

HERRAMIENTA DE GESTIÓN DE ESPACIOS Y HORARIOS EN UN CENTRO EDUCATIVO

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Julio de 2023

Autor

Diego Gutiérrez Martín

Tutoras

Ana Belén Gil González

Mercedes Maldonado Cordero

Certificado de los tutores TFG

Dña. Ana Belén Gil González, profesora del Departamento de Informática y Automática de la Universidad de Salamanca y Dña. Mercedes Maldonado Cordero, profesora del Departamento del Matemáticas de la Universidad de Salamanca

HACEN CONSTAR:

Que el trabajo titulado “Herramienta de gestión de espacios y horarios en un centro educativo”, que se presenta, ha sido realizado por Diego Gutiérrez Martín, con DNI 70922839Q y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en Ingeniería Informática en esta Universidad.

Salamanca, 3 de julio de 2023

Fdo.: Ana Belén Gil González _____ Fdo.: Mercedes Maldonado Cordero

Resumen

El presente trabajo tiene como objetivo desarrollar y presentar una herramienta de gestión de espacios y horarios específicamente diseñada para centros educativos (como caso de estudio se ha utilizado al Facultad de Ciencias de la Universidad de Salamanca). Se ha buscado crear una herramienta que sea útil tanto a los gestores encargados de la planificación y asignación de espacios y horarios, como a los docentes del centro.

El resultado ha sido una aplicación web que permite a los docentes visualizar tanto sus horarios personales como los horarios de otros docentes, los horarios generales de cada titulación y los horarios de todas las aulas. Además, dispone de una sección dedicada a la gestión de avisos o recordatorios, en la que los docentes y gestores podrán programar y recibir notificaciones personalizadas que aparecerán como ventanas emergentes en la pantalla principal en el momento que hayan indicado. Esto les ayudará a mantenerse al tanto de eventos o actividades importantes.

Por otro lado, los gestores podrán realizar modificaciones en los horarios, mientras consultan información sobre la ocupación de otras aulas permitidas para la asignatura seleccionada. Además, tendrán la opción de generar informes de ocupación de un aula en una determinada semana, estos informes estarán compuestos por diferentes gráficas cuyo objetivo será facilitar la visualización de diferentes apartados relevantes como la cantidad de alumnos por asignatura, media de alumnos por hora, mapa de calor, etc.

La aplicación también ofrece la posibilidad de cargar y eliminar datos de la base de datos mediante una interfaz sencilla e intuitiva. Por último, se ha implementado un algoritmo para la búsqueda de incompatibilidades, que detecta asignaciones simultáneas de clases en la misma aula o a un mismo docente.

Para el desarrollo del cliente se ha usado el framework de JavaScript Vue 3, con las bibliotecas Vue Router, Pinia, Apexcharts y Html2pdf.

En cuanto al backend (almacenamiento de datos y autenticación), se ha desarrollado mediante Supabase, una herramienta BaaS (Backend as a Service) de código abierto que utiliza centros de datos de AWS (Amazon Web Services).

Palabras clave: Aplicación web, Vue, Supabase, Scrum, desarrollo Full Stack

Summary

The present work aims to develop and present a space and time management tool specifically designed for educational institutions (the Faculty of Sciences at the University of Salamanca was used as a case study). The aim was to create a tool that would be useful both for managers in charge of planning and assigning spaces and schedules, and for teachers at the school.

The result is a web application that allows teachers to visualize both their personal schedules and the schedules of other teachers, as well as the general schedules of each degree program and the schedules of all classrooms. In addition, it includes a section dedicated to the management of notices or reminders, where teachers and administrators can schedule and receive personalized notifications that appear as pop-up windows on the main screen at the specified time. This will help them stay informed about important events or activities.

On the other hand, managers will be able to make modifications to the schedules while consulting information about the occupancy of other classrooms allowed for the selected subject. They will also have the option to generate occupancy reports for a specific classroom in a given week. These reports will consist of different charts aimed at facilitating the visualization of various relevant sections, such as the number of students per subject, average number of students per hour, heat map, etc.

The application also offers the possibility to upload and delete data from the database through a simple and intuitive interface. Finally, an algorithm has been implemented to search for incompatibilities, which detects simultaneous class assignments in the same classroom or to the same teacher.

For the client development, the JavaScript framework Vue 3 has been used, along with the Vue router, Pinia, Apexcharts, and Html2pdf libraries.

As for the backend (data storage and authentication), it has been developed using Supabase, an open-source Backend as a Service (BaaS) tool that utilizes AWS (Amazon Web Services) data centers.

Keywords: Web application, Vue, Supabase, Scrum, Full Stack development.

Índice

1.	Introducción	12
2.	Objetivos.....	13
2.1.	Objetivos del proyecto.....	13
2.2.	Objetivos personales	13
3.	Conceptos teóricos.....	14
4.	Técnicas y herramientas	15
4.1.	Visual Studio Code	15
4.2.	Adobe XD	15
4.3.	Visual paradigm	16
4.4.	Lucidchart	16
4.5.	Vue 3.....	17
4.6.	Supabase	18
4.7.	Miro	19
4.8.	Html2pdf.....	20
4.9.	Apexcharts.....	20
4.10.	Netlify	20
5.	Aspectos relevantes del desarrollo del proyecto.....	21
5.1.	Metodología de trabajo	21
5.1.1.	Roles en Scrum	21
5.1.2.	Artefactos de scrum.....	21
5.2.	Especificación de requisitos.....	22
5.2.1.	Participantes	22
5.2.2.	Actores.....	22
5.2.3.	Objetivos del sistema.....	23
5.2.4.	Requisitos de información	24
5.2.5.	Requisitos no funcionales	24
5.2.6.	Requisitos funcionales	25
5.3.	Especificación de diseño	27
5.3.1.	Colores y tipografía.....	27
5.3.2.	Patrón arquitectónico.....	28
5.3.3.	Subsistemas de diseño.....	30
5.3.4.	Clases de diseño	30
5.3.5.	Realización de caso de uso – diseño	34

5.3.6.	Diagrama de despliegue	35
5.3.7.	Diseño de la base de datos	35
5.4.	¿Por qué Supabase?	38
5.5.	Seguridad de la aplicación	40
5.5.1.	Row Level Security	41
5.5.2.	Acceso a la web	42
5.5.3.	With security invoker	43
5.6.	PostgreSQL Functions y PLV8	44
6.	Trabajos relacionados (estado del arte)	45
6.1.	Diaweb.....	45
6.2.	SIRA: Sistema Integral de Reserva de Aulas	45
6.3.	Robin.....	46
6.4.	Skedda.....	47
7.	Pruebas de usuario	48
8.	Funcionalidades de la aplicación	49
8.1.	Usuario.....	49
8.1.1.	Inicio de sesión	49
8.1.2.	Inicio	49
8.1.3.	Personal	52
8.1.4.	Horario general.....	54
8.1.5.	Horario aula	56
8.1.6.	Perfil	57
8.2.	Administrador	58
8.2.1.	Modificar horario.....	58
8.2.2.	Informes	60
8.2.3.	Cagar datos.....	63
8.2.4.	Eliminar datos	64
8.2.5.	Incompatibilidades	65
9.	Conclusiones y líneas de trabajo futuro.....	66
9.1.	Conclusiones.....	66
9.2.	Líneas de trabajo futuro	67
10.	Referencias	69

Índice de ilustraciones

<i>Ilustración 1. Adobe XD</i>	15
<i>Ilustración 2. Visual paradigm</i>	16
<i>Ilustración 3. Lucidchart</i>	17
<i>Ilustración 4. Logo Vue.js</i>	17
<i>Ilustración 5. Logo Supabase</i>	18
<i>Ilustración 6. Miro</i> https://miro.com/es/pizarra-virtual/	20
<i>Ilustración 7. Esquema de los actores</i>	23
<i>Ilustración 9. Tipografía Lato</i>	27
<i>Ilustración 10. Paleta de colores</i>	27
<i>Ilustración 11. Distribución cromática</i>	28
<i>Ilustración 12. Esquema del patrón Model-View-ModelView</i> https://www.geeksforgeeks.org/how-a-view-model-works-in-vue-js/	28
<i>Ilustración 13. Esquema "Single-File-Component" en Vue</i>	29
<i>Ilustración 14. Subsistemas de diseño</i>	30
<i>Ilustración 15. Subsistema Stores</i>	31
<i>Ilustración 16. Subsistema Views</i>	32
<i>Ilustración 17. Subsistema Components</i>	33
<i>Ilustración 18. Diagrama de secuencia de [UC-003] Ver horario</i>	34
<i>Ilustración 19. Diagrama de despliegue</i>	35
<i>Ilustración 20. Diseño de la base de datos</i>	36
<i>Ilustración 21. Ejemplo de la base de datos NoSQL de Firestore</i>	38
<i>Ilustración 22. Conexión dese JavaScript a Supabase</i>	39
<i>Ilustración 23. Ejemplo de consulta a Supabase</i>	39
<i>Ilustración 24. Interfaz de Supabase</i>	40
<i>Ilustración 25. Datos contenidos en una sesión de Supabase</i>	40
<i>Ilustración 26. Ejemplo de política de Row Level Security</i>	41
<i>Ilustración 27. Interfaz para la gestión de usuarios de Supabase</i>	42
<i>Ilustración 28. Desactivar el registro de nuevos usuarios en Supabase</i>	42
<i>Ilustración 29. Ejemplo de vista con "With (security_invoker) as"</i>	43
<i>Ilustración 30. Ejemplo de una PostgreSQL Function con PLV8</i>	44
<i>Ilustración 32. Diaweb</i>	45
<i>Ilustración 33. Interfaz de SIRA</i>	45
<i>Ilustración 34. Modos de visualización de SIRA</i>	46
<i>Ilustración 35. Interfaz de Robin</i>	47
<i>Ilustración 36. Skedda -</i> https://www.skedda.com	47
<i>Ilustración 37. Inicio de sesión</i>	49
<i>Ilustración 38. Inicio</i>	50
<i>Ilustración 39. Modales de los recordatorios</i>	50
<i>Ilustración 40. Docencia</i>	51
<i>Ilustración 41. Recordatorios</i>	51
<i>Ilustración 42. Modal añadir recordatorio</i>	51
<i>Ilustración 43. Personal seleccionar centro</i>	52
<i>Ilustración 44. Personal</i>	52

Ilustración 45. Opciones de filtrado Personal	52
Ilustración 46. Opciones de filtrado Personal - Nombre.....	52
Ilustración 47. Perfil de otro docente	53
Ilustración 48. Horario de otro docente.....	53
Ilustración 49. Docencia de otro docente	53
Ilustración 50. Horario general seleccionar titulación	54
Ilustración 51. Horario general	55
Ilustración 52. Horario general filtrado.....	56
Ilustración 53. Horario aula seleccionar aula	56
Ilustración 54. Horario aula.....	57
Ilustración 55. Perfil	57
Ilustración 56. Nuevas secciones Administrador	58
Ilustración 57. Modificar horario Inicio.....	58
Ilustración 58. Modificar horario	58
Ilustración 59. Opciones Hora inicio y Hora fin.....	59
Ilustración 60. Modificar horario con el horario del aula	59
Ilustración 61. Modificar horario con el horario del aula Segundo semestre.....	59
Ilustración 62. Modificar horario con el horario del aula Reservas/Exámenes	60
Ilustración 63. Informes seleccionar semana.....	60
Ilustración 64. Opciones primer gráfico	60
Ilustración 65. Diferentes opciones de visualización del primer gráfico	61
Ilustración 66. Informes	62
Ilustración 67. Opciones de descarga de los gráficos	62
Ilustración 68. Cargar datos	63
Ilustración 69. Cargar datos tabla contraída	63
Ilustración 70. Cargar datos tabla desplegada	63
Ilustración 71. Eliminar datos - Aulas permitidas por asignatura	64
Ilustración 72. Eliminar datos - Examen.....	64
Ilustración 73. Eliminar datos - Horario	64
Ilustración 74. Eliminar datos - Grupo	64
Ilustración 75. Eliminar datos - Personal por grupo.....	65
Ilustración 76. Buscar incompatibilidades	65

Índice de tablas

Tabla 1. Esquema para la especificación de los participantes	22
Tabla 2. Esquema para la especificación de usuarios	23
<i>Tabla 3. Esquema para la especificación de objetivos</i>	<i>23</i>
<i>Tabla 4. Esquema para la especificación de los requisitos de información</i>	<i>24</i>
<i>Tabla 5. Esquema para la especificación de los requisitos no funcionales</i>	<i>25</i>
<i>Tabla 6. Esquema de especificación de las historias de usuario</i>	<i>25</i>
<i>Tabla 7. Esquema para la especificación de los casos de uso</i>	<i>26</i>

1. Introducción

En los centros educativos, la gestión eficiente de los espacios y horarios es esencial para garantizar un entorno de aprendizaje óptimo y maximizar la utilización de los recursos disponibles. La asignación adecuada de aulas, laboratorios, salas de reuniones y otros espacios, así como la planificación de horarios para profesores y estudiantes, son aspectos cruciales que pueden influir en la calidad de la educación.

A menudo, la gestión de manual de estos aspectos resulta compleja, laboriosa y propensa a errores. Debido a esto, surge la necesidad de una herramienta capaz de simplificar estos procesos. Es aquí donde nuestra aplicación web para la gestión de espacios y horarios en un centro educativo entra en juego.

Se trata de una herramienta creada específicamente para la Facultad de Ciencias de la Universidad de Salamanca, de hecho, toma inspiración de otras plataformas web de la universidad como Diaweb [1] o SIRA [2]. Sin embargo, podría ser fácilmente adaptable a los requerimientos de otras instituciones o centros educativos y además, debido a su estructura modular basada en componentes, permitirá una fácil actualización y ampliación en el futuro.

En esta primera parte, se detallarán los objetivos tanto personales como del proyecto, los conceptos teóricos necesarios para el correcto entendimiento del proyecto, las técnicas y herramientas utilizadas durante el desarrollo, otro trabajos o aplicaciones relacionados y por último, las conclusiones y posibles futuras líneas de trabajo.

