comunicaciones III	GIM_5_COMMS3	5	3	ST	#DBB1DD
Inglés V	GIM_5_ING5	5	6	ST	#FD41D9
Operaciones Anfibas II	GIM_5_OA2	5	6	ST	#BF8F00
Táctica Anfibia IV	GIM_5_TAN4	5	6	ST	#F4B084
Artillería y Coordinación de Fuegos	GIM_5_ACF	5	6	ST	#D9D9D9
Comunicaciones II	GIM_5_COMMS2	5	6	ST	#C00000
Automóviles	GIM_5_AUT	5	6	ST	#8EA9DB

Tabla 3-1 Asignaturas 5º CGA - IM ST

El diseño de este archivo responde a la necesidad de almacenar y manipular la información académica de manera eficiente dentro de la aplicación. De esta forma se logra automatizar el proceso de asignación de asignaturas. Esto agiliza la configuración inicial del sistema y minimiza errores humanos en la introducción de datos, asegurando que la información se mantenga actualizada.

Además, la estructura flexible del archivo permite una gestión dinámica de las asignaturas. Cualquier cambio en el plan de estudios, como la edición o eliminación de asignaturas, se puede reflejar fácilmente con una simple actualización del archivo, sin requerir modificaciones en el código de la aplicación. Esto otorga una gran adaptabilidad al sistema, permitiendo su evolución en función de nuevas necesidades académicas.

Otra ventaja clave es la vinculación con otros módulos de la plataforma, como la asignación de profesores, aulas y sesiones. Gracias a la estructura del archivo, la aplicación puede relacionar automáticamente las asignaturas con los docentes y espacios correspondientes, asegurando la correcta integración dentro del sistema de gestión de horarios.

Por último, la información contenida en el archivo Excel permite optimizar la visualización del horario académico. El uso de códigos de colores predefinidos para cada asignatura facilita la diferenciación de materias dentro del horario interactivo, mejorando la experiencia del usuario y permitiendo una identificación rápida de las clases programadas.

3.2.1.2 Profesores

La Tabla 3-2 contiene toda la información de los profesores que imparten clases a los alumnos de quinto curso sin titulación de la Escuela Naval Militar.

Nombre completo	Código	Destino	Empleo	Correo
José María Núñez Ortuño	JNO1	CUD	Prof	████████@tud.uvigo.es
Rubén Nocelo López	RNL1	CUD	Prof	████████@tud.uvigo.es
José Pablo González Coma	JGC1	CUD	Prof	████████@tud.uvigo.es
Pedro Jesús Carrasco Pena	PCP1	CUD	TN	████████@tud.uvigo.es
Gerardo González-Cela Echevarría	GGE1	CUD	CF	████████@tud.uvigo.es
Xavier Núñez Nieto	XNN1	CUD	Prof	████████@tud.uvigo.es
Francisco Javier Rodríguez Rodríguez	FRR1	CUD	Prof	████████@tud.uvigo.es
Milagros Fernández Gavilanes	MFG1	CUD	Prof	████████@tud.uvigo.es
Norberto Fernández García	NFG1	CUD	Prof	████████@tud.uvigo.es
Pablo Sendín Raña	PSR1	CUD	Prof	████████@tud.uvigo.es
████████████████████	EDR1	ENM	TN	████████@fn.mde.es
████████████████████	JRC1	ENM	CC	████████@fn.mde.es
████████████████████	JAC1	ENM	TN	
████████████████████	JMS1	ENM	TN	████████@mde.es
Guillermo Rosales González	GRG1	ENM	Prof	████████@gmail.com
████████████████████	AGY1	ENM	CTE	████████@fn.mde.es
████████████████████	ACNI1	ENM	CTE	
Miguel Ángel Álvarez Feijoo	MAAF1	CUD	Prof	
████████████████████	MRM1	ENM	CAP	████████@fn.mde.es

Tabla 3-2 Profesores 5º CGA-IM ST

- **Nombre completo:** Contiene el nombre y los apellidos del profesor. Esta información es clave para identificar al docente dentro del sistema y vincularlo con sus asignaturas correspondientes.
- **Código:** Se asigna un código único a cada profesor (ejemplo: *JNO1* para José María Núñez Ortuño). Este código sirve como identificador dentro de la base de datos, lo que facilita la asignación automática de profesores a diferentes materias sin depender exclusivamente del nombre completo.
- **Destino:** Indica la institución a la que pertenece el profesor. En este caso, hay profesores asociados al **CUD - ENM (Centro Universitario de la Defensa)** y a la **ENM (Escuela Naval Militar)**. Este dato es fundamental para distinguir entre los profesores vinculados a ambos centros docentes, asegurando una correcta distribución del personal en la planificación académica.
- **Empleo:** Define el rango o cargo del profesor dentro de la institución. Puede ser *Prof* (Profesor), *TN* (Teniente de Navío), *CC* (Capitán de Corbeta), *CF* (Capitán de Fragata), *CTE* (Comandante), *CAP* (Capitán), entre otros. Este campo es especialmente relevante para gestionar asignaturas que pueden requerir la intervención de personal con formación y experiencia específicas.
- **Correo:** Se proporciona la dirección de correo electrónico del profesor. Este dato permite la comunicación entre el docente, los alumnos y la administración académica, facilitando la resolución de dudas y la coordinación de las clases.

Cuando este archivo se integra en el sistema, la gestión del personal docente se vuelve mucho más ágil y estructurada. En él se almacenan de manera clara y ordenada datos esenciales de cada profesor, desde su nombre y código identificador hasta su destino, rango o empleo, y correo electrónico. Gracias

a este formato, no es necesario introducir manualmente la información en la plataforma, lo que ahorra tiempo y reduce la posibilidad de errores.

Además, actualizar la plantilla docente resulta sencillo. Si un profesor se incorpora o deja de formar parte del equipo, basta con modificar el archivo y volver a cargarlo en el sistema. Este proceso inmediato evita configuraciones adicionales dentro de la aplicación, permitiendo que la información esté siempre al día sin complicaciones.

Otro aspecto fundamental de este archivo es su integración con los distintos módulos del sistema. Gracias a la información contenida en él, la plataforma puede vincular mediante un Excel del POD que relaciona a cada docente con las asignaturas que le corresponden. Así, las asignaciones se reflejan correctamente en los horarios, evitando conflictos y garantizando una planificación más organizada. Asimismo, la posibilidad de acceder rápidamente a los datos de contacto de los profesores favorece la comunicación entre el personal docente, los alumnos y la administración académica.

Por otro lado, la inclusión de un código único para cada profesor permite gestionar de manera más clara la asignación de sesiones y la distribución del horario. De este modo, se minimizan los solapamientos y se mantiene un control más detallado sobre la planificación académica. En definitiva, este archivo se convierte en una herramienta clave para estructurar y coordinar de manera eficaz la actividad docente dentro del sistema.

3.2.1.3 POD (profesores – asignaturas)

La Tabla 3-3 relaciona las asignaturas con los profesores de manera que dentro del *backend* se puedan incluir restricciones para que solo aparezcan a elegir los profesores de dichas asignaturas, evitando así errores manuales del usuario.

Nombre asignatura	Nombre profesor	Coordinador
Sensores Navales	José María Núñez Ortuño	SI
Sensores Navales	Rubén Nocelo López	NO
Sensores Navales	José Pablo González Coma	NO
Teoría del Buque y Construcción Naval	TN Pedro Jesús Carrasco Pena	SI
Teoría del Buque y Construcción Naval	CF Gerardo González-Cela Echevarría	NO
Oficina Técnica	Xavier Núñez Nieto	SI
Oficina Técnica	Francisco Javier Rodríguez Rodríguez	NO
Fundamentos de Redes de Ordenadores	Milagros Fernández Gavilanes	SI
Fundamentos de Redes de Ordenadores	Norberto Fernández García	NO
Fundamentos de Redes de Ordenadores	Pablo Sendín Raña	NO
Táctica Naval III	TN ████████████████████	SI
Tiro Naval II	CC ████████████████████	SI
Tiro Naval II	TN ████████████████████	NO
Comunicaciones III	TN ████████████████████	SI
Inglés V	Guillermo Rosales González	SI
Operaciones Anfibias II	CTE ████████████████████	SI
Táctica Anfibia IV	CTE ████████████████████	SI
Artillería y Coordinación de Fuegos	CAP ████████████████████	SI
Automóviles	Miguel Ángel Álvarez Feijoo	SI
Comunicaciones II	CTE ████████████████████	SI

Tabla 3-3 POD (profesores - asignaturas).

- **Nombre del profesor:** Nombre completo del profesor que facilita la vinculación con otros sistemas y bases de datos.
- **Nombre de la asignatura:** Nombre completo de cada materia, asegurando que no haya ambigüedades en la asignación.
- **Coordinador:** Indica si el profesor es el responsable de la materia, de esta manera se puede saber quién de los profesores va a tener más carga de trabajo y a quién poder hacer consultas en caso de dudas con la asignatura.

Gracias a este archivo, la plataforma es capaz de gestionar automáticamente la asignación de profesores a sus respectivas asignaturas, eliminando la necesidad de realizar esta tarea de forma manual dentro de la aplicación. Basta con cargarlo en el sistema para que cualquier actualización en la relación entre docentes y materias se refleje de inmediato, lo que permite adaptarse con rapidez a cambios en la plantilla o en la organización académica.

3.2.1.4 Aulas

La Tabla 3-4 representa todas las aulas que son usadas por alumnos en la Escuela Naval Militar, todas estas han sido extraídas del libro 14 Botones Manual del Aspirante [29].

Nombre	Código	Ubicación	Capacidad	Tipo	Pantalla	Pizarra	PC_CUD
Aula 1	A1	Patio de Aulas	50	teoría	Sí	Sí	Sí
Aula 2	A2	Patio de Aulas	50	teoría	Sí	Sí	Sí
Aula 3	A3_INF	Patio de Aulas	20	informática	Sí	Sí	Sí
Aula 4	A4_INF	Patio de Aulas	30	informática	Sí	Sí	Sí
Aula 5	A5	Patio de Aulas	40	teoría	Sí	Sí	Sí
Aula 6	A6	Patio de Aulas	15	teoría	Sí	No	No
Aula 7	A7	Patio de Aulas	50	teoría	Sí	Sí	Sí
Aula 8	A8	Patio de Aulas	30	teoría	Sí	Sí	Sí
Aula 9	A9	Patio de Aulas	80	teoría	Sí	Sí	Sí
Aula 10	A10	Patio de Aulas	50	teoría	Sí	Sí	Sí
Aula 11	A11	Patio de Aulas	50	teoría	Sí	Sí	No
Aula 12	A12	Patio de Aulas	20	teoría	Sí	Sí	No
Simulador táctico	SIMTAC	Patio de Aulas	25	laboratorio	Sí	No	No
Aula 13	A13_INF	Patio de Aulas	20	informática	Sí	Sí	No
Simulador navegación	SIMNAV	Patio de Aulas	20	laboratorio	No	No	No
Aula 14	A14_INF	Patio de Aulas	30	informática	Sí	Sí	Sí
Aula 15	A15_INF	Patio de Aulas	25	informática	Sí	Sí	Sí
Aula 16	A16	Patio de Aulas	40	teoría	Sí	Sí	Sí
Aula 17	A17_INF	Patio de Aulas	25	informática	Sí	Sí	Sí
Aula 18	A18_INF	Patio de Aulas	25	informática	Sí	Sí	Sí
Aula 19	A19	Patio de Aulas	30	teoría	Sí	Sí	No
Aula 20	A20	Patio de Aulas	30	teoría	Sí	Sí	No
Aula 40	A40	Isaac Peral	50	teoría	Sí	Sí	Sí
Aula 41	A41	Isaac Peral	40	teoría	Sí	No	Sí
Aula 42	A42_INF	Isaac Peral	50	informática	Sí	Sí	Sí
Aula 43	A43_INF	Isaac Peral	40	informática	Sí	Sí	Sí
Aula 44	A44_INF	Isaac Peral	30	informática	Sí	Sí	Sí
Aula 45	A45	Isaac Peral	60	teoría	Sí	Sí	Sí
Laboratorio química	LAB_QUI	Isaac Peral	25	laboratorio	No	Sí	No
Laboratorio física	LAB_FIS	Isaac Peral	25	laboratorio	No	Sí	No
Laboratorio auxiliares	LAB_AUX	Isaac Peral	25	laboratorio	No	Sí	No
Laboratorio turbinas	LAB_TUR	Isaac Peral	25	laboratorio	No	Sí	No
Laboratorio motores	LAB_MOT	Isaac Peral	25	laboratorio	No	Sí	No

Laboratorio materiales	LAB_MAT	Isaac Peral	25	laboratorio	No	Sí	No
Aula 50	A50	Isaac Peral	50	teoría	Sí	Sí	Sí
Aula 51	A51	Isaac Peral	50	teoría	Sí	Sí	Sí
Aula 52-54	A52_54	Isaac Peral	80	teoría	Sí	Sí	Sí
Aula 53	A53	Isaac Peral	50	teoría	Sí	Sí	Sí
Aula 55-57	A55_57	Isaac Peral	80	teoría	Sí	Sí	Sí
Aula 56	A56	Isaac Peral	80	teoría	Sí	Sí	Sí
Laboratorio electricidad	LAB_ELE	Isaac Peral	25	laboratorio	No	Sí	Sí

Tabla 3-4 Aulas ENM

- **Nombre del aula:** Nombra el espacio físico donde se imparten las clases, permitiendo una rápida referencia dentro del sistema.
- **Código del aula:** Un identificador único que evita duplicidades y facilita la integración con otros módulos de la plataforma.
- **Ubicación:** Indica la zona o edificio en el que se encuentra el aula, lo que permite una mejor organización de los espacios educativos.
- **Capacidad:** Especifica el número máximo de alumnos que pueden ocupar el aula simultáneamente, asegurando una correcta distribución de los estudiantes y la sobreasignación de alumnos.
- **Tipo de aula:** Define si se trata de un aula para dar clases de teoría, laboratorio, aula de informática, simulador, seminarios u otro tipo de espacio especializado, facilitando la asignación adecuada según las necesidades de cada materia.
- **Disponibilidad (pizarra, PC_CUD, pantalla) :** Indica si el aula está habilitada para su correcto uso o si presenta restricciones, lo que permite evitar conflictos en la planificación horaria.

Incorporar este archivo en el sistema supone un gran avance en la gestión de las aulas dentro de la plataforma. Su estructura organizada permite almacenar información esencial, como el nombre de cada aula, su identificador único y cualquier característica adicional relevante. Gracias a ello, se elimina la necesidad de introducir manualmente estos datos en la aplicación, lo que no solo agiliza el proceso, sino que también minimiza posibles errores.

La asignación y actualización de aulas se vuelve más sencilla y dinámica. Ante cualquier cambio en la infraestructura, ya sea la incorporación de nuevas aulas o ajustes en la disponibilidad de las existentes, basta con modificar el archivo y cargarlo nuevamente en el sistema. Esto permite que la distribución de espacios se actualice de manera inmediata sin necesidad de realizar modificaciones manuales dentro de la aplicación.

Además, este archivo no opera de manera aislada, sino que se integra con otros módulos de la plataforma, lo que garantiza una mejor organización del horario académico.

Por otro lado, el uso de identificadores únicos para cada aula resulta clave en la administración académica, ya que evita confusiones y posibles errores en la asignación de espacios. De esta forma, el sistema puede gestionar con precisión la distribución de las clases, asegurando que cada aula se utilice de manera eficiente y ordenada dentro del calendario académico.

3.2.1.5 Titulación

La Tabla 3-5 recoge los grupos de clase que serán representados en la sábana, en este caso los grupos pertenecientes al quinto curso del Grado de Ingeniería Mecánica del Cuerpo General de la Armada e Infantería de Marina.

Nombre completo	Código	Curso	Cuerpo	Ingreso	Grupos
Grado ingeniería mecánica	GIM	5	CGA	ST	G1
Grado ingeniería mecánica	GIM	5	CGA	ST	G2

Grado ingeniería mecánica	GIM	5	CGA	ST	G3
Grado ingeniería mecánica	GIM	5	CGA	ST	G4
Grado ingeniería mecánica	GIM	5	CGA	ST	G5
Grado ingeniería mecánica	GIM	5	CGA	ST	G6
Grado ingeniería mecánica	GIM	5	CIM	ST	IM1
Grado ingeniería mecánica	GIM	5	CIM	ST	IM2

Tabla 3-5 Titulación y grupos de clase

- **Nombre completo:** Nombra el plan de estudios que cursa el alumno, pudiéndose posteriormente añadir el resto de los planes de estudios.
- **Código:** Un identificador único que evita duplicidades y facilita la integración con otros módulos de la plataforma.
- **Curso:** Indica el curso al que pertenece el alumno dentro del plan de estudios, posteriormente se podrían añadir el resto de los cursos.
- **Cuerpo:** Especifica el cuerpo al que pertenecen los alumnos de dicho grupo, en este caso siendo CGA (Cuerpo General de la Armada) y CIM (Cuerpo de Infantería de Marina). Más tarde se podrían añadir CINA (Cuerpo Intendencia de la Armada) y CIA (Cuerpo de Ingenieros de la Armada).
- **Ingreso:** Expresa la forma de ingreso de los alumnos, pudiendo ser ST (Sin Titulación previa) o CT (Con Titulación previa), este último plan de estudios se podría añadir más adelante.
- **Grupos:** Código identificador que hace referencia a un grupo de alumnos, dependiendo del tamaño de la promoción será de entre unas 10-14 personas. Este identificador se mostrará en la parte superior de la sábana.

La incorporación de este documento se vuelve esencial para organizar de manera clara y estructurada la distribución de los alumnos en el sistema académico. Más allá de ofrecer una simple clasificación por plan de estudios, curso y grupo, su utilidad radica en su capacidad para integrarse con otros módulos de la plataforma, lo que permite una asignación automatizada de asignaturas, una correcta distribución de las aulas y una planificación horaria coherente y sin solapamientos.

Uno de los elementos más destacados de este archivo es su capacidad para diferenciar con precisión los distintos planes y cuerpos que cursan su estudios en la Escuela Naval Militar. Gracias a un código único para cada plan de estudios, se evita la redundancia en los datos y se garantiza que cada estudiante sea agrupado correctamente. Además, la distinción entre cuerpos, como el Cuerpo General de la Armada (CGA) y el Cuerpo de Infantería de Marina (CIM), facilita la personalización de la planificación académica, permitiendo en el futuro ampliar la clasificación para incluir otras especialidades, como el Cuerpo de Intendencia de la Armada (CINA) o el Cuerpo de Ingenieros de la Armada (CIA).

Otro aspecto clave que contempla este archivo es la forma en la que los alumnos acceden al sistema académico. Se establece una diferenciación entre aquellos que ingresan sin titulación previa (ST) y aquellos que ya poseen una formación universitaria (CT). Este criterio resulta fundamental para ajustar los itinerarios formativos de manera adecuada, asegurando que cada grupo reciba la formación específica que requiere su perfil de ingreso.

En cuanto a la organización interna, la segmentación por grupos de alumnos juega un papel fundamental en la planificación académica. Cada grupo, conformado generalmente por entre 10 y 14 estudiantes, cuenta con un identificador propio, lo que facilita su ubicación dentro de la sábana horaria y contribuye a que la información se muestre de manera ordenada y comprensible dentro del sistema.

En definitiva, este archivo no solo simplifica la administración académica, sino que también establece una base sólida para futuras ampliaciones y mejoras en la plataforma. Su capacidad para estructurar los datos de manera clara y su flexibilidad para adaptarse a cambios lo convierten en una

pieza clave dentro del desarrollo del sistema, garantizando que la gestión de los grupos de alumnos sea precisa, organizada y fácilmente escalable.

3.2.1.6 Horario

La Tabla 3-6 recoge las franjas horarias correspondientes a los lunes, reflejadas tanto en la sábana como en el Horario General de Alumnos de la Escuela Naval Militar escrito y promulgado por el Capitán de Fragata Jefe de Estudios.

Día	Número	Inicio	Fin	Código
Lunes	1	8:15	9:05	Lu1
Lunes	2	9:10	10:00	Lu2
Lunes	3	10:20	11:10	Lu3
Lunes	4	11:15	12:05	Lu4
Lunes	5A	12:10	13:00	Lu5A
Lunes	EFB	12:15	13:05	LuEFB
Lunes	EFA	13:10	14:00	LuEFA
Lunes	TCB	13:20	14:10	LuTCB
Lunes	TCA	14:15	15:05	LuTCA
Lunes	5B	14:30	15:20	Lu5B
Lunes	6	15:30	16:20	Lu6
Lunes	7	16:25	17:15	Lu7
Lunes	8	17:20	18:10	Lu8
Lunes	9	18:15	19:00	Lu9

Tabla 3-6 Horario de la sábana para el lunes

- **Día:** Indica el día específico de la semana en el que se desarrolla una sesión, permitiendo una correcta distribución de la carga académica a lo largo de la semana.
- **Inicio/Fin:** Define el intervalo de tiempo en el que se desarrolla cada sesión, asegurando la correcta planificación de las actividades sin solapamientos.
- **Número:** Establece el orden cronológico de las horas a lo largo del día, siendo el 1 la primera hora y la 9 la última hora
- **Código horario:** Reúne la información referente al horario en un código de manera que posteriormente se pueda referenciar en los *scripts*.

La tabla de horarios es una pieza central dentro del sistema, ya que permite estructurar la planificación académica.

Uno de sus principales valores radica en la capacidad de organizar la distribución de las sesiones en función de múltiples criterios. Cada franja horaria y cada día de la semana están definidos con precisión, lo que facilita la visualización del horario y su posterior gestión.

Este archivo también resulta fundamental para la correcta distribución de la carga docente y la gestión de los recursos disponibles, pudiendo visualizar si alguna asignatura, profesor o grupo de alumnos se excede en horas asignadas y carga lectiva.

Además de su utilidad en la planificación inicial, este archivo aporta una gran flexibilidad a la gestión académica. Los cambios en la programación, como ajustes en el horario, pueden implementarse de manera rápida y sencilla. Basta con modificar el documento y recargarlo en la plataforma para que las actualizaciones se apliquen de inmediato sin necesidad de intervención manual en la aplicación.

En definitiva, este archivo no solo facilita la organización del horario académico, sino que se convierte en una herramienta imprescindible para garantizar que el funcionamiento del sistema sea ágil, preciso y adaptable. Su estructura clara y su capacidad de integración permiten gestionar la asignación de clases de forma dinámica, asegurando que cada recurso se utilice de la mejor manera posible.

3.2.2 Frontend

El *frontend* de la aplicación ha sido desarrollado con **HTML**, **CSS** y **HTMX**, enfocándose en la simplicidad, interactividad y eficiencia en la actualización de contenido sin necesidad de recargar la página.

3.2.2.1 Configuración y estructura del <head>

Uno de los elementos fundamentales en la configuración del *frontend* es la correcta definición del encabezado (<head>), que contiene información clave para el funcionamiento de la aplicación. Dentro de este apartado, se ha incluido la codificación de caracteres **UTF-8** mediante la etiqueta <meta charset="UTF-8">. Esto garantiza que el sistema pueda interpretar y mostrar correctamente caracteres especiales, evitando problemas de visualización en distintos idiomas y asegurando una mayor compatibilidad a nivel internacional.

Otro aspecto esencial es la configuración del **viewport**, establecida a través de la etiqueta <meta name="viewport" content="width=device-width, initial-scale=1.0">. Esta configuración permite que la interfaz web se adapte correctamente a distintos tamaños de pantalla, desde ordenadores hasta dispositivos móviles. Gracias a esta propiedad, la experiencia de usuario se mantiene fluida e intuitiva sin importar el dispositivo desde el que se acceda a la plataforma.

Por último, se ha integrado **HTMX**, una biblioteca que simplifica la manipulación del contenido sin necesidad de utilizar JavaScript de forma explícita. Su inclusión en el sistema se realiza a través de <script src="https://cdnjs.cloudflare.com/ajax/libs/htmx/1.9.6/htmx.min.js"> lo que permite a la aplicación actualizar secciones específicas de la interfaz sin requerir una recarga completa de la página. Esta funcionalidad mejora notablemente la interactividad y el rendimiento del sistema, facilitando una experiencia de usuario más ágil y eficiente.

A continuación, se muestra el código descrito previamente en la Figura 3-2.

```

1 <!doctype html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <script src="https://cdnjs.cloudflare.com/ajax/libs/htmx/1.9.6/htmx.min.js"></script>
7 </head>

```

Figura 3-2 <head> HTML

3.2.2.2 Configuración y estructura de los estilos CSS

El diseño y apariencia de la aplicación web han sido desarrollados con hojas de estilo en cascada (CSS), permitiendo una presentación visual clara, atractiva y funcional. La personalización de la interfaz facilita la experiencia de usuario y mejora la organización de la información en la plataforma. A continuación, se detallan los principales aspectos del código CSS implementado en el sistema.

3.2.2.2.1 Diseño general y estilización del cuerpo

Desde el inicio, se ha definido una estructura global que garantiza una correcta presentación en cualquier dispositivo. El estilo del cuerpo de la página (*body*), que se observa en la Figura 3-3, emplea una tipografía estándar y de fácil lectura como Arial, acompañado de un fondo en color gris claro para reducir la fatiga visual. Además, se ha aplicado un margen alrededor del contenido y una alineación central para asegurar una distribución equilibrada de los elementos en pantalla.

```

9 <style type="text/css">
10
11 /* ===== Estilos Generales ===== */
12 body {
13     font-family: 'Arial', sans-serif; /* Fuente limpia y legible */
14     background-color: #f4f4f4; /* Color de fondo gris claro para mejorar la legibilidad */
15     margin: 20px; /* Espaciado alrededor del contenido */
16     text-align: center; /* Alinea el texto del cuerpo al centro */
17 }

```

Figura 3-3 Estilos CSS generales

3.2.2.2 Estilización de la tabla de horarios

Uno de los elementos clave dentro de la interfaz es la tabla de horarios, la cual se ha diseñado con un ancho del 90% para mantener una visualización óptima. Se han eliminado los espacios entre los bordes de las celdas (*border-collapse: collapse*) y se han aplicado detalles en esquinas (*border-radius: 10px*) para un acabado más moderno. Para mejorar la estética y legibilidad, se ha añadido un fondo blanco y una ligera sombra que proporciona un efecto de elevación sobre el fondo.

Cada celda de la tabla (*td*) ha sido configurada con un tamaño uniforme para mantener una distribución ordenada del contenido. Además, se han incorporado transiciones en la propiedad *background-color* y *transform*, lo que permite un efecto visual atractivo cuando el usuario pasa el cursor sobre ellas, aumentando ligeramente su tamaño para resaltar la información. Todo esto se observa en la Figura 3-4.

```

40 /* ===== Efecto Hover en las Celdas ===== */
41 table.horarios td:hover {
42     background-color: #e0e0e0; /* Cambia el color de fondo al pasar el ratón */
43     transform: scale(1.05); /* Aumenta ligeramente el tamaño */
44 }
45
46 /* ===== Estilo para las Celdas de Hora ===== */
47 table.horarios td.Hora {
48     background-color: #00BFFF; /* Color azul celeste */
49     color: white; /* Texto en blanco para contraste */
50     font-weight: bold; /* Texto en negrita */
51 }

```

Figura 3-4 Estilos CSS de la tabla de horarios

3.2.2.2.3 Estilización de las celdas de encabezados y horas

El diseño de los encabezados (*tr.Header td*) busca diferenciar de manera clara las secciones de la tabla. Para ello, se ha aplicado un fondo de color verde azulado y una tipografía en negrita de mayor tamaño, mejorando la visibilidad de los títulos. De manera similar, las celdas que contienen información sobre las horas de clase (*td.Hora*) han sido resaltadas con un fondo azul celeste y texto en blanco, permitiendo que los usuarios identifiquen rápidamente los bloques de tiempo dentro del horario. A continuación, se muestra el código que establece estos estilos en la Figura 3-5.

```

46  /* ===== Estilo para las Celdas de Hora ===== */
47  table.horarios td.Hora {
48      background-color: #00BFFF; /* Color azul celeste */
49      color: white; /* Texto en blanco para contraste */
50      font-weight: bold; /* Texto en negrita */
51  }
52
53  /* ===== Encabezado de la Tabla ===== */
54  tr.Header td {
55      background-color: #009688; /* Color de fondo verde azulado */
56      color: white; /* Texto en blanco */
57      font-size: 12pt; /* Aumenta el tamaño de la fuente */
58      font-weight: bold; /* Hace el texto más llamativo */
59      padding: 10px; /* Añade espacio dentro de las celdas del encabezado */
60  }

```

Figura 3-5 Estilos CSS de las franjas horarias y grupos de clase

3.2.2.2.4 Estilización de botones interactivos

Los botones juegan un papel fundamental en la aplicación, ya que permiten la selección de asignaturas, profesores, aulas y sesiones. Para su diseño, se ha utilizado un fondo en color verde con bordes redondeados y transiciones suaves en su interacción, brindando una experiencia fluida y moderna. Asimismo, se ha creado una variante de botones en color rojo (*button.delete*) para aquellas opciones que requieren eliminación de datos, asegurando que el usuario pueda diferenciarlos de manera intuitiva.

Para mejorar la interactividad, los botones cuentan con animaciones de entrada (*@keyframes fadeIn*), lo que genera una aparición progresiva en pantalla y contribuye a una experiencia visual más agradable. Este tipo de animaciones sutiles refuerzan los estilos de la aplicación. Estos estilos se describen en el *frontend* en la Figura 3-6.

```

62  /* ===== Estilos de los Botones ===== */
63  button {
64      padding: 8px 12px; /* Espaciado interno */
65      margin: 5px; /* Espaciado exterior */
66      border: none; /* Elimina bordes */
67      border-radius: 5px; /* Bordes redondeados */
68      background-color: #4CAF50; /* Color verde */
69      color: white; /* Texto en blanco */
70      cursor: pointer; /* Cambia el cursor a puntero */
71      transition: background 0.3s ease; /* Efecto de cambio de color al pasar el ratón */
72  }
73
74  button:hover {
75      background-color: #45a049; /* Color más oscuro al pasar el ratón */
76  }
77
78  /* ===== Botón para eliminar selección ===== */
79  button.delete {
80      background-color: #d9534f; /* Rojo */
81  }
82
83  button.delete:hover {
84      background-color: #c9302c; /* Rojo más oscuro al pasar el ratón */
85  }
86
87  /* ===== Animaciones y Transiciones ===== */
88  @keyframes fadeIn {
89      from {
90          opacity: 0; /* Comienza con opacidad cero */
91          transform: translateY(-5px); /* Se mueve ligeramente hacia arriba */
92      }
93      to {
94          opacity: 1; /* Se muestra completamente */
95          transform: translateY(0); /* Regresa a su posición normal */
96      }
97  }
98
99  button {
100     animation: fadeIn 1s ease-in-out; /* Aplica la animación de aparición */
101 }

```

Figura 3-6 Estilos CSS de los botones de selección

3.2.2.2.5 Estilización de los botones copiar y pegar

El contenedor `.copy-paste-controls` posiciona estos botones en la parte inferior derecha de cada celda con un pequeño espacio entre ellos.

El botón de copiar utiliza un fondo azul (`#2F00FF`) y el de pegar un fondo azul (`#007BFF`), ambos con texto blanco, bordes redondeados y un tamaño compacto (12px). Además, cuentan con un efecto *hover* que aumenta su brillo (`filter: brightness(1.2);`), mejorando la interactividad. Se muestra en la Figura 3-7.

Estos estilos garantizan que los controles sean visibles, intuitivos y funcionales, optimizando la experiencia del usuario en la gestión de datos.

```

87     /* Contenedor de los botones */
88     .copy-paste-controls {
89         bottom: 5px;
90         right: 5px;
91         gap: 5px;
92     }
93
94     /* Botón de Copiar */
95     .copy-paste-controls button:first-child {
96         background-color: #2f00ff; /* Naranja */
97         color: white;
98         border: none;
99         border-radius: 5px;
100        padding: 5px 10px;
101        font-size: 12px;
102        cursor: pointer;
103    }
104
105    /* Botón de Pegar */
106    .copy-paste-controls button:last-child {
107        background-color: #007bff; /* Azul */
108        color: white;
109        border: none;
110        border-radius: 5px;
111        padding: 5px 10px;
112        font-size: 12px;
113        cursor: pointer;
114    }
115
116    /* Efecto Hover para hacerlos más visibles */
117    .copy-paste-controls button:hover {
118        filter: brightness(1.2);
119    }

```

Figura 3-7 Estilos CSS de botones copiar y pegar

3.2.2.2.6 Jerarquización del texto y diferenciación de elementos

La estructura del sistema requiere que ciertos elementos tengan mayor relevancia que otros dentro de la interfaz. En este sentido, se han definido distintos tamaños de fuente según el tipo de contenido:

- **Asignaturas (.subject):** Se ha empleado una fuente más grande y en negrita para destacar la importancia de la materia que se imparte en cada bloque horario.
- **Profesores (.teacher):** Se utiliza un tamaño intermedio que permite su correcta identificación sin opacar la asignatura.
- **Aulas (.classroom) y sesiones (.session):** Se han establecido tamaños de fuente menores, ya que su información es complementaria dentro de la celda.
- **Estudio (.study):** Se usa un tamaño de letra adecuado de manera que su lectura sea a primera vista, puesto que aparece predefinido en la celda sin ningún otro elemento alrededor.

De igual manera, el título principal de la página (.titulo) ha sido diseñado con una tipografía destacada y un subrayado en color azul, logrando que se mantenga como un elemento visual clave en la interfaz. Todo esto se observa en la Figura 3-8.

```

103 /* ===== Tamaños de Fuente Diferenciados ===== */
104 .subject {
105     font-size: 20px; /* Mayor tamaño para la asignatura */
106     font-weight: bold; /* En negrita */
107     margin-bottom: 5px; /* Espaciado inferior */
108 }
109
110 .teacher {
111     font-size: 14px; /* Tamaño intermedio para profesor */
112     margin-bottom: 3px; /* Espaciado inferior */
113 }
114
115 /* ===== Estilo para el Título Principal ===== */
116 .titulo {
117     font-family: 'Arial', sans-serif; /* Fuente profesional */
118     font-size: 28px; /* Tamaño llamativo */
119     font-weight: bold; /* Texto en negrita */
120     color: #2c3e50; /* Azul oscuro elegante */
121     text-align: center; /* Centrado */
122     margin-top: 20px; /* Espaciado superior */
123     margin-bottom: 20px; /* Espaciado inferior */
124     padding-bottom: 10px; /* Espaciado interno */
125     border-bottom: 3px solid #2980b9; /* Subrayado con color azul */
126     text-transform: uppercase; /* Convierte el texto en mayúsculas */
127     letter-spacing: 1px; /* Espaciado entre letras */
128 }
129
130 /* ===== Estilo para Aula ===== */
131 .classroom {
132     font-size: 12px; /* Tamaño menor para aula */
133     margin-bottom: 2px; /* Espaciado inferior */
134 }
135
136 /* ===== Estilo para Sesión ===== */
137 .session {
138     font-size: 10px; /* Tamaño más pequeño para sesiones */
139 }
140
141 /* ===== Estilo para el Contenido de las Celdas (Estudio) ===== */
142 .study {
143     font-size: 14pt; /* Tamaño adecuado para lectura */
144 }

```

Figura 3-8 Estilos CSS de los elementos <div> de las celdas

3.2.2.2.7 Botones de exportación de datos

Para facilitar la gestión de la información, como se muestra Figura 3-9 se han incorporado dos botones de exportación (*.export-btn*) que permiten la descarga del horario en formato CSV y PDF. Su diseño se basa en un color azul vibrante, acompañado de un efecto de cambio de tonalidad al pasar el cursor. Con estos botones, los usuarios pueden obtener una copia de la planificación de clases de manera rápida y sencilla.

```

146 /* ===== Estilo para el Botón de Exportar CSV ===== */
147 .export-btn {
148     background-color: #007bff; /* Azul vibrante */
149     color: white; /* Texto en blanco */
150     border: none; /* Sin borde */
151     padding: 10px 20px; /* Espaciado interno */
152     font-size: 16px; /* Tamaño de fuente cómodo */
153     border-radius: 5px; /* Bordes redondeados */
154     cursor: pointer; /* Cursor de puntero */
155     margin-top: 20px; /* Espaciado superior */
156 }
157
158 .export-btn:hover {
159     background-color: #0056b3; /* Azul más oscuro al pasar el ratón */
160 }

```

Figura 3-9 Estilos CSS de los botones de exportación de datos en CSV y PDF

3.2.2.2.8 Estilos de imágenes y elementos gráficos

Para mejorar la disposición y el impacto visual de los elementos gráficos, se ha incorporado un contenedor de imágenes (`.contenedor`). Este elemento utiliza `display: flex;` para alinear las imágenes horizontalmente, asegurando que queden centradas tanto en el eje horizontal como en el vertical (`justify-content: center;` y `align-items: center;`). Además, se ha agregado un espacio entre las imágenes mediante `gap: 20px;`, lo que evita que queden demasiado juntas y mejora la estética general. Para hacer que este contenedor resalte dentro del diseño, se ha aplicado un fondo semitransparente (`background: rgba(255, 255, 255, 0.8);`), bordes redondeados (`border-radius: 10px;`) y un efecto de sombra (`box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);`), lo que le da un aspecto moderno y bien integrado con el resto de la interfaz.

En cuanto a las imágenes en sí, se han diseñado con la clase `.imagen`, que establece un ancho reducido (`width: 200px;`) para garantizar que ocupen un espacio controlado dentro de la página, manteniendo su proporción mediante `height: auto;`. Para armonizar con el diseño general, se han aplicado bordes redondeados (`border-radius: 10px;`) y un efecto de sombra (`box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.3);`), lo que contribuye a su integración visual sin resultar invasivo. Lo descrito anteriormente, se observa en la Figura 3-10.

```

161  /* Contenedor de las imágenes */
162  .contenedor {
163      display: flex;
164      justify-content: center; /* Centra horizontalmente */
165      align-items: center; /* Alinea verticalmente */
166      gap: 20px; /* Espacio entre imágenes */
167      padding: 20px; /* Espaciado interno */
168      background: rgba(255, 255, 255, 0.8); /* Fondo semitransparente para destacar imágenes */
169      border-radius: 10px; /* Bordes redondeados */
170      box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2); /* Sombra suave */
171  }
172
173  /* Estilo de las imágenes reducidas */
174  .imagen {
175      width: 200px; /* Tamaño más pequeño */
176      height: auto; /* Mantiene la proporción */
177      border-radius: 10px; /* Bordes redondeados */
178      box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.3); /* Sombra ligera */
179  }

```

Figura 3-10 Estilos CSS de imágenes del encabezado

Por otro lado, en la Figura 3-11, se ha incorporado un pequeño icono de marca de agua con la clase `.watermark`, el cual está destinado a acompañar el botón de exportación de datos. Este elemento ha sido diseñado con un tamaño reducido (`width: 30px;`), una opacidad disminuida (`opacity: 0.5;`), y un filtro de escala de grises (`filter: grayscale(100%);`) para evitar que distraiga la atención del usuario mientras mantiene una presencia sutil dentro de la interfaz. Su finalidad es aportar un detalle distintivo sin sobrecargar visualmente el diseño.

```

180  .watermark {
181      width: 30px; /* Tamaño reducido */
182      height: auto;
183      opacity: 0.5; /* Efecto de marca de agua */
184      filter: grayscale(100%); /* Hace que la imagen sea menos llamativa */
185  }

```

Figura 3-11 Estilo CSS de marca de agua

3.2.2.2.9 Estilización de las celdas Deporte/Comida

Los estilos para `.lunch`, que se observan en la Figura 3-12, mejoran la estética y usabilidad de las celdas de Deporte/Comida, aplicando un fondo azul mar (`#0077b6`), texto blanco y bordes redondeados

para mayor claridad. Se añade un efecto *hover*, cambiando el tono a #005f87 y aumentando ligeramente el tamaño, mejorando la interacción del usuario. Además, la sombra sutil (*box-shadow*) aporta profundidad.

```

234     .lunch {
235         background-color: #0077b6; /* Azul mar */
236         color: white; /* Texto en blanco */
237         font-weight: bold;
238         text-align: center;
239         padding: 10px;
240         border-radius: 8px; /* Bordes redondeados */
241         font-size: 14px;
242         transition: all 0.3s ease-in-out; /* Animación suave */
243         cursor: pointer;
244         box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2); /* Sombra sutil */
245     }
246
247     /* Efecto hover: hace que la celda resalte un poco */
248     .lunch:hover {
249         background-color: #005f87; /* Azul más oscuro al pasar el cursor */
250         transform: scale(1.05); /* Aumenta ligeramente el tamaño */
251     }
252

```

Figura 3-12 Estilos CSS celdas Deporte/Comida

3.2.2.3 Script para el restablecimiento de datos al recargar la página web

Dentro de la implementación del sistema, se ha incorporado un mecanismo para garantizar que los datos almacenados en memoria se reinicien automáticamente cada vez que el usuario recarga la página. Esto se logra mediante el fragmento de código JavaScript mostrado en la Figura 3-13.

```

188     <script>
189         window.addEventListener("load", function() {
190             fetch("/reset_selecciones", { method: "POST" }); // Borra SELECCIONES al cargar la página
191         });
192     </script>

```

Figura 3-13 Código JavaScript que elimina los datos al refrescar la página

Esta función se ejecuta tan pronto como la página ha terminado de cargarse, gracias al evento "load", que se activa automáticamente en el objeto *window*. Al detectarse este evento, se realiza una solicitud HTTP de tipo POST a la ruta */reset_selecciones*, la cual está definida en el *backend* para vaciar la estructura de datos SELECCIONES. De esta manera, cada vez que el usuario refresca la página, se elimina cualquier selección previa almacenada en memoria, asegurando que la plataforma comience desde un estado limpio.

Esta funcionalidad es particularmente útil en el contexto de la aplicación, ya que evita la acumulación de datos de sesiones anteriores, lo que podría generar inconsistencias en la asignación de horarios, aulas o profesores. Al implementar este reinicio automático, se mejora la integridad del sistema y se proporciona una experiencia de usuario más predecible y ordenada.

Además, este enfoque permite mantener la aplicación ligera en términos de memoria, ya que no se almacenan datos innecesarios entre recargas. Gracias a esta optimización, el sistema se mantiene ágil y evita posibles errores derivados de datos obsoletos que podrían permanecer en SELECCIONES de una sesión anterior, provocando que se exporten datos de otra sesión al exportar los datos en un archivo CSV.

3.2.2.4 Script para capturar en imagen el estado de la herramienta web

Este *script* en JavaScript permite la exportación de la vista actual de la página en formato PDF de manera automática. Para ello, utiliza *html2canvas* para capturar la pantalla con alta resolución (*scale*:

2), asegurando que los elementos visuales se mantengan con calidad. Además, permite la carga de imágenes externas mediante *useCORS: true*.

Una vez generada la imagen, esta se convierte a base64 y se envía al servidor Flask mediante una solicitud *fetch* a la ruta */exportar_pdf*, donde se procesa y se integra en un documento PDF. Al recibir la respuesta del servidor, el *script* genera un enlace temporal para descargar automáticamente el archivo como "sabana.pdf". Se muestra en la Figura 3-14.

Esta implementación facilita la generación de documentos PDF de forma rápida, garantizando que los usuarios puedan exportar la información visualizada en la aplicación sin complicaciones.

```

241 <script src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/1.4.1/html2canvas.min.js"></script>
242 <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>
243 <script src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/1.4.1/html2canvas.min.js"></script>
244
245 <script>
246   document.addEventListener("DOMContentLoaded", function() {
247     let botonExportar = document.getElementById("exportar-pdf");
248
249     if (botonExportar) {
250       botonExportar.addEventListener("click", function() {
251         console.log("Iniciando captura de pantalla...");
252
253         html2canvas(document.body, {
254           scale: 2, // Mejor resolución
255           useCORS: true, // Permitir imágenes externas (CORS)
256           allowTaint: true, // Forzar carga de imágenes externas
257         }).then(canvas => {
258           console.log("Captura de pantalla realizada.");
259
260           let imgData = canvas.toDataURL("image/png");
261
262           // Enviar imagen a Flask
263           fetch("/exportar_pdf", {
264             method: "POST",
265             body: JSON.stringify({ image: imgData }),
266             headers: { "Content-Type": "application/json" }
267           })
268             .then(response => response.blob())
269             .then(blob => {
270               let link = document.createElement("a");
271               link.href = URL.createObjectURL(blob);
272               link.download = "sabana.pdf";
273               link.click();
274               console.log("PDF descargado.");
275             })
276             .catch(error => console.error("Error en Fetch:", error));
277         });
278       });
279     } else {
280       console.error("Error: No se encontró el botón exportar-pdf.");
281     }
282   });
283 </script>

```

Figura 3-14 Código JavaScript que hace captura de la herramienta web

3.2.2.5 Configuración y estructura del <body>

La etiqueta <body> es un elemento fundamental dentro de la estructura de un documento web, ya que contiene todo el contenido visible para el usuario en el navegador.

El tag <body> define la estructura principal de la página web, albergando elementos como encabezados (<h1> - <h6>), párrafos (<p>), imágenes (), enlaces (<a>), listas (, ,), tablas (<table>), formularios (<form>) y cualquier otro contenido interactivo o visual que componga la experiencia de usuario. Todo lo que se presenta en la pantalla cuando un usuario accede a un sitio web se encuentra dentro de esta sección.

En el caso de la herramienta web, al comenzar el <body> del *frontend* se encuentra el encabezado de nivel 1 (<h1 class="titulo">Sábana General Alumnos 5º Curso CGA-IM ST</h1>) que representa el título principal de la interfaz, estableciendo de manera clara y concisa el propósito de la sección en la que se encuentra. Su denominación indica que la información contenida en este apartado está destinada a la gestión y visualización de los datos académicos de los alumnos de quinto curso del programa CGA-

IM ST. Desde una perspectiva estructural, este título no solo jerarquiza el contenido dentro del documento HTML, sino que también facilita la identificación de la funcionalidad específica que aborda la plataforma. El uso de la clase "titulo" en este encabezado permite la aplicación de estilos personalizados mediante CSS, asegurando coherencia estética con el resto de la interfaz.

```
196 <body>
197 <h1 class="titulo">Sábana General Alumnos 5º Curso CGA-IM ST</h1>
```

Figura 3-15 <h1> del <body>

Avanzando en el *frontend*, se encuentra un fragmento de código que un papel importante en la organización visual de la interfaz, particularmente en la presentación de imágenes representativas del entorno académico y de identidad institucional.

El fragmento de código corresponde a un contenedor definido con la etiqueta <div> y la clase "contenedor", dentro del cual se incluyen tres imágenes representativas de la identidad institucional del Centro Universitario de la Defensa. La función principal de esta sección es reforzar visualmente la identidad de la plataforma, aportando un valor simbólico y un contexto inmediato a los usuarios que interactúan con la interfaz.

Cada imagen es insertada utilizando la etiqueta , la cual define un recurso gráfico en la web. Se establecen los atributos *src*, que indica la URL de la imagen, y *alt*, que proporciona una descripción textual alternativa en caso de que la imagen no pueda ser cargada o para mejorar la accesibilidad. En este caso, las imágenes que se muestran en la Figura 3-16 incluyen:

1. El logotipo del Centro Universitario de la Defensa, permitiendo una identificación visual inmediata con la institución.
2. Una imagen complementaria de la Escuela Naval Militar donde se ubica el CUD - ENM, reforzando su identidad dentro del contexto académico y militar.
3. La bandera de España, que representa el carácter oficial e institucional de la entidad.

Desde una perspectiva de diseño y usabilidad, el uso de una clase CSS "contenedor" sugiere que la disposición y el estilo de estas imágenes serán gestionados mediante hojas de estilo, permitiendo un mayor control sobre su alineación, tamaño y comportamiento dentro de la estructura de la página. La clase "imagen" aplicada a cada también sugiere una estrategia de diseño coherente, donde todas las imágenes incluidas en este apartado mantendrán una apariencia homogénea dentro del conjunto visual de la plataforma.

```
307 
308 
309 
```

Figura 3-16 <div> de imágenes corporativas

Al final del *frontend* nos encontramos con un bloque de interacción. Este bloque cumple una doble función: proporcionar dos botones para la exportación de datos y presentar una imagen de marca utilizada como *watermark*.

Los elementos <button> incluidos en este bloque tienen la clase "export-btn", el botón para exportar en CSV está dotado de un evento *onclick* que redirige al usuario a la ruta *'/exportar_csv'*. Esta funcionalidad permite la generación y descarga de un archivo en formato CSV que contendrá la información del horario del usuario.

La exportación a CSV es una opción ampliamente utilizada en plataformas de gestión de datos académicos, ya que permite a los estudiantes y docentes almacenar y manipular los horarios con herramientas externas como Microsoft Excel o Google Sheets.

Desde el punto de vista técnico, el evento `onclick="window.location.href='/exportar_csv'"` utiliza JavaScript para redirigir al usuario a una URL específica cuando se pulsa el botón. Esto significa que, en el *backend*, existe una ruta programada que procesa la solicitud, extrae la información del horario desde la base de datos y la transforma en un archivo CSV para su descarga.

El elemento `` incluido dentro del `<div>` tiene la clase `"watermark"`, lo que indica que su propósito es visual. La imagen que se carga proviene de un enlace interno y muestra un bonsái, lo que aporta una visión estética y de branding para la herramienta web. Todo este código viene recogido en la Figura 3-17.

```

2957 <div>
2958 <button onclick="window.location.href='/exportar_csv'" class="export-btn">
2959   Exportar Horario a CSV
2960 </button>
2961 <button id="exportar-pdf" class="export-btn"> Exportar Sesión a PDF</button>
2962
2963 
2964 </div>

```

Figura 3-17 `<div>` con botones para exportar datos en CSV y PDF e imagen de marca de agua

3.2.2.5.1 Estructura y funcionalidad del `<table>`

Esta tabla, identificada con la clase `"horarios"` y el atributo `id="GIM_5_Lu"`, está específicamente diseñada para organizar de manera clara y sistemática la distribución de las actividades académicas a lo largo del lunes, el resto de los días cuenta con su identificador propio en el código. Su estructura sigue una jerarquía bien definida, facilitando la presentación de información de forma accesible.

La tabla inicia con una fila de encabezado (`<tr class="Header">`) que contiene las etiquetas para cada columna. La primera columna indica el día de la semana, en este caso `"Lunes"`, mientras que las siguientes corresponden a los distintos grupos de estudio identificados como G1, G2, G3, G4, G5, G6, IM1 e IM2. Este diseño permite la segmentación por grupos, asegurando que cada estudiante pueda identificar con rapidez su asignación horaria correspondiente.

A continuación, la tabla se compone de múltiples filas (`<tr>`) representando diferentes franjas horarias del día. Cada una de estas filas está identificada con un `id` único que combina el día de la semana y el número de sesión, como `GIM_5_Lu1a`, `GIM_5_Lu2a`, etc. La primera celda de cada fila contiene la franja horaria específica en la que se imparte la actividad, utilizando la clase `"Hora"` y el formato de presentación `hh:mmh - hh:mmh` con saltos de línea (`
`) para mejorar la legibilidad.

Dentro de cada fila, las celdas correspondientes a los distintos grupos incluyen un contenedor `<div>` con una `id` única. Estos contenedores están definidos con la clase `"study"` y emplean la tecnología HTMX, mediante los atributos `hx-get`, `hx-target` y `hx-swap`. HTMX permite realizar solicitudes dinámicas sin necesidad de recargar la página, mejorando así la interactividad del sitio web. En este caso, la estructura de la solicitud `hx-get="/menu?id=GIM_5_Lu1a_G1"` indica que, al interactuar con la celda correspondiente, se obtendrá dinámicamente información desde el servidor acerca de la actividad asignada a ese grupo en esa franja horaria. El atributo `hx-target` especifica el destino de la actualización del contenido, mientras que `hx-swap="innerHTML"` permite reemplazar el contenido de la celda con la información recibida.

El contenido por defecto de cada celda es la palabra `"Estudio"`, lo que significa que, en ausencia de una asignación específica, ese espacio horario se destina al estudio autónomo o actividades de refuerzo académico. Sin embargo, mediante la actualización dinámica proporcionada por HTMX, es posible que estas celdas muestren información adicional como asignaturas, profesores asignados o aulas específicas, dependiendo de los datos que se obtengan desde el servidor.

Desde el punto de vista del diseño y usabilidad, esta tabla presenta una solución eficiente para la visualización de horarios en entornos académicos, permitiendo a los gestores acceder de forma rápida y sencilla a la información relevante de la planificación semanal. Además, la implementación de

identificadores únicos en cada celda y fila asegura que la estructura sea fácilmente manipulable mediante HTMX.

```

204 <tr class="Header"><td>Lunes</td><td>G1</td><td>G2</td><td>G3</td><td>G4</td><td>G5</td><td>G6</td><td>IM1</td><td>IM2</td></tr>
205 <tr id="GIM_5_Lu1a"> <!-- Fila completa con el ID correcto -->
206 <!-- Hora -->
207 <td class="Hora" id="Lu1a">8:15h<br/> -<br/> 9:05h</td>
208 <!-- Celda principal con asignatura, profesor y aula -->
209 <td>
210 <div id="GIM_5_Lu1a_G1"
211 <div class="study" hx-get="/menu?id=GIM_5_Lu1a_G1"
212 <div hx-target="#GIM_5_Lu1a_G1" hx-swap="innerHTML">
213 <div Estudio
214 <div </div>
215 </div>
216 </td>
217 <td>
218 <div id="GIM_5_Lu1a_G2"
219 <div class="study" hx-get="/menu?id=GIM_5_Lu1a_G2"
220 <div hx-target="#GIM_5_Lu1a_G2" hx-swap="innerHTML">
221 <div Estudio
222 <div </div>
223 </div>
224 </td>
225 <td>
226 <div id="GIM_5_Lu1a_G3"
227 <div class="study" hx-get="/menu?id=GIM_5_Lu1a_G3"
228 <div hx-target="#GIM_5_Lu1a_G3" hx-swap="innerHTML">
229 <div Estudio
230 <div </div>
231 </div>
232 </td>
233 <td>
234 <div id="GIM_5_Lu1a_G4"
235 <div class="study" hx-get="/menu?id=GIM_5_Lu1a_G4"
236 <div hx-target="#GIM_5_Lu1a_G4" hx-swap="innerHTML">
237 <div Estudio
238 <div </div>
239 </div>
240 </td>
241 <td>
242 <div id="GIM_5_Lu1a_G5"
243 <div class="study" hx-get="/menu?id=GIM_5_Lu1a_G5"
244 <div hx-target="#GIM_5_Lu1a_G5" hx-swap="innerHTML">
245 <div Estudio
246 <div </div>
247 </div>
248 </td>
249 <td>
250 <div id="GIM_5_Lu1a_G6"
251 <div class="study" hx-get="/menu?id=GIM_5_Lu1a_G6"
252 <div hx-target="#GIM_5_Lu1a_G6" hx-swap="innerHTML">
253 <div Estudio
254 <div </div>
255 </div>
256 </td>

```

Figura 3-18 Estructura del <table> para el lunes a 1ª hora

La Figura 3-18 muestra la estructura general de las tablas que ve el usuario al abrir el enlace web para rellenar la sábana. En cuanto a estructura, las tablas del resto de los días de la semana no varían exceptuando la tabla del viernes debido a que cuenta con una hora menos por la tarde. Y en cuanto a contenido del código, lo que varía entre los distintos días de la semana, franjas horarias y grupos de clase es cómo se estructura el identificador de cada celda de la sábana.

Lo único que modifica la lógica del código que se ve en la Figura 3-18 en cada día, es cuando se llega a la mitad del día y llegan la hora de deporte y de la comida, para ello se incluye el siguiente código HTML, se muestra en la Figura 3-19, que permite la presentación para interactuar con rutas en el *backend* del servidor.

```

614 <tr id="GIM_5_LuTC1">
615 <td class="Hora" id="LuTC1">13:10h<br/> -<br/> 14:10h</td>
616 <td>
617 <div id="GIM_5_LuTC1_G1"
618 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1"
619 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
620 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
621 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
622 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
623 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
624 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
625 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
626 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
627 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
628 <div class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1" hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">

```

Figura 3-19 Estructura HTML de las celdas de Deporte/Comida

3.2.3 Backend

El *backend* de la aplicación se desarrolla utilizando Flask. Este fragmento de código reúne las importaciones necesarias para gestionar la funcionalidad del *backend* en Flask, integrando herramientas esenciales para la manipulación de archivos, la gestión de datos y la generación de documentos PDF.

```

1 import os, openpyxl, csv # Importamos módulos para interactuar con el sistema de archivos (os) y manejar archivos
2
3 from flask import Flask, request, render_template_string, Response, send_file, jsonify # Importamos funciones de Flask
4 from collections import defaultdict
5 from reportlab.lib.pagesizes import landscape, A4
6 from reportlab.pdfgen import canvas
7 from io import BytesIO
8 import datetime
9 import base64
10 from PIL import Image
11
12 app = Flask(__name__) # Inicializamos la aplicación Flask
13 app.secret_key = "apRh0hNiTnMfwUc8+4AecQ==" # Clave secreta para manejar sesiones seguras en Flask

```

Figura 3-20 Inicialización del *backend*

Esta Figura 3-20 representa la base del *backend*, sobre la cual se implementarán funcionalidades como la gestión de horarios, el almacenamiento de datos y la generación de archivos descargables. Las bibliotecas *os*, *openpyxl* y *csv* permiten interactuar con archivos del sistema, facilitando la lectura y procesamiento de datos en Excel y CSV. Flask y sus módulos (*request*, *Response*, *send_file*, etc.) gestionan las solicitudes HTTP y la comunicación entre el servidor y el cliente.

Para la generación de documentos PDF, ReportLab (*canvas*) permite crear archivos en distintos formatos, mientras que BytesIO optimiza su manejo en memoria sin necesidad de guardarlos en disco. Pillow (PIL) se encarga de la manipulación de imágenes antes de integrarlas en los documentos, y base64 facilita la transferencia de imágenes en formato codificado.

La última parte del *backend* inicia la aplicación Flask ejecutándola en modo *debug*. La condición `if __name__ == '__main__':` garantiza que el servidor solo se inicie si el script se ejecuta directamente y no si se importa como módulo en otro programa.

El parámetro `debug=True`, en la Figura 3-21, es fundamental durante el desarrollo, ya que permite la recarga automática del servidor ante cambios en el código y muestra mensajes detallados de error en caso de fallos.

```
648
649 # Iniciar la aplicación Flask en modo debug
650 if __name__ == '__main__':
651     app.run(debug=True)
652
```

Figura 3-21 Cierre del *backend*

3.2.3.1 Diccionarios

En el desarrollo del *backend*, los diccionarios juegan un papel fundamental al permitir la organización y almacenamiento estructurado de información clave para la aplicación. En este caso, se han definido diversos diccionarios que facilitan la gestión de los horarios, asignaturas, profesores, aulas y sesiones dentro del sistema de planificación académica. A continuación, se describe en detalle su estructura y función dentro del código.

El diccionario *SELECCIONES* actúa como un contenedor dinámico donde se almacenan las selecciones de asignaturas, profesores y aulas correspondientes a cada celda del horario. Este diccionario se irá rellenando a lo largo del uso de la aplicación en función de las interacciones del usuario con la interfaz.

Por otro lado, se define una función que permite cargar dinámicamente las asignaturas y sus colores desde un archivo Excel (.xlsx) utilizando la biblioteca *Openpyxl*. En lugar de definir manualmente las asignaturas en un diccionario estático, el sistema extrae estos datos directamente desde un archivo externo, lo que facilita su actualización sin modificar el código.

La función *cargar_asignaturas_excel(ruta_excel)* abre el archivo, selecciona la hoja activa y busca las columnas que contienen el nombre de la asignatura y su código de color en formato hexadecimal. Luego, recorre cada fila del archivo, almacenando los valores en un diccionario, donde la clave es el nombre de la asignatura y el valor es su color.

Si el archivo no contiene las columnas esperadas, se genera un error para evitar datos incompletos. Finalmente, el diccionario resultante se almacena en la variable global *ASIGNATURAS*, asegurando que la aplicación pueda acceder y utilizar esta información de manera dinámica. Todo lo descrito, se muestra en la Figura 3-22.

```

20 # Función para cargar asignaturas desde el archivo Excel
21 def cargar_asignaturas_excel(ruta_excel):
22     """
23     Carga las asignaturas y sus colores desde un archivo Excel (.xlsx) usando OpenPyXL.
24
25     Parámetros:
26     ruta_excel (str): Ruta del archivo Excel a leer.
27
28     Retorna:
29     dict: Diccionario con asignaturas como claves y colores en formato hexadecimal como valores.
30     """
31     # Cargar el archivo Excel
32     workbook = openpyxl.load_workbook(ruta_excel)
33     sheet = workbook.active # Seleccionar la hoja activa
34
35     # Obtener los nombres de las columnas
36     column_headers = [cell.value for cell in sheet[1]]
37
38     # Buscar las columnas correctas
39     if "Nombre Completo" in column_headers and "Código RGB" in column_headers:
40         col_nombre = column_headers.index("Nombre Completo")
41         col_color = column_headers.index("Código RGB")
42
43         # Leer los datos fila por fila (ignorando encabezados)
44         asignaturas = {
45             row[col_nombre]: row[col_color]
46             for row in sheet.iter_rows(min_row=2, values_only=True)
47             if row[col_nombre] and row[col_color] # Evitar filas vacías
48         }
49         return asignaturas
50     else:
51         raise ValueError("El archivo Excel no contiene las columnas requeridas.")
52
53 # Ruta del archivo Excel con asignaturas
54 RUTA_EXCEL_ASIGNATURAS = "Asignaturas.xlsx"
55
56 # Cargar asignaturas dinámicamente desde el Excel
57 ASIGNATURAS = cargar_asignaturas_excel(RUTA_EXCEL_ASIGNATURAS)
58

```

Figura 3-22 Diccionario de asignaturas

La relación entre asignaturas y profesores se establece en el diccionario *ASIGNATURAS_PROFESORES*, donde la función *cargar_asignaturas_profesores_excel(ruta_excel)* abre el archivo, selecciona la hoja activa y verifica que existen las columnas "Nombre asignatura" y "Nombre profesor". Luego, recorre cada fila del archivo y almacena la información en un diccionario, donde la clave es el nombre de la asignatura y el valor es una lista de profesores que imparten esa materia.

Para gestionar múltiples profesores por asignatura, se utiliza *defaultdict(list)*, lo que permite agregar profesores a una misma clave sin necesidad de inicializar listas manualmente. Si el archivo no contiene las columnas requeridas, se genera un error para evitar inconsistencias.

Finalmente, como se muestra en la Figura 3-23, el diccionario resultante se almacena en *ASIGNATURAS_PROFESORES*, asegurando que la aplicación acceda y utilice estos datos de manera dinámica, optimizando la gestión de la información académica.

```

60 def cargar_asignaturas_profesores_excel(ruta_excel):
61     """
62     Carga la relación Asignatura ↔ Profesores desde un archivo Excel (.xlsx) usando OpenPyXL.
63
64     Parámetros:
65     ruta_excel (str): Ruta del archivo Excel a leer.
66
67     Retorna:
68     dict: Diccionario con asignaturas como claves y listas de profesores como valores.
69     """
70     # Cargar el archivo Excel
71     workbook = openpyxl.load_workbook(ruta_excel)
72     sheet = workbook.active # Seleccionar la hoja activa
73
74     # Obtener los nombres de las columnas
75     column_headers = [cell.value for cell in sheet[1]]
76
77     # Verificar que existen al menos dos columnas (Asignatura y Profesor)
78     if "Nombre asignatura" in column_headers and "Nombre profesor" in column_headers:
79         col_asignatura = column_headers.index("Nombre asignatura")
80         col_profesor = column_headers.index("Nombre profesor")
81
82         # Diccionario para almacenar la relación Asignatura ↔ Profesores
83         asignaturas_profesores = defaultdict(list)
84
85         # Leer los datos fila por fila (ignorando encabezados)
86         for row in sheet.iter_rows(min_row=2, values_only=True):
87             asignatura = row[col_asignatura]
88             profesor = row[col_profesor]
89
90             if asignatura and profesor: # Evitar filas vacías
91                 asignaturas_profesores[asignatura].append(profesor)
92
93         return dict(asignaturas_profesores)
94
95     else:
96         raise ValueError("El archivo Excel no contiene las columnas requeridas.")
97
98 # Ruta del archivo Excel con Asignaturas ↔ Profesores
99 RUTA_EXCEL_ASIGNATURAS_PROFESORES = "Asignaturas_Profesores.xlsx"
100
101 # Cargar la relación dinámicamente desde el Excel
102 ASIGNATURAS_PROFESORES = cargar_asignaturas_profesores_excel(RUTA_EXCEL_ASIGNATURAS_PROFESORES)
103

```

Figura 3-23 Diccionario que relaciona profesores con asignaturas

En cuanto a las aulas, se muestra el código en la Figura 3-24, la función *cargar_aulas_excel(ruta_excel)* extrae los nombres y códigos de las aulas desde un archivo Excel, utilizando Openpyxl. Se selecciona la hoja activa y se verifica que el archivo contiene al menos dos columnas.

A continuación, se recorren las filas a partir de la segunda, almacenando los datos en un diccionario, donde el nombre del aula se usa como clave y su código como valor. Esto permite gestionar dinámicamente la asignación de aulas sin modificar manualmente el código.