Por otro lado, se han realizado 5 anexos en los que se explicará la planificación temporal del proyecto, la especificación de requisitos software, la especificación de diseño, la documentación técnica de programación y finalmente, el manual de usuario.

2. Objetivos

2.1. Objetivos del proyecto

El objetivo del proyecto es plantear una herramienta integral que permita construir el histórico de los datos del modelo con intención de incluir inteligencia en el modelo que mejore la gestión de espacios y horarios. De una forma más específica:

- Facilitar la visualización del horario de cada docente tanto a otros docentes como a los administradores (que son los que tienen permitida la gestión de los datos), además de otros apartados como los datos de los exámenes, asignaturas impartidas, etc.
- Modificación de los horarios de una forma fácil e intuitiva.
- Visualización de los horarios de cada titulación.
- Visualización de los horarios de cada aula.
- Descarga de horarios en PDF.
- Generación de informes de ocupación de las aulas.
- Carga de datos mediante CSV a la base de datos mediante una interfaz sencilla.
- Eliminación de datos de la base de datos mediante una interfaz sencilla.
- Sistema de recordatorios para avisar al usuario de los diferentes eventos que introduzca.

2.2. Objetivos personales

- Ampliar los conocimientos sobre metodologías ágiles adquiridos durante la carrera y aplicación de estos a un proyecto.
- Aprender a desarrollar aplicaciones web mediante frameworks (en este caso Vue.js) y conocer las diferentes opciones disponibles.
- Aprender a desarrollar en JavaScript.
- Ampliar mis habilidades en HTML y CSS.
- Aprender a desarrollar el backend de una aplicación web, en este caso mediante la utilización de una plataforma BaaS (Backend as a service) como es Supabase.

3. Conceptos teóricos

- DOM: [3] Estructura de objetos que el navegador crea en memoria para cada elemento de la página.
- Front End: Parte de un sistema o aplicación web que se encuentra en el lado del cliente. Se relaciona con todo lo que ven las personas y se encarga de presentar la información e interactuar con el usuario.
- Back End: Parte de un sistema o aplicación web que se encuentra en el lado del servidor y no es accesible para los usuarios. Se ocupa de la gestión y procesamiento de datos, lógica y comunicación con otros sistemas.
- API (Application Programming Interface): Es un conjunto de reglas y protocolos que permite la interacción y comunicación entre diferentes aplicaciones de software. Una API define cómo se deben solicitar y entregar datos y funcionalidades entre sistemas, actuando como un intermediario que permite que las aplicaciones se comuniquen entre sí.
- Stakeholder: [4] Se refiere a cualquier individuo, grupo u organización que tiene un interés o una influencia directa o indirecta en un proyecto, producto o empresa. Los stakeholders pueden incluir tanto a personas internas como externas, como clientes, empleados, proveedores, accionistas, socios comerciales, reguladores y la comunidad en general.
- Framework: [5] Conjunto de herramientas, bibliotecas y componentes de software prediseñados que proporcionan un marco y un entorno para el desarrollo de aplicaciones. Es una infraestructura que permite a los desarrolladores construir y organizar software de manera eficiente y consistente, al proporcionar una base sólida y reutilizable.
- Vue CLI: [6] Es una herramienta de línea de comandos que facilita la creación, configuración y gestión de proyectos de Vue.js, proporcionando una estructura inicial, servidor de desarrollo y herramientas de empaquetado para agilizar el desarrollo de aplicaciones Vue.

4. Técnicas y herramientas

En cuanto a las herramientas utilizadas para el desarrollo son las siguientes:

4.1. Visual Studio Code

Visual Studio Code [7] es un editor de código fuente desarrollado por Microsoft para Windows, GNU/Linux y macOS, lanzado en 2015. Incluye soporte para la depuración, control Git incorporado, resaltado de sintaxis, finalización de código inteligente, snippets (fragmentos de código) y refactorización de código. Aunque es una aplicación gratuita de código abierto, la descarga oficial está disponible como software propietario e incluye funciones personalizadas por Microsoft.

4.2. Adobe XD

Adobe XD [8] es una herramienta de diseño y prototipado de interfaces de usuario (UI) y experiencia de usuario (UX). Es parte del paquete de software de Adobe y está diseñado específicamente para ayudar a los diseñadores a crear diseños de alta fidelidad, prototipos interactivos y realizar pruebas de usuario sin necesidad de escribir código. Proporciona una interfaz intuitiva y herramientas especializadas para crear diseños de pantalla, definir flujos de navegación, agregar interacciones, animaciones y transiciones, y colaborar con otros miembros del equipo en tiempo real.

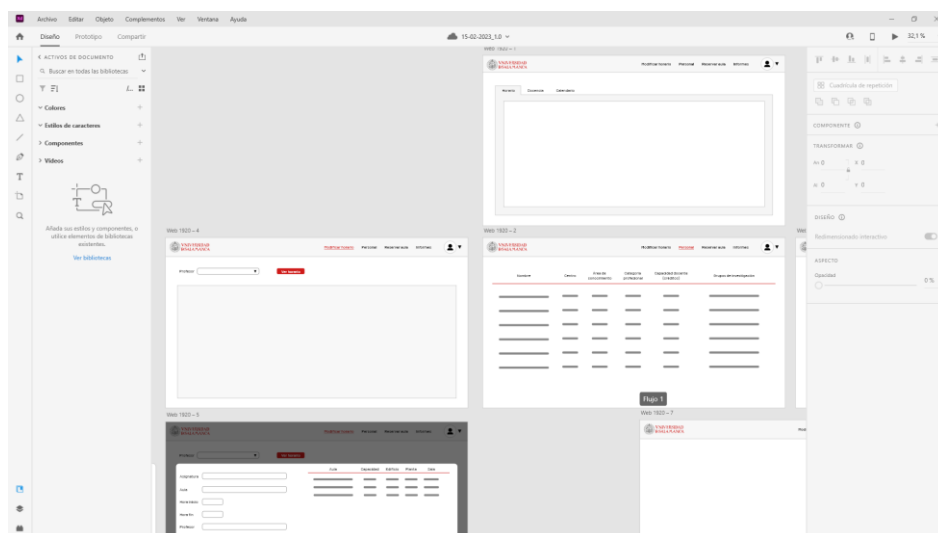


Ilustración 1. Adobe XD

Además, incluye la capacidad de diseñar y prototipar diseños responsivos, crear componentes reutilizables, definir estilos y reglas de diseño para mantener la coherencia, uniformidad y consistencia.

4.3. Visual paradigm

Visual Paradigm [9] es una herramienta de modelado y diseño de software que ayuda a los equipos de desarrollo a visualizar, diseñar y documentar sus sistemas y aplicaciones de manera efectiva. Proporciona una amplia gama de características y herramientas para el modelado de software, el diseño de diagramas y la generación de documentación técnica.

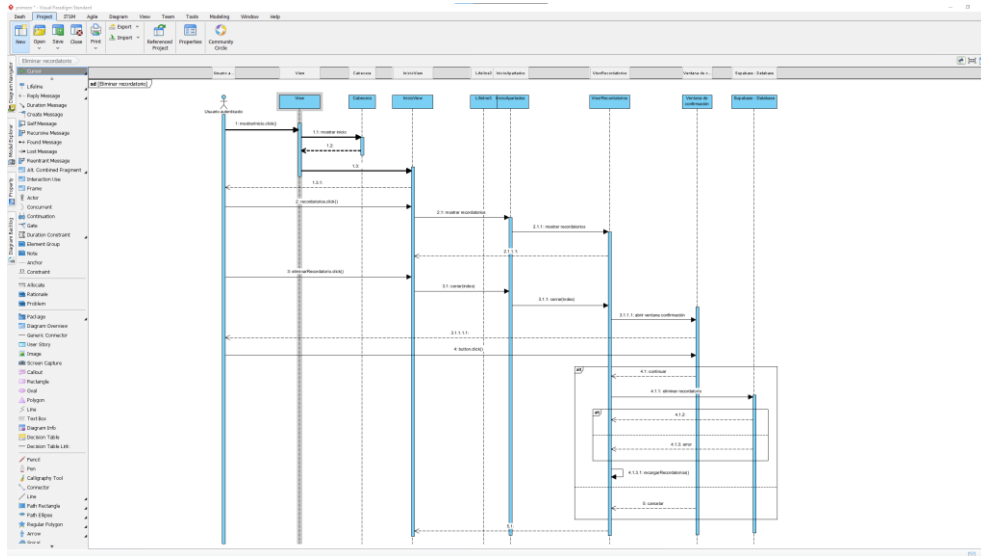


Ilustración 2. Visual paradigm

4.4. Lucidchart

Lucidchart [10] es una herramienta web de creación de diagramas que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas, esquemas de sitios web, diseños UML, mapas mentales, prototipos de software y muchos otros tipos de diagrama. Además, ofrece una amplia gama de plantillas y formas prediseñadas para agilizar el proceso de creación. Asimismo, puedes integrar Lucidchart con otras herramientas populares, como Google Drive, Microsoft Office y aplicaciones de gestión de proyectos.

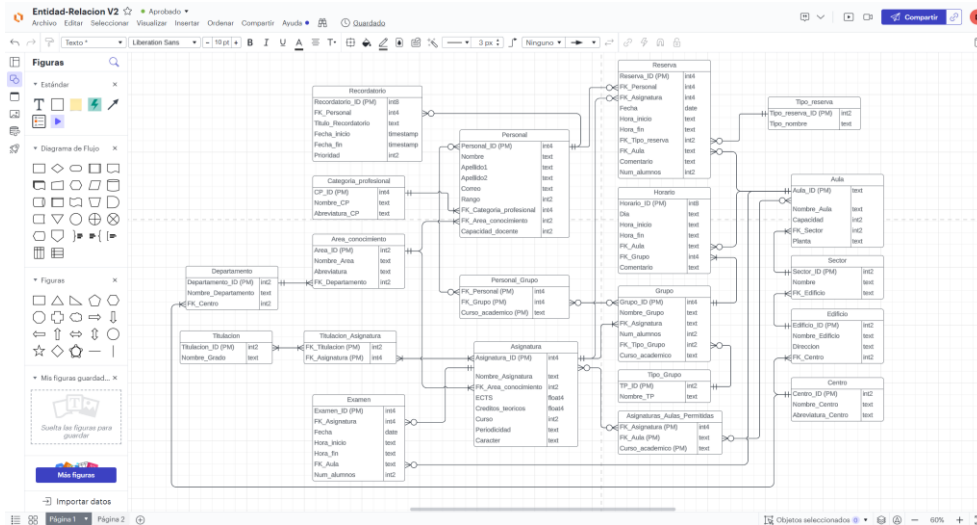


Ilustración 3. Lucidchart

4.5. Vue 3

[11] Se trata de un framework de JavaScript de código abierto utilizado para construir interfaces de usuario interactivas y componentes reutilizables. Proporciona un modelo de programación declarativo y basado en componentes que ayuda a desarrollar de manera eficiente interfaces de usuario, ya sean simples o complejas.



Ilustración 4. Logo Vue.js

Vue es reactivo, es decir, realiza un seguimiento automático de los cambios en el estado de JavaScript y actualiza eficientemente el DOM cuando ocurren cambios mediante la comparación entre el estado anterior y el actual, evitando la recarga de partes que no han cambiado.

En cuanto a los componentes, consisten en archivos “Single-File Components” (*.vue) que encapsulan la lógica del componente (JavaScript, HTML y CSS) en un mismo archivo.

- Vue Router

Vue Router [12] es la biblioteca oficial de enrutamiento para Vue.js. Proporciona una solución integral para definir y vincular rutas con componentes específicos dentro de una aplicación. Con Vue Router, se puede aprovechar características avanzadas como el enrutamiento anidado y dinámico, así como la capacidad de pasar parámetros en las rutas. Estas funcionalidades brindan un control avanzado sobre el comportamiento de la navegación, permitiendo una experiencia de usuario más personalizada y fluida.

- Pinia

[13] Es una biblioteca para Vue que permite compartir un estado entre todos los componentes o páginas. Se basa en el uso de “stores”, almacenes de estado que agrupan datos y lógica en un mismo lugar. Estas stores son reactivas, lo que implica que cualquier modificación en el estado se propagará automáticamente a todos los componentes que lo utilicen, asegurando la sincronización y simplificando la gestión de la información.

4.6. Supabase

Supabase [14] es una plataforma de código abierto para la construcción de backends seguros (BaaS o Backend as a Service) que combina una base de datos PostgreSQL [15] y una capa de API RESTful que permite realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para facilitar la construcción de aplicaciones web y móviles con una configuración mínima. Proporciona una alternativa a servicios como Firebase [16], ofreciendo una infraestructura escalable y flexible para el desarrollo de aplicaciones en tiempo real.



Ilustración 5. Logo Supabase

Los servicios utilizados en esta aplicación son:

- Supabase Database

[17] Como ya se ha mencionado antes, se trata de una base de datos con el motor PostgreSQL, que es conocido por su robustez, escalabilidad y soporte completo de SQL. Incorpora funcionalidades para la escucha en tiempo real (Supabase Realtime Server [18]), copias de seguridad y extensiones.

- Supabase Auth

[19] Proporciona un sistema de autenticación/autorización sencilla para aplicaciones web y móviles, pudiendo gestionar el registro de usuarios, el inicio de sesión, el restablecimiento de contraseñas y la gestión de perfiles de usuario. Para realizar la autenticación dispone de varias opciones:

- Correo electrónico y contraseña
- Magic Links (inicio de sesión con un clic)
- Proveedores externos (Google, Apple, GitHub, Azure, etc.)
- Inicio de sesión mediante móvil (Twilio, MessageBird y Vonage)

Por otro lado, utiliza un token de acceso JSON (JWT) generado después de la autenticación del usuario para administrar los derechos de usuario (permisos) en las solicitudes al backend.