Si el archivo no contiene suficiente información o presenta filas vacías, se ignoran automáticamente para evitar errores. Finalmente, el diccionario se almacena en *AULAS*, asegurando su disponibilidad para su uso en la aplicación.

```

105 def cargar_aulas_excel(ruta_excel):
106     """
107     Carga las aulas desde un archivo Excel (.xlsx) usando OpenPyXL.
108
109     Parámetros:
110     ruta_excel (str): Ruta del archivo Excel a leer.
111
112     Retorna:
113     dict: Diccionario con nombres de aula como claves y códigos de aula como valores.
114     """
115     # Cargar el archivo Excel
116     workbook = openpyxl.load_workbook(ruta_excel)
117     sheet = workbook.active # Seleccionar la hoja activa
118
119     # Obtener los nombres de las columnas
120     column_headers = [cell.value for cell in sheet[1]]
121
122     # Verificar que hay al menos dos columnas para leer
123     if len(column_headers) < 2:
124         raise ValueError("El archivo Excel no tiene suficientes columnas.")
125
126     # Leer los datos de la primera y segunda columna (pero intercambiando clave-valor)
127     aulas = {
128         row[1]: row[0] # Ahora el nombre del aula es la clave, y el código es el valor
129         for row in sheet.iter_rows(min_row=2, values_only=True)
130         if row[0] and row[1] # Evitar filas vacías
131     }
132     return aulas
133
134 # Ruta del archivo Excel con AULAS
135 RUTA_EXCEL_AULAS = "Aulas.xlsx"
136
137 # Cargar aulas dinámicamente desde el Excel
138 AULAS = cargar_aulas_excel(RUTA_EXCEL_AULAS)
139

```

Figura 3-24 Diccionario de aulas ENM

El diccionario *SESIONES*, Figura 3-25, categoriza los distintos tipos de sesiones disponibles dentro del plan de estudios, diferenciando entre seminarios, laboratorios y clases teóricas. Este aspecto es esencial para la organización académica, ya que permite asignar adecuadamente los recursos y metodologías de enseñanza.

```

57 # Tipos de sesión disponibles
58 SESIONES = {
59     "seminario": "Seminario",
60     "laboratorio": "Laboratorio",
61     "teoria": "Teoría"
62 }

```

Figura 3-25 Diccionario de tipos de sesión lectiva

Por su parte, *DIAS_SEMANA* actúa como un traductor de los códigos abreviados utilizados para referirse a los días de la semana en los identificadores de horarios. De este modo, se mejora la legibilidad del código y se facilita la manipulación de los datos en el *backend* y en el CSV exportable.

```

64 DIAS_SEMANA = {
65     "Lu": "Lunes",
66     "Ma": "Martes",
67     "Mi": "Miércoles",
68     "Ju": "Jueves",
69     "Vi": "Viernes"
70 }

```

Figura 3-26 Diccionario de días de la semana

Finalmente, *HORARIOS* proporciona una traducción entre los identificadores utilizados en el sistema y las franjas horarias reales, lo que permite una mejor interpretación de los datos y asegura la correcta estructuración de los horarios académicos en el archivo CSV exportado.

```

72 # Diccionario para traducir los horarios en el ID
73 HORARIOS = {
74     "1a": "08:15-09:05",
75     "2a": "09:10-10:00",
76     "3a": "10:20-11:10",
77     "4a": "11:15-12:05",
78     "5a": "12:10-13:00",
79     "6a": "13:10-14:00",
80     "7a": "15:30-16:20",
81     "8a": "16:25-17:15",
82     "9a": "17:20-18:10"
83 }

```

Figura 3-27 Diccionario de franjas horarias

En conjunto, estos diccionarios forman la base del sistema de gestión de horarios, permitiendo almacenar, procesar y visualizar la información de manera organizada dentro de la aplicación. Su implementación contribuye significativamente a la flexibilidad del sistema, facilitando futuras modificaciones y ampliaciones sin comprometer su funcionalidad.

3.2.3.2 Ruta /menú

Esta función permite la interacción dinámica entre el *frontend* y el *backend*. La función *mostrar_menu()* se encarga de recuperar el identificador único de la celda seleccionada a través del parámetro *id* en la solicitud HTTP. Este identificador es obtenido utilizando *request.args.get('id')*, lo que permite capturar la celda específica en la que el usuario ha hecho clic dentro del horario. Dicho identificador será fundamental para realizar modificaciones en el contenido de la celda sin necesidad de recargar toda la página.

A continuación, la función utiliza *render_template_string()* para generar dinámicamente el contenido HTML que se enviará como respuesta al cliente. En este caso, el código HTML generado consiste en un *div* con un identificador único correspondiente a la celda seleccionada. Dentro de este *div*, se incorpora la tecnología HTMX para permitir que el contenido se actualice de manera asincrónica. En particular, el atributo *hx-get="/desplegar_menu?id={{ ident }}"* define una solicitud HTTP GET al servidor cuando se activa la celda, lo que permite la carga dinámica del contenido. El atributo *hx-target="#{{ ident }}"* especifica que la respuesta de esta solicitud se insertará en la celda correspondiente,

y `hx-swap="outerHTML"` indica que el contenido completo de la celda será reemplazado con la nueva información enviada desde el *backend*.

Inicialmente, el contenido de la celda se establece con el texto "Estudio". Sin embargo, al hacer clic en ella, el sistema podrá actualizar dinámicamente el contenido con asignaturas, profesores o aulas según la selección realizada por el usuario. Todo lo descrito se muestra en la Figura 3-28.

```

86 @app.route('/menu')
87 def mostrar_menu():
88     identificador = request.args.get('id') # Obtener la celda seleccionada
89
90     return render_template_string('''
91 <div id="{{ ident }}"
92     hx-get="/desplegar_menu?id={{ ident }}"
93     hx-target="#{{ ident }}"
94     hx-swap="outerHTML">
95     Estudio <!-- Texto inicial -->
96 </div>
97     ''', ident=identificador)

```

Figura 3-28 Ruta /menú

3.2.3.3 Ruta /desplegar_menu

La ruta `@app.route('/desplegar_menu')`, mostrada en la Figura 3-29, constituye un componente importante dentro del *backend* de la aplicación, ya que es la encargada de generar dinámicamente la interfaz de selección de asignaturas, profesores, aulas y sesiones para cada celda dentro del horario académico.

Cuando un usuario interactúa con una celda del horario, esta ruta es invocada mediante una solicitud HTTP GET. La función `desplegar_menu()` extrae el identificador único de la celda mediante `request.args.get('id')`, lo que permite saber exactamente qué celda debe actualizarse y desplegar un menú de opciones en su lugar.

La respuesta generada por la función se construye dinámicamente utilizando `render_template_string()`, que genera un fragmento de código HTML en el que se definen distintas secciones interactivas para seleccionar información relevante a la programación académica.

El contenido del HTML generado se basa en que dentro del *div* principal, se crean cuatro secciones específicas, cada una con un identificador único basado en la celda seleccionada:

1. Selección de Asignatura:

- Representada por el *div* con `id="{{ ident }}_asignatura"`, permite que el usuario seleccione una asignatura.
- Utiliza `hx-get="/asignaturas?id={{ ident }}"`, lo que significa que, cuando el usuario haga clic en esta sección, se enviará una solicitud al servidor para obtener la lista de asignaturas.
- `hx-target="#{{ ident }}"` indica que la respuesta del servidor se insertará en la celda correspondiente.
- `hx-swap="outerHTML"` hace que el contenido actual sea reemplazado con la información obtenida desde el *backend*.

2. Selección de Profesor:

- Similar a la selección de asignaturas, este *div* (`id="{{ ident }}_profesor"`) permite elegir un profesor.

- Se obtiene la información desde la ruta `/profesores`, enviando el `id` de la celda.
- La actualización se realiza dinámicamente tras la selección de la asignatura.

3. Selección de Aula:

- Se implementa con `id="{{ ident }}_aula"`, y permite asignar un aula a la clase seleccionada.
- La solicitud `hx-get="/aulas?id={{ ident }}"` recupera los datos del *backend* en función de la selección del usuario.

4. Selección de Sesión:

- Mediante el `div id="{{ ident }}_sesion"`, el usuario puede definir el tipo de sesión (seminario, laboratorio o teoría).
- Se obtiene información a través de `hx-get="/sesion?id={{ ident }}"`, y se actualiza dinámicamente en la celda correspondiente.

5. Opción de pegar:

- Mediante el botón, el usuario puede pegar la información de una celda a otra.
- Se pega información de `hx-post="/pegar?id={{ ident }}"`, actualizándose dinámicamente en la celda al completo.

```

181 @app.route('/desplegar_menu')
182 def desplegar_menu():
183     identificador = request.args.get('id') # Obtener la celda seleccionada
184
185     return render_template_string('''
186 <div id="{{ ident }}"> <!-- Contenedor principal con un identificador único para la celda -->
187
188     <!-- Sección para mostrar la lista de asignaturas -->
189     <div id="{{ ident }}_asignatura" class="subject"
190         hx-get="/asignaturas?id={{ ident }}"
191         hx-target="#{{ ident }}"
192         hx-swap="outerHTML">
193         Asignatura
194     </div>
195
196     <!-- Sección para seleccionar un profesor (se actualizar dinámicamente) -->
197     <div id="{{ ident }}_profesor" class="teacher"
198         hx-get="/profesores?id={{ ident }}"
199         hx-target="#{{ ident }}"
200         hx-swap="outerHTML">
201         Profesor
202     </div>
203
204     <!-- Sección para seleccionar un aula (se actualizar dinámicamente) -->
205     <div id="{{ ident }}_aula" class="classroom"
206         hx-get="/aulas?id={{ ident }}"
207         hx-target="#{{ ident }}"
208         hx-swap="outerHTML">
209         Aula
210     </div>
211
212     <!-- Sección para seleccionar la sesión -->
213     <div id="{{ ident }}_sesion" class="session"
214         hx-get="/sesion?id={{ ident }}"
215         hx-target="#{{ ident }}"
216         hx-swap="outerHTML">
217         Sesión
218     </div>
219
220     <!-- Botones para Copiar y Pegar -->
221     <div class="copy-paste-controls">
222         <button hx-post="/pegar?id={{ ident }}"
223             hx-trigger="click"
224             hx-target="#{{ ident }}"
225             hx-swap="outerHTML">
226              Pegar
227         </button>
228     </div>
229 </div>
230 ''', ident=identificador)

```

Figura 3-29 Ruta `/desplegar_menu`

3.2.3.4 Ruta /copiar

La ruta `/copiar` en Flask (Figura 3-30) permite almacenar temporalmente la información de una celda en una variable global llamada `PORTAPAPELES`, facilitando la funcionalidad de copiar y pegar dentro de la aplicación.

Al recibir una solicitud POST, la función verifica si la celda identificada (`celda_id`) existe en el diccionario `SELECCIONES` y contiene datos. Si es así, su contenido se copia en `PORTAPAPELES` para su posterior uso.

La respuesta es 204 (*No Content*), ya que HTMX no requiere una respuesta visible para actualizar la interfaz. Si la celda está vacía, simplemente se devuelve el mismo código de estado sin modificar `PORTAPAPELES`.

```

236 @app.route('/copiar', methods=['POST'])
237 def copiar():
238     global PORTAPAPELES
239     celda_id = request.args.get('id')
240
241     if celda_id in SELECCIONES and SELECCIONES[celda_id]: # Solo copiar si la celda tiene datos
242         PORTAPAPELES = SELECCIONES[celda_id].copy() # Copiamos toda la informaci[n]
243
244     return "", 204 # Si la celda est[ ] vac[ ]a, tampoco mostramos nada

```

Figura 3-30 Ruta /copiar

3.2.3.5 Ruta /pegar

La ruta `/pegar` en Flask (Figura 3-31) permite insertar en una celda la información previamente copiada en `PORTAPAPELES`, facilitando la transferencia rápida de datos dentro de la aplicación.

Al recibir una solicitud POST, primero se verifica si `PORTAPAPELES` contiene datos. Si no hay información copiada, la función simplemente devuelve un código 204 (*No Content*) sin realizar cambios.

Si la celda de destino (`celda_id`) aún no existe en `SELECCIONES`, se inicializa con valores vacíos para evitar errores. Luego, la información almacenada en `PORTAPAPELES` se asigna a la celda, asegurando que los datos copiados se transfieran correctamente.

Finalmente, la función devuelve un fragmento HTML con los datos pegados, incluyendo la asignatura, profesor, aula y sesión, además de aplicar el color correspondiente a la asignatura.

```

246 @app.route('/pegar', methods=['POST'])
247 def pegar():
248     global PORTAPAPELES
249     celda_id = request.args.get('id')
250
251     if not PORTAPAPELES:
252         return "", 204 # No hacemos nada si no hay informaci3n copiada
253
254     # Si la celda NO existe en SELECCIONES, la creamos como vac3a antes de pegar
255     if celda_id not in SELECCIONES:
256         SELECCIONES[celda_id] = ["", "", "", ""] # Se inicializa la celda con valores vac3os
257
258     SELECCIONES[celda_id] = PORTAPAPELES.copy() # Pegamos la informaci3n copiada
259
260
261     # Obtener la informaci3n de la celda para actualizar la interfaz
262     asignatura, profesor, aula, sesion = SELECCIONES[celda_id]
263     color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Color de la asignatura
264
265     return render_template_string('''
266     <div id="{{ ident }}" class="study" style="background-color: {{ color }};">
267         <div id="{{ ident }}_asignatura" class="subject">
268             {{ asignatura if asignatura else "Asignatura" }}
269         </div>
270         <div id="{{ ident }}_profesor" class="teacher">
271             {{ profesor if profesor else "Profesor" }}
272         </div>
273         <div id="{{ ident }}_aula" class="classroom">
274             {{ aula if aula else "Aula" }}
275         </div>
276         <div id="{{ ident }}_sesion" class="session">
277             {{ sesion if sesion else "Sesi3n" }}
278         </div>
279     </div>
280 </div>
281 ''', ident=celda_id, asignatura=asignatura, profesor=profesor, aula=aula, sesion=sesion, color=color)
282

```

Figura 3-31 Ruta /pegar

3.2.3.6 Ruta /asignaturas

La ruta, mostrada en la Figura 3-32, tiene como finalidad que al hacer clic izquierdo el usuario sobre la palabra “*Asignatura*”, se listen las distintas asignaturas para después poder elegir entre las distintas asignaturas.

Para ello, primero la ruta, obtiene el identificador de la celda donde el usuario ha hecho clic izquierdo, posteriormente genera una respuesta HTML que consiste en que el código define un *div* con un *id* único basado en la celda seleccionada. Dentro de este *div*, se itera sobre la lista de asignaturas predefinidas en el *backend*, generando un botón para cada una de ellas.

Cada botón representa una asignatura y está configurado con un evento *hx-get* que, al ser presionado, envía una solicitud GET a la ruta */seleccion_asignatura*. Esta solicitud incluye dos parámetros en la URL:

1. *id={{ ident }}* → Identifica la celda específica en la que se debe actualizar la información.
2. *asignatura={{ asignatura }}* → Especifica la asignatura seleccionada.

El atributo *hx-target="#{{ ident }}"* indica que la respuesta del servidor se insertará en el contenedor de la celda correspondiente. Además, *hx-swap="outerHTML"* se encarga de reemplazar el contenido de la celda con la nueva información devuelta por la ruta */seleccion_asignatura*.

Después de cada botón se añade un *
* para separar visualmente las asignaturas, mejorando la legibilidad de la interfaz.

Por último, sigue devolviendo los valores de profesor, aula y sesión predefinidos.

```

141 # Ruta que devuelve un listado de asignaturas disponibles
142 @app.route('/asignaturas')
143 def listar_asignaturas():
144     # Obtener el identificador de la celda con la que se está interactuando
145     identificador = request.args.get('id')
146
147     # Retornamos una plantilla HTML con la lista de asignaturas disponibles
148     return render_template_string('''
149     <div id="{{ ident }}"> <!-- Contenedor principal con un identificador único para la celda -->
150
151     <!-- Sección para mostrar la lista de asignaturas -->
152     <div id="{{ ident }}_asignatura">
153         {% for asignatura in asignaturas %} <!-- Iteramos sobre todas las asignaturas disponibles -->
154             <button
155                 hx-get="/seleccion_asignatura?id={{ ident }}&asignatura={{ asignatura }}"
156                 hx-target="#{{ ident }}"
157                 hx-swap="outerHTML">
158                 {{ asignatura }} <!-- Mostramos el nombre de la asignatura en el botón -->
159             </button><br/> <!-- Insertamos un salto de línea entre cada botón -->
160         {% endfor %}
161     </div>
162
163     <!-- Sección para seleccionar un profesor (se actualizará dinámicamente) -->
164     <div id="{{ ident }}_profesor">
165         hx-get="/profesores"
166         hx-target="#{{ ident }}"
167         hx-swap="outerHTML">
168         Profesor <!-- Texto inicial antes de seleccionar un profesor -->
169     </div>
170
171     <!-- Sección para seleccionar un aula (se actualizará dinámicamente) -->
172     <div id="{{ ident }}_aula">
173         hx-get="/aulas"
174         hx-target="#{{ ident }}"
175         hx-swap="outerHTML">
176         Aula <!-- Texto inicial antes de seleccionar un aula -->
177     </div>
178
179     <!-- Sección para seleccionar la sesión -->
180     <div id="{{ ident }}_sesion">
181         hx-get="/sesion?id={{ ident }}"
182         hx-target="#{{ ident }}"
183         hx-swap="outerHTML">
184         Sesión
185     </div>
186 </div>
187 ''', asignaturas=ASIGNATURAS, ident=identificador) # Pasamos las asignaturas y el identificador como variables a la plantilla
188

```

Figura 3-32 Ruta /asignaturas

3.2.3.7 Ruta /selección_asignatura

Su función principal es recibir la asignatura seleccionada por el usuario y actualizar el estado de la celda correspondiente en la estructura de datos del *backend*, permitiendo una actualización visual sin necesidad de recargar la página.

Cuando un usuario selecciona una asignatura dentro del sistema de horarios, se genera una solicitud GET dirigida a la ruta */seleccion_asignatura* (Figura 3-33). Esta solicitud incluye dos parámetros clave: el identificador único de la celda en la tabla de horarios y el nombre de la asignatura elegida. Estos datos son procesados en el *backend* para actualizar la información correspondiente y reflejar los cambios en la interfaz de manera dinámica.

El primer paso en este proceso es la actualización del diccionario *SELECCIONES*, que actúa como una base de datos temporal para almacenar la información de cada celda. Si la celda ya existía en el diccionario, su valor se sobrescribe con la nueva asignatura seleccionada, asegurando que cualquier modificación del usuario se refleje correctamente. En caso de que la celda no haya sido previamente registrada en *SELECCIONES*, se crea una nueva entrada en la que se guarda la asignatura seleccionada, mientras que los valores correspondientes al profesor y al aula permanecen vacíos hasta que sean definidos posteriormente.

Otro aspecto fundamental es la asignación de un color de fondo asociado a la asignatura elegida. Este mecanismo permite diferenciar visualmente las materias dentro de la tabla de horarios, facilitando la identificación rápida de cada asignatura. Para ello, se consulta el diccionario *ASIGNATURAS*, que

almacena los colores predefinidos para cada materia. Si la asignatura seleccionada tiene un color asignado, se aplica automáticamente a la celda correspondiente.

Finalmente, el sistema puede generar dos respuestas en formato HTML, dependiendo de si la asignatura tiene profesor asociado o no da una respuesta u otra, que se inserta en la interfaz sin necesidad de recargar la página.

Si la asignatura tiene profesor asociado, esta respuesta incluye un contenedor *div* cuyo identificador coincide con el de la celda seleccionada y cuyo color de fondo se ajusta a la asignatura elegida. Además, dentro de este contenedor se presentan las opciones para seleccionar un profesor, un aula y una sesión, manteniendo una estructura interactiva que permite la actualización dinámica de cada sección.

Si la asignatura seleccionada no tiene profesor asociado, solo se devuelve un *div* con la asignatura asociada sin dar la opción de elegir profesor, aula y sesión, esto es útil cuando la asignatura seleccionada son actividades militares, estudio u otras.

```

191 @app.route('/seleccion_asignatura')
192 def seleccionar_asignatura():
193     identificador = request.args.get('id') # Identificador de la celda
194     asignatura = request.args.get('asignatura') # Asignatura seleccionada
195
196     # Definir color de la asignatura o blanco si no está en el diccionario
197     color = ASIGNATURAS.get(asignatura, "#FFFFFF")
198
199     # Actualizar el diccionario SELECCIONES
200     if identificador in SELECCIONES:
201         SELECCIONES[identificador][0] = asignatura # Sobrescribir asignatura
202     else:
203         SELECCIONES[identificador] = [asignatura, None, None]
204
205     # Comprobar si la asignatura tiene profesores asignados
206     tiene_profesor = asignatura in ASIGNATURAS_PROFESORES and ASIGNATURAS_PROFESORES[asignatura]
207
208     # Si la asignatura NO tiene profesor, solo actualizar la celda con la asignatura
209     if not tiene_profesor:
210         return render_template_string('''
211             <div id="{{ ident }}" style="background-color: {{ bg_color }}">
212                 <div id="{{ ident }}_asignatura" class="subject">
213                     {{ asig }}
214                 </div>
215             </div>
216         ''', asig=asignatura, ident=identificador, bg_color=color)
217
218     # Si la asignatura SÍ tiene profesor, mostrar las opciones completas
219     return render_template_string('''
220         <div id="{{ ident }}" style="background-color: {{ bg_color }}">
221             <!-- Mostrar la asignatura seleccionada en la celda -->
222             <div id="{{ ident }}_asignatura" class="subject"
223                 hx-get="/asignaturas?id={{ ident }}"
224                 hx-target="#{{ ident }}"
225                 hx-swap="outerHTML">
226                 {{ asig }}
227             </div>
228
229             <!-- Sección para la selección de profesores -->
230             <div id="{{ ident }}_profesor" class="teacher"
231                 hx-get="/profesores?id={{ ident }}"
232                 hx-target="#{{ ident }}"
233                 hx-swap="outerHTML">
234                 Profesor
235             </div>
236
237             <!-- Sección para la selección de aulas -->
238             <div id="{{ ident }}_aula" class="classroom"
239                 hx-get="/aulas?id={{ ident }}"
240                 hx-target="#{{ ident }}"
241                 hx-swap="outerHTML">
242                 Aula

```

Figura 3-33 Ruta /selección_asignatura

3.2.3.8 Ruta /profesores

La función principal de la ruta */profesores* (Figura 3-34) es mostrar la lista de profesores disponibles en función de la asignatura previamente seleccionada en una celda específica. De este modo, el usuario puede avanzar en la personalización del horario, eligiendo a los docentes correspondientes para cada materia.

Cuando un usuario interactúa con una celda en la tabla, el sistema recupera su identificador y consulta el diccionario *SELECCIONES* para determinar qué asignatura está vinculada a dicha celda. Si aún no se ha seleccionado ninguna materia, la ruta simplemente devuelve una respuesta vacía con el código HTTP 204, lo que evita errores innecesarios en la interfaz. En caso contrario, se procede a buscar los profesores asociados a la asignatura en el diccionario *ASIGNATURAS_PROFESORES*, garantizando que solo se muestren opciones relevantes para el usuario.

Un aspecto interesante de esta funcionalidad es la coherencia visual que mantiene el sistema. Cada asignatura tiene un color predefinido que se mantiene al listar los profesores disponibles, reforzando la asociación entre las materias y los docentes de una manera clara e intuitiva. En la respuesta HTML generada dinámicamente, los nombres de los profesores aparecen como botones interactivos. Al hacer clic en uno de ellos, se activa una solicitud *hx-get*, la cual actualiza la celda con la selección realizada, sin necesidad de recargar la página.

```

257 # Definir la ruta que maneja la visualización de los profesores disponibles
258 @app.route('/profesores')
259 def listar_profesores():
260     # Obtener el identificador de la celda que hizo la solicitud
261     identificador = request.args.get('id')
262
263     # Obtener la asignatura seleccionada previamente en la celda correspondiente
264     asignatura = SELECCIONES.get(identificador, [None])[0] # Extrae solo la asignatura
265
266     # Si no hay asignatura seleccionada, no se muestran los profesores y se devuelve una respuesta vacía
267     if asignatura is None:
268         return "", 204 # Código 204 indica "No Content", evitando errores en la interfaz
269
270     # Obtener la lista de profesores asignados a la asignatura seleccionada
271     profesores = ASIGNATURAS_PROFESORES.get(asignatura, [])
272
273     # Obtener el color de fondo de la celda según la asignatura seleccionada
274     color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Si no hay color definido, usa blanco por defecto
275
276     # Generar y devolver el HTML que se insertará en la interfaz
277     return render_template_string("""
278 <div id="{{ ident }}" style="background-color: {{ color }}"> <!-- Se mantiene el color de fondo de la asignatura -->
279
280 <!-- Mostrar la asignatura seleccionada en la celda -->
281 <div id="{{ ident }}_asignatura"
282     |   hx-get="/asignaturas?id={{ ident }}"
283     |   hx-target="#{{ ident }}"
284     |   hx-swap="outerHTML">
285     |   {{ asignatura }}
286 </div>
287
288 <!-- Sección para mostrar la lista de profesores disponibles -->
289 <div id="{{ ident }}_profesor">
290     |   {% for profesor in profesores %} <!-- Itera sobre la lista de profesores disponibles -->
291     |       <button hx-get="/seleccion_profesor?id={{ ident }}&profesor={{ profesor }}"
292     |           |   hx-target="#{{ ident }}"
293     |           |   hx-swap="outerHTML">
294     |           |   {{ profesor }} <!-- Muestra el nombre del profesor en un botón -->
295     |       </button><br/> <!-- Inserta un salto de línea después de cada botón -->
296     |   {% endfor %}
297 </div>

```

Figura 3-34 Ruta /profesores

3.2.3.9 Ruta /selección_profesor

La ruta `/seleccion_profesor` (Figura 3-35) entra en juego cuando un usuario elige un profesor de la lista. El sistema recupera la celda afectada y actualiza la entrada correspondiente en `SELECCIONES`, asegurando que la asignación de docentes quede registrada de manera eficiente. En caso de que la celda aún no estuviera en el diccionario, se crea una nueva entrada con la asignatura en `None` (si aún no ha sido seleccionada), el profesor elegido y un aula en blanco, permitiendo que el usuario complete la configuración en pasos posteriores.

El HTML generado para reflejar esta selección mantiene la estructura visual previa, con la asignatura y el profesor mostrados en la celda de manera destacada. Además, se conservan las opciones para elegir un aula y una sesión, facilitando la continuidad del proceso sin interrupciones.

```

318 # Definir la ruta que maneja la selección de un profesor
319 @app.route('/seleccion_profesor')
320 def seleccionar_profesor():
321     # Obtener el identificador de la celda que hizo la solicitud
322     identificador = request.args.get('id')
323
324     # Obtener el nombre del profesor seleccionado desde la solicitud GET
325     profesor = request.args.get('profesor')
326
327     # Verificar si la celda ya está en SELECCIONES (es decir, si ya tiene una asignatura asignada)
328     if identificador in SELECCIONES:
329         SELECCIONES[identificador][1] = profesor # Asigna el profesor en la posición 1 del array
330     else:
331         # Si la celda aún no ha sido registrada en SELECCIONES, se crea una nueva entrada con:
332         # - None en la posición de asignatura (porque aún no se ha seleccionado ninguna)
333         # - El profesor seleccionado
334         # - None en la posición de aula
335         SELECCIONES[identificador] = [None, profesor, None]
336
337     # Obtener la asignatura almacenada en SELECCIONES para esta celda (si existe)
338     asignatura = SELECCIONES.get(identificador, [None])[0]
339
340     # Obtener el color correspondiente a la asignatura seleccionada (si no hay asignatura, se usa blanco)
341     color = ASIGNATURAS.get(asignatura, "#FFFFFF")
342
343     # Generar y devolver el HTML actualizado para la celda del horario
344     return render_template_string('''
345     <div id="{{ ident }}" style="background-color: {{ color }}">
346
347         <!-- Mostrar la asignatura seleccionada (si hay una) -->
348         <div id="{{ ident }}_asignatura" class="subject"
349             hx-get="/asignaturas?id={{ ident }}"
350             hx-target="#{{ ident }}"
351             hx-swap="outerHTML">
352             {{ asignatura }}
353         </div>
354
355         <!-- Mostrar el profesor seleccionado en la celda -->
356         <div id="{{ ident }}_profesor" class="teacher"
357             hx-get="/profesores?id={{ ident }}"
358             hx-target="#{{ ident }}"
359             hx-swap="outerHTML">
360             {{ profesor }}
361         </div>

```

Figura 3-35 Ruta /selección_profesor

3.2.3.10 Ruta /aulas

La ruta `/aulas` (Figura 3-36) es un paso fundamental dentro del flujo de asignación de horarios, ya que permite a los usuarios seleccionar el aula en la que se impartirá cada materia. A diferencia de las rutas anteriores, en las que el usuario elige primero una asignatura y luego un profesor, aquí la lógica impone un requisito adicional: solo se mostrarán las aulas disponibles si ya se han seleccionado tanto una asignatura como un docente. Esto evita que se asigne un espacio sin contexto previo y garantiza una estructura organizada en la planificación académica.

Cuando el usuario interactúa con una celda del horario, el sistema recupera su identificador y consulta el diccionario `SELECCIONES` para verificar si ya se han definido la asignatura y el profesor. En caso de que alguno de estos elementos aún no haya sido escogido, la respuesta devuelta es un código HTTP 204 (*No Content*), lo que impide que la interfaz muestre opciones erróneas y ayuda a mantener la integridad de los datos. Una vez confirmada la presencia de estos elementos, se procede a recuperar el color correspondiente a la asignatura seleccionada.

El sistema genera una respuesta HTML en la que cada aula aparece representada como un botón interactivo. Estos botones están asociados a una solicitud `hx-get`, que permite la selección dinámica del aula sin necesidad de recargar la página. Además, la estructura mantiene la coherencia con las etapas anteriores: se siguen mostrando la asignatura y el profesor ya seleccionados, permitiendo al usuario confirmar su elección antes de avanzar.

```

382 # Definir la ruta que maneja la visualización de las aulas disponibles
383 @app.route('/aulas')
384 def listar_aulas():
385     # Obtener el identificador de la celda que hizo la solicitud
386     identificador = request.args.get('id')
387
388     # Obtener la asignatura seleccionada previamente en la celda correspondiente
389     asignatura = SELECCIONES.get(identificador, [None])[0] # Extrae solo la asignatura
390
391     # Obtener el profesor seleccionado previamente en la celda correspondiente
392     profesor = SELECCIONES.get(identificador, [None, None])[1] # Extrae solo el profesor
393
394     # Si no hay asignatura o profesor seleccionados, no se muestran las aulas y se devuelve una respuesta vacía
395     if asignatura is None or profesor is None:
396         return "", 204 # Código 204 indica "No Content", evitando errores en la interfaz
397
398     # Obtener el color de fondo de la celda según la asignatura seleccionada
399     color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Si no hay color definido, usa blanco por defecto
400
401     # Generar y devolver el HTML que se insertará en la interfaz
402     return render_template_string('''
403 <div id="{{ ident }}" style="background-color: {{ color }}"> <!-- Se mantiene el color de fondo de la asignatura -->
404
405 <!-- Mostrar la asignatura seleccionada en la celda -->
406 <div id="{{ ident }}_asignatura"
407     hx-get="/asignaturas?id={{ ident }}"
408     hx-target="#{{ ident }}"
409     hx-swap="outerHTML">
410     {{ asignatura }}
411 </div>
412
413 <!-- Mostrar el profesor seleccionado en la celda -->
414 <div id="{{ ident }}_profesor"
415     hx-get="/profesores?id={{ ident }}"
416     hx-target="#{{ ident }}"
417     hx-swap="outerHTML">
418     {{ profesor if profesor else "Profesor" }} <!-- Si no hay profesor, mostrar "Profesor" como marcador de posición -->
419 </div>
420
421 <!-- Sección para mostrar la lista de aulas disponibles -->
422 <div id="{{ ident }}_aula">
423     {% for aula_id, aula_nombre in aulas.items() %} <!-- Itera sobre la lista de aulas disponibles -->
424     <button hx-get="/seleccion_aula?id={{ ident }}&aula={{ aula_id }}"
425         hx-target="#{{ ident }}"
426         hx-swap="outerHTML">
427         {{ aula_id }} <!-- Muestra el código del aula en un botón -->
428     </button><br/> <!-- Inserta un salto de línea después de cada botón -->
429     {% endfor %}

```

Figura 3-36 Ruta /aulas

3.2.3.11 Ruta /selección_aula

La ruta `/seleccion_aula` (Figura 3-37) es el paso final en la asignación de los elementos dentro del horario antes de seleccionar la sesión. Su función es registrar el aula elegida por el usuario y actualizar la interfaz con esta información. Al igual que en rutas anteriores, el sistema recibe el identificador de la celda y el código del aula seleccionada a través de los parámetros de la solicitud GET. Estos datos permiten modificar la entrada correspondiente en el diccionario `SELECCIONES`, donde se almacenan las configuraciones de cada celda.

Cuando el usuario selecciona un aula, el *backend* primero verifica si la celda ya está registrada en `SELECCIONES`. Si es así, simplemente actualiza el campo correspondiente con el aula elegida. En caso contrario, se crea una nueva entrada con valores predeterminados para los demás campos, asegurando que la estructura del diccionario permanezca organizada y coherente. Este enfoque evita inconsistencias y garantiza que todas las celdas contengan información válida.

Una vez almacenada la selección, la aplicación genera una respuesta HTML que refleja los cambios en la interfaz. La celda del horario se actualiza dinámicamente, mostrando la asignatura, el profesor y el aula elegidos, junto con la opción de seleccionar la sesión.

```

443 # Definir la ruta que maneja la selección de un aula
444 @app.route('/seleccion_aula')
445 def seleccionar_aula():
446     # Obtener el identificador de la celda que hizo la solicitud
447     identificador = request.args.get('id')
448
449     # Obtener el código del aula seleccionada desde la solicitud GET
450     aula = request.args.get('aula')
451
452     # Verificar si la celda ya tiene información guardada en SELECCIONES
453     if identificador in SELECCIONES:
454         SELECCIONES[identificador][2] = aula # Se asigna el aula en la posición [2] del array
455     else:
456         # Si la celda aún no ha sido registrada en SELECCIONES, se crea una nueva entrada con:
457         # - None en la posición de asignatura (porque aún no se ha seleccionado ninguna)
458         # - None en la posición de profesor (porque aún no se ha seleccionado ninguno)
459         # - El aula seleccionada
460         SELECCIONES[identificador] = [None, None, aula]
461
462     # Obtener la asignatura almacenada en SELECCIONES para esta celda (si existe)
463     asignatura = SELECCIONES.get(identificador, [None])[0]
464
465     # Obtener el profesor almacenado en SELECCIONES para esta celda (si existe)
466     profesor = SELECCIONES.get(identificador, [None, None])[1]
467
468     # Obtener el color correspondiente a la asignatura seleccionada (si no hay asignatura, se usa blanco)
469     color = ASIGNATURAS.get(asignatura, "#FFFFFF")
470
471     # Generar y devolver el HTML actualizado para la celda del horario
472     return render_template_string('''
473     <div id="{{ ident }}" style="background-color: {{ color }}">
474
475         <!-- Mostrar la asignatura seleccionada en la celda -->
476         <div id="{{ ident }}_asignatura" class="subject"
477             hx-get="/asignaturas?id={{ ident }}"
478             hx-target="#{{ ident }}"
479             hx-swap="outerHTML">
480             {{ asignatura }}
481         </div>
482
483         <!-- Mostrar el profesor seleccionado en la celda -->
484         <div id="{{ ident }}_profesor" class="teacher"
485             hx-get="/profesores?id={{ ident }}"
486             hx-target="#{{ ident }}"
487             hx-swap="outerHTML">
488             {{ profesor if profesor else "Profesor" }} <!-- Si no hay profesor, mostrar "Profesor" como marcador de posición -->
489         </div>
490
491         <!-- Mostrar el aula seleccionada en la celda -->
492         <div id="{{ ident }}_aula" class="classroom"
493             hx-get="/aulas?id={{ ident }}"
494             hx-target="#{{ ident }}"
495             hx-swap="outerHTML">
496             {{ aula }}
497         </div>

```

Figura 3-37 Ruta /selección_aula

3.2.3.12 Ruta /sesión

La ruta `/sesión` (Figura 3-38) completa el proceso de asignación de horarios al permitir la selección del tipo de sesión asociado a una clase. Al igual que en las rutas anteriores, esta recibe el identificador de la celda en la tabla y verifica qué asignatura, profesor y aula han sido seleccionados previamente. Solo cuando todos estos elementos han sido definidos, la opción de elegir la sesión se habilita, garantizando que la planificación se realice de manera estructurada.