- Supabase Storage

[20] Está diseñado para permitir el almacenamiento y la gestión de archivos en la nube de manera sencilla e integrada con otras funcionalidades de Supabase. Con Supabase Storage, los desarrolladores pueden cargar, almacenar y recuperar archivos en la nube de forma segura y sencilla. Además, dispone de un sistema de políticas de seguridad y opciones para la manipulación de imágenes.

- Supabase Edge Functions

[21] Son funciones en TypeScript que se ejecutan en el lado del servidor y son distribuidas globalmente, lo que permite reducir al mínimo las latencias. Han sido desarrolladas mediante Deno [22], un entorno de ejecución para JavaScript y TypeScript en el lado del servidor.

Otro aspecto muy interesante es que debido a que se trata de una plataforma de código abierto, muchos usuarios han aportado código al proyecto, permitiendo su compatibilidad con múltiples lenguajes y librerías, y añadiendo extensiones de todo tipo.

4.7. Miro

[23] Es una plataforma colaborativa en línea que ofrece un entorno virtual con una pizarra en la que se puede incluir todo tipo esquemas y diagramas. Permite colaborar en tiempo real, pudiendo ver y editar contenido, realizar comentarios y trabajar de forma conjunta en la resolución de problemas, lo que facilita la comunicación e interacción.

Como ya se ha comentado, se basa en la utilización de una pizarra en blanco, en la que se pueden incluir los diferentes elementos deseados. Para facilitar la planificación, dispone de una amplia variedad de plantillas prediseñadas creadas por la plataforma o por usuarios de la misma.

Además, dispone de más de 100 integraciones con otras herramientas para facilitar la organización.



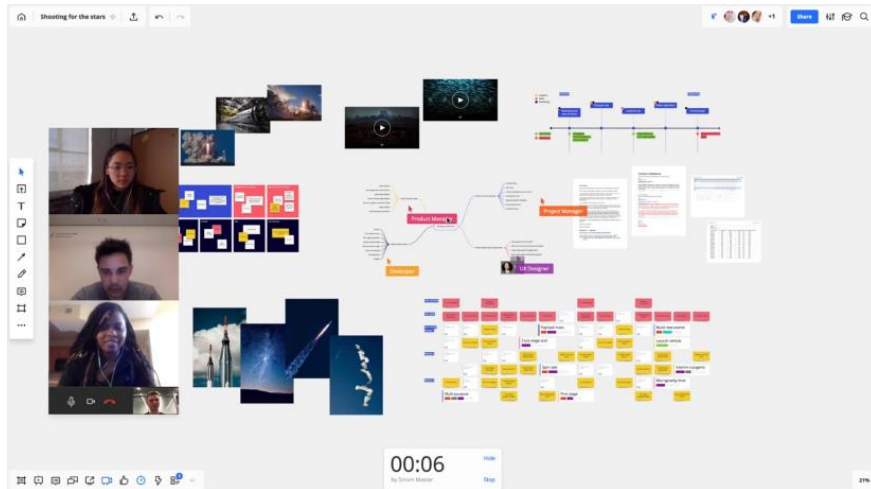


Ilustración 6. Miro <https://miro.com/es/pizarra-virtual/>

4.8. Html2pdf

[24] Biblioteca para convertir un documento HTML en un archivo PDF.

4.9. Apexcharts

[25] Es una biblioteca de código abierto para la creación de gráficos interactivos. Ofrece una amplia gama de opciones y configuraciones y es compatible con múltiples bibliotecas de JavaScript como Vue.js, React [26] o Angular [27]. Además, es flexible y reactiva.

4.10. Netlify

Netlify [28] es una plataforma de alojamiento web y despliegue de aplicaciones en la nube. Proporciona una forma sencilla de publicar sitios web estáticos y ofrece funcionalidades avanzadas para el despliegue de aplicaciones web y servicios backend.

Permite a los desarrolladores conectar su repositorio Git y desplegar automáticamente la web cada vez que el usuario actualiza el repositorio.

En este caso, el enlace a la aplicación web desplegada es: <https://gestor-de-espacios-y-horarios.netlify.app>

Para acceder hay que utilizar las credenciales:

- Modo usuario: usuario@tfg.es usuario2023
- Modo administrador: admin@tfg.es admin2023

5. Aspectos relevantes del desarrollo del proyecto

5.1. Metodología de trabajo

La metodología de trabajo escogida es Scrum [29], una metodología ágil de gestión de proyectos ampliamente utilizada. Se enfoca en lograr mejoras continuas y graduales a través de la realización de publicaciones pequeñas y frecuentes.

En Scrum se da mucha importancia a la experiencia, da por hecho que, al inicio de un proyecto, el equipo no es capaz de tener todas las variables en cuenta y que a medida que se desarrolla el proyecto se aprende y se realizan modificaciones al mismo.

Para conseguir esta flexibilidad, se realizan sprints de longitud fija, normalmente de 1 a 4 semanas, además de reuniones diarias, con el objetivo de entregar una versión funcional al final de cada sprint.

5.1.1. Roles en Scrum

- Product Owner (o Propietario del producto): Trabaja con los stakeholders para recopilar los requisitos y priorizar el Product backlog. Además, se asegura de que el producto se desarrolla de acuerdo a los requisitos.
- Scrum Master (o Facilitador): Es el encargado de que todo el equipo siga el marco de desarrollo Scrum y de resolver cualquier problema que pueda surgir. Ayuda al equipo a mejorar sus habilidades y prácticas en Scrum.
- Desarrollador: Profesional encargado del desarrollo del producto mediante la entrega de incrementos al final de cada sprint.

5.1.2. Artefactos de scrum

Los artefactos de scrum ofrecen información importante que el equipo de scrum utiliza para definir el producto y el trabajo que hay que hacer para crearlo. Existen tres artefactos:

- Product backlog: Lista priorizada de requisitos, funcionalidades y mejoras deseadas para el producto. Lo gestiona el Product Owner y debe centrarse en definir los requisitos de una forma general, siendo un documento que representa el qué va a ser construido en su totalidad. Se trata de un elemento que será constantemente revisado durante todo el proyecto y se irá refinando y ajustando a medida que se obtiene un mayor entendimiento.
- Sprint backlog: Durante la planificación de cada sprint, el equipo selecciona un conjunto de elementos del Product Backlog y los desglosa en tareas más específicas en el Sprint Backlog. Al igual que el Product Backlog, puede evolucionar durante el propio sprint sin modificar el objetivo fundamental del sprint.

- **Incremento:** Una vez finalizadas todas las tareas de cada sprint, el equipo debe entregar un incremento funcional y potencialmente desplegable del producto.

5.2. Especificación de requisitos

En este apartado se muestra la especificación de requisitos de una forma general, si se quiere conocer más información sobre la realización de este apartado se puede consultar el *Anexo II – Especificación de requisitos*. Se ha seguido la metodología de Durán y Bernárdez [29].

Para efectuar esta especificación, se ha se ha recogido información acerca de los principales objetivos que debe cumplir el sistema, los requisitos de información, requisitos funcionales y requisitos no funcionales.

5.2.1. Participantes

Los participantes del proyecto son:

- Diego Gutiérrez Martín
- Ana Belén Gil González
- Mercedes Maldonado Cordero

Para su especificación se utiliza el siguiente esquema:

Participante	Diego Gutiérrez Martín
Organización	Universidad de Salamanca
Rol	Analista, diseñador y programador
Desarrollador	Sí
Cliente	No
Usuario	No
Comentarios	

Tabla 1. Esquema para la especificación de los participantes

5.2.2. Actores

Los actores que van a interactuar con el sistema son los siguientes:

- Usuario sin autenticar
- Usuario autenticado
- Administrador

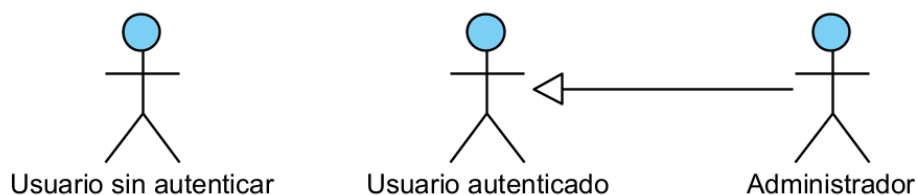


Ilustración 7. Esquema de los actores

Para su especificación se utiliza el siguiente esquema:

ACT-001	Usuario sin autentificar
<i>Versión</i>	1.0
<i>Autores</i>	Diego Gutiérrez Martín
<i>Fuentes</i>	
<i>Descripción</i>	Representa a cualquier usuario que no se haya autentificado aún en el sistema.
<i>Comentarios</i>	

Tabla 2. Esquema para la especificación de usuarios

5.2.3. Objetivos del sistema

Los objetivos del sistema son:

- Gestión de usuarios
- Gestión de horarios
- Gestión de exámenes
- Gestión de reservas
- Gestión de recordatorios

Para su especificación se utiliza el siguiente esquema:

OBJ-001	Gestión de usuarios
<i>Versión</i>	1.0
<i>Autores</i>	Diego Gutiérrez Martín
<i>Fuentes</i>	Ana Belén Gil González Mercedes Maldonado Cordero
<i>Descripción</i>	El sistema debe permitir a los usuarios iniciar y cerrar sesión y debe además gestionar sus permisos.
<i>Importancia</i>	Esencial
<i>Urgencia</i>	Crítica
<i>Estado</i>	Validado
<i>Estabilidad</i>	Alta
<i>Comentarios</i>	

Tabla 3. Esquema para la especificación de objetivos

5.2.4.Requisitos de información

Los requisitos de información son:

- Información de usuarios
- Información de horarios
- Información de grupos
- Información de aulas
- Información de asignaturas
- Información de departamentos
- Información de centros
- Información de exámenes
- Información de reservas
- Información de recordatorios

Para su especificación se utiliza el siguiente esquema:

IRQ-001	Información de usuarios
<i>Versión</i>	1.0
<i>Autores</i>	Diego Gutiérrez Martín
<i>Fuentes</i>	Ana Belén Gil González Mercedes Maldonado Cordero
<i>Dependencias</i>	
<i>Descripción</i>	El sistema debe almacenar información sobre los usuarios registrados en el sistema.
<i>Datos específicos</i>	<ul style="list-style-type: none">- Nombre- Correo- Rango- Categoría profesional- Área conocimiento- Categoría profesional- Capacidad docente
<i>Importancia</i>	Esencial
<i>Estabilidad</i>	Alta
<i>Comentarios</i>	

Tabla 4. Esquema para la especificación de los requisitos de información

5.2.5.Requisitos no funcionales

Los requisitos no funcionales son:

- Compatibilidad
- Seguridad
- Confidencialidad
- Usabilidad
- Rendimiento
- Escalabilidad

- Mantenimiento

Para su especificación se utiliza el siguiente esquema:

NFR-001	Compatibilidad
<i>Versión</i>	1.0
<i>Autores</i>	Diego Gutiérrez Martín
<i>Fuentes</i>	Ana Belén Gil González Mercedes Maldonado Cordero
<i>Dependencias</i>	
<i>Descripción</i>	El sistema debe ser compatible con diferentes navegadores web.
<i>Importancia</i>	Esencial
<i>Urgencia</i>	Alta
<i>Estado</i>	Validado
<i>Estabilidad</i>	Alta
<i>Comentarios</i>	

Tabla 5. Esquema para la especificación de los requisitos no funcionales

5.2.6.Requisitos funcionales

Los requisitos funcionales se han recogido en forma de historias de usuario y casos de uso en el *Anexo II - Especificación de Requisitos del Software*.

Para especificar los requisitos del Product backlog, se utilizan las historias de usuario [30].

Una historia de usuario es una técnica utilizada para capturar los requisitos o las necesidades del usuario en un formato conciso y comprensible, supone la unidad de trabajo más pequeña en un marco ágil. Es una descripción breve y simple de una funcionalidad o característica que el sistema o producto debe proporcionar, desde la perspectiva del usuario.

Para definir las se suele utilizar el siguiente esquema:

"Como [tipo de usuario], quiero [alguna funcionalidad], para [algún objetivo o beneficio]"

A continuación, se puede observar un par de ejemplos del Product backlog:

ID	Descripción	Urgencia
1	Como usuario sin autenticar, quiero poder iniciar sesión (y cerrarla) en la plataforma para acceder a la web.	Crítica
7	Como usuario autenticado, quiero poder ver los datos del personal registrado en la plataforma y ordenarlos para facilitar la búsqueda.	Crítica

Tabla 6. Esquema de especificación de las historias de usuario

Por otro lado, para la especificación de los casos de uso se utiliza el siguiente esquema:

UC-001		Iniciar sesión	
<i>Versión</i>	1.0		
<i>Autores</i>	Diego Gutiérrez Martín		
<i>Fuentes</i>	Ana Belén Gil González Mercedes Maldonado Cordero		
<i>Dependencias</i>	OBJ-001 Gestión de usuarios IRQ-001 Información de usuarios		
<i>Descripción</i>	El usuario inicia sesión para acceder a la página.		
<i>Precondición</i>	El usuario no está autenticado.		
<i>Secuencia normal</i>	<i>Paso</i>	<i>Acción</i>	
	1	El actor Usuario no autenticado introduce su correo y contraseña y envía la solicitud.	
	2	El sistema comprueba los datos proporcionados y le permite el acceso a la página.	
<i>Postcondición</i>			
<i>Excepciones</i>	<i>Paso</i>	<i>Acción</i>	
	2	Si los datos proporcionados no existen en la base de datos se muestra un mensaje de error.	
<i>Frecuencia</i>			
<i>Importancia</i>	Esencial		
<i>Urgencia</i>	Crítica		
<i>Estado</i>	Validado		
<i>Estabilidad</i>	Alta		
<i>Comentarios</i>			

Tabla 7. Esquema para la especificación de los casos de uso

5.3. Especificación de diseño

En este apartado se detallan los aspectos específicos relacionados con el diseño del producto, sistema o proceso en cuestión. Proporciona una visión general de las decisiones y consideraciones de diseño que se han tenido en cuenta durante el desarrollo del trabajo.

Este apartado está más detallado en el *Anexo III – Especificación de diseño*.

5.3.1. Colores y tipografía

Para escoger los colores y tipografía de la aplicación se ha seguido el Manual de Identidad Corporativa de la Universidad de Salamanca [46].

En este manual se indica que la familia tipográfica a utilizar es la familia “Lato” debido a que:

- Es muy extensa.
- Tiene una alta legibilidad (incluso en tamaños pequeños).
- Gran variedad de pesos y opciones en cursiva que permite jerarquizar y distinguir informaciones es un documento.
- Alta compatibilidad debido a que está alojada en Google Fonts.

LATO REGULAR
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
.,:;-_'"``^*\|/()[]{}¿?;!#\$%&+°ª@

Ilustración 8. Tipografía Lato.

En cuanto a los colores (paleta cromática), se indican tres colores principales a utilizar, Rojo Vítor, Gris aluminio y Azul VIII Centenario.

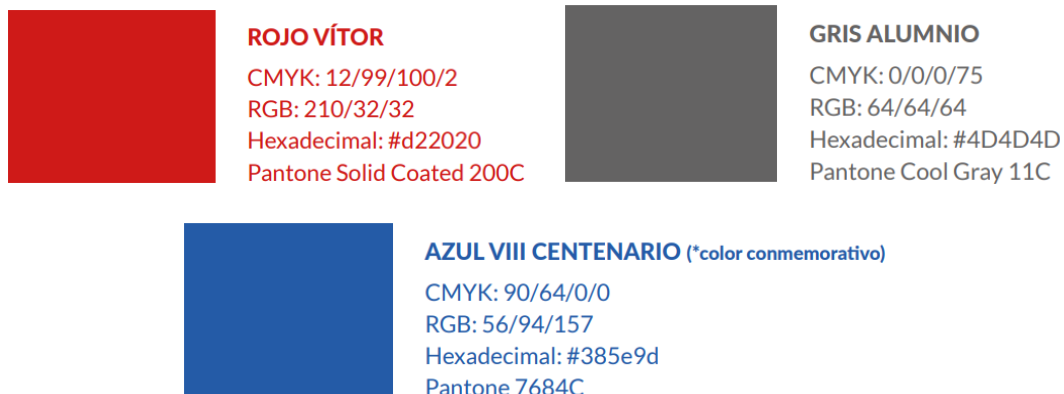
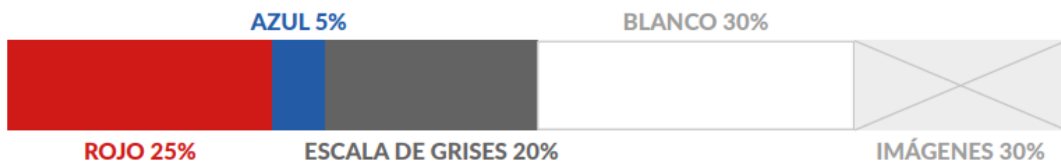


Ilustración 9. Paleta de colores

Además, el manual también indica las distribuciones cromáticas en porcentajes con y sin uso de imágenes, lo que permite hacerse una idea de cuanto debería aparecer cada color.

DISTRIBUCIÓN DEL USO DE LA PALETA CROMÁTICA EN PORCENTAJES RESPECTO AL 100% DE LA SUPERFICIE UTILIZADA INCLUYENDO IMÁGENES



DISTRIBUCIÓN DEL USO DE LA PALETA CROMÁTICA EN PORCENTAJES RESPECTO AL 100% DE LA SUPERFICIE UTILIZADA SIN IMÁGENES



Ilustración 10. Distribución cromática

Como se puede apreciar, los colores principales deberán ser el rojo, el gris y el blanco, siendo el azul un color de “apoyo” para resaltar aspectos más importantes que lo indicado en gris pero menos que lo indicado en rojo.

5.3.2. Patrón arquitectónico

En Vue.js, el patrón arquitectónico seguido es el MVVM o Model-View-ViewModel (Modelo-Vista-Vista del modelo), haciendo especial hincapié en la comunicación bidireccional entre Vista y Modelo, es decir, el ViewModel [32]. En este patrón, el Modelo los datos y la lógica subyacente de la aplicación, la Vista representa la interfaz gráfica y la Vista Modelo es una capa de abstracción que hace de puente entre ambos. Esto permite una vinculación bidireccional, lo que facilita el manejo de bloques HTML en comparación con otros patrones [33].

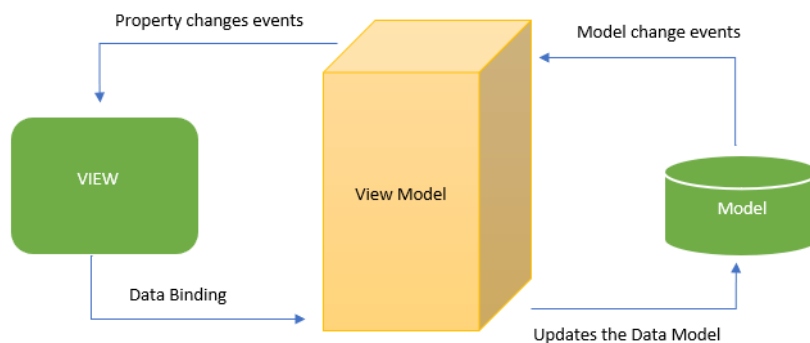



Ilustración 11. Esquema del patrón Model-View-ModelView <https://www.geeksforgeeks.org/how-a-view-model-works-in-vue-js/>

Para conseguir esta vinculación bidireccional se usan las siguientes directivas [34]:

- v-model: Se utiliza para crear enlaces bidireccionales de datos en inputs, áreas de texto, etc.
- v-bind: Se utiliza para manipular atributos HTML, cambiar el estilo o asignar clases.
- v-if: Se utiliza para el renderizado condicional. Si se cumple una condición se renderiza.
- v-on: Se utiliza para que los elementos del DOM escuchen los eventos.

Como ya se ha comentado en la sección de técnicas y herramientas, Vue utiliza archivos "Single-File Components" (*.vue) que encapsulan la lógica del componente (JavaScript, HTML y CSS) en un mismo archivo. En el caso de Vue, la estructura del código es la siguiente:



```
<script>
  // Código JavaScript/TypeScript
</script>

<template>
  <!-- HTML -->
</template>

<style>
  /* CSS */
</style>
```

Ilustración 12. Esquema "Single-File-Component" en Vue

Teniendo esto en cuenta, el Modelo generalmente se representa mediante la sección de JavaScript. La Vista generalmente se representa mediante un componente de Vue. El ViewModel generalmente se representa mediante una instancia de Vue.

Sus principales diferencias con el famoso MVC (Modelo-Vista-Controlador) son [35]:

- En MVC, el mediador entre la Vista y el Modelo es el Controlador, mientras que en MVVM el mediador es la VistaModelo.
- En MVC, la Vista contiene lógica. MVVM busca disminuir al mínimo la lógica presente en la Vista y trasladarla a la VistaModelo.
- La comunicación unidireccional de MVC (el Controlador tiene una referencia a la Vista) se sustituye por la comunicación bidireccional entre la Vista y la VistaModelo, lo que mejora la interactividad.

En resumen, Vue es un framework orientado al trabajo con componentes, que en su interior implementa la arquitectura MVVM.

5.3.3.Subsistemas de diseño

Reparto del sistema en subsistemas específicos y cómo se relacionan entre ellos.

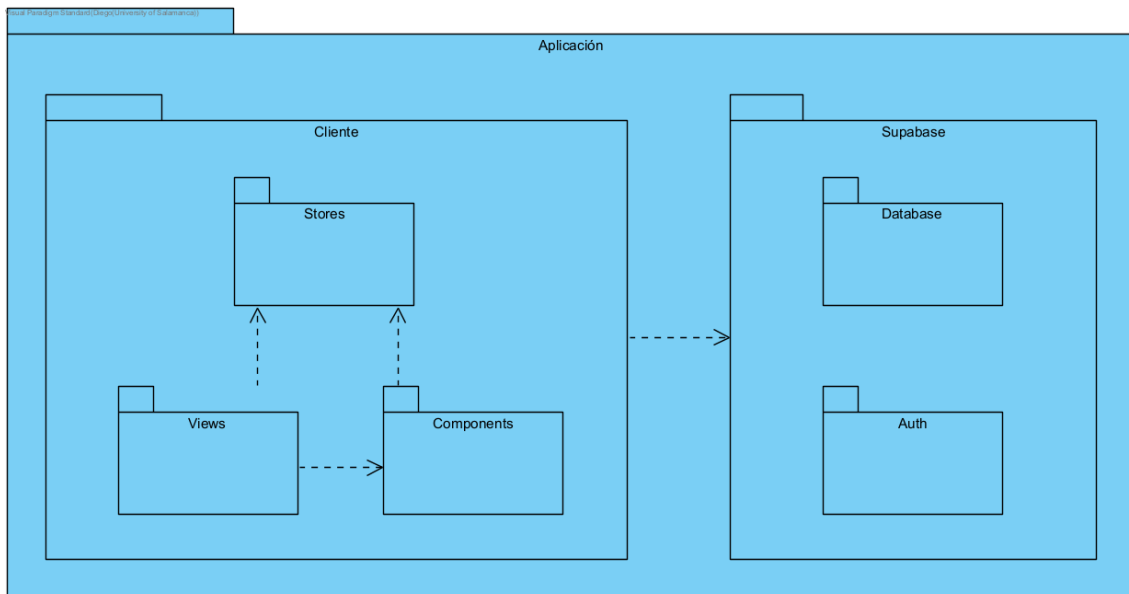


Ilustración 13. Subsistemas de diseño

5.3.4.Clases de diseño

Como ya se ha visto, el proyecto se ha dividido en los siguientes subsistemas de diseño en el cliente:

- Stores: Aquí se encuentran los almacenes de estado de Pinia. Se ha dividido en múltiples stores para mantener la modularidad.
- Views: Representan las páginas o secciones de la aplicación.
- Components: Donde se deben colocar los componentes de Vue.js reutilizables.