El *backend* obtiene las opciones disponibles de sesión desde el diccionario *SESIONES*, donde se establecen categorías como seminario, laboratorio o teoría. Estas opciones se presentan en la interfaz mediante botones interactivos, que al ser seleccionados envían una nueva solicitud GET para registrar la elección en el sistema. Este diseño asegura que el usuario pueda modificar y visualizar su planificación en tiempo real sin necesidad de recargar la página.

```

509 @app.route('/sesion')
510 def listar_sesion():
511     identificador = request.args.get('id') # Obtener la celda donde se hizo clic
512     # Obtener la asignatura seleccionada previamente en la celda correspondiente
513     asignatura = SELECCIONES.get(identificador, [None])[0] # Extrae solo la asignatura
514
515     # Obtener el profesor seleccionado previamente en la celda correspondiente
516     profesor = SELECCIONES.get(identificador, [None, None])[1] # Extrae solo el profesor
517     # Obtener el aula seleccionado previamente en la celda correspondiente
518     aula = SELECCIONES.get(identificador, [None, None, None])[2] # Extrae solo el aula
519     # Si no hay asignatura, profesor o aula seleccionados, no se muestran las sesiones y se devuelve una respuesta vacía
520     if asignatura is None or profesor is None or aula is None:
521         return "", 204 # Código 204 indica "No Content", evitando errores en la interfaz
522
523     # Obtener el color de fondo de la celda según la asignatura seleccionada
524     color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Si no hay color definido, usa blanco por defecto
525     return render_template_string('''
526 <div id="{{ ident }}" style="background-color: {{ color }}"> <!-- Se mantiene el color de fondo de la asignatura -->
527
528 <!-- Mostrar la asignatura seleccionada en la celda -->
529 <div id="{{ ident }}_asignatura"
530     class="subject"
531     hx-get="/asignaturas?id={{ ident }}"
532     hx-target="#{{ ident }}"
533     hx-swap="outerHTML">
534     {{ asignatura }}
535 </div>
536
537 <!-- Mostrar el profesor seleccionado en la celda -->
538 <div id="{{ ident }}_profesor"
539     class="teacher"
540     hx-get="/profesores?id={{ ident }}"
541     hx-target="#{{ ident }}"
542     hx-swap="outerHTML">
543     {{ profesor if profesor else "Profesor" }} <!-- Si no hay profesor, mostrar "Profesor" como marcador de posición -->
544 </div>
545
546 <div id="{{ ident }}_aula" class="classroom"
547     hx-get="/aulas?id={{ ident }}"
548     hx-target="#{{ ident }}"
549     hx-swap="outerHTML">
550     {{ aula }}
551 </div>
552
553 <div id="{{ ident }}_sesion">
554     {% for clave, nombre in sesiones.items() %}
555     <button hx-get="/seleccionar_sesion?id={{ ident }}&sesion={{ clave }}"
556         hx-target="#{{ ident }}"
557         hx-swap="outerHTML">
558         {{ nombre }}
559     </button><br/>
560     {% endfor %}
561 </div>
562 ''', sesiones=SESIONES, ident=identificador, aula=aula, asignatura=asignatura, profesor=profesor, color=color)

```

Figura 3-38 Ruta /sesión

3.2.3.13 Ruta /selección_sesión

La ruta `/seleccionar_sesion` (Figura 3-39) finaliza el proceso de asignación de horarios al registrar la sesión elegida por el usuario en la celda correspondiente. Una vez que el usuario selecciona el tipo de sesión (seminario, laboratorio o teoría), esta información se almacena en el diccionario `SELECCIONES`, asegurando que la celda contenga todos los elementos necesarios: asignatura, profesor, aula y sesión.

Por último, incluye un botón que da la opción de copiar la información de la celda, para posteriormente poder pegar la información a otras celdas de manera mucho más rápida.

```

707 @app.route('/seleccionar_sesion')
708 def seleccionar_sesion():
709     identificador = request.args.get('id') # Leer la celda seleccionada
710     sesion = request.args.get('sesion') # Leer la sesión seleccionada
711
712     # Guardar la sesión en la selección
713     if identificador in SELECCIONES:
714         # Asegurar que la lista tiene al menos 4 elementos
715         while len(SELECCIONES[identificador]) < 4:
716             SELECCIONES[identificador].append(None) # Agrega 'None' hasta que tenga 4 posiciones
717             SELECCIONES[identificador][3] = sesion # Ahora sí podemos asignar la sesión
718     else:
719         SELECCIONES[identificador] = [None, None, None, sesion] # Se asegura de que siempre haya 4 elementos
720     # Obtener la asignatura almacenada en SELECCIONES para esta celda (si existe)
721     asignatura = SELECCIONES.get(identificador, [None])[0]
722     # Obtener el profesor almacenado en SELECCIONES para esta celda (si existe)
723     profesor = SELECCIONES.get(identificador, [None, None])[1]
724     # Obtener el aula almacenada en SELECCIONES para esta celda (si existe)
725     aula = SELECCIONES.get(identificador, [None, None, None])[2]
726     # Obtener el color correspondiente a la asignatura seleccionada (si no hay asignatura, se usa blanco)
727     color = ASIGNATURAS.get(asignatura, "#FFFFFF")
728
729
730     return render_template_string('''
731     <div id="{{ ident }}" style="background-color: {{ color }}">
732         <div id="{{ ident }}_asignatura" class="subject" hx-get="/asignaturas?id={{ ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">
733             {{ asignatura }}
734         </div>
735         <div id="{{ ident }}_profesor" class="teacher" hx-get="/profesores?id={{ ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">
736             {{ profesor if profesor else "Profesor" }}
737         </div>
738         <div id="{{ ident }}_aula" class="classroom" hx-get="/aulas?id={{ ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">
739             {{ aula }}
740         </div>
741         <div id="{{ ident }}_sesion" class="session" hx-get="/sesion?id={{ ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">
742             {{ sesion }}
743         </div>
744         <!-- Botones de Copiar y Pegar -->
745         <div class="copy-paste-controls">
746             <button hx-post="/copiar?id={{ ident }}"
747                 hx-trigger="click"
748                 hx-on::after-request="this.style.display='none'">
749                 Copiar
750             </button>
751         </div>
752     </div>
753     ''', ident=identificador, asignatura=asignatura, profesor=profesor, aula=aula, sesion=sesion, color=color)
754

```

Figura 3-39 Ruta `/selección_sesión`

3.2.3.14 Ruta /deporte_comida

La ruta `/deporte_comida` (Figura 3-40) permite al usuario seleccionar entre Deporte o Comida al hacer clic en una celda específica del horario. Al activarse, genera dinámicamente un menú con ambas opciones utilizando HTMX.

```

761 @app.route('/deporte_comida')
762 def deporte_comida():
763     identificador = request.args.get('id')
764     |
765     return render_template_string('''
766     <div id="{{ ident }}" class="lunch-options">
767         <button hx-get="/seleccionar_opcion?id={{ ident }}&opcion=Deporte"
768             hx-target="#{{ ident }}"
769             hx-swap="outerHTML">
770             Deporte
771         </button>
772         <button hx-get="/seleccionar_opcion?id={{ ident }}&opcion=Comida"
773             hx-target="#{{ ident }}"
774             hx-swap="outerHTML">
775             Comida
776         </button>
777     </div>
778     ''', ident=identificador)
779

```

Figura 3-40 Ruta /deporte_comida

3.2.3.15 Ruta /seleccionar_opción

La ruta `/seleccionar_opcion` (Figura 3-41) permite registrar y actualizar la elección entre Deporte o Comida en una celda del horario. La opción seleccionada se almacena en `SELECCIONES`, garantizando su persistencia y su posterior exportación en CSV. Además, la celda sigue siendo interactiva, permitiendo al usuario cambiar su selección con un solo clic, manteniendo la coherencia con el resto del sistema.

```

782 @app.route('/seleccionar_opcion')
783 def seleccionar_opcion():
784     identificador = request.args.get('id')
785     opcion = request.args.get('opcion')
786     # Guardar la opción en SELECCIONES
787     SELECCIONES[identificador] = [opcion]
788
789     return render_template_string('''
790     <div id="{{ ident }}" class="lunch"
791         hx-get="/deporte_comida?id={{ ident }}"
792         hx-target="#{{ ident }}"
793         hx-swap="outerHTML">
794         {{ opcion }}
795     </div>
796     ''', ident=identificador, opcion=opcion)
797

```

Figura 3-41 Ruta /seleccionar_opción

3.2.3.16 Ruta /exportar_csv

La ruta `/exportar_csv`, reflejada en la Figura 3-42, tiene como objetivo generar y proporcionar un archivo CSV con la información estructurada de los horarios asignados en la aplicación. Este archivo puede ser descargado por los usuarios, permitiéndoles obtener una representación organizada y fácilmente procesable por otras aplicaciones (como Excel) de sus selecciones.

El proceso comienza con la creación de una lista `output`, donde se almacena la información en formato tabular. La primera fila corresponde a los encabezados del CSV, incluyendo las columnas: Día,

Horario, Grupo, Asignatura, Profesor, Aula, Sesión y Deporte/Comida, lo que facilita la interpretación del archivo resultante. Esta última columna se ha añadido para reflejar la nueva funcionalidad de selección entre Deporte y Comida, asegurando que esta información también pueda ser exportada junto con el resto de los datos del horario.

Luego, el *backend* recorre todas las entradas almacenadas en *SELECCIONES*, que contienen los datos seleccionados por el usuario en cada celda del horario. Para estructurar correctamente la información, se desglosa el identificador de la celda (*celda*), que sigue un formato específico, por ejemplo: *GIM_5_Lu1a_G1*. De este identificador se extraen varios elementos clave: el día de la semana (abreviado como "Lu", "Ma", etc.), el horario (identificado con códigos como "1a", "2a"), y el grupo (como "G1", "G2"). Con la ayuda de los diccionarios *DIAS_SEMANA* y *HORARIOS*, estos códigos son traducidos a su forma completa para mejorar la legibilidad en el archivo exportado.

Asimismo, para garantizar que los periodos de comida y deporte se reflejen correctamente en la exportación, se ha introducido un diccionario adicional que permite la traducción de los horarios especiales TC1 y TC2, convirtiéndolos en "13:10 - 14:10" y "14:20 - 15:20", respectivamente. De este modo, la estructura del CSV mantiene la coherencia con el resto del horario.

Una vez estructurada la información, se convierte la lista en una cadena de texto en formato CSV, donde cada fila se representa como una línea de valores separados por comas. Finalmente, Flask envía esta información como una respuesta HTTP con el tipo de contenido *text/csv*, permitiendo que el usuario descargue automáticamente el archivo *horario.csv*, solo mostrándose las celdas que han sido rellenas por el usuario.

```

798 @app.route('/exportar_csv')
799 def exportar_csv():
800     # Crear la respuesta HTTP con el contenido CSV
801     output = []
802     output.append(["Día", "Horario", "Grupo", "Asignatura", "Profesor", "Aula", "Sesión", "Deporte/Comida"]) # Agregamos la nueva columna
803
804     # Diccionario para traducir horarios TC1 y TC2
805     HORARIOS_ESPECIALES = {
806         "TC1": "13:10-14:10",
807         "TC2": "14:20-15:20"
808     }
809
810     # Recorrer todas las celdas guardadas en SELECCIONES
811     for celda, valores in SELECCIONES.items():
812         # Extraer los valores, asegurando que haya suficientes elementos
813         asignatura = valores[0] if len(valores) > 0 else ""
814         profesor = valores[1] if len(valores) > 1 else ""
815         aula = valores[2] if len(valores) > 2 else ""
816         sesion = valores[3] if len(valores) > 3 else ""
817         deporte_comida = valores[4] if len(valores) > 4 else "" # Nueva columna para deporte/comida
818
819         # Desglosar el ID de la celda (Ejemplo: GIM_5_Lu1a_G1 o GIM_5_LuTC1_G1)
820         partes = celda.split("_")
821
822         if len(partes) < 4:
823             continue # Si el formato del ID no es correcto, evitar errores
824
825         # Obtener y traducir el día de la semana
826         dia_abrev = partes[2][:2] # Extraer los dos primeros caracteres (Lu, Ma, etc.)
827         dia_completo = DIAS_SEMANA.get(dia_abrev, dia_abrev) # Traducir al nombre completo
828
829         # Obtener y traducir el horario (caso normal y caso especial TC1/TC2)
830         horario_abrev = partes[2][2:] # Extraer la parte del horario (1a, 2a, etc.)
831         horario_completo = HORARIOS.get(horario_abrev, HORARIOS_ESPECIALES.get(horario_abrev, horario_abrev)) # Traducir la hora
832
833         # Mantener el grupo (Ejemplo: G1, G2, etc.)
834         grupo = partes[3] # Extraer la última parte del ID
835
836         # Agregar la información al CSV con la nueva columna
837         output.append([dia_completo, horario_completo, grupo, asignatura, profesor, aula, sesion, deporte_comida])
838
839     # Crear el CSV en memoria
840     si = '\n'.join([','.join(map(str, row)) for row in output])
841
842     # Enviar el archivo CSV como respuesta
843     response = Response(si, mimetype="text/csv")
844     response.headers["Content-Disposition"] = "attachment; filename=horario.csv"
845     return response
846

```

Figura 3-42 Ruta /exportar_csv

3.2.3.17 Ruta `/resetear_selecciones`

La ruta `/resetear_selecciones` permite a los usuarios reiniciar todas las selecciones realizadas en la tabla de horarios. Su implementación es sencilla pero crucial para garantizar flexibilidad en la asignación de horarios y evitar errores en la planificación.

El funcionamiento de esta ruta se basa en vaciar completamente el diccionario `SELECCIONES`, eliminando todas las asignaciones de asignaturas, profesores, aulas y sesiones registradas hasta el momento. Al hacer esto, el sistema restablece el estado inicial, permitiendo a los usuarios comenzar de nuevo sin necesidad de eliminar manualmente cada celda.

Esta funcionalidad está estrechamente ligada con el *frontend*, donde está establecido que al recargar la página se dispara una solicitud hacia esta ruta. Una vez ejecutada la acción, se actualiza la interfaz, asegurando que todas las celdas vuelvan a su estado original.

```

642 @app.route('/reset_selecciones', methods=['POST'])
643 def reset_selecciones():
644     SELECCIONES.clear() # Vaciar el diccionario cuando se recarga la página
645     return '', 204 # Responder con un código HTTP 204 (sin contenido)

```

Figura 3-43 Ruta `/resetear_selecciones`

3.2.3.18 Ruta `/exportar_pdf`

La ruta `/exportar_pdf` en Flask (Figura 3-44) permite la generación y descarga de un PDF a partir de una captura de pantalla enviada desde el *frontend*. Esta funcionalidad es clave para exportar la vista del horario en un documento portable sin pérdida de calidad.

El proceso comienza con la recepción de la imagen en formato base64 a través de una solicitud POST desde JavaScript. Si no se recibe la imagen correctamente, se devuelve un error 400 (*Bad Request*) para evitar un procesamiento incorrecto. Luego, la imagen se decodifica y convierte en un objeto PIL, asegurando su compatibilidad para la inserción en el PDF.

Para evitar problemas de formato, la imagen se convierte a RGB y se guarda temporalmente en disco como `temp_image.png`, permitiendo su reapertura antes de ser integrada en el PDF. A continuación, se crea un documento PDF en memoria, ajustando la imagen para que encaje proporcionalmente en la página, dejando márgenes adecuados.

Una vez que el PDF es generado y guardado en memoria, se le asigna un nombre dinámico con la fecha actual (`sabana_YYYY-MM-DD.pdf`) y se envía como un archivo descargable al usuario con `send_file`. En caso de error, la ruta devuelve un código HTTP 500 (*Internal Server Error*) junto con la descripción del problema, garantizando un manejo adecuado de excepciones.

```

797 @app.route('/exportar_pdf', methods=['POST'])
798 def exportar_pdf():
799     try:
800         data = request.json # Recibir la imagen desde JavaScript
801         if not data or "image" not in data:
802             return jsonify({"error": "No se recibió la imagen"}), 400
803
804         # Decodificar la imagen base64
805         img_data = base64.b64decode(data["image"].split(",")[1])
806
807         # Convertir la imagen en un objeto PIL
808         img = Image.open(BytesIO(img_data))
809         img = img.convert("RGB") # Convertimos a RGB para evitar errores
810
811         # Guardar la imagen temporalmente en disco y volver a abrirla
812         img_temp_path = "temp_image.png"
813         img.save(img_temp_path, format="PNG")
814         img = Image.open(img_temp_path) # Volvemos a abrir la imagen desde el disco
815
816         # Crear un PDF en memoria
817         pdf_bytes = BytesIO()
818         pdf = canvas.Canvas(pdf_bytes, pagesize=A4)
819
820         # Obtener tamaño de la imagen y redimensionar para ajustarla al PDF
821         img_width, img_height = img.size
822         aspect_ratio = img_width / img_height
823         new_width = A4[0] - 40 # Dejamos márgenes de 20px a cada lado
824         new_height = new_width / aspect_ratio
825
826         if new_height > A4[1] - 40: # Si la imagen es muy alta, ajustarla a la altura del PDF
827             new_height = A4[1] - 40
828             new_width = new_height * aspect_ratio
829
830         # Insertar la imagen correctamente en el PDF desde el archivo temporal
831         pdf.drawInlineImage(img_temp_path, 20, 20, width=new_width, height=new_height)
832         pdf.save()
833         pdf_bytes.seek(0)
834
835         # Generar nombre del PDF con fecha
836         fecha_actual = datetime.datetime.now().strftime("%Y-%m-%d")
837         nombre_pdf = f"sabana_{fecha_actual}.pdf"
838
839         print("PDF generado correctamente.")
840         return send_file(pdf_bytes, mimetype="application/pdf", as_attachment=True, download_name=nombre_pdf)
841
842     except Exception as e:
843         return jsonify({"error": str(e)}), 500
844
845

```

Figura 3-44 Ruta /exportar_pdf

3.2.4 Comunicación entre el servidor y el navegador

El funcionamiento de la herramienta web se basa en la comunicación entre el servidor (*backend* en Flask) y el navegador (*frontend* con HTML, JavaScript y HTMX). Esta comunicación permite la manipulación del horario de manera dinámica sin necesidad de recargar la página. A continuación, se describen ejemplos de los dos tipos de rutas que encontramos en el código (*GET* y *POST*), la función del HTMX en la interfaz y la interacción entre JavaScript y Flask para la exportación del horario en PDF.

3.2.4.1 Rutas GET

Las rutas *GET* se utilizan para solicitar información del servidor. Para entenderlas mejor, se procede a explicar el flujo de comunicación que se lleva a cabo cuando el usuario hace clic en una celda del horario, en este caso, la aplicación solicita al servidor un fragmento HTML con un menú de opciones (Figura 3-29). El flujo sería el siguiente:

- 1 El usuario hace clic en una celda del horario.

- HTMX envía una solicitud *GET* al servidor Flask con el *id* de la celda seleccionada (Figura 3-45).

Estado	Método	Dominio	Archivo	Iniciador	Tipo
204	POST	127.0.0.1:5000	reset_selecciones	index.html:257 (fetch)	html
200	GET	127.0.0.1:5000	menu?id=GIM_5_Lu1a_G1	htmx.min.js:1 (xhr)	html
200	GET	127.0.0.1:5000	desplegar_menu?id=GIM_5_Lu1a_G1	htmx.min.js:1 (xhr)	html
200	GET	127.0.0.1:5000	menu?id=GIM_5_Lu1a_G1	htmx.min.js:1 (xhr)	html

Figura 3-45 Solicitud GET de HTMX al servidor Flask

- Flask procesa la solicitud y devuelve un fragmento HTML con las opciones disponibles (Figura 3-46).

```
127.0.0.1 - - [24/Mar/2025 23:44:20] "GET /menu?id=GIM_5_Lu1a_G1 HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2025 23:44:24] "GET /desplegar_menu?id=GIM_5_Lu1a_G1 HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2025 23:44:24] "GET /menu?id=GIM_5_Lu1a_G1 HTTP/1.1" 200 -
```

Figura 3-46 Respuesta del servidor Flask al navegador

- HTMX reemplaza el contenido de la celda en el navegador, mostrando el menú sin necesidad de recargar la página.

3.2.4.2 Rutas POST

Las rutas POST permiten enviar datos al servidor. Dentro del *backend*, como ejemplo, se observa la ruta */pegar* (Figura 3-31), la cual se utiliza para pegar la información dentro del horario de una celda a otra. Para entenderla mejor, se continúa explicando el flujo de información entre navegador y servidor.

- El usuario copia información de una celda y la pega en otra.
- El navegador envía una solicitud *POST* al servidor Flask con los datos pegados (Figura 3-47).

200	GET	127.0.0.1:5000	desplegar_menu?id=GIM_5_Lu1a_G2	htmx.min.js:1 (xhr)	html
200	GET	127.0.0.1:5000	menu?id=GIM_5_Lu1a_G2	htmx.min.js:1 (xhr)	html
200	POST	127.0.0.1:5000	pegar?id=GIM_5_Lu1a_G2	htmx.min.js:1 (xhr)	html

Figura 3-47 Solicitud POST del navegador al servidor Flask

- Flask almacena la información en la estructura de datos *SELECCIONES* y devuelve un fragmento HTML actualizado (Figura 3-48).

```
127.0.0.1 - - [25/Mar/2025 00:07:59] "GET /desplegar_menu?id=GIM_5_Lu1a_G2 HTTP/1.1" 200 -
127.0.0.1 - - [25/Mar/2025 00:07:59] "GET /menu?id=GIM_5_Lu1a_G2 HTTP/1.1" 200 -
127.0.0.1 - - [25/Mar/2025 00:08:00] "POST /pegar?id=GIM_5_Lu1a_G2 HTTP/1.1" 200 -
```

Figura 3-48 Respuesta del navegador

- HTMX reemplaza la celda en el navegador, mostrando los datos pegados sin recargar la página.

3.2.4.3 Interacción entre JavaScript y Flask

Cuando un usuario desea exportar el horario en PDF (Figura 3-14), se ejecuta un proceso en el cual JavaScript envía una solicitud al servidor Flask, que genera el archivo y lo devuelve al navegador para su descarga. Para explicar más en profundidad, se procede a desglosar el flujo de comunicación.

Primero, tiene lugar el flujo de comunicación entre navegador y servidor. En el cual, el usuario hace clic en el botón exportar a PDF, esto provoca que se active el código JavaScript y captura el contenido HTML de la tabla del horario. Después, el navegador envía una solicitud *POST* a Flask, en el que envía el contenido HTML del horario dentro de un objeto *FormData* (Figura 3-49).

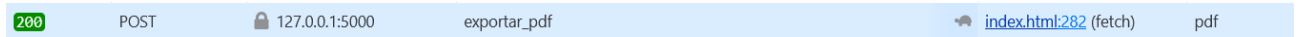


Figura 3-49 Solicitud POST del navegador al servidor para exportar en PDF

Posteriormente, Flask procesa la solicitud y genera el PDF. Para ello, primero recibe los datos, extrae el contenido HTML enviado en `request.form.get("html")` y lo convierte en un PDF con `weasyprint`. Después, Flask devuelve una respuesta HTTP con el archivo PDF (Figura 3-50), y por último el navegador recibe el PDF y lo descarga automáticamente.

```
PDF generado correctamente.  
127.0.0.1 - - [25/Mar/2025 00:25:36] "POST /exportar_pdf HTTP/1.1" 200 -
```

Figura 3-50 Respuesta del servidor con archivo PDF

4 RESULTADOS

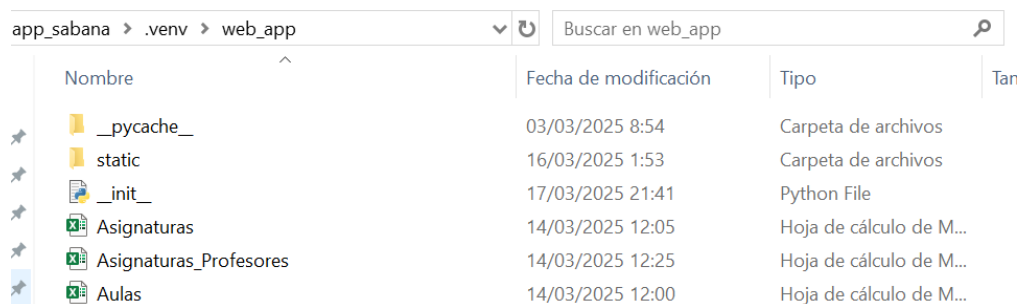
Una vez completado el desarrollo de la aplicación, se comprueban los resultados y funcionalidades usando la herramienta web para crear una sábana de ejemplo.

4.1 Inicializar la herramienta web

Para ejecutar la aplicación desarrollada en Flask, es necesario seguir un proceso de configuración que permite organizar correctamente los archivos del proyecto. En primer lugar, se crea un directorio de trabajo (*app_sabana*) y se accede a él desde el terminal. A continuación, se genera un entorno virtual con el comando `python -m venv .venv`, el cual facilita la gestión de dependencias del proyecto sin interferir con otras instalaciones de Python en el sistema. Para activar este entorno en Windows, se utiliza el comando `.venv\Scripts\Activate`, lo que permite instalar paquetes de manera aislada. Posteriormente, se instala Flask mediante `pip install Flask`, asegurando que el *framework* esté disponible en el entorno de desarrollo.

Una vez configurado el entorno, se crea una carpeta (*web_app*) para almacenar la aplicación, dentro de la cual se copia el archivo `__init__.py`, que contiene el código de inicialización de la aplicación, también se incluyen los archivos Excel que se leen con la librería Openpyxl. Además, se genera un directorio *static* dentro de *web_app*, donde se almacenarán los archivos HTML y otros recursos estáticos, como imágenes. Finalmente, el archivo `index.html` se copia a este directorio. Todos estos directorios se observan en la Figura 4-1.

Para ejecutar la aplicación Flask y obtener el resultado de la Figura 4-2, se emplea el comando `flask --app web_app run --debug`, lo que inicia un servidor web local en la dirección *localhost* y el puerto 5000, permitiendo acceder a la aplicación desde un navegador web.



Nombre	Fecha de modificación	Tipo	Tan
__pycache__	03/03/2025 8:54	Carpeta de archivos	
static	16/03/2025 1:53	Carpeta de archivos	
__init__.py	17/03/2025 21:41	Python File	
Asignaturas	14/03/2025 12:05	Hoja de cálculo de M...	
Asignaturas_Profesores	14/03/2025 12:25	Hoja de cálculo de M...	
Aulas	14/03/2025 12:00	Hoja de cálculo de M...	

Figura 4-1 Archivos necesarios para correr la herramienta

```
C:\Users\Raul Urbano Burrel\app_sabana\.venv\web_app>python __init__.py
* Serving Flask app '__init__'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 243-509-302
```

Figura 4-2 Ruta del servidor web local en el terminal

Lo único que faltaría por hacer, sería copiar y pegar la ruta `http://127.0.0.1:5000/static/index.html` en un navegador web, como, por ejemplo, Mozilla Firefox. Tras copiar, pegar y ejecutar la ruta en el navegador, aparecería la interfaz de la herramienta web con todas las celdas predeterminadas mostrando estudio, las franjas horarias y la sábana para cada día a rellenar, tal y como se muestra en la Figura 4-3.

Lunes	G1	G2	G3	G4	G5	G6	IM1	IM2
8:15h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
9:05h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
9:55h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
10:45h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
11:35h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
12:25h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
13:15h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
14:05h	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida
14:55h	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida	Deporte/Comida
15:45h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
16:35h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
17:25h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio
18:10h	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio	Estudio

Figura 4-3 Apariencia de la herramienta web al ejecutar la ruta local

4.2 Interacción con la herramienta web para rellenar una celda

Tras la inicialización, se va a ir mostrando qué ocurre según el usuario vaya interactuando con la interfaz

4.2.1 Cambiar estudio por una asignatura

Para cambiar en una celda el estudio por una asignatura, se procede a hacer doble clic izquierdo sobre el estudio que se desee cambiar (en este caso lunes a primera hora del grupo 1 de clase) y se mostraría el siguiente menú en la celda (Figura 4-4).

Lunes	G1
8:15h - 9:05h	Asignatura Profesor Aula Sesión

Figura 4-4 Menú en la celda

Como se ha establecido en las rutas del *backend*, no se puede elegir un campo sin haber seleccionado previamente el campo superior, es decir no se puede escoger profesor si primero no se ha elegido asignatura.

Posteriormente para acceder al listado de asignaturas seleccionables, se procede a hacer clic izquierdo sobre la palabra “Asignatura”, al hacerlo, se muestran una lista de botones con el nombre de las asignaturas (Figura 4-5).



Figura 4-5 Listado de asignaturas dentro de la celda

Por último, se procede a elegir una asignatura de entre todas las del listado, en este punto hay dos posibles opciones. Por un lado, que la asignatura seleccionada tenga un profesor asociado que la imparta, o por otro lado que la asignatura seleccionada no tenga un profesor asociado. Se procede a estudiar los resultados para ambas opciones.

4.2.1.1 Asignatura con profesor asociado

Cuando se selecciona una asignatura con un profesor asociado (en este caso “Oficina Técnica” (Figura 4-6)) se actualiza el nombre de la asignatura con su color de fondo en la celda correspondiente, y sigue ofreciendo las opciones de “profesor”, “aula” y “sesión”.



Figura 4-6 Asignatura con profesores asociados

4.2.1.2 Asignatura sin profesor asociado

Cuando se selecciona una asignatura sin ningún profesor asociado (en este caso “Actividades Militares”) se actualiza el nombre de la asignatura con su color de fondo en la celda al completo, y no ofrece las opciones “profesor”, “aula” y “sesión”, ya que solo se pretende rellenar la celda con la actividad seleccionada. En la Figura 4-7, se observan las diferencias dependiendo de la asignatura seleccionada.

Lunes	G1	G2
8:15h - 9:05h	Oficina Técnica Profesor Aula Sesión	Actividades Militares

Figura 4-7 Diferencia entre asignaturas seleccionadas

4.2.2 Escoger profesor

Para escoger profesor se procede a hacer un clic izquierdo sobre la palabra “Profesor”, a continuación (Figura 4-8), se listan los profesores disponibles para impartir la asignatura previamente seleccionada.

Lunes	G1
8:15h - 9:05h	Oficina Técnica Xavier Núñez Nieto Francisco Javier Rodríguez Rodríguez Aula Sesión

Figura 4-8 Listado de profesores asociados a la asignatura

Una vez escogido al profesor haciendo clic izquierdo sobre el botón correspondiente (en este caso “Xavier Núñez Nieto” (Figura 4-9)), se devuelve el valor a la celda correspondiente, mostrándose tanto la asignatura seleccionada como el profesor, y dando las opciones de escoger aula y sesión.

Lunes	G1
8:15h - 9:05h	Oficina Técnica Xavier Núñez Nieto Aula Sesión

Figura 4-9 Celda con asignatura y profesor seleccionado

4.2.3 Escoger aula

Tras haber escogido profesor se continúa con la selección de aula, para ello se hace un clic izquierdo sobre la palabra “Aula”, y se listan las aulas disponibles en la Escuela Naval Militar (Figura 4-10).

Lunes	G1
8:15h - 9:05h	Oficina Técnica Xavier Núñez Nieto
	PA1
	PA2
	PA3_INF
	PA4_INF
	PA5
	PA6
	PA7
	PA8
	PA9
	PA10
	PA11
	PA12
	SINTAC
	PA13_INF
	SIMNAV
	PA14_INF
	PA15_INF
	PA16
	PA17_INF
PA18_INF	
PA19	
PA20	

Figura 4-10 Listado de aulas ENM

Una vez se escoge el aula haciendo clic izquierdo sobre el botón correspondiente (en este caso “PA1” haciendo referencia al Aula 1 del Patio de Aulas (Figura 4-11)), se devuelve el valor a la celda correspondiente, mostrándose tanto la asignatura seleccionada como el profesor y el aula, y dando la opción de escoger sesión.

Lunes	G1
8:15h - 9:05h	Oficina Técnica Xavier Núñez Nieto PA1 Sesión

Figura 4-11 Celda con asignatura, profesor y aula seleccionada

4.2.4 Escoger sesión

Por último, faltaría seleccionar la sesión a impartir que corresponda, para ello se hará un clic izquierdo sobre la palabra “Sesión”, tras haber hecho clic se listan las tres opciones de sesión a impartir en botones: teoría, laboratorio y seminario (Figura 4-12).