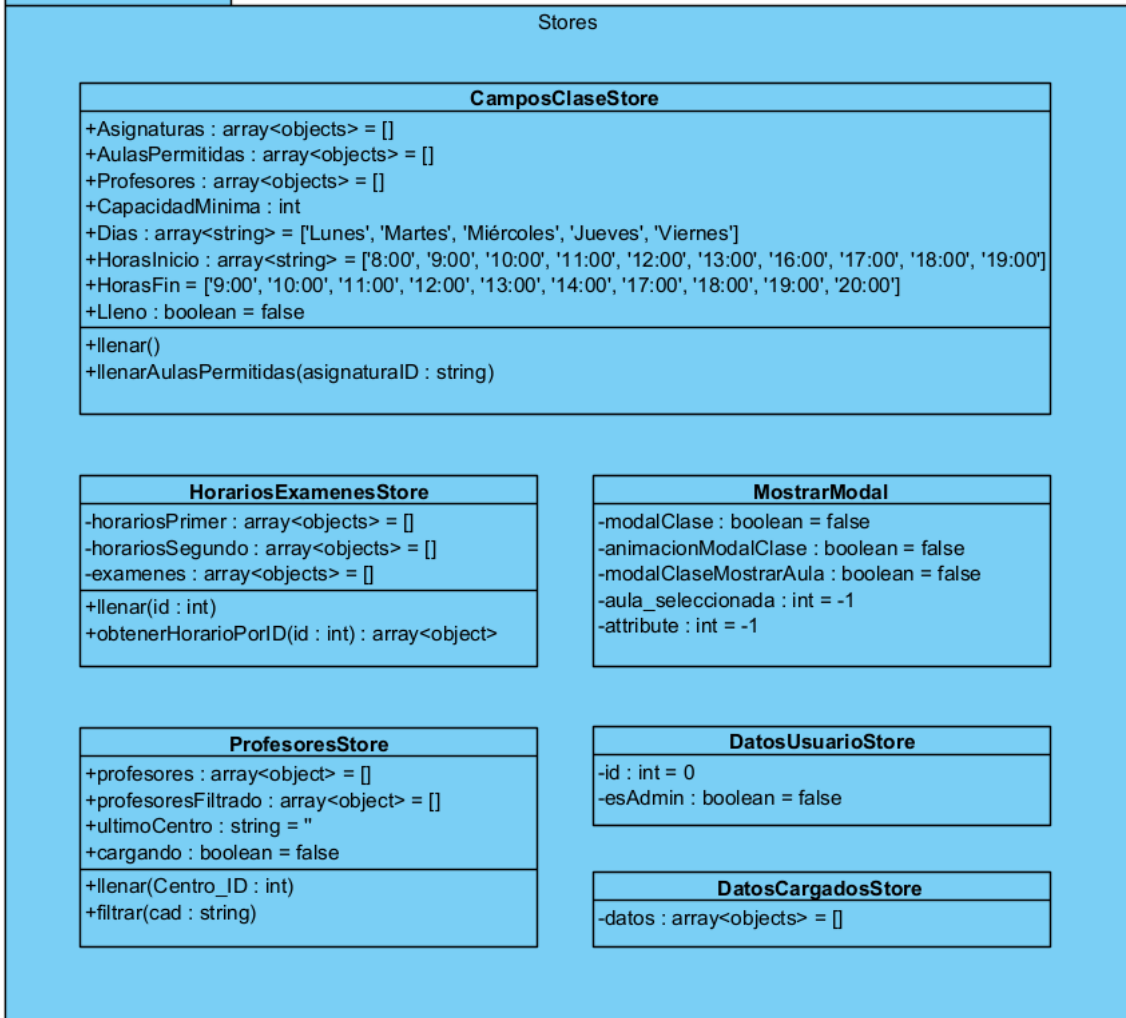


Ilustración 14. Subsistema Stores

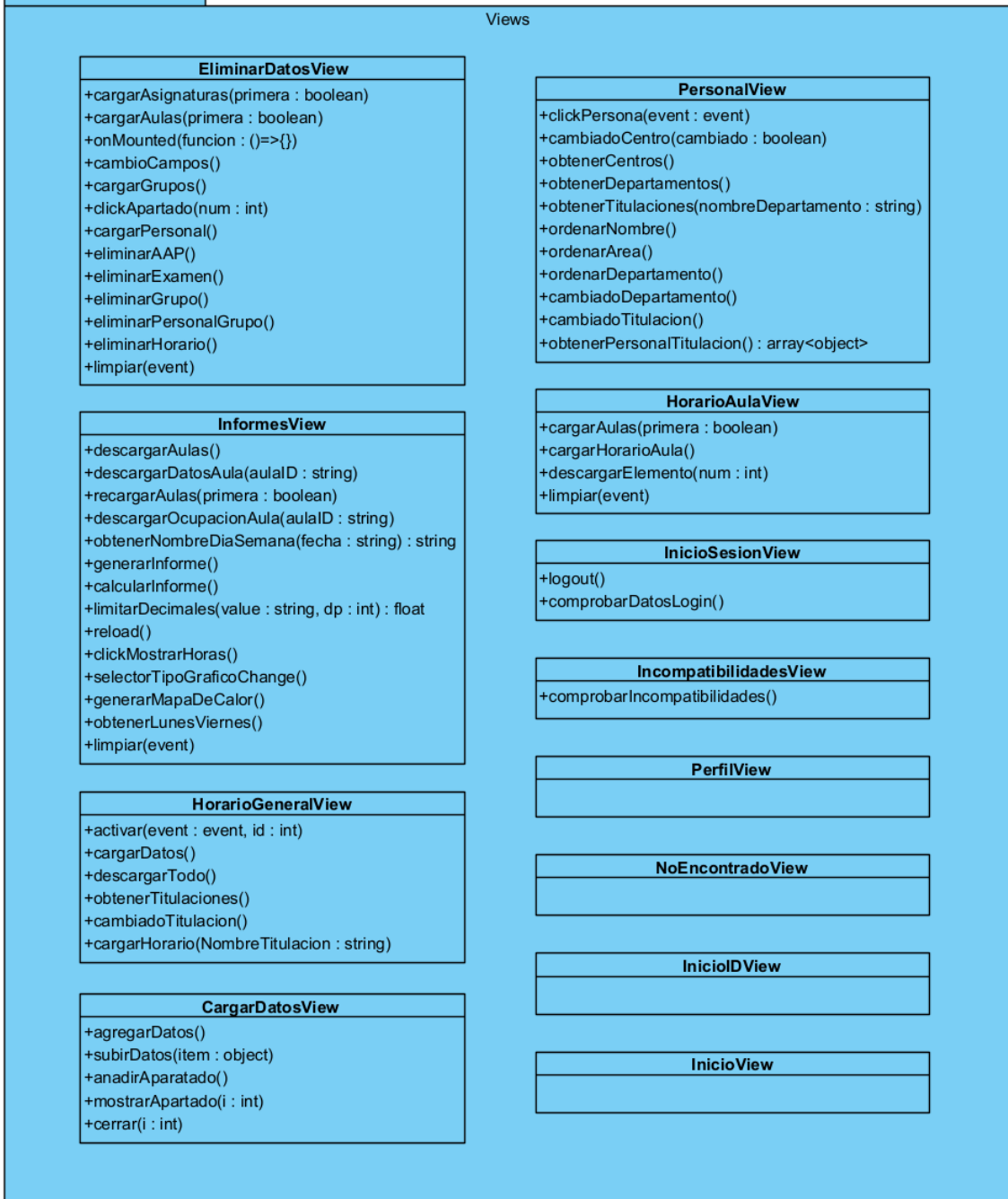


Ilustración 15. Subsistema Views

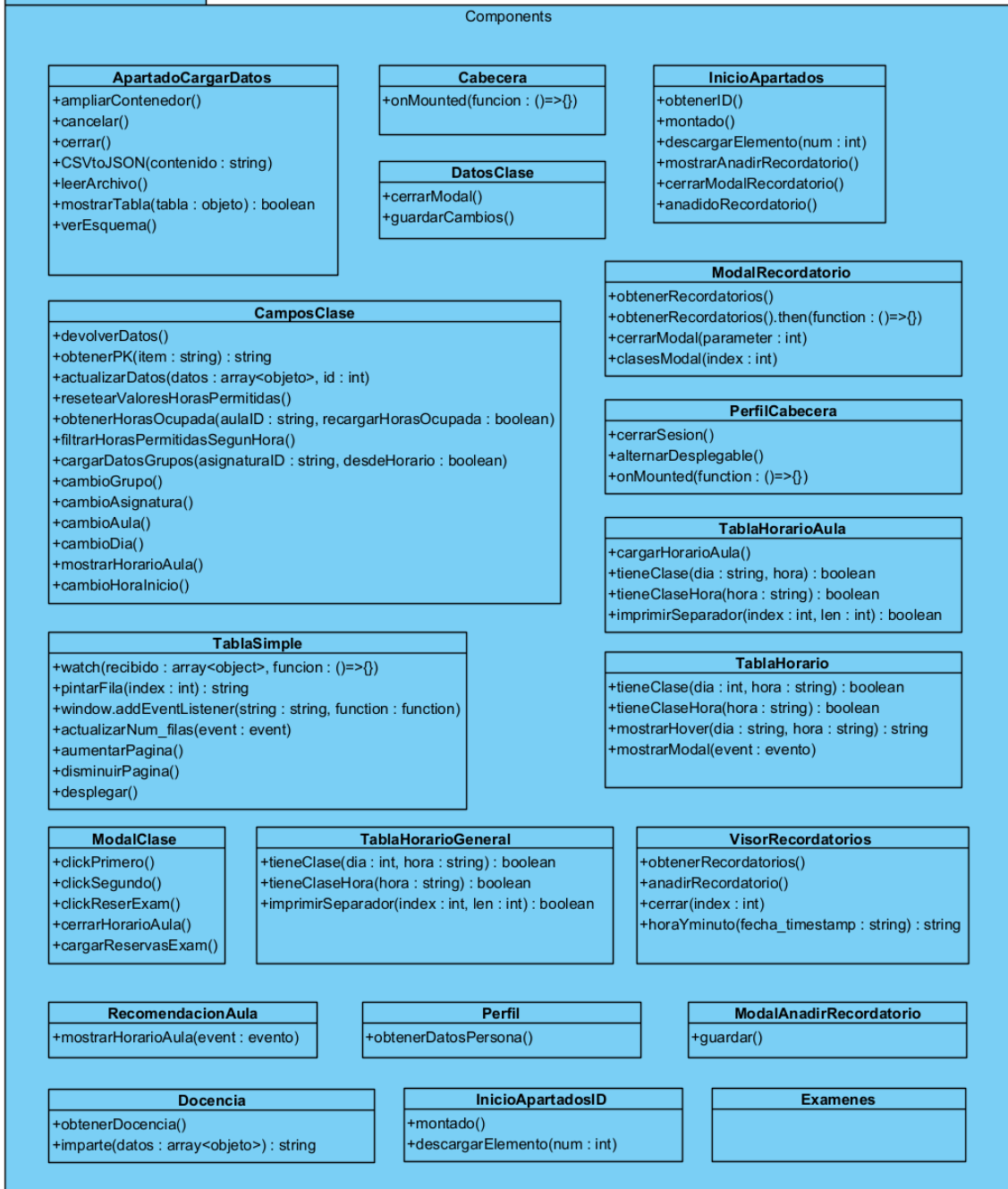


Ilustración 16.Subsistema Components

5.3.5. Realización de caso de uso – diseño

Los diagramas de secuencia representan interacciones entre los actores (usuarios) y el sistema, describiendo cómo se utiliza el sistema para lograr un objetivo específico.

En ejemplo de cómo se han realizado estos diagramas es:

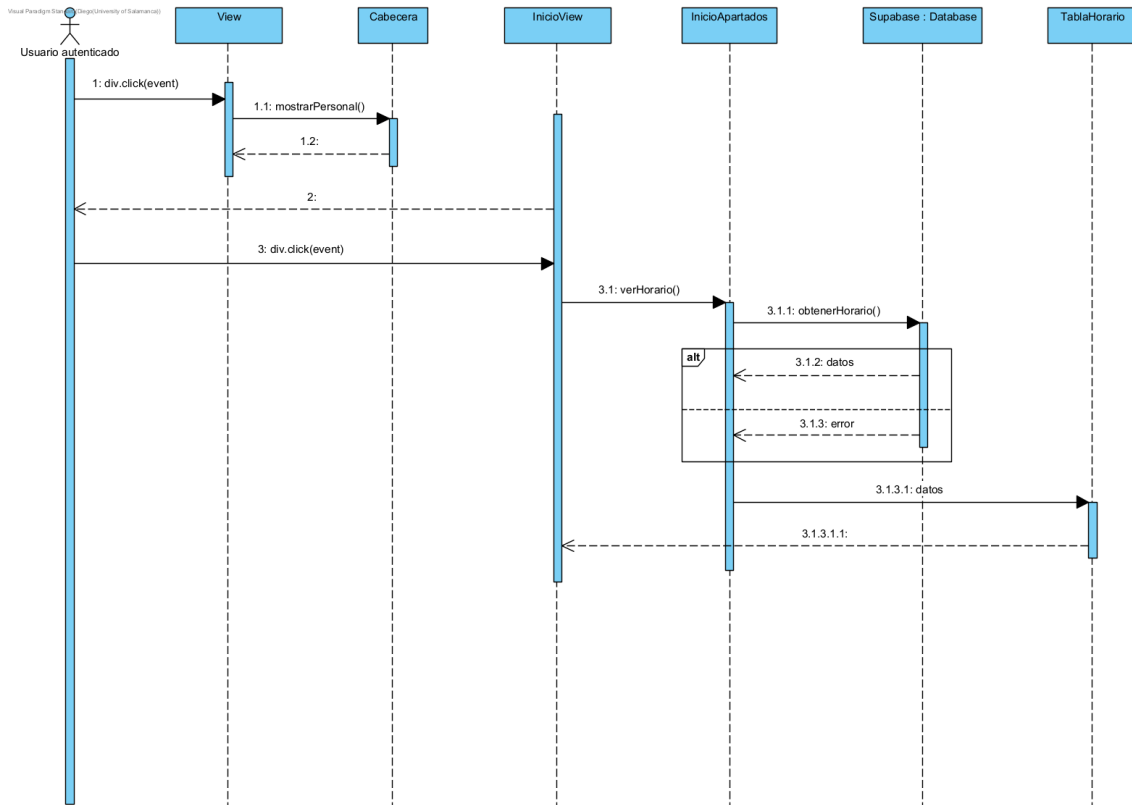


Ilustración 17. Diagrama de secuencia de [UC-003] Ver horario

5.3.6. Diagrama de despliegue

Se ha completado la fase de diseño con la creación de un diagrama de despliegue, el cual proporciona una visualización de la estructura y conexión de los nodos del sistema.

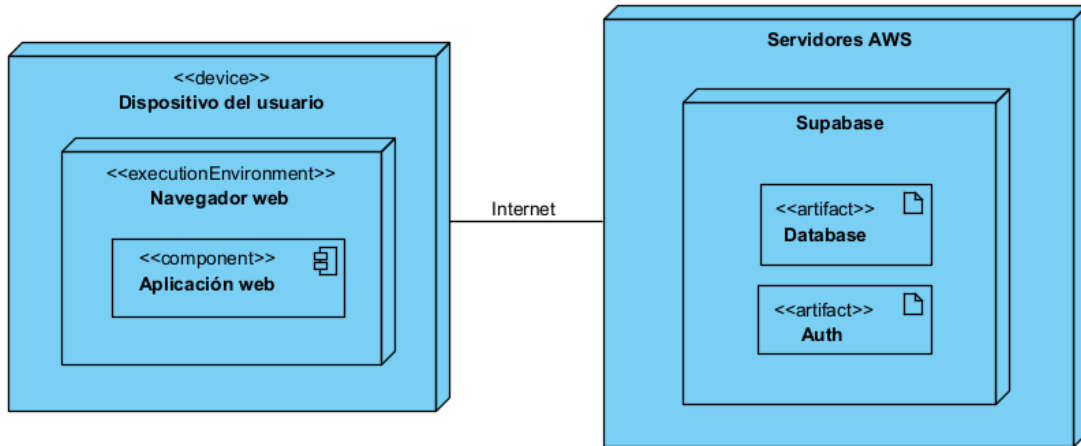


Ilustración 18. Diagrama de despliegue

5.3.7. Diseño de la base de datos

Para almacenar los datos de la aplicación se utiliza el servicio Supabase Database que como ya se ha mencionado antes, utiliza el motor PostgreSQL.

Para más detalles consultar el *Anexo III: Especificación de diseño* en el apartado de Diseño de datos.

El esquema de la base de datos es el siguiente:

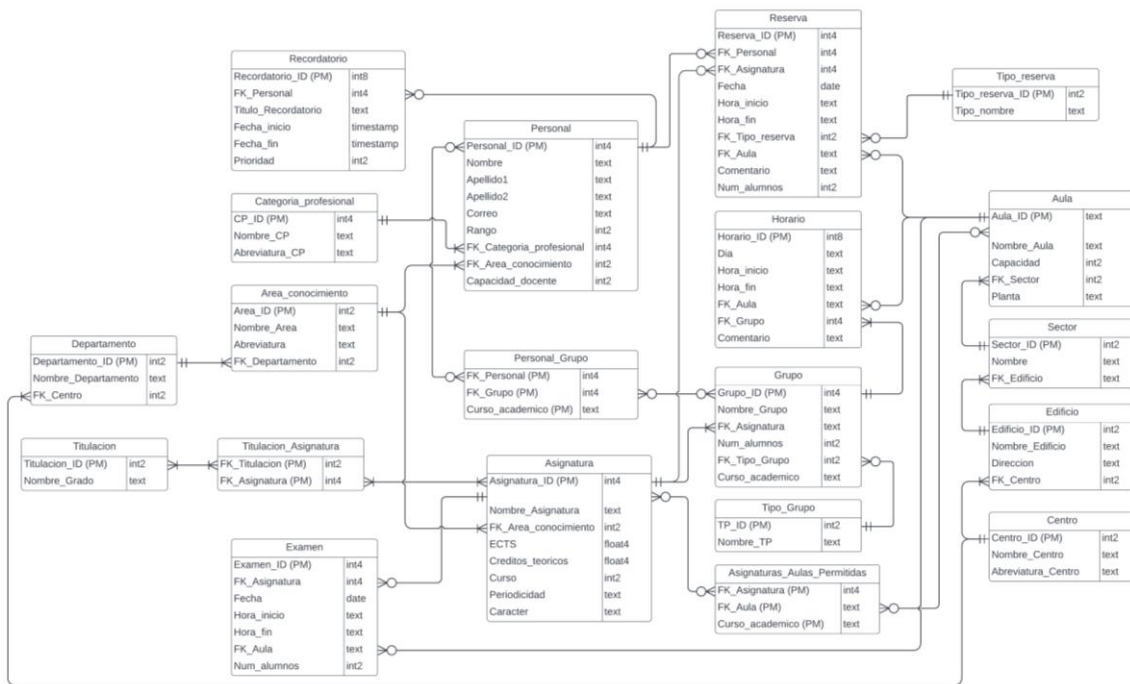


Ilustración 19. Diseño de la base de datos

Las claves primarias están indicadas como (PM) y las indicadas a continuación utilizan un entero autoincrementado como clave primaria:

- Area_conocimiento
- Categoria_profesional
- Centro
- Departamento
- Edificio
- Examen
- Grupo
- Horario
- Personal
- Recordatorio
- Tipo_grupo
- Titulación

La tabla que hace de nexo o unión es Grupo que representa los datos de un grupo de cualquier tipo (prácticas, teoría, seminario, etc.). Por ejemplo: TA1 de Programación 1.

Esta tabla grupo se enlaza con otras dos tablas muy importantes como son Personal y Horario para obtener los docentes que imparten clase en ese grupo y los horarios del grupo.

Otros datos a tener en cuenta son:

- Cada asignatura tiene una serie de aulas permitidas donde impartirla.

- Cada asignatura está asignada a un área de conocimiento (la cual está vinculada a un departamento, que a su vez está vinculado a un centro) y titulación.
- Cada titulación tiene una serie de asignaturas asignadas.
- Cada docente está asignado a un área de conocimiento y a una categoría profesional.
- Las reservas tienen asignado un docente que será el responsable de la misma y una asignatura. En caso de que por ejemplo se tratara de una charla, se podría añadir una asignatura cuyo nombre sea el tema a tratar.
- Los centros tienen edificios, los edificios están divididos en sectores y los sectores están formados por aulas. Esto se debe a que se barajó la posibilidad de incluir un sistema de recomendación de aulas basado en distancias, que finalmente fue desestimado, pero decidí mantener la descomposición.

5.4. ¿Por qué Supabase?

En un primer momento la plataforma más recomendada que encuentro que puede facilitar el desarrollo del back-end es Firebase. Firebase es una plataforma de Google que proporciona servicios y herramientas para el almacenamiento de datos en la nube, autenticación de usuarios, hosting de aplicaciones y ejecución de funciones entre otras.

Para el almacenamiento Firebase Firestore utiliza una base de datos NoSQL. A diferencia de las bases de datos SQL, no utiliza tablas ni filas. En su lugar, se almacenan los datos en documentos, que a su vez se organizan en colecciones. Cada documento contiene un par clave-valor (como en JSON) y puede contener subcolecciones y objetos anidados.

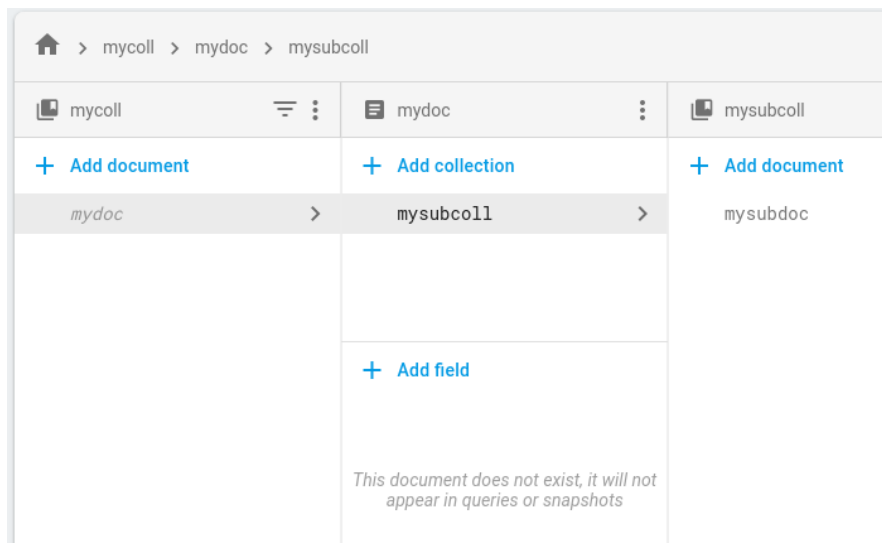


Ilustración 20. Ejemplo de la base de datos NoSQL de Firestore

Sin embargo, tras unos días haciendo pruebas y leyendo la documentación, sigo sin acabar de visualizar cómo organizar los datos, por lo que comienzo a investigar otras opciones BaaS con una base de datos SQL, siendo Supabase la escogida.

Supabase como se ha mencionado antes proporciona una base de datos PostgreSQL, autenticación y autorización, almacenamiento de archivos, eventos en tiempo real, etc. De hecho, el eslogan que utilizan es “Construye en un fin de semana, escala a millones: Supabase es una alternativa de código abierto a Firebase para construir backends seguros y eficientes utilizando Postgres con una configuración mínima.”

Todo esto facilita mucho el desarrollo del back-end siendo necesarias para realizar la conexión a la base de datos apenas 5 líneas de código [36]:

```
src - supabase.js
1 import { createClient } from '@supabase/supabase-js'
2
3 const supabaseUrl = 'https:// --- .supabase.co'
4 const supabaseKey = ''
5 const supabase = createClient(supabaseUrl, supabaseKey)
6
7 export default supabase
```

Ilustración 21. Conexión desde JavaScript a Supabase

En cuanto a las consultas mediante la API [37]:

```
let { data, error } = await supabase.from('Asignatura').select('*').eq('Asignatura_ID', 101120)
if (error) {
  //Manejar el error
} else {
  ...
}
```

Ilustración 22. Ejemplo de consulta a Supabase

El código anterior recupera todos los valores de la tabla Asignatura donde Asignatura_ID es igual a 101120.

Además, la documentación es muy completa, dispone de vídeos de los creadores explicando los diferentes apartados, una comunidad grande y activa, compatibilidad con múltiples lenguajes (Javascript, Flutter, Python, C#, Swift y Kotlin) e incluso dispone de una Inteligencia Artificial (Supabase AI) para la resolución de dudas.

Proporciona también una interfaz muy intuitiva que simplifica la gestión.

En cada llamada a la base de datos, automáticamente la API de Supabase envía también el token del usuario, garantizando en todas las consultas que usuario es quien dice ser para autorizar o no el acceso a los datos.

5.5.1. Row Level Security

Para garantizar la seguridad de las tablas de la base de datos, se utiliza Row Level Security (RLS) [40]. En un primer momento decidí utilizar roles, pero tras investigar y realizar varias pruebas desestimé esta idea, ya que el uso de diferentes roles para los permisos de consulta actualmente no funciona correctamente.

RLS es un sistema de seguridad implementado en sistemas de gestión de bases de datos para controlar y restringir el acceso a filas o registros individuales dentro de una tabla de la base de datos, basándose en ciertos criterios o condiciones específicas llamadas políticas.

Este sistema es muy interesante ya que es muy potente y flexible, permite definir privilegios de accesos a nivel de fila en lugar de a nivel de tabla, permitiendo incluso que usuarios con el mismo rol, tengan diferentes niveles de acceso [41]. En la siguiente imagen se puede ver un ejemplo de una política que permite actualizar los datos de un recordatorio, solamente si el recordatorio fue creado por el mismo usuario:

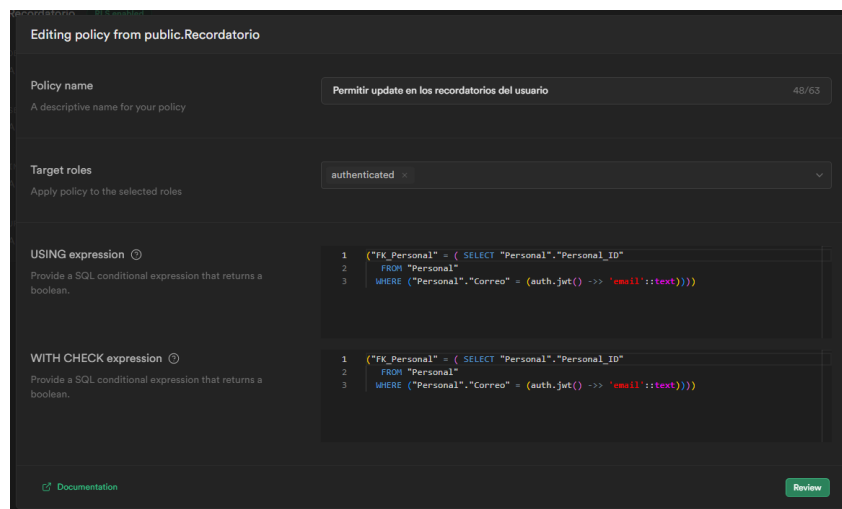


Ilustración 25. Ejemplo de política de Row Level Security

5.5.2. Acceso a la web

En un primer momento y para facilitar el inicio de sesión, decido implementar la autenticación con Google.

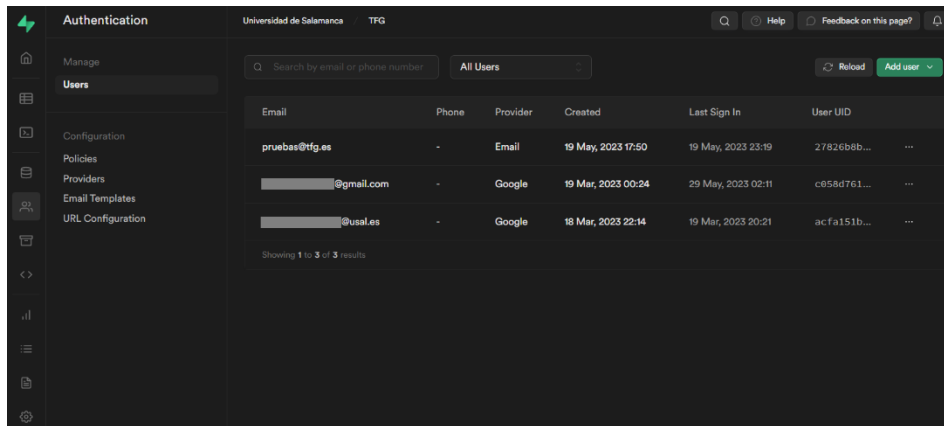


Ilustración 26. Interfaz para la gestión de usuarios de Supabase

Pero el problema con este sistema es que la plataforma no dispone en este momento de forma alguna de limitar quién puede y quién no, registrarse en la web. De forma que cualquier persona podría utilizar su cuenta de Google para crear una cuenta y acceder a la plataforma. Esto sería una brecha de seguridad, ya que la web está pensada para ser solamente accesible a personas (en un principio solo docentes) ligadas a la Universidad de Salamanca.

Para evitarlo la única manera que encuentro es activar solo el inicio de sesión con correo electrónico y contraseña, desactivando la opción que permite el registro de nuevos usuarios en Supabase y permitiendo solamente la adición de usuarios desde el panel de control.

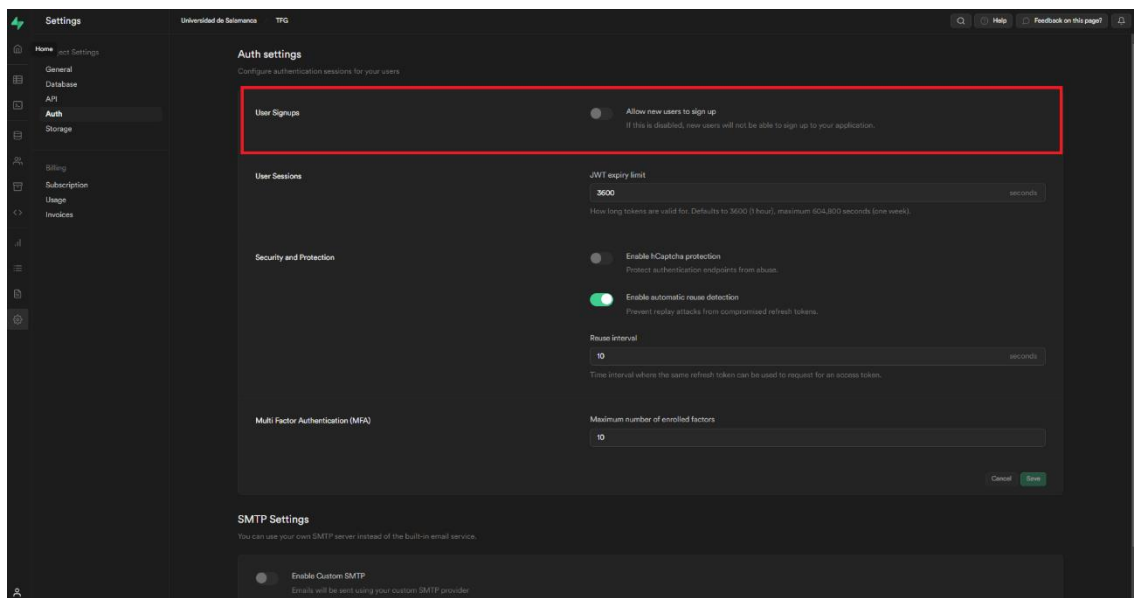
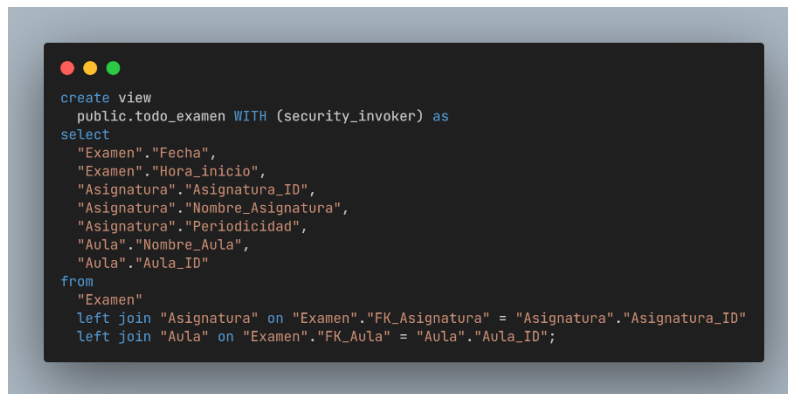


Ilustración 27. Desactivar el registro de nuevos usuarios en Supabase

5.5.3. With security invoker

Las vistas o views en PostgreSQL, por defecto, se crean con permisos del usuario que las crea, lo que supone un grave problema de seguridad, ya que cualquiera podría obtener los datos de la vista a pesar de que no tienen permisos.