Lunes	G1
8:15h - 9:05h	Oficina Técnica Xavier Núñez Nieto PA1 <div style="display: flex; flex-direction: column; align-items: center; gap: 10px;"> <div style="background-color: #4CAF50; color: white; padding: 5px 15px; border-radius: 5px;">Seminario</div> <div style="background-color: #4CAF50; color: white; padding: 5px 15px; border-radius: 5px;">Laboratorio</div> <div style="background-color: #4CAF50; color: white; padding: 5px 15px; border-radius: 5px;">Teoría</div> </div>

Figura 4-12 Listado de sesiones

Al hacer clic izquierdo sobre el botón de la sesión deseada (en este caso “Seminario”), se devuelve el valor a la celda correspondiente (Figura 4-13), de esta manera ya se mostraría al completo la información en la celda y se mostraría un botón en la interfaz, el cual permite copiar toda la información de la celda en un portapapeles para posteriormente poder pegar la información al completo en otra celda, al hacer clic izquierdo sobre el botón *copiar* desaparece.

Lunes	G1
8:15h - 9:05h	Oficina Técnica Xavier Núñez Nieto PA1 seminario <div style="background-color: #007bff; color: white; padding: 5px; display: inline-block; border-radius: 5px;"> Copiar </div>

Figura 4-13 Celda rellena al completo

4.2.5 Pegar información en otra celda

Para pegar la información en otra celda, simplemente habría que hacer clic izquierdo en una celda sin rellenar, al hacer clic aparecerá un botón que da la opción de pegar la información del portapapeles (Figura 4-14).

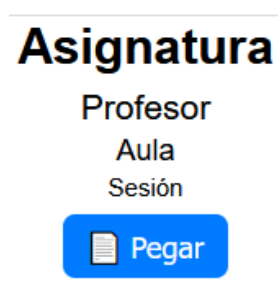


Figura 4-14 Interfaz antes de pegar información

Al hacer clic sobre el botón de pegar, se pega toda la información del portapapeles en la celda seleccionada (Figura 4-15).

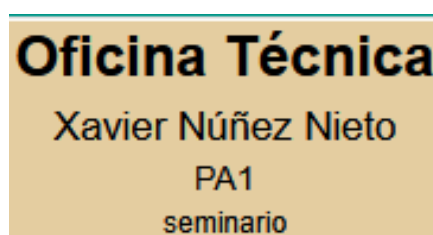


Figura 4-15 Celda al hacer clic sobre Pegar

4.3 Interacción con la herramienta web para exportar la sábana

Para observar los resultados al exportar la sábana, primero se procede a rellenar la sábana con una cierta cantidad de celdas completas para observar resultados concluyentes, se muestra en la Figura 4-16. Para ello se ha asignado las siguientes clases para los distintos grupos el lunes hasta la hora de la comida/deporte.

Lunes	G1	G2	G3	G4	G5	G6	IM1	IM2
8:15h - 9:05h	Oficina Técnica Xavier Núñez Nieto PA1 seminario	Actividades Militares	Teoría del Buque y Construcción Naval CF Gerardo González-Cela Echevarría PA17_INF laboratorio	Teoría del Buque y Construcción Naval CF Gerardo González-Cela Echevarría PA17_INF laboratorio	Fundamentos de Redes de Ordenadores Pablo Sendin Raña PA15_INF laboratorio	Fundamentos de Redes de Ordenadores Pablo Sendin Raña PA15_INF laboratorio	Operaciones Anfibia II CTE teoría	Operaciones Anfibia II CTE PA2 teoría
9:10h - 10:00h	Actividades Militares	Oficina Técnica Xavier Núñez Nieto PA1 seminario	Teoría del Buque y Construcción Naval CF Gerardo González-Cela Echevarría PA17_INF laboratorio	Teoría del Buque y Construcción Naval CF Gerardo González-Cela Echevarría PA17_INF laboratorio	Fundamentos de Redes de Ordenadores Pablo Sendin Raña PA15_INF laboratorio	Fundamentos de Redes de Ordenadores Pablo Sendin Raña PA15_INF laboratorio	Operaciones Anfibia II CTE PA2 teoría	Operaciones Anfibia II CTE PA2 teoría
10:20h - 11:10h	Táctica Naval III TN PA1 teoría	Táctica Naval III TN PA1 teoría	Táctica Naval III TN PA1 teoría	Táctica Naval III TN PA1 teoría	Sensores Navales Rubén Nocelo López PA2 teoría	Sensores Navales Rubén Nocelo López PA2 teoría	Sensores Navales Rubén Nocelo López PA2 teoría	Sensores Navales Rubén Nocelo López PA2 teoría
11:15h - 12:05h	Táctica Naval III TN PA1 teoría	Táctica Naval III TN PA1 teoría	Táctica Naval III TN PA1 teoría	Táctica Naval III TN PA1 teoría	Sensores Navales Rubén Nocelo López PA2 teoría	Sensores Navales Rubén Nocelo López PA2 teoría	Sensores Navales Rubén Nocelo López PA2 teoría	Sensores Navales Rubén Nocelo López PA2 teoría
12:10h - 13:00h	Sensores Navales Rubén Nocelo López PA1 teoría	Sensores Navales Rubén Nocelo López PA1 teoría	Sensores Navales Rubén Nocelo López PA1 teoría	Sensores Navales Rubén Nocelo López PA1 teoría	Táctica Naval III TN PA2 teoría	Táctica Naval III TN PA2 teoría	Estudio	Estudio
13:10h - 14:10h	Deporte	Deporte	Deporte	Deporte	Comida	Comida	Comida	Comida

Figura 4-16 Ejemplo de sábana rellena a exportar

4.3.1 Exportar sábana en archivo Excel formato CSV

Para exportar la sábana en un archivo CSV, se desliza la barra lateral hasta abajo y se observa el botón para generar el documento, se hace clic izquierdo sobre el botón y el servidor devuelve el archivo descargado (Figura 4-18) con el nombre de *horario.csv*.

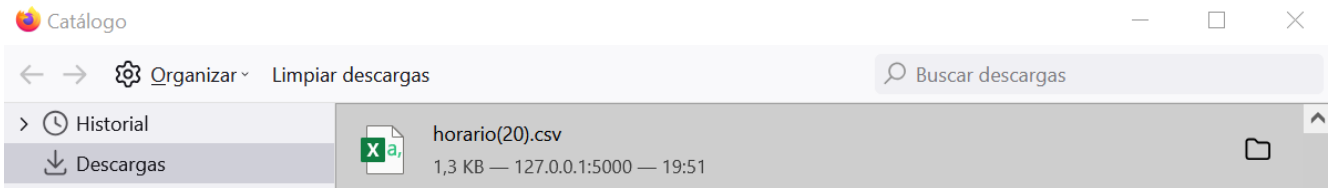


Figura 4-17 Archivo CSV con la sábana devuelto por el servidor

Para finalizar, se abre el archivo con la aplicación bloc de notas, y se puede observar la siguiente información ordenada en orden cronológico (Figura 4-18).

```
Día,Horario,Grupo,Asignatura,Profesor,Aula,Sesión,Deporte/Comida
Lunes,08:15-09:05,G1,Oficina Técnica,Xavier Núñez Nieto,PA1,seminario,
Lunes,09:10-10:00,G2,Oficina Técnica,Xavier Núñez Nieto,PA1,seminario,
Lunes,08:15-09:05,G2,Actividades Militares,None,None,,
Lunes,09:10-10:00,G1,Actividades Militares,None,None,,
Lunes,10:20-11:10,G1,Táctica Naval III,TN,PA1,teoría,
Lunes,10:20-11:10,G2,Táctica Naval III,TN,PA1,teoría,
Lunes,10:20-11:10,G3,Táctica Naval III,TN,PA1,teoría,
Lunes,10:20-11:10,G4,Táctica Naval III,TN,PA1,teoría,
Lunes,11:15-12:05,G1,Táctica Naval III,TN,PA1,teoría,
Lunes,11:15-12:05,G2,Táctica Naval III,TN,PA1,teoría,
Lunes,11:15-12:05,G3,Táctica Naval III,TN,PA1,teoría,
Lunes,11:15-12:05,G4,Táctica Naval III,TN,PA1,teoría,
Lunes,12:10-13:00,G1,Sensores Navales,Rubén Nocelo López,PA1,teoría,
Lunes,12:10-13:00,G2,Sensores Navales,Rubén Nocelo López,PA1,teoría,
Lunes,12:10-13:00,G3,Sensores Navales,Rubén Nocelo López,PA1,teoría,
Lunes,12:10-13:00,G4,Sensores Navales,Rubén Nocelo López,PA1,teoría,
Lunes,11:15-12:05,G5,Sensores Navales,Rubén Nocelo López,PA2,teoría,
Lunes,11:15-12:05,G6,Sensores Navales,Rubén Nocelo López,PA2,teoría,
Lunes,11:15-12:05,G7,Sensores Navales,Rubén Nocelo López,PA2,teoría,
Lunes,11:15-12:05,G8,Sensores Navales,Rubén Nocelo López,PA2,teoría,
Lunes,12:10-13:00,G5,Táctica Naval III,TN,PA2,teoría,
Lunes,12:10-13:00,G6,Táctica Naval III,TN,PA2,teoría,
Lunes,12:10-13:00,G7,Estudio,None,None,,
Lunes,12:10-13:00,G8,Estudio,None,None,,
```

Figura 4-18 Bloc de notas exportado con la sábana

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Desde la fase inicial, se identificaron los problemas asociados a la gestión manual del horario, tales como la complejidad en la asignación de aulas, profesores y asignaturas, los cambios habituales de planificación y la falta de una herramienta interactiva. Con base en esta problemática, se establecieron objetivos específicos que guiaron el desarrollo del proyecto, entre los que destacan la elección de tecnologías adecuadas, la creación de una base de datos y la implementación de una interfaz intuitiva para facilitar la gestión de los horarios.

La implementación del sistema se estructuró en tres partes principales: el almacenamiento y procesamiento de datos en Excel, la lógica de negocio a través de Flask y la visualización e interacción en la web mediante una interfaz dinámica (HTML, CSS y HTMX).

La herramienta creada cumple con el objetivo general de servir de apoyo y a la gestión de la planificación horaria de la sábana en la ENM. Por otra parte, también se han cumplido los objetivos específicos. La herramienta facilita la creación y consulta del horario a la persona encargada de hacer la sábana, dejando de lado de esta manera la confección de la sábana en un archivo Excel.

Además de la posibilidad de exportar los horarios en formato CSV y PDF facilita la documentación y distribución de este, asegurando que los usuarios puedan disponer de la información en distintos formatos según sus necesidades.

5.2 Líneas futuras

En cuanto a las líneas futuras y limitaciones, existen aspectos que podrían mejorarse en futuros proyectos. Entre ellos, se encuentra la posibilidad de integrar una base de datos como MySQL, la implementación de un sistema de autenticación para diferentes tipos de usuarios y la ampliación de la base de datos para incluir todos los cursos y planes de estudios, no solo 5° ST.

Actualmente, la herramienta está diseñada con una estructura específica que permite gestionar la sábana de quinto curso sin titulación, pero la ENM cuenta con alumnos de diferentes promociones, escalas y planes de estudios. Para adaptar la aplicación a esta realidad, sería necesario expandir la base de datos para incluir estos diferentes planes de estudio.

La ampliación requeriría una adaptación del *backend*, estableciendo criterios de filtrado y segmentación que permitan seleccionar cursos específicos dentro de la interfaz. Esto significaría desarrollar nuevas rutas que procesen los datos en función del plan de estudios y el curso del alumno. Por ejemplo, se podrían filtrar asignaturas, aulas y profesores en función del curso a planificar, estableciendo desde el principio esas restricciones.

La implementación de un sistema de autenticación de usuarios sería un paso fundamental en la transición a un entorno multiusuario. En este momento, la herramienta permite la edición del horario sin restricciones, pero en una futura versión sería necesario integrar un sistema de roles y permisos, donde ciertos usuarios puedan modificar el horario como administradores, mientras que los alumnos y profesores solo tengan permisos de lectura y consultas específicas.

Otra mejora clave en futuras versiones de la aplicación sería su despliegue en un servidor accesible desde la red, en lugar de depender exclusivamente de una ejecución local. Actualmente, la herramienta se ejecuta en un servidor Flask en modo de desarrollo, lo que limita su uso a la máquina donde se aloja. Para hacer que la aplicación esté disponible para más usuarios, sería necesario implementar un despliegue en un servidor de producción, utilizando tecnologías como Apache, permitiendo que profesores y alumnos puedan acceder al sistema desde cualquier dispositivo dentro de la red de la ENM.

Por último, podría resultar útil también añadir las siguientes nuevas funcionalidades:

- Control de versiones de la sábana, basándose en la funcionalidad de exportar a CSV y usando herramientas como Git.
- Importar una sábana previamente guardada en un archivo CSV, lo que permitiría, por ejemplo, empezar a trabajar sobre una sábana tipo, en lugar de empezar con la sábana vacía.
- Verificación de colisiones en la asignación de aulas y profesores, de tal manera que no sea posible asignar, por error, la misma aula a dos clases distintas, ni tampoco asignar clases al mismo profesor en sitios distintos a la vez.
- Verificar que se han asignado todas las sesiones previstas en una semana para un grupo, por ejemplo, detectar si a uno de los grupos no se le asigna un seminario que le corresponde esa semana.

6 BIBLIOGRAFÍA

- [1] «TIOBE Index», TIOBE. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.tiobe.com/tiobe-index/>
- [2] «Applications for Python», Python.org. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.python.org/about/apps/>
- [3] «▷ El lenguaje de programación C++ **【actualizado marzo 2025】**», Lenguajes de programación. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://lenguajesdeprogramacion.net/cpp/>
- [4] «¿Que es Java? | IBM». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.ibm.com/mx-es/topics/java>
- [5] M. Benito, «Descubre las características de la programación en C», FP Online. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://fp.uoc.fje.edu/blog/descubre-las-caracteristicas-de-la-programacion-en-c/>
- [6] «JavaScript | MDN». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [7] «HTML5 Tutorial - An Ultimate Guide for Beginners». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.tutorialrepublic.com/html-tutorial/>
- [8] «HTML: Lenguaje de etiquetas de hipertexto | MDN». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTML>
- [9] «The Essentials of HTML: How to Write, Learn & Use It». Accedido: 24 de marzo de 2025. [En línea]. Disponible en: <https://blog.hubspot.com/website/html>
- [10] «Extensible Markup Language (XML)». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.w3.org/XML/>
- [11] «CommonMark». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://commonmark.org/>
- [12] «openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.1.3 documentation». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://openpyxl.readthedocs.io/en/stable/>
- [13] «Pandas : La biblioteca de Python dedicada a la Data Science», Formación en ciencia de datos | DataScientest.com. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://datascientest.com/es/pandas-python>
- [14] «XlsxWriter: A Python module for creating Excel XLSX files.» Python. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://github.com/jmcnamara/XlsxWriter>

- [15] «La Historia De Excel: Desde Su Lanzamiento Hasta Convertirse En Una Herramienta Esencial En El Mundo Empresarial | Aula 10 Centro De Formación». Accedido: 20 de marzo de 2025. [En línea]. Disponible en: <https://aula10formacion.com/blog/la-historia-de-excel/>
- [16] «Welcome to Flask — Flask Documentation (3.1.x)». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://flask.palletsprojects.com/en/stable/>
- [17] «CSS | MDN». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
- [18] «HTMX: La tecnología HTML para crear apps web avanzadas - Vicente SG». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://vicentesg.com/htmx-la-tecnologia-html-para-crear-apps-web-avanzadas/>
- [19] A. Navarro, «Jinja2 en Flask: Introducción - Programación», Junco TIC. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://juncotic.com/jinja2-en-flask-introduccion/>
- [20] «ReportLab». 19 de septiembre de 2024. Accedido: 15 de marzo de 2025. [En línea]. Disponible en: <https://www.reportlab.com/docs/reportlab-userguide.pdf>
- [21] «Smarter und schneller Planen – mit Untis, der führenden Stundenplansoftware.», Untis. Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.untis.at/es>
- [22] «aSc Horario de clases - horario software». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://www.asctimetables.com/>
- [23] «Open Timetables». Accedido: 13 de marzo de 2025. [En línea]. Disponible en: <https://opentimetable.tu-ilmenau.de/>
- [24] «Multi-week timetable», Untis. Accedido: 24 de marzo de 2025. [En línea]. Disponible en: <https://www.untis.at/en/products/untis-basic-software/multi-week-timetable-1>
- [25] J. L. Tamarit Cutiño y M. Santiesteban Díaz, «Asistente de ayuda para la confección de horarios docentes». junio de 2007. Accedido: 8 de febrero de 2025. [En línea]. Disponible en: https://repositorio.uci.cu/jspui/bitstream/ident/TD_0346_07/1/TD_0346_07.pdf?
- [26] J. Albaladejo Carreño, «Sistema de gestión de información en la ENM», Trabajo de Fin de Grado, Centro Universitario de la Defensa, Escuela Naval Militar, 2023.
- [27] F. J. Ruiz Fontán, «Desarrollo de una herramienta web para optimización de procesos académicos en la ENM», Trabajo de Fin de Grado, Centro Universitario de la Defensa, Escuela Naval Militar, 2022.
- [28] «Download Python», Python.org. Accedido: 23 de marzo de 2025. [En línea]. Disponible en: <https://www.python.org/downloads/>
- [29] G. Villar Rodríguez y G. Leira Neira, *14 BOTONES MANUAL DEL ASPIRANTE*. 2014.

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

La gestión eficiente de los horarios en una institución de formación militar como la Escuela Naval Militar (ENM) tiene un impacto significativo en diversos ámbitos más allá del ámbito académico y organizativo. Un sistema bien estructurado y optimizado no solo mejora la formación de los futuros oficiales de la Armada Española, sino que también repercute en la sociedad, la economía y el medioambiente.

Un horario correctamente diseñado garantiza que los futuros oficiales de la Armada reciban una formación estructurada y sin interrupciones, lo que mejora su preparación y competencia profesional. Esto es especialmente importante en el contexto militar, donde la disciplina, la organización y la eficiencia en la gestión del tiempo son valores fundamentales. Un mejor sistema de planificación horaria:

- Reduce los conflictos en la distribución de clases y entrenamientos, asegurando que los alumnos puedan aprovechar al máximo su tiempo de aprendizaje.
- Permite una mejor coordinación entre las actividades académicas y la instrucción militar, facilitando un desarrollo integral del alumnado.
- Contribuye a la formación de oficiales altamente cualificados que, una vez egresados, desempeñarán un papel clave en la defensa y seguridad del país, impactando directamente en la sociedad española y europea.

Además, la mejora en la organización de horarios evita el estrés y la sobrecarga de trabajo en el personal encargado de la planificación, beneficiando tanto a docentes como a estudiantes y promoviendo un entorno educativo más saludable.

Desde un punto de vista económico, una mejor planificación horaria en la ENM supone una optimización de recursos y reducción de costes. La digitalización del sistema de horarios mediante la herramienta web propuesta en este TFG tiene ventajas significativas:

- Ahorro de tiempo y esfuerzo en la gestión de horarios, reduciendo el trabajo manual y minimizando errores administrativos.
- Optimización del uso de infraestructuras como aulas, laboratorios y simuladores, evitando la infrutilización o saturación de espacios.

Estos factores se traducen en un impacto económico positivo tanto para la ENM como para la administración pública, que destina recursos a la formación de los futuros oficiales.

Desde la perspectiva ambiental, la digitalización y automatización de la planificación horaria permite reducir el consumo de recursos físicos, contribuyendo a una institución más sostenible, como por ejemplo la disminución del uso de papel y tinta al sustituir los horarios impresos por una plataforma digital.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

La correcta planificación y gestión del tiempo es un pilar fundamental en la formación de cualquier individuo, especialmente en entornos de alta exigencia como la Escuela Naval Militar (ENM). Un horario bien estructurado no solo optimiza el rendimiento académico y profesional, sino que también tiene importantes implicaciones éticas y sociales en la vida de los estudiantes y la comunidad en general.

Desde un punto de vista ético, la planificación eficiente del tiempo fomenta la responsabilidad, la disciplina y la equidad. Garantizar que todos los alumnos tengan acceso a una distribución justa de las horas de clase, adiestramientos y descansos es una cuestión de justicia organizativa. Un sistema de horarios transparente evita favoritismos, minimiza la confusión y promueve una formación estructurada y equitativa para todos los estudiantes.

Además, una correcta organización del tiempo respalda el bienestar del individuo. La sobrecarga de actividades o la descoordinación de horarios pueden generar fatiga, estrés y una disminución del rendimiento. Un horario equilibrado permite a los alumnos gestionar mejor su tiempo, favoreciendo la concentración, el aprendizaje efectivo y un mejor equilibrio entre la vida académica y personal.

Desde una perspectiva social, la planificación del tiempo tiene un impacto directo en la productividad y la eficacia de las instituciones. En el caso de la ENM, una mejor estructuración de los horarios contribuye a la formación de oficiales más preparados, lo que, a su vez, beneficia a la sociedad española y europea. Oficiales con una formación más eficiente podrán desempeñar mejor sus funciones en la defensa y la seguridad nacional, asegurando un servicio público más competente y profesional.

Por otro lado, un sistema de horarios bien diseñado también impacta en la cohesión social y la colaboración dentro de la institución. Una estructura organizada facilita la comunicación entre profesores y alumnos, mejora la convivencia en el entorno académico y promueve un sentido de cooperación y trabajo en equipo, valores esenciales en cualquier ámbito profesional, especialmente en el ámbito militar.

En conclusión, la correcta gestión del tiempo no solo es un aspecto técnico de la administración académica, sino que también responde a principios éticos y valores sociales fundamentales. Implementar un sistema eficiente de horarios no solo mejora el funcionamiento de la ENM, sino que también contribuye a la formación de individuos más organizados, responsables y preparados para servir a la sociedad de manera humana y justa.

ANEXO III: CÓDIGO BACKEND – FLASK (__INIT__.PY)

```

import os, openpyxl, csv # Importamos módulos para interactuar con el sistema de
archivos (os) y manejar archivos CSV (csv)

from flask import Flask, request, render_template_string, Response, send_file,
jsonify # Importamos funciones esenciales de Flask
from collections import defaultdict
from reportlab.lib.pagesizes import landscape, A4
from reportlab.pdfgen import canvas
from io import BytesIO
import datetime
import base64
from PIL import Image

app = Flask(__name__) # Inicializamos la aplicación Flask
app.secret_key = "apRh0hNiTnMfwUc8+4AEcQ==" # Clave secreta para manejar sesiones
seguras en Flask

SELECCIONES = {} # Diccionario donde se almacenarán las selecciones de
asignaturas, profesores y aulas de cada celda

# Función para cargar asignaturas desde el archivo Excel
def cargar_asignaturas_excel(ruta_excel):
    """
    Carga las asignaturas y sus colores desde un archivo Excel (.xlsx) usando
    OpenPyXL.

    Parámetros:
    ruta_excel (str): Ruta del archivo Excel a leer.

    Retorna:
    dict: Diccionario con asignaturas como claves y colores en formato hexadecimal
    como valores.
    """
    # Cargar el archivo Excel
    workbook = openpyxl.load_workbook(ruta_excel)
    sheet = workbook.active # Seleccionar la hoja activa

    # Obtener los nombres de las columnas
    column_headers = [cell.value for cell in sheet[1]]

    # Buscar las columnas correctas
    if "Nombre Completo" in column_headers and "Código RGB" in column_headers:
        col_nombre = column_headers.index("Nombre Completo")
        col_color = column_headers.index("Código RGB")

        # Leer los datos fila por fila (ignorando encabezados)
        asignaturas = {
            row[col_nombre]: row[col_color]
            for row in sheet.iter_rows(min_row=2, values_only=True)
            if row[col_nombre] and row[col_color] # Evitar filas vacías
        }
        return asignaturas
    else:
        raise ValueError("El archivo Excel no contiene las columnas requeridas.")

# Ruta del archivo Excel con asignaturas
RUTA_EXCEL_ASIGNATURAS = "Asignaturas.xlsx"

```

```

# Cargar asignaturas dinámicamente desde el Excel
ASIGNATURAS = cargar_asignaturas_excel(RUTA_EXCEL_ASIGNATURAS)

def cargar_asignaturas_profesores_excel(ruta_excel):
    """
    Carga la relación Asignatura ↔ Profesores desde un archivo Excel (.xlsx)
    usando OpenPyXL.

    Parámetros:
    ruta_excel (str): Ruta del archivo Excel a leer.

    Retorna:
    dict: Diccionario con asignaturas como claves y listas de profesores como
    valores.
    """
    # Cargar el archivo Excel
    workbook = openpyxl.load_workbook(ruta_excel)
    sheet = workbook.active # Seleccionar la hoja activa

    # Obtener los nombres de las columnas
    column_headers = [cell.value for cell in sheet[1]]

    # Verificar que existen al menos dos columnas (Asignatura y Profesor)
    if "Nombre asignatura" in column_headers and "Nombre profesor" in
column_headers:
        col_asignatura = column_headers.index("Nombre asignatura")
        col_profesor = column_headers.index("Nombre profesor")

        # Diccionario para almacenar la relación Asignatura ↔ Profesores
        asignaturas_profesores = defaultdict(list)

        # Leer los datos fila por fila (ignorando encabezados)
        for row in sheet.iter_rows(min_row=2, values_only=True):
            asignatura = row[col_asignatura]
            profesor = row[col_profesor]

            if asignatura and profesor: # Evitar filas vacías
                asignaturas_profesores[asignatura].append(profesor)

        return dict(asignaturas_profesores)

    else:
        raise ValueError("El archivo Excel no contiene las columnas requeridas.")

# Ruta del archivo Excel con Asignaturas ↔ Profesores
RUTA_EXCEL_ASIGNATURAS_PROFESORES = "Asignaturas_Profesores.xlsx"

# Cargar la relación dinámicamente desde el Excel
ASIGNATURAS_PROFESORES =
cargar_asignaturas_profesores_excel(RUTA_EXCEL_ASIGNATURAS_PROFESORES)

def cargar_aulas_excel(ruta_excel):
    """
    Carga las aulas desde un archivo Excel (.xlsx) usando OpenPyXL.

    Parámetros:
    ruta_excel (str): Ruta del archivo Excel a leer.

    Retorna:

```

```

dict: Diccionario con nombres de aula como claves y códigos de aula como
valores.
"""
# Cargar el archivo Excel
workbook = openpyxl.load_workbook(ruta_excel)
sheet = workbook.active # Seleccionar la hoja activa

# Obtener los nombres de las columnas
column_headers = [cell.value for cell in sheet[1]]

# Verificar que hay al menos dos columnas para leer
if len(column_headers) < 2:
    raise ValueError("El archivo Excel no tiene suficientes columnas.")

# Leer los datos de la primera y segunda columna (pero intercambiando clave-
valor)
aulas = {
    row[1]: row[0] # Ahora el nombre del aula es la clave, y el código es el
valor
    for row in sheet.iter_rows(min_row=2, values_only=True)
    if row[0] and row[1] # Evitar filas vacías
}
return aulas

# Ruta del archivo Excel con AULAS
RUTA_EXCEL_AULAS = "Aulas.xlsx"

# Cargar aulas dinámicamente desde el Excel
AULAS = cargar_aulas_excel(RUTA_EXCEL_AULAS)

# Tipos de sesión disponibles
SESIONES = {
    "seminario": "Seminario",
    "laboratorio": "Laboratorio",
    "teoria": "Teoría"
}

# Diccionario para traducir los días de la semana
DIAS_SEMANA = {
    "Lu": "Lunes",
    "Ma": "Martes",
    "Mi": "Miércoles",
    "Ju": "Jueves",
    "Vi": "Viernes"
}

# Diccionario para traducir los horarios en el ID
HORARIOS = {
    "1a": "08:15-09:05",
    "2a": "09:10-10:00",
    "3a": "10:20-11:10",
    "4a": "11:15-12:05",
    "5a": "12:10-13:00",
    "6a": "13:10-14:00",
    "7a": "15:30-16:20",
    "8a": "16:25-17:15",
    "9a": "17:20-18:10"
}

@app.route('/menu')
def mostrar_menu():

```

```

identificador = request.args.get('id') # Obtener la celda seleccionada
return render_template_string('''
<div id="{{ ident }}"
    hx-get="/desplegar_menu?id={{ ident }}"
    hx-target="#{{ ident }}"
    hx-swap="outerHTML">
    Estudio <!-- Texto inicial -->
</div>
''', ident=identificador)

@app.route('/desplegar_menu')
def desplegar_menu():
    identificador = request.args.get('id') # Obtener la celda seleccionada

    return render_template_string('''
<div id="{{ ident }}"> <!-- Contenedor principal con un identificador único
para la celda -->

    <!-- Sección para mostrar la lista de asignaturas -->
    <div id="{{ ident }}_asignatura" class="subject"
        hx-get="/asignaturas?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Asignatura
    </div>

    <!-- Sección para seleccionar un profesor (se actualizará dinámicamente) -
->
    <div id="{{ ident }}_profesor" class="teacher"
        hx-get="/profesores?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Profesor
    </div>

    <!-- Sección para seleccionar un aula (se actualizará dinámicamente) -->
    <div id="{{ ident }}_aula" class="classroom"
        hx-get="/aulas?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Aula
    </div>

    <!-- Sección para seleccionar la sesión -->
    <div id="{{ ident }}_sesion" class="session"
        hx-get="/sesion?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Sesión
    </div>

    <!-- Botones para Copiar y Pegar -->
    <div class="copy-paste-controls">
        <button hx-post="/pegar?id={{ ident }}"
            hx-trigger="click"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            F&Pegar
        </button>
    </div>
</div>
''', ident=identificador)

```

```

PORTAPAPELES = None # Variable para almacenar temporalmente la información
copiada

@app.route('/copiar', methods=['POST'])
def copiar():
    global PORTAPAPELES
    celda_id = request.args.get('id')

    if celda_id in SELECCIONES and SELECCIONES[celda_id]: # Solo copiar si la
celda tiene datos
        PORTAPAPELES = SELECCIONES[celda_id].copy() # Copiamos toda la
información

        return "", 204 # No devolvemos contenido (HTMX no necesita respuesta)

    return "", 204 # Si la celda está vacía, tampoco mostramos nada

@app.route('/pegar', methods=['POST'])
def pegar():
    global PORTAPAPELES
    celda_id = request.args.get('id')

    if not PORTAPAPELES:
        return "", 204 # No hacemos nada si no hay información copiada

    # Si la celda NO existe en SELECCIONES, la creamos como vacía antes de pegar
    if celda_id not in SELECCIONES:
        SELECCIONES[celda_id] = ["", "", "", ""] # Se inicializa la celda con
valores vacíos

    SELECCIONES[celda_id] = PORTAPAPELES.copy() # Pegamos la información copiada

    # Obtener la información de la celda para actualizar la interfaz
    asignatura, profesor, aula, sesion = SELECCIONES[celda_id]
    color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Color de la asignatura

    return render_template_string('''
<div id="{{ ident }}" class="study" style="background-color: {{ color }};">
    <div id="{{ ident }}_asignatura" class="subject">
        {{ asignatura if asignatura else "Asignatura" }}
    </div>
    <div id="{{ ident }}_profesor" class="teacher">
        {{ profesor if profesor else "Profesor" }}
    </div>
    <div id="{{ ident }}_aula" class="classroom">
        {{ aula if aula else "Aula" }}
    </div>
    <div id="{{ ident }}_sesion" class="session">
        {{ sesion if sesion else "Sesión" }}
    </div>
</div>
</div>
''', ident=celda_id, asignatura=asignatura, profesor=profesor, aula=aula,
sesion=sesion, color=color)

# Ruta que devuelve un listado de asignaturas disponibles
@app.route('/asignaturas')
def listar_asignaturas():
    # Obtener el identificador de la celda con la que se está interactuando
    identificador = request.args.get('id')

```

```

# Retornamos una plantilla HTML con la lista de asignaturas disponibles
return render_template_string('''
<div id="{{ ident }}"> <!-- Contenedor principal con un identificador único
para la celda -->

    <!-- Sección para mostrar la lista de asignaturas -->
    <div id="{{ ident }}_asignatura">
        {% for asignatura in asignaturas %} <!-- Iteramos sobre todas las
asignaturas disponibles -->
            <button
                hx-get="/seleccion_asignatura?id={{ ident }}&asignatura={{
asignatura }}"
                hx-target="#{{ ident }}"
                hx-swap="outerHTML">
                    {{ asignatura }} <!-- Mostramos el nombre de la asignatura en
el botón -->
            </button><br/> <!-- Insertamos un salto de línea entre cada botón
-->
        {% endfor %}
    </div>

    <!-- Sección para seleccionar un profesor (se actualizará dinámicamente) -
->
    <div id="{{ ident }}_profesor"
        hx-get="/profesores"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Profesor <!-- Texto inicial antes de seleccionar un profesor -->
    </div>

    <!-- Sección para seleccionar un aula (se actualizará dinámicamente) -->
    <div id="{{ ident }}_aula"
        hx-get="/aulas"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Aula <!-- Texto inicial antes de seleccionar un aula -->
    </div>

    <!-- Sección para seleccionar la sesión -->
    <div id="{{ ident }}_sesion"
        hx-get="/sesion?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Sesión
    </div>
</div>
''', asignaturas=ASIGNATURAS, ident=identificador) # Pasamos las asignaturas
y el identificador como variables a la plantilla

@app.route('/seleccion_asignatura')
def seleccionar_asignatura():
    identificador = request.args.get('id') # Identificador de la celda
    asignatura = request.args.get('asignatura') # Asignatura seleccionada

    # Definir color de la asignatura o blanco si no está en el diccionario
    color = ASIGNATURAS.get(asignatura, "#FFFFFF")

    # Actualizar el diccionario SELECCIONES
    if identificador in SELECCIONES:
        SELECCIONES[identificador][0] = asignatura # Sobrescribir asignatura