Para evitar esto, desde PostgreSQL 15 existe la opción de especificar que se ejecute la consulta con los permisos del usuario que la invoca mediante la opción “WITH (security_invoker) as” [42].



```
create view
public.todo_examen WITH (security_invoker) as
select
"Examen"."Fecha",
"Examen"."Hora_inicio",
"Asignatura"."Asignatura_ID",
"Asignatura"."Nombre_Asignatura",
"Asignatura"."Periodicidad",
"Aula"."Nombre_Aula",
"Aula"."Aula_ID"
from
"Examen"
left join "Asignatura" on "Examen"."FK_Asignatura" = "Asignatura"."Asignatura_ID"
left join "Aula" on "Examen"."FK_Aula" = "Aula"."Aula_ID";
```

Ilustración 28. Ejemplo de vista con "With (security_invoker) as"

5.6. PostgreSQL Functions y PLV8

Las funciones PostgreSQL [43] permiten realizar operaciones que requerirían de múltiples consultas e iteraciones y se almacenan dentro de la propia base de datos. Pueden realizar operaciones complejas y cálculos en la base de datos de manera eficiente.

Funcionan como cualquier otra función, pueden recibir parámetros de entrada, crear variables, realizar consultas a la base de datos y devolver resultados.

PLV8 [44] es una biblioteca compartida que proporciona un lenguaje procedimental de PostgreSQL basado en el motor de JavaScript V8. Supabase dispone de una extensión que activa esta librería, con la que se puede escribir en JavaScript una función que puede ser invocada desde SQL.

Se ha hecho uso de esta biblioteca para la creación de todas las funciones de la base de datos. La creación de dichas funciones ha sido necesaria en los casos en los que no era posible realizar la operación con la API de JavaScript de Supabase o debido a consideraciones relacionadas con la seguridad, como es el caso de la función que se puede observar a continuación, en la que se comprueba si el usuario es administrador:

A screenshot of a code editor showing a PostgreSQL function definition. The code is written in SQL and uses PLV8 (PostgreSQL Language Version 8) for procedural logic. The function is named 'esAdmin' and takes a 'correo' parameter of type 'text'. It returns a boolean value. The logic involves executing a SQL query using 'plv8.execute' to check for a user with a specific email and role in a table named 'Personal'. If the query returns no results, it returns false; otherwise, it returns true. The function is defined using the 'language plv8' clause.

```
create or replace function esAdmin(correo text)
returns boolean as $$
    var result = plv8.execute(
        'select * from "Personal" where "Correo" = $1 and "Rango" = 0;', [correo]
    );
    if(result.length == 0)
        return false
    else
        return true;
$$ language plv8;
```

Ilustración 29. Ejemplo de una PostgreSQL Function con PLV8

Para más detalles consultar el *Anexo IV: Documentación técnica de programación*.

6. Trabajos relacionados (estado del arte)

6.1. Diaweb

Diaweb [1] es la página web del departamento de Departamento de Informática y Automática de la Universidad de Salamanca. De ella he extraído los esquemas y apartados para la disposición de los datos de las páginas Horario, Docencia y Personal.



Ilustración 30. Diaweb

Se trata de una página con múltiples opciones y no se centra solamente en la gestión de horarios, pero me ha parecido correcto seguir una disposición de los datos ya existente en la universidad para facilitar la navegación y entendimiento de la información mostrada.

6.2. SIRA: Sistema Integral de Reserva de Aulas

SIRA [2] es el sistema de reserva de espacios de la Universidad de Salamanca. Permite a los usuarios de los centros consultar la disponibilidad de un espacio y hacer la reserva llamando o enviando un correo a la conserjería correspondiente.



Ilustración 31. Interfaz de SIRA

Como se puede apreciar en la imagen muestra la disponibilidad por horas de cada aula y día, pudiendo el usuario filtrar por fecha, capacidad mínima o centro. Además, dispone de varios modos de visualización del horario.

The image shows a complex timetable interface. At the top, there are tabs for different days: 'Lunes, 22/05/2023', 'Martes, 23/05/2023', and 'Miércoles, 24/05/2023'. Below these, a grid of rooms (Aula A1, Aula A2) is shown against a time axis from 00:00 to 21:00. Each cell in the grid contains text representing a subject or course, such as 'Perfiles de Ingeniería', 'Medicina', 'Seguridad', 'Química', 'Física', 'Matemáticas', 'Ingeniería', 'Química Orgánica', 'Física II', 'Química Analítica', 'Química Orgánica', 'Física III', 'Química Orgánica', 'Física IV', 'Química Orgánica', 'Física V', 'Química Orgánica', 'Física VI', 'Química Orgánica', 'Física VII', 'Química Orgánica', 'Física VIII', 'Química Orgánica', 'Física IX', 'Química Orgánica', 'Física X', 'Química Orgánica', 'Física XI', 'Química Orgánica', 'Física XII', 'Química Orgánica'. The grid is color-coded to indicate availability or occupancy status.

Ilustración 32. Modos de visualización de SIRA

6.3. Robin

Robin [46] es una plataforma de administración de espacios de trabajo y reservas. Permite a las empresas u organizaciones gestionar y optimizar el uso de sus espacios físicos, como salas de reuniones, escritorios, espacios comunes y recursos compartidos.

Ofrece una interfaz sencilla e intuitiva y proporciona información en tiempo real sobre la ocupación de los espacios, además, dispone de funciones de automatización gracias a su IA, lo que agiliza los flujos de trabajo.

Se trata de una plataforma con algunos aspectos en común o parecidos al trabajo planteado, pero muy orientada gestión de espacios en empresas

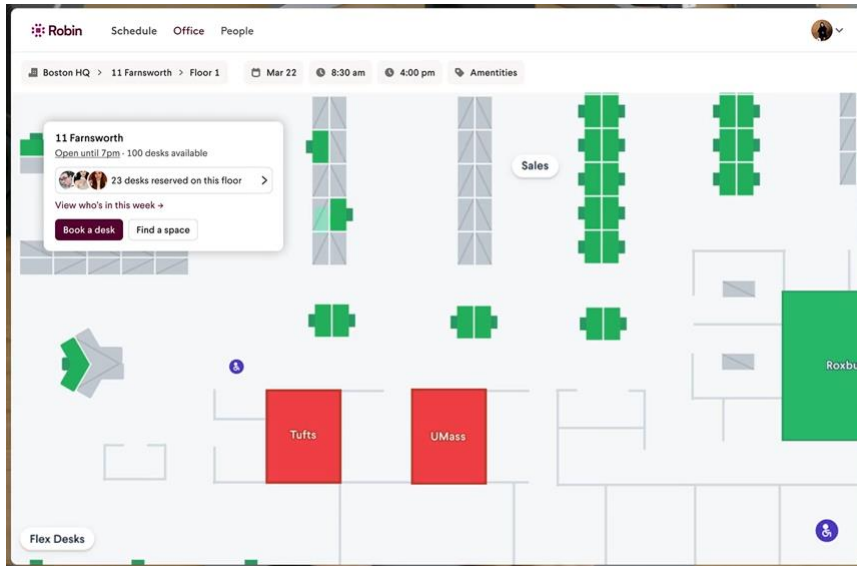


Ilustración 33. Interfaz de Robin

6.4. Skedda

Software diseñado para gestionar y optimizar el uso de espacios, instalaciones y recursos. Se utiliza principalmente para lugares como espacios de coworking, salas de reuniones, escritorios, laboratorios, instalaciones deportivas, etc.

Permite a los administradores configurar los espacios y disponibilidad, definir reglas y administrar las reservas.

Está enfocado para el uso tanto en empresas como en universidades, de hecho lo utilizan algunas universidades conocidas internacionalmente como la Universidad de Harvard, la Universidad de California o la Universidad de Washington.

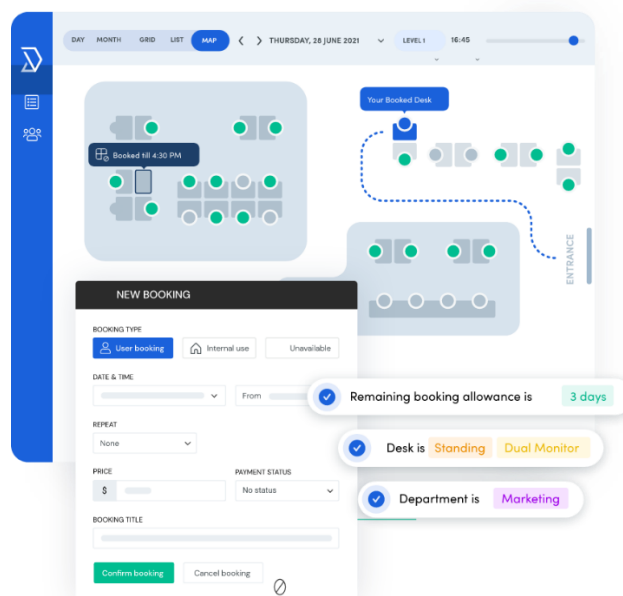


Ilustración 34. Skedda - <https://www.skedda.com>

7. Pruebas de usuario

En esta sección, se describen las pruebas llevadas a cabo con usuarios para validar el funcionamiento del sistema y reducir los posibles fallos.

Durante el desarrollo de cada elemento del sprint backlog, se realizaron pruebas unitarias para garantizar su correcto funcionamiento. También se llevaron a cabo pruebas de integración cuando se finalizaban las pruebas unitarias para garantizar que no hubiera ningún error a la hora de combinar los diferentes elementos del sistema.

Al finalizar cada sprint y antes de entregar el incremento, se realizaron pruebas exhaustivas en todo el sistema, asegurándose de que no se producían comportamientos anómalos en ningún aspecto. Esto permitió agilizar y facilitar las reuniones con los stakeholders.

Una vez la aplicación web alcanzó una versión cercana a la versión final, se llevaron a cabo pruebas con usuarios externos para validar la interfaz gráfica y los flujos de los casos de uso. En estas pruebas se les pidió a los usuarios, sin ninguna explicación anterior, que realizaran algunos casos de uso y que fueran explicando el porqué de sus decisiones, lo que me permitió comprender si el diseño era sencillo y entendible. Además, se les pidió que comentaran aspectos a mejorar.

Los aspectos más comentados son:

- Modificación del sistema de recordatorios

Varios usuarios comentan que podría ser interesante cambiar el esquema de ventanas modales en la esquina inferior derecha por otro en el que se muestre algún icono que cuando tenga algún recordatorio abierto permita hacer clic y desplegar una lista con los avisos. Esta idea es finalmente desechada ya que se podría parecer mucho a un menú de notificaciones y el objetivo es que el usuario lo vea.

- Modificación del aspecto de la ventana modal de los recordatorios en el inicio

El primer boceto de la ventana modal utilizaba el mismo esquema de color pero aplicado al fondo (*Anexo III - Página 47*). Este esquema no gusta a los usuarios y me sugieren sustituirlo por otro en el que el color sea menos visible, por ejemplo, en una esquina.

- Botones de filtrado en horario general

Varios usuarios no parecen entender de primeras que los elementos de la parte superior de la sección son botones, pero una vez descubierto comentan que se entiende fácilmente el funcionamiento.

- Apartado con los recordatorios

Me comentan que se puede simplificar la disposición de los datos de los recordatorios podría simplificarse (*Anexo III – Página 47*). Además, cuando les pregunto acerca de dónde pondrían este apartado, la mayoría creen que la pantalla de inicio sería un buen lugar.

8. Funcionalidades de la aplicación

La aplicación se podría dividir en dos grupos de usuarios, los usuarios y los administradores. Hay que tener en cuenta que los administradores tienen acceso también a todo lo de los usuarios.

8.1. Usuario

8.1.1. Inicio de sesión

Se proporciona el siguiente formulario de inicio de sesión mediante correo y contraseña:



El formulario de inicio de sesión de la Universidad de Salamanca presenta el escudo de la institución y el nombre "UNIVERSIDAD DE SALAMANCA" en rojo. Incluye dos campos de entrada: "CORREO ELECTRÓNICO" y "CONTRASEÑA", cada uno con una línea de subrayado. Debajo de estos campos hay un enlace que dice "¿HAS OLVIDADO LA CONTRASEÑA?". En la parte inferior del formulario, hay un botón rojo con el texto "ACCEDER" en blanco.