```

```

else:
    SELECCIONES[identificador] = [asignatura, None, None]

# Comprobar si la asignatura tiene profesores asignados
tiene_profesor = asignatura in ASIGNATURAS_PROFESORES and
ASIGNATURAS_PROFESORES[asignatura]

# Si la asignatura NO tiene profesor, solo actualizar la celda con la
asignatura
if not tiene_profesor:
    return render_template_string('''
        <div id="{{ ident }}" style="background-color: {{ bg_color }}">
            <div id="{{ ident }}_asignatura" class="subject">
                {{ asig }}
            </div>
        </div>
    ''', asig=asignatura, ident=identificador, bg_color=color)

# Si la asignatura SÍ tiene profesor, mostrar las opciones completas
return render_template_string('''
    <div id="{{ ident }}" style="background-color: {{ bg_color }}">
        <!-- Mostrar la asignatura seleccionada en la celda -->
        <div id="{{ ident }}_asignatura" class="subject"
            hx-get="/asignaturas?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ asig }}
        </div>

        <!-- Sección para la selección de profesores -->
        <div id="{{ ident }}_profesor" class="teacher"
            hx-get="/profesores?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            Profesor
        </div>

        <!-- Sección para la selección de aulas -->
        <div id="{{ ident }}_aula" class="classroom"
            hx-get="/aulas?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            Aula
        </div>

        <!-- Sección para seleccionar la sesión -->
        <div id="{{ ident }}_sesion"
            hx-get="/sesion?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            Sesión
        </div>
    </div>
    ''', asig=asignatura, ident=identificador, bg_color=color)

# Definir la ruta que maneja la visualización de los profesores disponibles
@app.route('/profesores')
def listar_profesores():
    # Obtener el identificador de la celda que hizo la solicitud
    identificador = request.args.get('id')

```

```

# Obtener la asignatura seleccionada previamente en la celda correspondiente
asignatura = SELECCIONES.get(identificador, [None])[0] # Extrae solo la
asignatura

# Si no hay asignatura seleccionada, no se muestran los profesores y se
devuelve una respuesta vacía
if asignatura is None:
    return "", 204 # Código 204 indica "No Content", evitando errores en la
interfaz

# Obtener la lista de profesores asignados a la asignatura seleccionada
profesores = ASIGNATURAS_PROFESORES.get(asignatura, [])

# Obtener el color de fondo de la celda según la asignatura seleccionada
color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Si no hay color definido,
usa blanco por defecto

# Generar y devolver el HTML que se insertará en la interfaz
return render_template_string('''
<div id="{{ ident }}" style="background-color: {{ color }}"> <!-- Se mantiene
el color de fondo de la asignatura -->

    <!-- Mostrar la asignatura seleccionada en la celda -->
    <div id="{{ ident }}_asignatura"
        hx-get="/asignaturas?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        {{ asignatura }}
    </div>

    <!-- Sección para mostrar la lista de profesores disponibles -->
    <div id="{{ ident }}_profesor">
        {% for profesor in profesores %} <!-- Itera sobre la lista de
profesores disponibles -->
            <button hx-get="/seleccion_profesor?id={{ ident }}&profesor={{
profesor }}"
                hx-target="#{{ ident }}"
                hx-swap="outerHTML">
                {{ profesor }} <!-- Muestra el nombre del profesor en un
botón -->
            </button><br/> <!-- Inserta un salto de línea después de cada
botón -->
        {% endfor %}
    </div>

    <!-- Sección para seleccionar el aula -->
    <div id="{{ ident }}_aula"
        hx-get="/aulas?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Aula
    </div>

    <!-- Sección para seleccionar la sesión -->
    <div id="{{ ident }}_sesion"
        hx-get="/sesion?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Sesión
    </div>

</div>
''', profesores=profesores, ident=identificador, asignatura=asignatura,
color=color)

```

```

# Definir la ruta que maneja la selección de un profesor
@app.route('/seleccion_profesor')
def seleccionar_profesor():
    # Obtener el identificador de la celda que hizo la solicitud
    identificador = request.args.get('id')

    # Obtener el nombre del profesor seleccionado desde la solicitud GET
    profesor = request.args.get('profesor')

    # Verificar si la celda ya está en SELECCIONES (es decir, si ya tiene una
    asignatura asignada)
    if identificador in SELECCIONES:
        SELECCIONES[identificador][1] = profesor # Asigna el profesor en la
        posición 1 del array
    else:
        # Si la celda aún no ha sido registrada en SELECCIONES, se crea una nueva
        entrada con:
        # - None en la posición de asignatura (porque aún no se ha seleccionado
        ninguna)
        # - El profesor seleccionado
        # - None en la posición de aula
        SELECCIONES[identificador] = [None, profesor, None]

    # Obtener la asignatura almacenada en SELECCIONES para esta celda (si existe)
    asignatura = SELECCIONES.get(identificador, [None])[0]

    # Obtener el color correspondiente a la asignatura seleccionada (si no hay
    asignatura, se usa blanco)
    color = ASIGNATURAS.get(asignatura, "#FFFFFF")

    # Generar y devolver el HTML actualizado para la celda del horario
    return render_template_string('''
        <div id="{{ ident }}" style="background-color: {{ color }}">

            <!-- Mostrar la asignatura seleccionada (si hay una) -->
            <div id="{{ ident }}_asignatura" class="subject"
                hx-get="/asignaturas?id={{ ident }}"
                hx-target="#{{ ident }}"
                hx-swap="outerHTML">
                {{ asignatura }}
            </div>

            <!-- Mostrar el profesor seleccionado en la celda -->
            <div id="{{ ident }}_profesor" class="teacher"
                hx-get="/profesores?id={{ ident }}"
                hx-target="#{{ ident }}"
                hx-swap="outerHTML">
                {{ profesor }}
            </div>

            <!-- Sección para seleccionar un aula -->
            <div id="{{ ident }}_aula" class="classroom"
                hx-get="/aulas?id={{ ident }}"
                hx-target="#{{ ident }}"
                hx-swap="outerHTML">
                Aula
            </div>

            <!-- Sección para seleccionar la sesión -->
            <div id="{{ ident }}_sesion"
                hx-get="/sesion?id={{ ident }}"
                hx-target="#{{ ident }}"
    ''')

```

```

        hx-swap="outerHTML">
        Sesión
    </div>

</div>
''' , profesor=profesor, ident=identificador, asignatura=asignatura,
color=color)

# Definir la ruta que maneja la visualización de las aulas disponibles
@app.route('/aulas')
def listar_aulas():
    # Obtener el identificador de la celda que hizo la solicitud
    identificador = request.args.get('id')

    # Obtener la asignatura seleccionada previamente en la celda correspondiente
    asignatura = SELECCIONES.get(identificador, [None])[0] # Extrae solo la
asignatura

    # Obtener el profesor seleccionado previamente en la celda correspondiente
    profesor = SELECCIONES.get(identificador, [None, None])[1] # Extrae solo el
profesor

    # Si no hay asignatura o profesor seleccionados, no se muestran las aulas y se
devuelve una respuesta vacía
    if asignatura is None or profesor is None:
        return "", 204 # Código 204 indica "No Content", evitando errores en la
interfaz

    # Obtener el color de fondo de la celda según la asignatura seleccionada
    color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Si no hay color definido,
usa blanco por defecto

    # Generar y devolver el HTML que se insertará en la interfaz
    return render_template_string('''
    <div id="{{ ident }}" style="background-color: {{ color }}"> <!-- Se mantiene
el color de fondo de la asignatura -->

        <!-- Mostrar la asignatura seleccionada en la celda -->
        <div id="{{ ident }}_asignatura"
            hx-get="/asignaturas?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ asignatura }}
        </div>

        <!-- Mostrar el profesor seleccionado en la celda -->
        <div id="{{ ident }}_profesor"
            hx-get="/profesores?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ profesor if profesor else "Profesor" }} <!-- Si no hay profesor,
mostrar "Profesor" como marcador de posición -->
        </div>

        <!-- Sección para mostrar la lista de aulas disponibles -->
        <div id="{{ ident }}_aula">
            {% for aula_id, aula_nombre in aulas.items() %} <!-- Itera sobre la
lista de aulas disponibles -->
                <button hx-get="/seleccion_aula?id={{ ident }}&aula={{ aula_id }}"
                    hx-target="#{{ ident }}"
                    hx-swap="outerHTML">
                    {{ aula_id }} <!-- Muestra el código del aula en un botón -->

```

```

        </button><br/> <!-- Inserta un salto de línea después de cada
botón -->
        {% endfor %}
    </div>
    <!-- Sección para seleccionar la sesión -->
    <div id="{{ ident }}_sesion"
        hx-get="/sesion?id={{ ident }}"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Sesión
    </div>

</div>
''' , aulas=AULAS, ident=identificador, asignatura=asignatura,
profesor=profesor, color=color)

# Definir la ruta que maneja la selección de un aula
@app.route('/seleccion_aula')
def seleccionar_aula():
    # Obtener el identificador de la celda que hizo la solicitud
    identificador = request.args.get('id')

    # Obtener el código del aula seleccionada desde la solicitud GET
    aula = request.args.get('aula')

    # Verificar si la celda ya tiene información guardada en SELECCIONES
    if identificador in SELECCIONES:
        SELECCIONES[identificador][2] = aula # Se asigna el aula en la posición
[2] del array
    else:
        # Si la celda aún no ha sido registrada en SELECCIONES, se crea una nueva
entrada con:
        # - None en la posición de asignatura (porque aún no se ha seleccionado
ninguna)
        # - None en la posición de profesor (porque aún no se ha seleccionado
ninguno)
        # - El aula seleccionada
        SELECCIONES[identificador] = [None, None, aula]

    # Obtener la asignatura almacenada en SELECCIONES para esta celda (si existe)
    asignatura = SELECCIONES.get(identificador, [None])[0]

    # Obtener el profesor almacenado en SELECCIONES para esta celda (si existe)
    profesor = SELECCIONES.get(identificador, [None, None])[1]

    # Obtener el color correspondiente a la asignatura seleccionada (si no hay
asignatura, se usa blanco)
    color = ASIGNATURAS.get(asignatura, "#FFFFFF")

    # Generar y devolver el HTML actualizado para la celda del horario
    return render_template_string('''
        <div id="{{ ident }}" style="background-color: {{ color }}">

        <!-- Mostrar la asignatura seleccionada en la celda -->
        <div id="{{ ident }}_asignatura" class="subject"
            hx-get="/asignaturas?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ asignatura }}
        </div>

        <!-- Mostrar el profesor seleccionado en la celda -->

```

```

        <div id="{{ ident }}" class="teacher"
            hx-get="/profesores?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ profesor if profesor else "Profesor" }} <!-- Si no hay profesor,
mostrar "Profesor" como marcador de posición -->
        </div>

        <!-- Mostrar el aula seleccionada en la celda -->
        <div id="{{ ident }}" class="classroom"
            hx-get="/aulas?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ aula }}
        </div>
        <!-- Sección para seleccionar la sesión -->
        <div id="{{ ident }}"_sesion"
            hx-get="/sesion?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            Sesión
        </div>

    </div>
    ''' , aula=aula, ident=identificador, asignatura=asignatura, profesor=profesor,
color=color)

@app.route('/sesion')
def listar_sesion():
    identificador = request.args.get('id') # Obtener la celda donde se hizo clic
    # Obtener la asignatura seleccionada previamente en la celda correspondiente
    asignatura = SELECCIONES.get(identificador, [None])[0] # Extrae solo la
asignatura

    # Obtener el profesor seleccionado previamente en la celda correspondiente
    profesor = SELECCIONES.get(identificador, [None, None])[1] # Extrae solo el
profesor
    # Obtener el aula seleccionado previamente en la celda correspondiente
    aula = SELECCIONES.get(identificador, [None, None, None])[2] # Extrae solo el
aula
    # Si no hay asignatura, profesor o aula seleccionados, no se muestran las
sesiones y se devuelve una respuesta vacía
    if asignatura is None or profesor is None or aula is None:
        return "", 204 # Código 204 indica "No Content", evitando errores en la
interfaz

    # Obtener el color de fondo de la celda según la asignatura seleccionada
    color = ASIGNATURAS.get(asignatura, "#FFFFFF") # Si no hay color definido,
usa blanco por defecto
    return render_template_string('''
        <div id="{{ ident }}" style="background-color: {{ color }}"> <!-- Se mantiene
el color de fondo de la asignatura -->

        <!-- Mostrar la asignatura seleccionada en la celda -->
        <div id="{{ ident }}"_asignatura"
            class="subject"
            hx-get="/asignaturas?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ asignatura }}
        </div>

        <!-- Mostrar el profesor seleccionado en la celda -->
    
```

```

        <div id="{{ ident }}"_profesor"
            class="teacher"
            hx-get="/profesores?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ profesor if profesor else "Profesor" }} <!-- Si no hay profesor,
mostrar "Profesor" como marcador de posición -->
        </div>

        <div id="{{ ident }}"_aula" class="classroom"
            hx-get="/aulas?id={{ ident }}"
            hx-target="#{{ ident }}"
            hx-swap="outerHTML">
            {{ aula }}
        </div>

        <div id="{{ ident }}"_sesion">
        {% for clave, nombre in sesiones.items() %}
            <button hx-get="/seleccionar_sesion?id={{ ident }}&sesion={{ clave }}"
                hx-target="#{{ ident }}"
                hx-swap="outerHTML">
                {{ nombre }}
            </button><br/>
        {% endfor %}
        </div>
''' , sesiones=SESIONES, ident=identificador, aula=aula, asignatura=asignatura,
profesor=profesor, color=color)

@app.route('/seleccionar_sesion')
def seleccionar_sesion():
    identificador = request.args.get('id') # Leer la celda seleccionada
    sesion = request.args.get('sesion') # Leer la sesión seleccionada

    # Guardar la sesión en la selección
    if identificador in SELECCIONES:
        # Asegurar que la lista tiene al menos 4 elementos
        while len(SELECCIONES[identificador]) < 4:
            SELECCIONES[identificador].append(None) # Agrega 'None' hasta que
tenga 4 posiciones
        SELECCIONES[identificador][3] = sesion # Ahora sí podemos asignar la
sesión
    else:
        SELECCIONES[identificador] = [None, None, None, sesion] # Se asegura de
que siempre haya 4 elementos
        # Obtener la asignatura almacenada en SELECCIONES para esta celda (si existe)
        asignatura = SELECCIONES.get(identificador, [None])[0]
        # Obtener el profesor almacenado en SELECCIONES para esta celda (si existe)
        profesor = SELECCIONES.get(identificador, [None, None])[1]
        # Obtener el aula almacenada en SELECCIONES para esta celda (si existe)
        aula = SELECCIONES.get(identificador, [None, None, None])[2]
        # Obtener el color correspondiente a la asignatura seleccionada (si no hay
asignatura, se usa blanco)
        color = ASIGNATURAS.get(asignatura, "#FFFFFF")

    return render_template_string('''
        <div id="{{ ident }}" style="background-color: {{ color }}">
            <div id="{{ ident }}"_asignatura" class="subject" hx-
get="/asignaturas?id={{ ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">
                {{ asignatura }}
            </div>
            <div id="{{ ident }}"_profesor" class="teacher" hx-
get="/profesores?id={{ ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">

```

```

        {{ profesor if profesor else "Profesor" }}
    </div>
    <div id="{{ ident }}"_aula" class="classroom" hx-get="/aulas?id={{
ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">
        {{ aula }}
    </div>
    <div id="{{ ident }}"_sesion" class="session" hx-get="/sesion?id={{
ident }}" hx-target="#{{ ident }}" hx-swap="outerHTML">
        {{ sesion }}
    </div>
    <!-- Botones de Copiar y Pegar -->
    <div class="copy-paste-controls">
        <button hx-post="/copiar?id={{ ident }}"
            hx-trigger="click"
            hx-on::after-request="this.style.display='none'">
            FvCopiar
        </button>
    </div>
</div>
''' , ident=identificador, asignatura=asignatura, profesor=profesor, aula=aula,
sesion=sesion, color=color)

@app.route('/deporte_comida')
def deporte_comida():
    identificador = request.args.get('id')

    return render_template_string('''
<div id="{{ ident }}" class="lunch-options">
    <button hx-get="/seleccionar_opcion?id={{ ident }}&opcion=Deporte"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Deporte
    </button>
    <button hx-get="/seleccionar_opcion?id={{ ident }}&opcion=Comida"
        hx-target="#{{ ident }}"
        hx-swap="outerHTML">
        Comida
    </button>
</div>
''' , ident=identificador)

@app.route('/seleccionar_opcion')
def seleccionar_opcion():
    identificador = request.args.get('id')
    opcion = request.args.get('opcion')
    # Guardar la opción en SELECCIONES
    SELECCIONES[identificador] = [opcion]

    return render_template_string('''
<div id="{{ ident }}" class="lunch"
    hx-get="/deporte_comida?id={{ ident }}"
    hx-target="#{{ ident }}"
    hx-swap="outerHTML">
    {{ opcion }}
</div>
''' , ident=identificador, opcion=opcion)

```

```
@app.route('/exportar_csv')
def exportar_csv():
    # Crear la respuesta HTTP con el contenido CSV
    output = []
    output.append(["Día", "Horario", "Grupo", "Asignatura", "Profesor", "Aula",
                  "Sesión", "Deporte/Comida"]) # Agregamos la nueva columna

    # Diccionario para traducir horarios TC1 y TC2
    HORARIOS_ESPECIALES = {
        "TC1": "13:10-14:10",
        "TC2": "14:20-15:20"
    }

    # Recorrer todas las celdas guardadas en SELECCIONES
    for celda, valores in SELECCIONES.items():
        # Extraer los valores, asegurando que haya suficientes elementos
        asignatura = valores[0] if len(valores) > 0 else ""
        profesor = valores[1] if len(valores) > 1 else ""
        aula = valores[2] if len(valores) > 2 else ""
        sesion = valores[3] if len(valores) > 3 else ""
        deporte_comida = valores[4] if len(valores) > 4 else "" # Nueva columna
        para deporte/comida

        # Desglosar el ID de la celda (Ejemplo: GIM_5_Lula_G1 o GIM_5_LuTC1_G1)
        partes = celda.split("_")

        if len(partes) < 4:
            continue # Si el formato del ID no es correcto, evitar errores

        # Obtener y traducir el día de la semana
        dia_abrev = partes[2][:2] # Extraer los dos primeros caracteres (Lu, Ma,
        etc.)
        dia_completo = DIAS_SEMANA.get(dia_abrev, dia_abrev) # Traducir al nombre
        completo

        # Obtener y traducir el horario (caso normal y caso especial TC1/TC2)
        horario_abrev = partes[2][2:] # Extraer la parte del horario (1a, 2a,
        etc.)
        horario_completo = HORARIOS.get(horario_abrev,
        HORARIOS_ESPECIALES.get(horario_abrev, horario_abrev)) # Traducir la hora

        # Mantener el grupo (Ejemplo: G1, G2, etc.)
        grupo = partes[3] # Extraer la última parte del ID

        # Agregar la información al CSV con la nueva columna
        output.append([dia_completo, horario_completo, grupo, asignatura,
        profesor, aula, sesion, deporte_comida])

    # Crear el CSV en memoria
    si = '\n'.join([''.join(map(str, row)) for row in output])

    # Enviar el archivo CSV como respuesta
    response = Response(si, mimetype="text/csv")
    response.headers["Content-Disposition"] = "attachment; filename=horario.csv"
    return response

@app.route('/exportar_pdf', methods=['POST'])
def exportar_pdf():
    try:
        data = request.json # Recibir la imagen desde JavaScript
        if not data or "image" not in data:
```

```

    return jsonify({"error": "No se recibió la imagen"}), 400

# Decodificar la imagen base64
img_data = base64.b64decode(data["image"].split(",")[1])

# Convertir la imagen en un objeto PIL
img = Image.open(BytesIO(img_data))
img = img.convert("RGB") # Convertimos a RGB para evitar errores

# Guardar la imagen temporalmente en disco y volver a abrirla
img_temp_path = "temp_image.png"
img.save(img_temp_path, format="PNG")
img = Image.open(img_temp_path) # Volvemos a abrir la imagen desde el
disco

# Crear un PDF en memoria
pdf_bytes = BytesIO()
pdf = canvas.Canvas(pdf_bytes, pagesize=A4)

# Obtener tamaño de la imagen y redimensionar para ajustarla al PDF
img_width, img_height = img.size
aspect_ratio = img_width / img_height
new_width = A4[0] - 40 # Dejamos márgenes de 20px a cada lado
new_height = new_width / aspect_ratio

if new_height > A4[1] - 40: # Si la imagen es muy alta, ajustarla a la
altura del PDF
    new_height = A4[1] - 40
    new_width = new_height * aspect_ratio

# Insertar la imagen correctamente en el PDF desde el archivo temporal
pdf.drawInlineImage(img_temp_path, 20, 20, width=new_width,
height=new_height)
pdf.save()
pdf_bytes.seek(0)

# Generar nombre del PDF con fecha
fecha_actual = datetime.datetime.now().strftime("%Y-%m-%d")
nombre_pdf = f"sabana_{fecha_actual}.pdf"

print("PDF generado correctamente.")
return send_file(pdf_bytes, mimetype="application/pdf",
as_attachment=True, download_name=nombre_pdf)

except Exception as e:
    return jsonify({"error": str(e)}), 500

@app.route('/reset_selecciones', methods=['POST'])
def reset_selecciones():
    SELECCIONES.clear() # Vaciar el diccionario cuando se recarga la página
    return '', 204 # Responder con un código HTTP 204 (sin contenido)

# Iniciar la aplicación Flask en modo debug
if __name__ == '__main__':
    app.run(debug=True)

```

ANEXO IV: CÓDIGO FRONTEND – HTML (INDEX.HTML)

```

<!doctype html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/htmx/1.9.6/htmx.min.js"></script>
</head>

<style type="text/css">

/* ===== Estilos Generales ===== */
  body {
    font-family: 'Arial', sans-serif; /* Fuente limpia y legible */
    background-color: #f4f4f4; /* Color de fondo gris claro para mejorar la
legibilidad */
    margin: 20px; /* Espaciado alrededor del contenido */
    text-align: center; /* Alinea el texto del cuerpo al centro */
  }

  /* ===== Estilos de la Tabla ===== */
  table.horarios {
    width: 90%; /* Ocupa el 90% del ancho de la pantalla */
    margin: auto; /* Centra la tabla en la página */
    border-collapse: collapse; /* Elimina el espacio entre bordes de celdas */
    border-radius: 10px; /* Bordos redondeados */
    overflow: hidden; /* Evita que el contenido sobresalga de los bordes
redondeados */
    background-color: #ffffff; /* Fondo blanco para mayor contraste */
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1); /* Sombra para efecto visual
atractivo */
  }

  /* ===== Estilos para las Celdas ===== */
  table.horarios td {
    width: 300px; /* Establece un ancho uniforme para las celdas */
    height: 60px; /* Altura de las celdas */
    text-align: center; /* Alineación del contenido en el centro */
    font-size: 10pt; /* Tamaño de fuente adecuado */
    border: 1px solid #ddd; /* Bordos sutiles en gris claro */
    transition: background-color 0.3s ease, transform 0.2s; /* Animación suave
al pasar el cursor */
  }

  /* ===== Efecto Hover en las Celdas ===== */
  table.horarios td:hover {
    background-color: #e0e0e0; /* Cambia el color de fondo al pasar el ratón
*/
    transform: scale(1.05); /* Aumenta ligeramente el tamaño */
  }

  /* ===== Estilo para las Celdas de Hora ===== */
  table.horarios td.Hora {
    background-color: #00BFFF; /* Color azul celeste */
    color: white; /* Texto en blanco para contraste */
    font-weight: bold; /* Texto en negrita */
  }

  /* ===== Encabezado de la Tabla ===== */
  tr.Header td {

```

```

    background-color: #009688; /* Color de fondo verde azulado */
    color: white; /* Texto en blanco */
    font-size: 12pt; /* Aumenta el tamaño de la fuente */
    font-weight: bold; /* Hace el texto más llamativo */
    padding: 10px; /* Añade espacio dentro de las celdas del encabezado */
}

/* ===== Estilos de los Botones ===== */
button {
    padding: 8px 12px; /* Espaciado interno */
    margin: 5px; /* Espaciado exterior */
    border: none; /* Elimina bordes */
    border-radius: 5px; /* Bordes redondeados */
    background-color: #4CAF50; /* Color verde */
    color: white; /* Texto en blanco */
    cursor: pointer; /* Cambia el cursor a puntero */
    transition: background 0.3s ease; /* Efecto de cambio de color al pasar el
ratón */
}

button:hover {
    background-color: #45a049; /* Color más oscuro al pasar el ratón */
}

/* ===== Botón para eliminar selección ===== */
button.delete {
    background-color: #d9534f; /* Rojo */
}

button.delete:hover {
    background-color: #c9302c; /* Rojo más oscuro al pasar el ratón */
}

/* Contenedor de los botones */
.copy-paste-controls {
    bottom: 5px;
    right: 5px;
    gap: 5px;
}

/* Botón de Copiar */
.copy-paste-controls button:first-child {
    background-color: #2f00ff; /* Naranja */
    color: white;
    border: none;
    border-radius: 5px;
    padding: 5px 10px;
    font-size: 12px;
    cursor: pointer;
}

/* Botón de Pegar */
.copy-paste-controls button:last-child {
    background-color: #007bff; /* Azul */
    color: white;
    border: none;
    border-radius: 5px;
    padding: 5px 10px;
    font-size: 12px;
    cursor: pointer;
}

/* Efecto Hover para hacerlos más visibles */

```

```

.copy-paste-controls button: hover {
  filter: brightness(1.2);
}

.export-button {
  display: inline-block;
  padding: 10px 15px;
  background-color: #007BFF;
  color: white;
  text-decoration: none;
  border-radius: 5px;
  font-size: 14px;
}

.export-button: hover {
  background-color: #0056b3;
}

/* ===== Animaciones y Transiciones ===== */
@keyframes fadeIn {
  from {
    opacity: 0; /* Comienza con opacidad cero */
    transform: translateY(-5px); /* Se mueve ligeramente hacia arriba */
  }
  to {
    opacity: 1; /* Se muestra completamente */
    transform: translateY(0); /* Regresa a su posición normal */
  }
}

button {
  animation: fadeIn 1s ease-in-out; /* Aplica la animación de aparición */
}

/* ===== Tamaños de Fuente Diferenciados ===== */
.subject {
  font-size: 20px; /* Mayor tamaño para la asignatura */
  font-weight: bold; /* En negrita */
  margin-bottom: 5px; /* Espaciado inferior */
}

.teacher {
  font-size: 14px; /* Tamaño intermedio para profesor */
  margin-bottom: 3px; /* Espaciado inferior */
}

/* ===== Estilo para el Título Principal ===== */
.titulo {
  font-family: 'Arial', sans-serif; /* Fuente profesional */
  font-size: 28px; /* Tamaño llamativo */
  font-weight: bold; /* Texto en negrita */
  color: #2c3e50; /* Azul oscuro elegante */
  text-align: center; /* Centrado */
  margin-top: 20px; /* Espaciado superior */
  margin-bottom: 20px; /* Espaciado inferior */
  padding-bottom: 10px; /* Espaciado interno */
  border-bottom: 3px solid #2980b9; /* Subrayado con color azul */
  text-transform: uppercase; /* Convierte el texto en mayúsculas */
  letter-spacing: 1px; /* Espaciado entre letras */
}

/* ===== Estilo para Aula ===== */
.classroom {

```

```

    font-size: 12px; /* Tamaño menor para aula */
    margin-bottom: 2px; /* Espaciado inferior */
}

/* ===== Estilo para Sesión ===== */
.session {
    font-size: 10px; /* Tamaño más pequeño para sesiones */
}

/* ===== Estilo para el Contenido de las Celdas (Estudio) ===== */
.study {
    font-size: 14pt; /* Tamaño adecuado para lectura */
}

/* ===== Estilo para el Botón de Exportar CSV ===== */
.export-btn {
    background-color: #007BFF; /* Azul vibrante */
    color: white; /* Texto en blanco */
    border: none; /* Sin borde */
    padding: 10px 20px; /* Espaciado interno */
    font-size: 16px; /* Tamaño de fuente cómodo */
    border-radius: 5px; /* Bordes redondeados */
    cursor: pointer; /* Cursor de puntero */
    margin-top: 20px; /* Espaciado superior */
}

.export-btn:hover {
    background-color: #0056b3; /* Azul más oscuro al pasar el ratón */
}

/* Contenedor de las imágenes */
.contenedor {
    display: flex;
    justify-content: center; /* Centra horizontalmente */
    align-items: center; /* Alinea verticalmente */
    gap: 20px; /* Espacio entre imágenes */
    padding: 20px; /* Espaciado interno */
    background: rgba(255, 255, 255, 0.8); /* Fondo semitransparente para
destacar imágenes */
    border-radius: 10px; /* Bordes redondeados */
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2); /* Sombra suave */
}

/* Estilo de las imágenes reducidas */
.imagen {
    width: 200px; /* Tamaño más pequeño */
    height: auto; /* Mantiene la proporción */
    border-radius: 10px; /* Bordes redondeados */
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.3); /* Sombra ligera */
}

.watermark {
    width: 30px; /* Tamaño reducido */
    height: auto;
    opacity: 0.5; /* Efecto de marca de agua */
    filter: grayscale(100%); /* Hace que la imagen sea menos llamativa */
}

.lunch {
    background-color: #0077b6; /* Azul mar */
    color: white; /* Texto en blanco */
    font-weight: bold;
    text-align: center;
    padding: 10px;
    border-radius: 8px; /* Bordes redondeados */
    font-size: 14px;
}

```

```
        transition: all 0.3s ease-in-out; /* Animación suave */
        cursor: pointer;
        box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2); /* Sombra sutil */
    }

    /* Efecto hover: hace que la celda resalte un poco */
    .lunch:hover {
        background-color: #005f87; /* Azul más oscuro al pasar el cursor */
        transform: scale(1.05); /* Aumenta ligeramente el tamaño */
    }

</style>
<script>
window.addEventListener("load", function() {
    fetch("/reset_selecciones", { method: "POST" }); // Borra SELECCIONES
al cargar la página
    });
</script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/1.4.1/html2canvas.min.js">
</script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script
>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/1.4.1/html2canvas.min.js">
</script>

<script>
document.addEventListener("DOMContentLoaded", function() {
    let botonExportar = document.getElementById("exportar-pdf");

    if (botonExportar) {
        botonExportar.addEventListener("click", function() {
            console.log("Iniciando captura de pantalla...");

            html2canvas(document.body, {
                scale: 2, // Mejor resolución
                useCORS: true, // Permitir imágenes externas (CORS)
                allowTaint: true, // Forzar carga de imágenes externas
            }).then(canvas => {
                console.log("Captura de pantalla realizada.");

                let imgData = canvas.toDataURL("image/png");

                // Enviar imagen a Flask
                fetch("/exportar_pdf", {
                    method: "POST",
                    body: JSON.stringify({ image: imgData }),
                    headers: { "Content-Type": "application/json" }
                })
                .then(response => response.blob())
                .then(blob => {
                    let link = document.createElement("a");
                    link.href = URL.createObjectURL(blob);
                    link.download = "sabana.pdf";
                    link.click();
                    console.log("PDF descargado.");
                })
                .catch(error => console.error("Error en Fetch:", error));
            });
        });
    } else {
```

```

        console.error("Error: No se encontró el botón exportar-pdf.");
    }
});
</script>

<body>
<h1 class="titulo">Sábana General Alumnos 5° Curso CGA-IM ST</h1>
<div class="contenedor">
    
    
    
</div>
    <table class="horarios" id="GIM_5_Lu">
    <tr
class="Header"><td>Lunes</td><td>G1</td><td>G2</td><td>G3</td><td>G4</td><td>G5</t
d><td>G6</td><td>IM1</td><td>IM2</td></tr>
    <tr id="GIM_5_Lula"> <!-- Fila completa con el ID correcto -->
    <!-- Hora -->
    <td class="Hora" id="Lula">8:15h<br/> -<br/> 9:05h</td>
    <!-- Celda principal con asignatura, profesor y aula -->
    <td>
        <div id="GIM_5_Lula_G1"
            class="study" hx-get="/menu?id=GIM_5_Lula_G1"
            hx-target="#GIM_5_Lula_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lula_G2"
            class="study" hx-get="/menu?id=GIM_5_Lula_G2"
            hx-target="#GIM_5_Lula_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lula_G3"
            class="study" hx-get="/menu?id=GIM_5_Lula_G3"
            hx-target="#GIM_5_Lula_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lula_G4"
            class="study" hx-get="/menu?id=GIM_5_Lula_G4"
            hx-target="#GIM_5_Lula_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lula_G5"
            class="study" hx-get="/menu?id=GIM_5_Lula_G5"
            hx-target="#GIM_5_Lula_G5" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lula_G6"
            class="study" hx-get="/menu?id=GIM_5_Lula_G6"
            hx-target="#GIM_5_Lula_G6" hx-swap="innerHTML">
            Estudio
    </div>
    </td>
</table>

```

```

</div>
</td>
<td>
  <div id="GIM_5_Lula_G7"
    class="study" hx-get="/menu?id=GIM_5_Lula_G7"
    hx-target="#GIM_5_Lula_G7" hx-swap="innerHTML">
    Estudio
  </div>

</td>
<td>
  <div id="GIM_5_Lula_G8"
    class="study" hx-get="/menu?id=GIM_5_Lula_G8"
    hx-target="#GIM_5_Lula_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>

</tr>
  <tr id="GIM_5_Lu2a">
    <td class="Hora" id="Lu2a">9:10h<br/> -<br/> 10:00h</td>
      <td>
        <div id="GIM_5_Lu2a_G1"
          class="study" hx-get="/menu?id=GIM_5_Lu2a_G1"
          hx-target="#GIM_5_Lu2a_G1" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Lu2a_G2"
          class="study" hx-get="/menu?id=GIM_5_Lu2a_G2"
          hx-target="#GIM_5_Lu2a_G2" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Lu2a_G3"
          class="study" hx-get="/menu?id=GIM_5_Lu2a_G3"
          hx-target="#GIM_5_Lu2a_G3" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Lu2a_G4"
          class="study" hx-get="/menu?id=GIM_5_Lu2a_G4"
          hx-target="#GIM_5_Lu2a_G4" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Lu2a_G5"
          class="study" hx-get="/menu?id=GIM_5_Lu2a_G5"
          hx-target="#GIM_5_Lu2a_G5" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Lu2a_G6"
          class="study" hx-get="/menu?id=GIM_5_Lu2a_G6"
          hx-target="#GIM_5_Lu2a_G6" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
    </tr>
  </td>
  </td>
  <td>
    <div id="GIM_5_Lu2a_G6"
      class="study" hx-get="/menu?id=GIM_5_Lu2a_G6"
      hx-target="#GIM_5_Lu2a_G6" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
</tr>

```

```

<td>
  <div id="GIM_5_Lu2a_G7"
    class="study" hx-get="/menu?id=GIM_5_Lu2a_G7"
    hx-target="#GIM_5_Lu2a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Lu2a_G8"
    class="study" hx-get="/menu?id=GIM_5_Lu2a_G8"
    hx-target="#GIM_5_Lu2a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_Lu3a">
<td class="Hora" id="Lu3a">10:20h<br/> -<br/> 11:10h</td>
<td>
  <div id="GIM_5_Lu3a_G1"
    class="study" hx-get="/menu?id=GIM_5_Lu3a_G1"
    hx-target="#GIM_5_Lu3a_G1" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Lu3a_G2"
    class="study" hx-get="/menu?id=GIM_5_Lu3a_G2"
    hx-target="#GIM_5_Lu3a_G2" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Lu3a_G3"
    class="study" hx-get="/menu?id=GIM_5_Lu3a_G3"
    hx-target="#GIM_5_Lu3a_G3" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Lu3a_G4"
    class="study" hx-get="/menu?id=GIM_5_Lu3a_G4"
    hx-target="#GIM_5_Lu3a_G4" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Lu3a_G5"
    class="study" hx-get="/menu?id=GIM_5_Lu3a_G5"
    hx-target="#GIM_5_Lu3a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Lu3a_G6"
    class="study" hx-get="/menu?id=GIM_5_Lu3a_G6"
    hx-target="#GIM_5_Lu3a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Lu3a_G7"
    class="study" hx-get="/menu?id=GIM_5_Lu3a_G7"
    hx-target="#GIM_5_Lu3a_G7" hx-swap="innerHTML">

```

```

        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Lu3a_G8"
        class="study" hx-get="/menu?id=GIM_5_Lu3a_G8"
        hx-target="#GIM_5_Lu3a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Lu4a">
    <td class="Hora" id="Lu4a">11:15h<br/> -<br/> 12:05h</td>
    <td>
        <div id="GIM_5_Lu4a_G1"
            class="study" hx-get="/menu?id=GIM_5_Lu4a_G1"
            hx-target="#GIM_5_Lu4a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu4a_G2"
            class="study" hx-get="/menu?id=GIM_5_Lu4a_G2"
            hx-target="#GIM_5_Lu4a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu4a_G3"
            class="study" hx-get="/menu?id=GIM_5_Lu4a_G3"
            hx-target="#GIM_5_Lu4a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu4a_G4"
            class="study" hx-get="/menu?id=GIM_5_Lu4a_G4"
            hx-target="#GIM_5_Lu4a_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu4a_G5"
            class="study" hx-get="/menu?id=GIM_5_Lu4a_G5"
            hx-target="#GIM_5_Lu4a_G5" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu4a_G6"
            class="study" hx-get="/menu?id=GIM_5_Lu4a_G6"
            hx-target="#GIM_5_Lu4a_G6" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu4a_G7"
            class="study" hx-get="/menu?id=GIM_5_Lu4a_G7"
            hx-target="#GIM_5_Lu4a_G7" hx-swap="innerHTML">
            Estudio
        </div>
    </td>

```

```

<td>
<div id="GIM_5_Lu4a_G8"
class="study" hx-get="/menu?id=GIM_5_Lu4a_G8"
hx-target="#GIM_5_Lu4a_G8" hx-swap="innerHTML">
Estudio
</div>
</td>
</tr>
<tr id="GIM_5_Lu5a">
<td class="Hora" id="Lu5a">12:10h<br/> -<br/> 13:00h</td>
<td>
<div id="GIM_5_Lu5a_G1"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G1"
hx-target="#GIM_5_Lu5a_G1" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Lu5a_G2"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G2"
hx-target="#GIM_5_Lu5a_G2" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Lu5a_G3"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G3"
hx-target="#GIM_5_Lu5a_G3" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Lu5a_G4"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G4"
hx-target="#GIM_5_Lu5a_G4" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Lu5a_G5"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G5"
hx-target="#GIM_5_Lu5a_G5" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Lu5a_G6"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G6"
hx-target="#GIM_5_Lu5a_G6" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Lu5a_G7"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G7"
hx-target="#GIM_5_Lu5a_G7" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Lu5a_G8"
class="study" hx-get="/menu?id=GIM_5_Lu5a_G8"

```

```

hx-target="#GIM_5_Lu5a_G8" hx-swap="innerHTML">
Estudio
</div>
</td>
</tr>
<tr id="GIM_5_LuTC1">
<td class="Hora" id="LuTC1">13:10h<br/> -<br/> 14:10h</td>
<td>
<div id="GIM_5_LuTC1_G1"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G1"
hx-target="#GIM_5_LuTC1_G1" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>
<td>
<div id="GIM_5_LuTC1_G2"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G2"
hx-target="#GIM_5_LuTC1_G2" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>
<td>
<div id="GIM_5_LuTC1_G3"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G3"
hx-target="#GIM_5_LuTC1_G3" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>
<td>
<div id="GIM_5_LuTC1_G4"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G4"
hx-target="#GIM_5_LuTC1_G4" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>
<td>
<div id="GIM_5_LuTC1_G5"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G5"
hx-target="#GIM_5_LuTC1_G5" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>
<td>
<div id="GIM_5_LuTC1_G6"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G6"
hx-target="#GIM_5_LuTC1_G6" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>
<td>
<div id="GIM_5_LuTC1_G7"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G7"
hx-target="#GIM_5_LuTC1_G7" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>
<td>
<div id="GIM_5_LuTC1_G8"
class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC1_G8"
hx-target="#GIM_5_LuTC1_G8" hx-swap="innerHTML">
Deporte/Comida
</div>
</td>

```

```

</tr>
<tr id="GIM_5_LuTC2">
  <td class="Hora" id="TC2">14:20h<br/> -<br/> 15:20h</td>
  <td>
    <div id="GIM_5_LuTC2_G1" class="lunch" hx-
get="/deporte_comida?id=GIM_5_LuTC2_G1"
    hx-target="#GIM_5_LuTC2_G1" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_LuTC2_G2"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC2_G2"
      hx-target="#GIM_5_LuTC2_G2" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_LuTC2_G3"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC2_G3"
      hx-target="#GIM_5_LuTC2_G3" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_LuTC2_G4"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC2_G4"
      hx-target="#GIM_5_LuTC2_G4" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_LuTC2_G5"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC2_G5"
      hx-target="#GIM_5_LuTC2_G5" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_LuTC2_G6"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC2_G6"
      hx-target="#GIM_5_LuTC2_G6" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_LuTC2_G7"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC2_G7"
      hx-target="#GIM_5_LuTC2_G7" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_LuTC2_G8"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_LuTC2_G8"
      hx-target="#GIM_5_LuTC2_G8" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
</tr>
<tr id="GIM_5_Lu7a">
  <td class="Hora" id="Lu7a">15:30h<br/> -<br/> 16:20h</td>
  <td>

```

```

        <div id="GIM_5_Lu7a_G1"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G1"
            hx-target="#GIM_5_Lu7a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu7a_G2"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G2"
            hx-target="#GIM_5_Lu7a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu7a_G3"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G3"
            hx-target="#GIM_5_Lu7a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu7a_G4"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G4"
            hx-target="#GIM_5_Lu7a_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu7a_G5"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G5"
            hx-target="#GIM_5_Lu7a_G5" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu7a_G6"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G6"
            hx-target="#GIM_5_Lu7a_G6" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu7a_G7"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G7"
            hx-target="#GIM_5_Lu7a_G7" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Lu7a_G8"
            class="study" hx-get="/menu?id=GIM_5_Lu7a_G8"
            hx-target="#GIM_5_Lu7a_G8" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
</tr>
<tr id="GIM_5_Lu8a">
    <td class="Hora" id="Lu8a">16:25h<br/> -<br/> 17:15h</td>
    <td>
        <div id="GIM_5_Lu8a_G1"
            class="study" hx-get="/menu?id=GIM_5_Lu8a_G1"
            hx-target="#GIM_5_Lu8a_G1" hx-swap="innerHTML">

```

```

                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu8a_G2"
                class="study" hx-get="/menu?id=GIM_5_Lu8a_G2"
                hx-target="#GIM_5_Lu8a_G2" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu8a_G3"
                class="study" hx-get="/menu?id=GIM_5_Lu8a_G3"
                hx-target="#GIM_5_Lu8a_G3" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu8a_G4"
                class="study" hx-get="/menu?id=GIM_5_Lu8a_G4"
                hx-target="#GIM_5_Lu8a_G4" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu8a_G5"
                class="study" hx-get="/menu?id=GIM_5_Lu8a_G5"
                hx-target="#GIM_5_Lu8a_G5" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu8a_G6"
                class="study" hx-get="/menu?id=GIM_5_Lu8a_G6"
                hx-target="#GIM_5_Lu8a_G6" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu8a_G7"
                class="study" hx-get="/menu?id=GIM_5_Lu8a_G7"
                hx-target="#GIM_5_Lu8a_G7" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu8a_G8"
                class="study" hx-get="/menu?id=GIM_5_Lu8a_G8"
                hx-target="#GIM_5_Lu8a_G8" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>
    <tr id="GIM_5_Lu9a">
        <td class="Hora" id="Lu9a">17:20h<br/> -<br/> 18:10h</td>
        <td>
            <div id="GIM_5_Lu9a_G1"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G1"
                hx-target="#GIM_5_Lu9a_G1" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>

```

```

        <td>
            <div id="GIM_5_Lu9a_G2"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G2"
                hx-target="#GIM_5_Lu9a_G2" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu9a_G3"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G3"
                hx-target="#GIM_5_Lu9a_G3" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu9a_G4"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G4"
                hx-target="#GIM_5_Lu9a_G4" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu9a_G5"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G5"
                hx-target="#GIM_5_Lu9a_G5" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu9a_G6"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G6"
                hx-target="#GIM_5_Lu9a_G6" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu9a_G7"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G7"
                hx-target="#GIM_5_Lu9a_G7" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Lu9a_G8"
                class="study" hx-get="/menu?id=GIM_5_Lu9a_G8"
                hx-target="#GIM_5_Lu9a_G8" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>
</table>

<table class="horarios" id="GIM_5_Ma">
    <tr
class="Header"><td>Martes</td><td>G1</td><td>G2</td><td>G3</td><td>G4</td><td>G5</
td><td>G6</td><td>IM1</td><td>IM2</td></tr>
    <tr id="GIM_5_Mala"> <!-- Fila completa con el ID correcto -->
        <!-- Hora -->
        <td class="Hora" id="Mala">8:15h<br/> -<br/> 9:05h</td>
        <!-- Celda principal con asignatura, profesor y aula -->

```

```

<td>
  <div id="GIM_5_Mala_G1"
    class="study" hx-get="/menu?id=GIM_5_Mala_G1"
    hx-target="#GIM_5_Mala_G1" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mala_G2"
    class="study" hx-get="/menu?id=GIM_5_Mala_G2"
    hx-target="#GIM_5_Mala_G2" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mala_G3"
    class="study" hx-get="/menu?id=GIM_5_Mala_G3"
    hx-target="#GIM_5_Mala_G3" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mala_G4"
    class="study" hx-get="/menu?id=GIM_5_Mala_G4"
    hx-target="#GIM_5_Mala_G4" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mala_G5"
    class="study" hx-get="/menu?id=GIM_5_Mala_G5"
    hx-target="#GIM_5_Mala_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mala_G6"
    class="study" hx-get="/menu?id=GIM_5_Mala_G6"
    hx-target="#GIM_5_Mala_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mala_G7"
    class="study" hx-get="/menu?id=GIM_5_Mala_G7"
    hx-target="#GIM_5_Mala_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mala_G8"
    class="study" hx-get="/menu?id=GIM_5_Mala_G8"
    hx-target="#GIM_5_Mala_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_Ma2a">
  <td class="Hora" id="Ma2a">9:10h<br/> -<br/> 10:00h</td>
  <td>
    <div id="GIM_5_Ma2a_G1"

```

```

        class="study" hx-get="/menu?id=GIM_5_Ma2a_G1"
        hx-target="#GIM_5_Ma2a_G1" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma2a_G2"
        class="study" hx-get="/menu?id=GIM_5_Ma2a_G2"
        hx-target="#GIM_5_Ma2a_G2" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma2a_G3"
        class="study" hx-get="/menu?id=GIM_5_Ma2a_G3"
        hx-target="#GIM_5_Ma2a_G3" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma2a_G4"
        class="study" hx-get="/menu?id=GIM_5_Ma2a_G4"
        hx-target="#GIM_5_Ma2a_G4" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma2a_G5"
        class="study" hx-get="/menu?id=GIM_5_Ma2a_G5"
        hx-target="#GIM_5_Ma2a_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma2a_G6"
        class="study" hx-get="/menu?id=GIM_5_Ma2a_G6"
        hx-target="#GIM_5_Ma2a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma2a_G7"
        class="study" hx-get="/menu?id=GIM_5_Ma2a_G7"
        hx-target="#GIM_5_Ma2a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma2a_G8"
        class="study" hx-get="/menu?id=GIM_5_Ma2a_G8"
        hx-target="#GIM_5_Ma2a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Ma3a">
<td class="Hora" id="Ma3a">10:20h<br/> --<br/> 11:10h</td>
<td>
    <div id="GIM_5_Ma3a_G1"
        class="study" hx-get="/menu?id=GIM_5_Ma3a_G1"
        hx-target="#GIM_5_Ma3a_G1" hx-swap="innerHTML">
        Estudio
    </div>

```

```

</td>
<td>
  <div id="GIM_5_Ma3a_G2"
    class="study" hx-get="/menu?id=GIM_5_Ma3a_G2"
    hx-target="#GIM_5_Ma3a_G2" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma3a_G3"
    class="study" hx-get="/menu?id=GIM_5_Ma3a_G3"
    hx-target="#GIM_5_Ma3a_G3" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma3a_G4"
    class="study" hx-get="/menu?id=GIM_5_Ma3a_G4"
    hx-target="#GIM_5_Ma3a_G4" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma3a_G5"
    class="study" hx-get="/menu?id=GIM_5_Ma3a_G5"
    hx-target="#GIM_5_Ma3a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma3a_G6"
    class="study" hx-get="/menu?id=GIM_5_Ma3a_G6"
    hx-target="#GIM_5_Ma3a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma3a_G7"
    class="study" hx-get="/menu?id=GIM_5_Ma3a_G7"
    hx-target="#GIM_5_Ma3a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma3a_G8"
    class="study" hx-get="/menu?id=GIM_5_Ma3a_G8"
    hx-target="#GIM_5_Ma3a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_Ma4a">
  <td class="Hora" id="Ma4a">11:15h<br/> -<br/> 12:05h</td>
  <td>
    <div id="GIM_5_Ma4a_G1"
      class="study" hx-get="/menu?id=GIM_5_Ma4a_G1"
      hx-target="#GIM_5_Ma4a_G1" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma4a_G2"
      class="study" hx-get="/menu?id=GIM_5_Ma4a_G2"

```

```

        hx-target="#GIM_5_Ma4a_G2" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma4a_G3"
        class="study" hx-get="/menu?id=GIM_5_Ma4a_G3"
        hx-target="#GIM_5_Ma4a_G3" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma4a_G4"
        class="study" hx-get="/menu?id=GIM_5_Ma4a_G4"
        hx-target="#GIM_5_Ma4a_G4" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma4a_G5"
        class="study" hx-get="/menu?id=GIM_5_Ma4a_G5"
        hx-target="#GIM_5_Ma4a_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma4a_G6"
        class="study" hx-get="/menu?id=GIM_5_Ma4a_G6"
        hx-target="#GIM_5_Ma4a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma4a_G7"
        class="study" hx-get="/menu?id=GIM_5_Ma4a_G7"
        hx-target="#GIM_5_Ma4a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma4a_G8"
        class="study" hx-get="/menu?id=GIM_5_Ma4a_G8"
        hx-target="#GIM_5_Ma4a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Ma5a">
    <td class="Hora" id="Ma5a">12:10h<br/> -<br/> 13:00h</td>
    <td>
        <div id="GIM_5_Ma5a_G1"
            class="study" hx-get="/menu?id=GIM_5_Ma5a_G1"
            hx-target="#GIM_5_Ma5a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Ma5a_G2"
            class="study" hx-get="/menu?id=GIM_5_Ma5a_G2"
            hx-target="#GIM_5_Ma5a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
</tr>

```

```

</td>
<td>
  <div id="GIM_5_Ma5a_G3"
    class="study" hx-get="/menu?id=GIM_5_Ma5a_G3"
    hx-target="#GIM_5_Ma5a_G3" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma5a_G4"
    class="study" hx-get="/menu?id=GIM_5_Ma5a_G4"
    hx-target="#GIM_5_Ma5a_G4" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma5a_G5"
    class="study" hx-get="/menu?id=GIM_5_Ma5a_G5"
    hx-target="#GIM_5_Ma5a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma5a_G6"
    class="study" hx-get="/menu?id=GIM_5_Ma5a_G6"
    hx-target="#GIM_5_Ma5a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma5a_G7"
    class="study" hx-get="/menu?id=GIM_5_Ma5a_G7"
    hx-target="#GIM_5_Ma5a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma5a_G8"
    class="study" hx-get="/menu?id=GIM_5_Ma5a_G8"
    hx-target="#GIM_5_Ma5a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_MaTC1">
  <td class="Hora" id="MaTC1">13:10h<br/> -<br/> 14:10h</td>
  <td>
    <div id="GIM_5_MaTC1_G1"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G1"
      hx-target="#GIM_5_MaTC1_G1" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_MaTC1_G2"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G2"
      hx-target="#GIM_5_MaTC1_G2" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_MaTC1_G3"

```

```

        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G3"
        hx-target="#GIM_5_MaTC1_G3" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MaTC1_G4"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G4"
        hx-target="#GIM_5_MaTC1_G4" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MaTC1_G5"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G5"
        hx-target="#GIM_5_MaTC1_G5" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MaTC1_G6"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G6"
        hx-target="#GIM_5_MaTC1_G6" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MaTC1_G7"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G7"
        hx-target="#GIM_5_MaTC1_G7" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MaTC1_G8"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC1_G8"
        hx-target="#GIM_5_MaTC1_G8" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
</tr>
<tr id="GIM_5_MaTC2">
<td class="Hora" id="TC2">14:20h<br/> -<br/> 15:20h</td>
<td>
    <div id="GIM_5_MaTC2_G1" class="lunch" hx-
get="/deporte_comida?id=GIM_5_MaTC2_G1"
        hx-target="#GIM_5_MaTC2_G1" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MaTC2_G2"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC2_G2"
        hx-target="#GIM_5_MaTC2_G2" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MaTC2_G3"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC2_G3"
        hx-target="#GIM_5_MaTC2_G3" hx-swap="innerHTML">
        Deporte/Comida
    </div>

```

```

</td>
<td>
  <div id="GIM_5_MaTC2_G4"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC2_G4"
    hx-target="#GIM_5_MaTC2_G4" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_MaTC2_G5"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC2_G5"
    hx-target="#GIM_5_MaTC2_G5" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_MaTC2_G6"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC2_G6"
    hx-target="#GIM_5_MaTC2_G6" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_MaTC2_G7"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC2_G7"
    hx-target="#GIM_5_MaTC2_G7" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_MaTC2_G8"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MaTC2_G8"
    hx-target="#GIM_5_MaTC2_G8" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
</tr>
<tr id="GIM_5_Ma7a">
  <td class="Hora" id="Ma7a">15:30h<br/> -<br/> 16:20h</td>
  <td>
    <div id="GIM_5_Ma7a_G1"
      class="study" hx-get="/menu?id=GIM_5_Ma7a_G1"
      hx-target="#GIM_5_Ma7a_G1" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma7a_G2"
      class="study" hx-get="/menu?id=GIM_5_Ma7a_G2"
      hx-target="#GIM_5_Ma7a_G2" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma7a_G3"
      class="study" hx-get="/menu?id=GIM_5_Ma7a_G3"
      hx-target="#GIM_5_Ma7a_G3" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma7a_G4"
      class="study" hx-get="/menu?id=GIM_5_Ma7a_G4"

```

```

        hx-target="#GIM_5_Ma7a_G4" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma7a_G5"
        class="study" hx-get="/menu?id=GIM_5_Ma7a_G5"
        hx-target="#GIM_5_Ma7a_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma7a_G6"
        class="study" hx-get="/menu?id=GIM_5_Ma7a_G6"
        hx-target="#GIM_5_Ma7a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma7a_G7"
        class="study" hx-get="/menu?id=GIM_5_Ma7a_G7"
        hx-target="#GIM_5_Ma7a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma7a_G8"
        class="study" hx-get="/menu?id=GIM_5_Ma7a_G8"
        hx-target="#GIM_5_Ma7a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Ma8a">
    <td class="Hora" id="Ma8a">16:25h<br/> -<br/> 17:15h</td>
    <td>
        <div id="GIM_5_Ma8a_G1"
            class="study" hx-get="/menu?id=GIM_5_Ma8a_G1"
            hx-target="#GIM_5_Ma8a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Ma8a_G2"
            class="study" hx-get="/menu?id=GIM_5_Ma8a_G2"
            hx-target="#GIM_5_Ma8a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Ma8a_G3"
            class="study" hx-get="/menu?id=GIM_5_Ma8a_G3"
            hx-target="#GIM_5_Ma8a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Ma8a_G4"
            class="study" hx-get="/menu?id=GIM_5_Ma8a_G4"
            hx-target="#GIM_5_Ma8a_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
</tr>
</div>

```

```

</td>
<td>
  <div id="GIM_5_Ma8a_G5"
      class="study" hx-get="/menu?id=GIM_5_Ma8a_G5"
      hx-target="#GIM_5_Ma8a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma8a_G6"
      class="study" hx-get="/menu?id=GIM_5_Ma8a_G6"
      hx-target="#GIM_5_Ma8a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma8a_G7"
      class="study" hx-get="/menu?id=GIM_5_Ma8a_G7"
      hx-target="#GIM_5_Ma8a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ma8a_G8"
      class="study" hx-get="/menu?id=GIM_5_Ma8a_G8"
      hx-target="#GIM_5_Ma8a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_Ma9a">
  <td class="Hora" id="Ma9a">17:20h<br/> -<br/> 18:10h</td>
  <td>
    <div id="GIM_5_Ma9a_G1"
        class="study" hx-get="/menu?id=GIM_5_Ma9a_G1"
        hx-target="#GIM_5_Ma9a_G1" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma9a_G2"
        class="study" hx-get="/menu?id=GIM_5_Ma9a_G2"
        hx-target="#GIM_5_Ma9a_G2" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma9a_G3"
        class="study" hx-get="/menu?id=GIM_5_Ma9a_G3"
        hx-target="#GIM_5_Ma9a_G3" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma9a_G4"
        class="study" hx-get="/menu?id=GIM_5_Ma9a_G4"
        hx-target="#GIM_5_Ma9a_G4" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ma9a_G5"

```

```

                class="study" hx-get="/menu?id=GIM_5_Ma9a_G5"
                hx-target="#GIM_5_Ma9a_G5" hx-swap="innerHTML">
                Estudio
            </div>
</td>
<td>
    <div id="GIM_5_Ma9a_G6"
        class="study" hx-get="/menu?id=GIM_5_Ma9a_G6"
        hx-target="#GIM_5_Ma9a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma9a_G7"
        class="study" hx-get="/menu?id=GIM_5_Ma9a_G7"
        hx-target="#GIM_5_Ma9a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ma9a_G8"
        class="study" hx-get="/menu?id=GIM_5_Ma9a_G8"
        hx-target="#GIM_5_Ma9a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
</table>

<table class="horarios" id="GIM_5_Mi">
    <tr
class="Header"><td>Miércoles</td><td>G1</td><td>G2</td><td>G3</td><td>G4</td><td>G
5</td><td>G6</td><td>IM1</td><td>IM2</td></tr>
    <tr id="GIM_5_Mila"> <!-- Fila completa con el ID correcto -->
        <!-- Hora -->
        <td class="Hora" id="Mila">8:15h<br/> -<br/> 9:05h</td>
        <!-- Celda principal con asignatura, profesor y aula -->
        <td>
            <div id="GIM_5_Mila_G1"
                class="study" hx-get="/menu?id=GIM_5_Mila_G1"
                hx-target="#GIM_5_Mila_G1" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Mila_G2"
                class="study" hx-get="/menu?id=GIM_5_Mila_G2"
                hx-target="#GIM_5_Mila_G2" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Mila_G3"
                class="study" hx-get="/menu?id=GIM_5_Mila_G3"
                hx-target="#GIM_5_Mila_G3" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Mila_G4"
                class="study" hx-get="/menu?id=GIM_5_Mila_G4"
                hx-target="#GIM_5_Mila_G4" hx-swap="innerHTML">

```

```

        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Mila_G5"
        class="study" hx-get="/menu?id=GIM_5_Mila_G5"
        hx-target="#GIM_5_Mila_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Mila_G6"
        class="study" hx-get="/menu?id=GIM_5_Mila_G6"
        hx-target="#GIM_5_Mila_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Mila_G7"
        class="study" hx-get="/menu?id=GIM_5_Mila_G7"
        hx-target="#GIM_5_Mila_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Mila_G8"
        class="study" hx-get="/menu?id=GIM_5_Mila_G8"
        hx-target="#GIM_5_Mila_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
    <tr id="GIM_5_Mi2a">
    <td class="Hora" id="Mi2a">9:10h<br/> -<br/> 10:00h</td>
    <td>
    <div id="GIM_5_Mi2a_G1"
        class="study" hx-get="/menu?id=GIM_5_Mi2a_G1"
        hx-target="#GIM_5_Mi2a_G1" hx-swap="innerHTML">
        Estudio
    </div>
</td>
    <td>
    <div id="GIM_5_Mi2a_G2"
        class="study" hx-get="/menu?id=GIM_5_Mi2a_G2"
        hx-target="#GIM_5_Mi2a_G2" hx-swap="innerHTML">
        Estudio
    </div>
</td>
    <td>
    <div id="GIM_5_Mi2a_G3"
        class="study" hx-get="/menu?id=GIM_5_Mi2a_G3"
        hx-target="#GIM_5_Mi2a_G3" hx-swap="innerHTML">
        Estudio
    </div>
</td>
    <td>
    <div id="GIM_5_Mi2a_G4"
        class="study" hx-get="/menu?id=GIM_5_Mi2a_G4"
        hx-target="#GIM_5_Mi2a_G4" hx-swap="innerHTML">
        Estudio
    </div>

```

```

</td>
<td>
  <div id="GIM_5_Mi2a_G5"
    class="study" hx-get="/menu?id=GIM_5_Mi2a_G5"
    hx-target="#GIM_5_Mi2a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mi2a_G6"
    class="study" hx-get="/menu?id=GIM_5_Mi2a_G6"
    hx-target="#GIM_5_Mi2a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mi2a_G7"
    class="study" hx-get="/menu?id=GIM_5_Mi2a_G7"
    hx-target="#GIM_5_Mi2a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mi2a_G8"
    class="study" hx-get="/menu?id=GIM_5_Mi2a_G8"
    hx-target="#GIM_5_Mi2a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_Mi3a">
<td class="Hora" id="Mi3a">10:20h<br/> -<br/> 11:10h</td>
<td>
  <div id="GIM_5_Mi3a_G1"
    class="study" hx-get="/menu?id=GIM_5_Mi3a_G1"
    hx-target="#GIM_5_Mi3a_G1" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mi3a_G2"
    class="study" hx-get="/menu?id=GIM_5_Mi3a_G2"
    hx-target="#GIM_5_Mi3a_G2" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mi3a_G3"
    class="study" hx-get="/menu?id=GIM_5_Mi3a_G3"
    hx-target="#GIM_5_Mi3a_G3" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mi3a_G4"
    class="study" hx-get="/menu?id=GIM_5_Mi3a_G4"
    hx-target="#GIM_5_Mi3a_G4" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Mi3a_G5"
    class="study" hx-get="/menu?id=GIM_5_Mi3a_G5"

```

```

        hx-target="#GIM_5_Mi3a_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Mi3a_G6"
        class="study" hx-get="/menu?id=GIM_5_Mi3a_G6"
        hx-target="#GIM_5_Mi3a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Mi3a_G7"
        class="study" hx-get="/menu?id=GIM_5_Mi3a_G7"
        hx-target="#GIM_5_Mi3a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Mi3a_G8"
        class="study" hx-get="/menu?id=GIM_5_Mi3a_G8"
        hx-target="#GIM_5_Mi3a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Mi4a">
    <td class="Hora" id="Mi4a">11:15h<br/> -<br/> 12:05h</td>
    <td>
        <div id="GIM_5_Mi4a_G1"
            class="study" hx-get="/menu?id=GIM_5_Mi4a_G1"
            hx-target="#GIM_5_Mi4a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi4a_G2"
            class="study" hx-get="/menu?id=GIM_5_Mi4a_G2"
            hx-target="#GIM_5_Mi4a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi4a_G3"
            class="study" hx-get="/menu?id=GIM_5_Mi4a_G3"
            hx-target="#GIM_5_Mi4a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi4a_G4"
            class="study" hx-get="/menu?id=GIM_5_Mi4a_G4"
            hx-target="#GIM_5_Mi4a_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi4a_G5"
            class="study" hx-get="/menu?id=GIM_5_Mi4a_G5"
            hx-target="#GIM_5_Mi4a_G5" hx-swap="innerHTML">
            Estudio
        </div>
    </td>

```

```

<td>
<div id="GIM_5_Mi4a_G6"
class="study" hx-get="/menu?id=GIM_5_Mi4a_G6"
hx-target="#GIM_5_Mi4a_G6" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi4a_G7"
class="study" hx-get="/menu?id=GIM_5_Mi4a_G7"
hx-target="#GIM_5_Mi4a_G7" hx-swap="innerHTML">
Estudio
</div>

</td>
<td>
<div id="GIM_5_Mi4a_G8"
class="study" hx-get="/menu?id=GIM_5_Mi4a_G8"
hx-target="#GIM_5_Mi4a_G8" hx-swap="innerHTML">
Estudio
</div>
</td>
</tr>
<tr id="GIM_5_Mi5a">
<td class="Hora" id="Mi5a">12:10h<br/> -<br/> 13:00h</td>
<td>
<div id="GIM_5_Mi5a_G1"
class="study" hx-get="/menu?id=GIM_5_Mi5a_G1"
hx-target="#GIM_5_Mi5a_G1" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi5a_G2"
class="study" hx-get="/menu?id=GIM_5_Mi5a_G2"
hx-target="#GIM_5_Mi5a_G2" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi5a_G3"
class="study" hx-get="/menu?id=GIM_5_Mi5a_G3"
hx-target="#GIM_5_Mi5a_G3" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi5a_G4"
class="study" hx-get="/menu?id=GIM_5_Mi5a_G4"
hx-target="#GIM_5_Mi5a_G4" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi5a_G5"
class="study" hx-get="/menu?id=GIM_5_Mi5a_G5"
hx-target="#GIM_5_Mi5a_G5" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi5a_G6"
class="study" hx-get="/menu?id=GIM_5_Mi5a_G6"

```

```

hx-target="#GIM_5_Mi5a_G6" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
  <div id="GIM_5_Mi5a_G7"
    class="study" hx-get="/menu?id=GIM_5_Mi5a_G7"
    hx-target="#GIM_5_Mi5a_G7" hx-swap="innerHTML">
    Estudio
  </div>

</td>
<td>
  <div id="GIM_5_Mi5a_G8"
    class="study" hx-get="/menu?id=GIM_5_Mi5a_G8"
    hx-target="#GIM_5_Mi5a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_MiTC1">
  <td class="Hora" id="MiTC1">13:10h<br/> -<br/> 14:10h</td>
  <td>
    <div id="GIM_5_MiTC1_G1"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G1"
      hx-target="#GIM_5_MiTC1_G1" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_MiTC1_G2"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G2"
      hx-target="#GIM_5_MiTC1_G2" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_MiTC1_G3"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G3"
      hx-target="#GIM_5_MiTC1_G3" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_MiTC1_G4"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G4"
      hx-target="#GIM_5_MiTC1_G4" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_MiTC1_G5"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G5"
      hx-target="#GIM_5_MiTC1_G5" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_MiTC1_G6"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G6"
      hx-target="#GIM_5_MiTC1_G6" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>

```

```

</td>
<td>
  <div id="GIM_5_MiTC1_G7"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G7"
    hx-target="#GIM_5_MiTC1_G7" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_MiTC1_G8"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC1_G8"
    hx-target="#GIM_5_MiTC1_G8" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
</tr>
<tr id="GIM_5_MiTC2">
  <td class="Hora" id="TC2">14:20h<br/> -<br/> 15:20h</td>
  <td>
    <div id="GIM_5_MiTC2_G1" class="lunch" hx-
get="/deporte_comida?id=GIM_5_MiTC2_G1"
    hx-target="#GIM_5_MiTC2_G1" hx-swap="innerHTML">
    Deporte/Comida
  </div>
  </td>
  <td>
    <div id="GIM_5_MiTC2_G2"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC2_G2"
    hx-target="#GIM_5_MiTC2_G2" hx-swap="innerHTML">
    Deporte/Comida
  </div>
  </td>
  <td>
    <div id="GIM_5_MiTC2_G3"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC2_G3"
    hx-target="#GIM_5_MiTC2_G3" hx-swap="innerHTML">
    Deporte/Comida
  </div>
  </td>
  <td>
    <div id="GIM_5_MiTC2_G4"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC2_G4"
    hx-target="#GIM_5_MiTC2_G4" hx-swap="innerHTML">
    Deporte/Comida
  </div>
  </td>
  <td>
    <div id="GIM_5_MiTC2_G5"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC2_G5"
    hx-target="#GIM_5_MiTC2_G5" hx-swap="innerHTML">
    Deporte/Comida
  </div>
  </td>
  <td>
    <div id="GIM_5_MiTC2_G6"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC2_G6"
    hx-target="#GIM_5_MiTC2_G6" hx-swap="innerHTML">
    Deporte/Comida
  </div>
  </td>
  <td>
    <div id="GIM_5_MiTC2_G7"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC2_G7"

```

```

        hx-target="#GIM_5_MiTC2_G7" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
<td>
    <div id="GIM_5_MiTC2_G8"
        class="lunch" hx-get="/deporte_comida?id=GIM_5_MiTC2_G8"
        hx-target="#GIM_5_MiTC2_G8" hx-swap="innerHTML">
        Deporte/Comida
    </div>
</td>
</tr>
<tr id="GIM_5_Mi7a">
    <td class="Hora" id="Mi7a">15:30h<br/> -<br/> 16:20h</td>
    <td>
        <div id="GIM_5_Mi7a_G1"
            class="study" hx-get="/menu?id=GIM_5_Mi7a_G1"
            hx-target="#GIM_5_Mi7a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi7a_G2"
            class="study" hx-get="/menu?id=GIM_5_Mi7a_G2"
            hx-target="#GIM_5_Mi7a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi7a_G3"
            class="study" hx-get="/menu?id=GIM_5_Mi7a_G3"
            hx-target="#GIM_5_Mi7a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi7a_G4"
            class="study" hx-get="/menu?id=GIM_5_Mi7a_G4"
            hx-target="#GIM_5_Mi7a_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi7a_G5"
            class="study" hx-get="/menu?id=GIM_5_Mi7a_G5"
            hx-target="#GIM_5_Mi7a_G5" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi7a_G6"
            class="study" hx-get="/menu?id=GIM_5_Mi7a_G6"
            hx-target="#GIM_5_Mi7a_G6" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Mi7a_G7"
            class="study" hx-get="/menu?id=GIM_5_Mi7a_G7"
            hx-target="#GIM_5_Mi7a_G7" hx-swap="innerHTML">
            Estudio
        </div>
    </td>

```

```

</td>
<td>
  <div id="GIM_5_Mi7a_G8"
    class="study" hx-get="/menu?id=GIM_5_Mi7a_G8"
    hx-target="#GIM_5_Mi7a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_Mi8a">
  <td class="Hora" id="Mi8a">16:25h<br/> -<br/> 17:15h</td>
  <td>
    <div id="GIM_5_Mi8a_G1"
      class="study" hx-get="/menu?id=GIM_5_Mi8a_G1"
      hx-target="#GIM_5_Mi8a_G1" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Mi8a_G2"
      class="study" hx-get="/menu?id=GIM_5_Mi8a_G2"
      hx-target="#GIM_5_Mi8a_G2" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Mi8a_G3"
      class="study" hx-get="/menu?id=GIM_5_Mi8a_G3"
      hx-target="#GIM_5_Mi8a_G3" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Mi8a_G4"
      class="study" hx-get="/menu?id=GIM_5_Mi8a_G4"
      hx-target="#GIM_5_Mi8a_G4" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Mi8a_G5"
      class="study" hx-get="/menu?id=GIM_5_Mi8a_G5"
      hx-target="#GIM_5_Mi8a_G5" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Mi8a_G6"
      class="study" hx-get="/menu?id=GIM_5_Mi8a_G6"
      hx-target="#GIM_5_Mi8a_G6" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Mi8a_G7"
      class="study" hx-get="/menu?id=GIM_5_Mi8a_G7"
      hx-target="#GIM_5_Mi8a_G7" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Mi8a_G8"

```

```

class="study" hx-get="/menu?id=GIM_5_Mi8a_G8"
hx-target="#GIM_5_Mi8a_G8" hx-swap="innerHTML">
Estudio
</div>
</td>
</tr>
<tr id="GIM_5_Mi9a">
<td class="Hora" id="Mi9a">17:20h<br/> -<br/> 18:10h</td>
<td>
<div id="GIM_5_Mi9a_G1"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G1"
hx-target="#GIM_5_Mi9a_G1" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi9a_G2"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G2"
hx-target="#GIM_5_Mi9a_G2" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi9a_G3"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G3"
hx-target="#GIM_5_Mi9a_G3" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi9a_G4"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G4"
hx-target="#GIM_5_Mi9a_G4" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi9a_G5"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G5"
hx-target="#GIM_5_Mi9a_G5" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi9a_G6"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G6"
hx-target="#GIM_5_Mi9a_G6" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi9a_G7"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G7"
hx-target="#GIM_5_Mi9a_G7" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Mi9a_G8"
class="study" hx-get="/menu?id=GIM_5_Mi9a_G8"
hx-target="#GIM_5_Mi9a_G8" hx-swap="innerHTML">
Estudio

```

```

</div>
</td>
</tr>
</table>

<table class="horarios" id="GIM_5_Ju">
  <tr
class="Header"><td>Jueves</td><td>G1</td><td>G2</td><td>G3</td><td>G4</td><td>G5</
td><td>G6</td><td>IM1</td><td>IM2</td></tr>
  <tr id="GIM_5_Jula"> <!-- Fila completa con el ID correcto -->
    <!-- Hora -->
    <td class="Hora" id="Jula">8:15h<br/> -<br/> 9:05h</td>
    <!-- Celda principal con asignatura, profesor y aula -->
    <td>
      <div id="GIM_5_Jula_G1"
        class="study" hx-get="/menu?id=GIM_5_Jula_G1"
        hx-target="#GIM_5_Jula_G1" hx-swap="innerHTML">
        Estudio
      </div>
    </td>
    <td>
      <div id="GIM_5_Jula_G2"
        class="study" hx-get="/menu?id=GIM_5_Jula_G2"
        hx-target="#GIM_5_Jula_G2" hx-swap="innerHTML">
        Estudio
      </div>
    </td>
    <td>
      <div id="GIM_5_Jula_G3"
        class="study" hx-get="/menu?id=GIM_5_Jula_G3"
        hx-target="#GIM_5_Jula_G3" hx-swap="innerHTML">
        Estudio
      </div>
    </td>
    <td>
      <div id="GIM_5_Jula_G4"
        class="study" hx-get="/menu?id=GIM_5_Jula_G4"
        hx-target="#GIM_5_Jula_G4" hx-swap="innerHTML">
        Estudio
      </div>
    </td>
    <td>
      <div id="GIM_5_Jula_G5"
        class="study" hx-get="/menu?id=GIM_5_Jula_G5"
        hx-target="#GIM_5_Jula_G5" hx-swap="innerHTML">
        Estudio
      </div>
    </td>
    <td>
      <div id="GIM_5_Jula_G6"
        class="study" hx-get="/menu?id=GIM_5_Jula_G6"
        hx-target="#GIM_5_Jula_G6" hx-swap="innerHTML">
        Estudio
      </div>
    </td>
    <td>
      <div id="GIM_5_Jula_G7"
        class="study" hx-get="/menu?id=GIM_5_Jula_G7"
        hx-target="#GIM_5_Jula_G7" hx-swap="innerHTML">
        Estudio
      </div>
    </td>
  </tr>
</table>

```

```

<td>
  <div id="GIM_5_Ju1a_G8"
    class="study" hx-get="/menu?id=GIM_5_Ju1a_G8"
    hx-target="#GIM_5_Ju1a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>

</tr>
  <tr id="GIM_5_Ju2a">
    <td class="Hora" id="Ju2a">9:10h<br/> -<br/> 10:00h</td>
      <td>
        <div id="GIM_5_Ju2a_G1"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G1"
          hx-target="#GIM_5_Ju2a_G1" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Ju2a_G2"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G2"
          hx-target="#GIM_5_Ju2a_G2" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Ju2a_G3"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G3"
          hx-target="#GIM_5_Ju2a_G3" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Ju2a_G4"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G4"
          hx-target="#GIM_5_Ju2a_G4" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Ju2a_G5"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G5"
          hx-target="#GIM_5_Ju2a_G5" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Ju2a_G6"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G6"
          hx-target="#GIM_5_Ju2a_G6" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Ju2a_G7"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G7"
          hx-target="#GIM_5_Ju2a_G7" hx-swap="innerHTML">
          Estudio
        </div>
      </td>
      <td>
        <div id="GIM_5_Ju2a_G8"
          class="study" hx-get="/menu?id=GIM_5_Ju2a_G8"

```

```

                hx-target="#GIM_5_Ju2a_G8" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>
    <tr id="GIM_5_Ju3a">
        <td class="Hora" id="Ju3a">10:20h<br/> -<br/> 11:10h</td>
        <td>
            <div id="GIM_5_Ju3a_G1"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G1"
                hx-target="#GIM_5_Ju3a_G1" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju3a_G2"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G2"
                hx-target="#GIM_5_Ju3a_G2" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju3a_G3"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G3"
                hx-target="#GIM_5_Ju3a_G3" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju3a_G4"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G4"
                hx-target="#GIM_5_Ju3a_G4" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju3a_G5"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G5"
                hx-target="#GIM_5_Ju3a_G5" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju3a_G6"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G6"
                hx-target="#GIM_5_Ju3a_G6" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju3a_G7"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G7"
                hx-target="#GIM_5_Ju3a_G7" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju3a_G8"
                class="study" hx-get="/menu?id=GIM_5_Ju3a_G8"
                hx-target="#GIM_5_Ju3a_G8" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>

```

```

</tr>
<tr id="GIM_5_Ju4a">
  <td class="Hora" id="Ju4a">11:15h<br/> -<br/> 12:05h</td>
  <td>
    <div id="GIM_5_Ju4a_G1"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G1"
      hx-target="#GIM_5_Ju4a_G1" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju4a_G2"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G2"
      hx-target="#GIM_5_Ju4a_G2" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju4a_G3"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G3"
      hx-target="#GIM_5_Ju4a_G3" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju4a_G4"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G4"
      hx-target="#GIM_5_Ju4a_G4" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju4a_G5"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G5"
      hx-target="#GIM_5_Ju4a_G5" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju4a_G6"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G6"
      hx-target="#GIM_5_Ju4a_G6" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju4a_G7"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G7"
      hx-target="#GIM_5_Ju4a_G7" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju4a_G8"
      class="study" hx-get="/menu?id=GIM_5_Ju4a_G8"
      hx-target="#GIM_5_Ju4a_G8" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
</tr>
<tr id="GIM_5_Ju5a">
  <td class="Hora" id="Ju5a">12:10h<br/> -<br/> 13:00h</td>

```

```

<td>
  <div id="GIM_5_Ju5a_G1"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G1"
    hx-target="#GIM_5_Ju5a_G1" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju5a_G2"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G2"
    hx-target="#GIM_5_Ju5a_G2" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju5a_G3"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G3"
    hx-target="#GIM_5_Ju5a_G3" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju5a_G4"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G4"
    hx-target="#GIM_5_Ju5a_G4" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju5a_G5"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G5"
    hx-target="#GIM_5_Ju5a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju5a_G6"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G6"
    hx-target="#GIM_5_Ju5a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju5a_G7"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G7"
    hx-target="#GIM_5_Ju5a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju5a_G8"
    class="study" hx-get="/menu?id=GIM_5_Ju5a_G8"
    hx-target="#GIM_5_Ju5a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_JuTC1">
  <td class="Hora" id="JuTC1">13:10h<br/> -<br/> 14:10h</td>
  <td>
    <div id="GIM_5_JuTC1_G1"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G1"

```

```

                hx-target="#GIM_5_JuTC1_G1" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
        <td>
            <div id="GIM_5_JuTC1_G2"
                class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G2"
                hx-target="#GIM_5_JuTC1_G2" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
        <td>
            <div id="GIM_5_JuTC1_G3"
                class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G3"
                hx-target="#GIM_5_JuTC1_G3" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
        <td>
            <div id="GIM_5_JuTC1_G4"
                class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G4"
                hx-target="#GIM_5_JuTC1_G4" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
        <td>
            <div id="GIM_5_JuTC1_G5"
                class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G5"
                hx-target="#GIM_5_JuTC1_G5" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
        <td>
            <div id="GIM_5_JuTC1_G6"
                class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G6"
                hx-target="#GIM_5_JuTC1_G6" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
        <td>
            <div id="GIM_5_JuTC1_G7"
                class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G7"
                hx-target="#GIM_5_JuTC1_G7" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
        <td>
            <div id="GIM_5_JuTC1_G8"
                class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC1_G8"
                hx-target="#GIM_5_JuTC1_G8" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
    </tr>
    <tr id="GIM_5_JuTC2">
        <td class="Hora" id="TC2">14:20h<br/> -<br/> 15:20h</td>
        <td>
            <div id="GIM_5_JuTC2_G1" class="lunch" hx-
get="/deporte_comida?id=GIM_5_JuTC2_G1"
                hx-target="#GIM_5_JuTC2_G1" hx-swap="innerHTML">
                Deporte/Comida
            </div>
        </td>
    </tr>

```

```

<td>
  <div id="GIM_5_JuTC2_G2"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC2_G2"
    hx-target="#GIM_5_JuTC2_G2" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_JuTC2_G3"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC2_G3"
    hx-target="#GIM_5_JuTC2_G3" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_JuTC2_G4"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC2_G4"
    hx-target="#GIM_5_JuTC2_G4" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_JuTC2_G5"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC2_G5"
    hx-target="#GIM_5_JuTC2_G5" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_JuTC2_G6"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC2_G6"
    hx-target="#GIM_5_JuTC2_G6" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_JuTC2_G7"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC2_G7"
    hx-target="#GIM_5_JuTC2_G7" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
<td>
  <div id="GIM_5_JuTC2_G8"
    class="lunch" hx-get="/deporte_comida?id=GIM_5_JuTC2_G8"
    hx-target="#GIM_5_JuTC2_G8" hx-swap="innerHTML">
    Deporte/Comida
  </div>
</td>
</tr>
<tr id="GIM_5_Ju7a">
  <td class="Hora" id="Ju7a">15:30h<br/> -<br/> 16:20h</td>
  <td>
    <div id="GIM_5_Ju7a_G1"
      class="study" hx-get="/menu?id=GIM_5_Ju7a_G1"
      hx-target="#GIM_5_Ju7a_G1" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju7a_G2"
      class="study" hx-get="/menu?id=GIM_5_Ju7a_G2"
      hx-target="#GIM_5_Ju7a_G2" hx-swap="innerHTML">

```

```

        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ju7a_G3"
        class="study" hx-get="/menu?id=GIM_5_Ju7a_G3"
        hx-target="#GIM_5_Ju7a_G3" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ju7a_G4"
        class="study" hx-get="/menu?id=GIM_5_Ju7a_G4"
        hx-target="#GIM_5_Ju7a_G4" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ju7a_G5"
        class="study" hx-get="/menu?id=GIM_5_Ju7a_G5"
        hx-target="#GIM_5_Ju7a_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ju7a_G6"
        class="study" hx-get="/menu?id=GIM_5_Ju7a_G6"
        hx-target="#GIM_5_Ju7a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ju7a_G7"
        class="study" hx-get="/menu?id=GIM_5_Ju7a_G7"
        hx-target="#GIM_5_Ju7a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Ju7a_G8"
        class="study" hx-get="/menu?id=GIM_5_Ju7a_G8"
        hx-target="#GIM_5_Ju7a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Ju8a">
    <td class="Hora" id="Ju8a">16:25h<br/> -<br/> 17:15h</td>
    <td>
        <div id="GIM_5_Ju8a_G1"
            class="study" hx-get="/menu?id=GIM_5_Ju8a_G1"
            hx-target="#GIM_5_Ju8a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Ju8a_G2"
            class="study" hx-get="/menu?id=GIM_5_Ju8a_G2"
            hx-target="#GIM_5_Ju8a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
</tr>

```

```

<td>
  <div id="GIM_5_Ju8a_G3"
      class="study" hx-get="/menu?id=GIM_5_Ju8a_G3"
      hx-target="#GIM_5_Ju8a_G3" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju8a_G4"
      class="study" hx-get="/menu?id=GIM_5_Ju8a_G4"
      hx-target="#GIM_5_Ju8a_G4" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju8a_G5"
      class="study" hx-get="/menu?id=GIM_5_Ju8a_G5"
      hx-target="#GIM_5_Ju8a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju8a_G6"
      class="study" hx-get="/menu?id=GIM_5_Ju8a_G6"
      hx-target="#GIM_5_Ju8a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju8a_G7"
      class="study" hx-get="/menu?id=GIM_5_Ju8a_G7"
      hx-target="#GIM_5_Ju8a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Ju8a_G8"
      class="study" hx-get="/menu?id=GIM_5_Ju8a_G8"
      hx-target="#GIM_5_Ju8a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_Ju9a">
  <td class="Hora" id="Ju9a">17:20h<br/> -<br/> 18:10h</td>
  <td>
    <div id="GIM_5_Ju9a_G1"
        class="study" hx-get="/menu?id=GIM_5_Ju9a_G1"
        hx-target="#GIM_5_Ju9a_G1" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju9a_G2"
        class="study" hx-get="/menu?id=GIM_5_Ju9a_G2"
        hx-target="#GIM_5_Ju9a_G2" hx-swap="innerHTML">
      Estudio
    </div>
  </td>
  <td>
    <div id="GIM_5_Ju9a_G3"
        class="study" hx-get="/menu?id=GIM_5_Ju9a_G3"

```

```

                hx-target="#GIM_5_Ju9a_G3" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju9a_G4"
                class="study" hx-get="/menu?id=GIM_5_Ju9a_G4"
                hx-target="#GIM_5_Ju9a_G4" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju9a_G5"
                class="study" hx-get="/menu?id=GIM_5_Ju9a_G5"
                hx-target="#GIM_5_Ju9a_G5" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju9a_G6"
                class="study" hx-get="/menu?id=GIM_5_Ju9a_G6"
                hx-target="#GIM_5_Ju9a_G6" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju9a_G7"
                class="study" hx-get="/menu?id=GIM_5_Ju9a_G7"
                hx-target="#GIM_5_Ju9a_G7" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Ju9a_G8"
                class="study" hx-get="/menu?id=GIM_5_Ju9a_G8"
                hx-target="#GIM_5_Ju9a_G8" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>
</table>
<table class="horarios" id="GIM_5_Vi">
    <tr
class="Header"><td>Viernes</td><td>G1</td><td>G2</td><td>G3</td><td>G4</td><td>G5<
/td><td>G6</td><td>IM1</td><td>IM2</td></tr>
    <tr id="GIM_5_Vila"> <!-- Fila completa con el ID correcto -->
        <!-- Hora -->
        <td class="Hora" id="Vila">8:15h<br/> -<br/> 9:05h</td>
        <!-- Celda principal con asignatura, profesor y aula -->
        <td>
            <div id="GIM_5_Vila_G1"
                class="study" hx-get="/menu?id=GIM_5_Vila_G1"
                hx-target="#GIM_5_Vila_G1" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Vila_G2"
                class="study" hx-get="/menu?id=GIM_5_Vila_G2"
                hx-target="#GIM_5_Vila_G2" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>

```

```

        </td>
        <td>
            <div id="GIM_5_Vila_G3"
                class="study" hx-get="/menu?id=GIM_5_Vila_G3"
                hx-target="#GIM_5_Vila_G3" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Vila_G4"
                class="study" hx-get="/menu?id=GIM_5_Vila_G4"
                hx-target="#GIM_5_Vila_G4" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Vila_G5"
                class="study" hx-get="/menu?id=GIM_5_Vila_G5"
                hx-target="#GIM_5_Vila_G5" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Vila_G6"
                class="study" hx-get="/menu?id=GIM_5_Vila_G6"
                hx-target="#GIM_5_Vila_G6" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Vila_G7"
                class="study" hx-get="/menu?id=GIM_5_Vila_G7"
                hx-target="#GIM_5_Vila_G7" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Vila_G8"
                class="study" hx-get="/menu?id=GIM_5_Vila_G8"
                hx-target="#GIM_5_Vila_G8" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
    </tr>
    <tr id="GIM_5_Vi2a">
        <td class="Hora" id="Vi2a">9:10h<br/> -<br/> 10:00h</td>
        <td>
            <div id="GIM_5_Vi2a_G1"
                class="study" hx-get="/menu?id=GIM_5_Vi2a_G1"
                hx-target="#GIM_5_Vi2a_G1" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
            <div id="GIM_5_Vi2a_G2"
                class="study" hx-get="/menu?id=GIM_5_Vi2a_G2"
                hx-target="#GIM_5_Vi2a_G2" hx-swap="innerHTML">
                Estudio
            </div>
        </td>
        <td>
    
```

```

        <div id="GIM_5_Vi2a_G3"
            class="study" hx-get="/menu?id=GIM_5_Vi2a_G3"
            hx-target="#GIM_5_Vi2a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
</td>
<td>
    <div id="GIM_5_Vi2a_G4"
        class="study" hx-get="/menu?id=GIM_5_Vi2a_G4"
        hx-target="#GIM_5_Vi2a_G4" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Vi2a_G5"
        class="study" hx-get="/menu?id=GIM_5_Vi2a_G5"
        hx-target="#GIM_5_Vi2a_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Vi2a_G6"
        class="study" hx-get="/menu?id=GIM_5_Vi2a_G6"
        hx-target="#GIM_5_Vi2a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Vi2a_G7"
        class="study" hx-get="/menu?id=GIM_5_Vi2a_G7"
        hx-target="#GIM_5_Vi2a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Vi2a_G8"
        class="study" hx-get="/menu?id=GIM_5_Vi2a_G8"
        hx-target="#GIM_5_Vi2a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Vi3a">
<td class="Hora" id="Vi3a">10:20h<br/> -<br/> 11:10h</td>
<td>
    <div id="GIM_5_Vi3a_G1"
        class="study" hx-get="/menu?id=GIM_5_Vi3a_G1"
        hx-target="#GIM_5_Vi3a_G1" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Vi3a_G2"
        class="study" hx-get="/menu?id=GIM_5_Vi3a_G2"
        hx-target="#GIM_5_Vi3a_G2" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
    <div id="GIM_5_Vi3a_G3"
        class="study" hx-get="/menu?id=GIM_5_Vi3a_G3"
        hx-target="#GIM_5_Vi3a_G3" hx-swap="innerHTML">
        Estudio
    </div>
</td>

```

```

        </div>
    </td>
    <td>
        <div id="GIM_5_Vi3a_G4"
            class="study" hx-get="/menu?id=GIM_5_Vi3a_G4"
            hx-target="#GIM_5_Vi3a_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi3a_G5"
            class="study" hx-get="/menu?id=GIM_5_Vi3a_G5"
            hx-target="#GIM_5_Vi3a_G5" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi3a_G6"
            class="study" hx-get="/menu?id=GIM_5_Vi3a_G6"
            hx-target="#GIM_5_Vi3a_G6" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi3a_G7"
            class="study" hx-get="/menu?id=GIM_5_Vi3a_G7"
            hx-target="#GIM_5_Vi3a_G7" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi3a_G8"
            class="study" hx-get="/menu?id=GIM_5_Vi3a_G8"
            hx-target="#GIM_5_Vi3a_G8" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
</tr>
<tr id="GIM_5_Vi4a">
    <td class="Hora" id="Vi4a">11:15h<br/> -<br/> 12:05h</td>
    <td>
        <div id="GIM_5_Vi4a_G1"
            class="study" hx-get="/menu?id=GIM_5_Vi4a_G1"
            hx-target="#GIM_5_Vi4a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi4a_G2"
            class="study" hx-get="/menu?id=GIM_5_Vi4a_G2"
            hx-target="#GIM_5_Vi4a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi4a_G3"
            class="study" hx-get="/menu?id=GIM_5_Vi4a_G3"
            hx-target="#GIM_5_Vi4a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi4a_G4"

```

```

        class="study" hx-get="/menu?id=GIM_5_Vi4a_G4"
        hx-target="#GIM_5_Vi4a_G4" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
<div id="GIM_5_Vi4a_G5"
        class="study" hx-get="/menu?id=GIM_5_Vi4a_G5"
        hx-target="#GIM_5_Vi4a_G5" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
<div id="GIM_5_Vi4a_G6"
        class="study" hx-get="/menu?id=GIM_5_Vi4a_G6"
        hx-target="#GIM_5_Vi4a_G6" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
<div id="GIM_5_Vi4a_G7"
        class="study" hx-get="/menu?id=GIM_5_Vi4a_G7"
        hx-target="#GIM_5_Vi4a_G7" hx-swap="innerHTML">
        Estudio
    </div>
</td>
<td>
<div id="GIM_5_Vi4a_G8"
        class="study" hx-get="/menu?id=GIM_5_Vi4a_G8"
        hx-target="#GIM_5_Vi4a_G8" hx-swap="innerHTML">
        Estudio
    </div>
</td>
</tr>
<tr id="GIM_5_Vi5a">
    <td class="Hora" id="Vi5a">12:10h<br/> -<br/> 13:00h</td>
    <td>
        <div id="GIM_5_Vi5a_G1"
            class="study" hx-get="/menu?id=GIM_5_Vi5a_G1"
            hx-target="#GIM_5_Vi5a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi5a_G2"
            class="study" hx-get="/menu?id=GIM_5_Vi5a_G2"
            hx-target="#GIM_5_Vi5a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi5a_G3"
            class="study" hx-get="/menu?id=GIM_5_Vi5a_G3"
            hx-target="#GIM_5_Vi5a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi5a_G4"
            class="study" hx-get="/menu?id=GIM_5_Vi5a_G4"
            hx-target="#GIM_5_Vi5a_G4" hx-swap="innerHTML">
            Estudio

```

```

</div>
</td>
<td>
  <div id="GIM_5_Vi5a_G5"
    class="study" hx-get="/menu?id=GIM_5_Vi5a_G5"
    hx-target="#GIM_5_Vi5a_G5" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Vi5a_G6"
    class="study" hx-get="/menu?id=GIM_5_Vi5a_G6"
    hx-target="#GIM_5_Vi5a_G6" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Vi5a_G7"
    class="study" hx-get="/menu?id=GIM_5_Vi5a_G7"
    hx-target="#GIM_5_Vi5a_G7" hx-swap="innerHTML">
    Estudio
  </div>
</td>
<td>
  <div id="GIM_5_Vi5a_G8"
    class="study" hx-get="/menu?id=GIM_5_Vi5a_G8"
    hx-target="#GIM_5_Vi5a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
<tr id="GIM_5_ViTC1">
  <td class="Hora" id="ViTC1">13:10h<br/> -<br/> 14:10h</td>
  <td>
    <div id="GIM_5_ViTC1_G1"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G1"
      hx-target="#GIM_5_ViTC1_G1" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_ViTC1_G2"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G2"
      hx-target="#GIM_5_ViTC1_G2" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_ViTC1_G3"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G3"
      hx-target="#GIM_5_ViTC1_G3" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
  <td>
    <div id="GIM_5_ViTC1_G4"
      class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G4"
      hx-target="#GIM_5_ViTC1_G4" hx-swap="innerHTML">
      Deporte/Comida
    </div>
  </td>
</td>

```

```

        <div id="GIM_5_ViTC1_G5"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G5"
            hx-target="#GIM_5_ViTC1_G5" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC1_G6"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G6"
            hx-target="#GIM_5_ViTC1_G6" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC1_G7"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G7"
            hx-target="#GIM_5_ViTC1_G7" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC1_G8"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC1_G8"
            hx-target="#GIM_5_ViTC1_G8" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
</tr>
<tr id="GIM_5_ViTC2">
    <td class="Hora" id="TC2">14:20h<br/> -<br/> 15:20h</td>
    <td>
        <div id="GIM_5_ViTC2_G1" class="lunch" hx-
get="/deporte_comida?id=GIM_5_ViTC2_G1"
            hx-target="#GIM_5_ViTC2_G1" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC2_G2"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC2_G2"
            hx-target="#GIM_5_ViTC2_G2" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC2_G3"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC2_G3"
            hx-target="#GIM_5_ViTC2_G3" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC2_G4"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC2_G4"
            hx-target="#GIM_5_ViTC2_G4" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC2_G5"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC2_G5"
            hx-target="#GIM_5_ViTC2_G5" hx-swap="innerHTML">
            Deporte/Comida

```

```

        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC2_G6"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC2_G6"
            hx-target="#GIM_5_ViTC2_G6" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC2_G7"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC2_G7"
            hx-target="#GIM_5_ViTC2_G7" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
    <td>
        <div id="GIM_5_ViTC2_G8"
            class="lunch" hx-get="/deporte_comida?id=GIM_5_ViTC2_G8"
            hx-target="#GIM_5_ViTC2_G8" hx-swap="innerHTML">
            Deporte/Comida
        </div>
    </td>
</tr>
<tr id="GIM_5_Vi7a">
    <td class="Hora" id="Vi7a">15:30h<br/> -<br/> 16:20h</td>
    <td>
        <div id="GIM_5_Vi7a_G1"
            class="study" hx-get="/menu?id=GIM_5_Vi7a_G1"
            hx-target="#GIM_5_Vi7a_G1" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi7a_G2"
            class="study" hx-get="/menu?id=GIM_5_Vi7a_G2"
            hx-target="#GIM_5_Vi7a_G2" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi7a_G3"
            class="study" hx-get="/menu?id=GIM_5_Vi7a_G3"
            hx-target="#GIM_5_Vi7a_G3" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi7a_G4"
            class="study" hx-get="/menu?id=GIM_5_Vi7a_G4"
            hx-target="#GIM_5_Vi7a_G4" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi7a_G5"
            class="study" hx-get="/menu?id=GIM_5_Vi7a_G5"
            hx-target="#GIM_5_Vi7a_G5" hx-swap="innerHTML">
            Estudio
        </div>
    </td>
    <td>
        <div id="GIM_5_Vi7a_G6"

```

```

class="study" hx-get="/menu?id=GIM_5_Vi7a_G6"
hx-target="#GIM_5_Vi7a_G6" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Vi7a_G7"
class="study" hx-get="/menu?id=GIM_5_Vi7a_G7"
hx-target="#GIM_5_Vi7a_G7" hx-swap="innerHTML">
Estudio
</div>

</td>
<td>
<div id="GIM_5_Vi7a_G8"
class="study" hx-get="/menu?id=GIM_5_Vi7a_G8"
hx-target="#GIM_5_Vi7a_G8" hx-swap="innerHTML">
Estudio
</div>
</td>
</tr>
<tr id="GIM_5_Vi8a">
<td class="Hora" id="Vi8a">16:25h<br/> -<br/> 17:15h</td>
<td>
<div id="GIM_5_Vi8a_G1"
class="study" hx-get="/menu?id=GIM_5_Vi8a_G1"
hx-target="#GIM_5_Vi8a_G1" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Vi8a_G2"
class="study" hx-get="/menu?id=GIM_5_Vi8a_G2"
hx-target="#GIM_5_Vi8a_G2" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Vi8a_G3"
class="study" hx-get="/menu?id=GIM_5_Vi8a_G3"
hx-target="#GIM_5_Vi8a_G3" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Vi8a_G4"
class="study" hx-get="/menu?id=GIM_5_Vi8a_G4"
hx-target="#GIM_5_Vi8a_G4" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Vi8a_G5"
class="study" hx-get="/menu?id=GIM_5_Vi8a_G5"
hx-target="#GIM_5_Vi8a_G5" hx-swap="innerHTML">
Estudio
</div>
</td>
<td>
<div id="GIM_5_Vi8a_G6"
class="study" hx-get="/menu?id=GIM_5_Vi8a_G6"
hx-target="#GIM_5_Vi8a_G6" hx-swap="innerHTML">
Estudio

```

```

</div>
</td>
<td>
  <div id="GIM_5_Vi8a_G7"
    class="study" hx-get="/menu?id=GIM_5_Vi8a_G7"
    hx-target="#GIM_5_Vi8a_G7" hx-swap="innerHTML">
    Estudio
  </div>

</td>
<td>
  <div id="GIM_5_Vi8a_G8"
    class="study" hx-get="/menu?id=GIM_5_Vi8a_G8"
    hx-target="#GIM_5_Vi8a_G8" hx-swap="innerHTML">
    Estudio
  </div>
</td>
</tr>
</table>
<div>
<button onclick="window.location.href='/exportar_csv'" class="export-btn">
Exportar Horario a CSV
</button>
<button id="exportar-pdf" class="export-btn">Exportar Sábana a PDF</button>

  
</div>

</body>
</html>

```