Ilustración 35. Inicio de sesión

8.1.2. Inicio

Nada más iniciar sesión el usuario visualizará la página de inicio donde podrá consultar los apartados de "Horario", "Docencia" y "Recordatorios".

Tras dos segundos, el sistema muestra las ventanas modales con los avisos de los recordatorios, estas ventanas se pueden cerrar.

UNIVERSIDAD DE SALAMANCA

Personal Horario general Horario aula

Horario Docencia Recordatorios

Primer cuatrimestre

	Lunes	Martes	Miércoles	Jueves	Viernes
10-11	Diseño de Bases de Datos Aula C-1 TA	Programación I Aula Magna II TA	Diseño de Bases de Datos Aula C-1 TA	Programación III Aula C-1 TA	
12-13		Programación I Aula Magna II TB		Programación I Aula Magna II TB	
16-17		Programación I Aula Informática 3 PB1			
17-18		Programación I Aula Informática 3 PB1			

Segundo cuatrimestre

	Lunes	Martes	Miércoles	Jueves	Viernes
12-13					Análisis Matemático III Aula D-01 TB
13-14					Análisis Matemático III Aula D-01 TB
18-19		Programación II Aula Informática 2 PA4			
19-20		Programación II Aula Informática 2 PA4			

Exámenes

Asignatura	Tipo	Fecha	Hora	Aula
Diseño de Bases de Datos	Primer Semestre	2023-01-13	10:00	Aula A-1
Programación I	Primer Semestre	2023-12-12	16:00	Aula Magna II

- Prueba de recordatorio con un título largo. El máximo es 70 caracteres
- Prueba de recordatorio con prioridad media.
- Prueba con prioridad baja.

Ilustración 36. Inicio

i Prueba de recordatorio con un título largo. El máximo es 70 caracteres ×

i Prueba de recordatorio con prioridad media. ×

i Prueba con prioridad baja. ×

Ilustración 37. Modales de los recordatorios

En la pestaña “Docencia”, se muestran las asignaturas que imparte el docente por titulación y algunos datos de las mismas. El apartado Imparte indica los grupos que imparte y el número de alumnos.

Nombre	Curso	Cuatrimestre	Carácter	CT/CP	Imparte
Programación I (101102)	1	Primer Semestre	Básica	3/3	PB1(28) / TB(84) / TA(90)
Programación II (101107)	1	Segundo Semestre	Básica	0/6	PA4(27)
Diseño de Bases de Datos (101110)	2	Primer Semestre	Obligatoria	0/6	TA(B6)
Programación III (101112)	2	Primer Semestre	Obligatoria	0/6	TA(78)

Nombre	Curso	Cuatrimestre	Carácter	CT/CP	Imparte
Análisis Matemático III (100212)	2	Segundo Semestre	Obligatoria	4/2	TB(30)

Ilustración 38. Docencia

En la pestaña “Recordatorios”, se muestran los recordatorios que hay actualmente almacenados en la base de datos y se permite añadir nuevos recordatorios o eliminarlos.

Añadir recordatorio

- Prueba de recordatorio con un título largo. El máximo es 70 caracteres
 - Día 2023-07-01 (16:42 - 23:57)
 - Prioridad alta
- Prueba de recordatorio con prioridad media.
 - 2023-06-29 (16:47) - 2023-07-02 (16:51)
 - Prioridad media
- Prueba con prioridad baja.
 - 2023-06-30 (16:52) - 2023-07-04 (16:52)
 - Prioridad baja

Ilustración 39. Recordatorios

Añadir recordatorio

Título

Fecha inicio Fecha fin

Prioridad

Calendar: julio de 2023. Selected date: 1.

Ilustración 40. Modal añadir recordatorio

8.1.3. Personal

En el apartado del personal una vez seleccionado el centro se muestran los datos de los docentes:



Ilustración 41. Personal seleccionar centro



Ilustración 42. Personal

En cuanto a las opciones de filtrado y ordenación, se tiene la opción de seleccionar o el departamento o la titulación, buscar por nombre o ordenar los docentes según su Nombre, Área de conocimiento y Departamento:

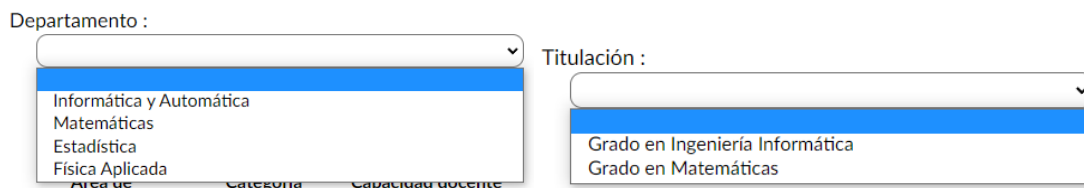


Ilustración 43. Opciones de filtrado Personal

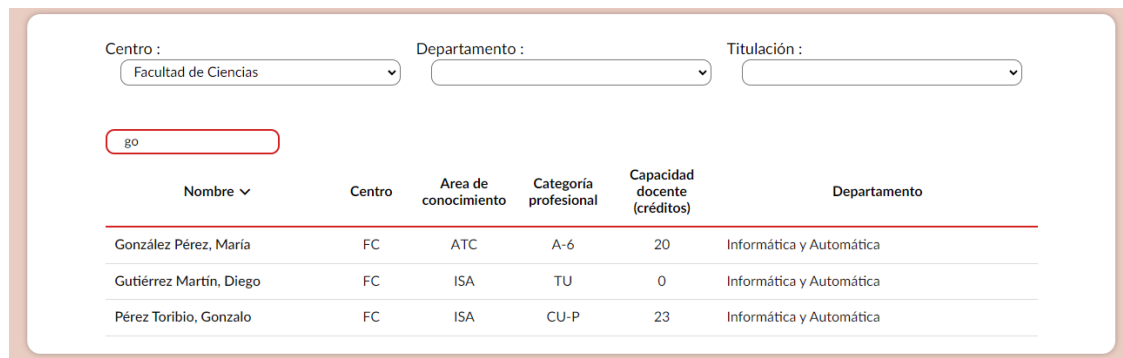


Ilustración 44. Opciones de filtrado Personal - Nombre

Si se hace clic en el nombre del docente, se mostrará la una página con los datos del docente y tres pestañas, Perfil, Horario y Docencia:

UNIVERSIDAD SALAMANCA Personal Horario general Horario aula

Perfil Horario Docencia

Datos personales

Nombre: María González Pérez
 Correo: maria@ufe.es
 Categoría: Asociado a tiempo parcial (6 horas)
 Área de conocimiento: Arquitectura y Tecnología de los Computadores
 Capacidad docente: 20

Ilustración 45. Perfil de otro docente

Perfil **Horario** Docencia

Primer cuatrimestre

	Lunes	Martes	Miércoles	Jueves	Viernes
10-11	Computadores I Aula Magna II TA		Computadores I Aula Magna II TA		
11-12		Sistemas Operativos I Aula C-1 TA		Sistemas Operativos I Aula C-1 TA	
12-13	Computadores I Aula Magna II TB		Computadores I Aula Magna II TB		

Segundo cuatrimestre

	Lunes	Martes	Miércoles	Jueves	Viernes

Exámenes

Asignatura	Tipo	Fecha	Hora	Aula

Ilustración 46. Horario de otro docente

Perfil Horario **Docencia**

Grado en Ingeniería Informática

Nombre	Curso	Cuatrimestre	Carácter	CT/CP	Imparte
Computadores I (101104)	1	Primer Semestre	Básica	0/6	TB(104) / TA(90)
Sistemas Operativos I (101114)	2	Primer Semestre	Obligatoria	0/6	TA(79)

Ilustración 47. Docencia de otro docente

8.1.4. Horario general

En el apartado “Horario general” se muestra el horario de todos los cursos de una titulación:

Titulacion :

- Grado en Ingeniería Informática
- Grado en Matemáticas
- Grado en Física

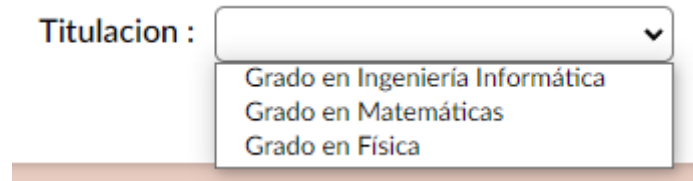


Ilustración 48. Horario general seleccionar titulación

Titulación: Grado en Ingeniería Informática

Primer semestre Segundo semestre

Descargar

PRIMER CURSO - 1º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
8-9	Álgebra Lineal y Geometría TA Aula Magna II Inicio 8:30	Álgebra Lineal y Geometría TA Aula Magna II			
9-10	Álgebra Lineal y Geometría TA Aula Magna II Inicio 8:30	Álgebra Lineal y Geometría TA Aula Magna II	Estadística TA Aula Magna II	Estadística TA Aula Magna II	
10-11	Computadores I TA Aula Magna II	Programación I TA Aula Magna II	Computadores I TA Aula Magna II	Programación I TA Aula Magna II	
11-12	Fundamentos Físicos TB Aula Magna II	Fundamentos Físicos TB Aula Magna II	Fundamentos Físicos TA Aula Magna II	Fundamentos Físicos TB Aula Magna II	
12-13	Computadores I TB Aula Magna II	Programación I TB Aula Magna II	Computadores I TB Aula Magna II	Programación I TB Aula Magna II	
13-14	Álgebra Lineal y Geometría TB Aula Magna II	Estadística TB Aula Magna II	Álgebra Lineal y Geometría TB Aula Magna II 14:30	Estadística TB Aula Magna II	
16-17		Programación I PB1 Aula Informática 3			
17-18		Programación I PB1 Aula Informática 3			

PRIMER CURSO - 2º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
18-19		Programación II PA4 Aula Informática 2			
19-20		Programación II PA4 Aula Informática 2			

SEGUNDO CURSO - 1º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
9-10	Señales y Sistemas TA Aula C-1	Señales y Sistemas TA Aula C-1	Señales y Sistemas TA Aula C-1		
10-11	Diseño de Bases de Datos TA Aula C-1	Programación III TA Aula C-1	Diseño de Bases de Datos TA Aula C-1	Programación III TA Aula C-1	
11-12	Estructura de Datos y Algoritmos I TA Aula C-1	Sistemas Operativos I TA Aula C-1	Estructura de Datos y Algoritmos I TA Aula C-1	Sistemas Operativos I TA Aula C-1	

SEGUNDO CURSO - 2º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes

TERCER CURSO - 1º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
9-10	Control de Procesos TA Aula C-2	Ingeniería del Software II TA Aula C-3	Ingeniería del Software II TA Aula Magna I	Ingeniería del Software II TA Aula C-3	
10-11	Programación Avanzada TB Aula C-2	Interfaces Gráficas de Usuario TA Aula C-3	Programación Avanzada TA Aula Magna I	Interfaces Gráficas de Usuario TA Aula C-3	
11-12	Redes de Computadores I TA Aula C-2		Redes de Computadores I TA Aula Magna I		
12-13			Modelado y Simulación TA Aula C-2		
13-14			Modelado y Simulación TA Aula C-2	Control de Procesos TA Aula C-2	

TERCER CURSO - 2º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes

CUARTO CURSO - 1º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
9-10		Animación Digital PA2 Laboratorio de Informática ----- Fundamentos de Sistemas Inteligentes TA Aula Magna I	Fundamentos de Sistemas Inteligentes TA Aula C-3	Animación Digital PA2 Laboratorio de Informática ----- Fundamentos de Sistemas Inteligentes TA Aula Magna I	Animación Digital PA1 Laboratorio de Informática
10-11		Seguridad en Sistemas Informáticos TA Aula Magna I		Seguridad en Sistemas Informáticos TA Aula Magna I	Animación Digital PA1 Laboratorio de Informática
11-12	Gestión de Proyectos TA Aula Magna I	Gestión de Proyectos TA Aula Magna I		Gestión de Proyectos TA Aula Magna I	
12-13	Aspectos Legales y Profesionales de la Informática TA Aula Magna I	Fundamentos de Sistemas Inteligentes TB Aula Magna I	Animación Digital PA1 Aula SUN	Fundamentos de Sistemas Inteligentes TA Aula Magna I	Animación Digital PA2 Aula SUN
13-14	Aspectos Legales y Profesionales de la Informática TA Aula Magna I	Fundamentos de Sistemas Inteligentes TB Aula Magna I	Animación Digital PA1 Aula SUN		Animación Digital PA2 Aula SUN

CUARTO CURSO - 2º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes

Ilustración 49. Horario general

Se puede indicar a la web que muestre solo los horarios de determinados cursos o semestres:

The screenshot shows the 'Horario general' page for the 'Grado en Ingeniería Informática' at the University of Salamanca. The interface includes navigation tabs for 'Primer semestre' and 'Segundo semestre', and a 'Descargar' button. Two tables are displayed:

PRIMER CURSO - 1º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
8-9	Álgebra Lineal y Geometría TA Aula Magna II Inicio 8:30	Álgebra Lineal y Geometría TA Aula Magna II			
9-10	Álgebra Lineal y Geometría TA Aula Magna II Inicio 8:30	Álgebra Lineal y Geometría TA Aula Magna II	Estadística TA Aula Magna II	Estadística TA Aula Magna II	
10-11	Computadores I TA Aula Magna II	Programación I TA Aula Magna II	Computadores I TA Aula Magna II	Programación I TA Aula Magna II	
11-12	Fundamentos Físicos TA Aula Magna II	Fundamentos Físicos TB Aula Magna II	Fundamentos Físicos TA Aula Magna II	Fundamentos Físicos TB Aula Magna II	
12-13	Computadores I TA Aula Magna II	Programación I TB Aula Magna II	Computadores I TB Aula Magna II	Programación I TB Aula Magna II	
13-14	Álgebra Lineal y Geometría TB Aula Magna II	Estadística TB Aula Magna II	Álgebra Lineal y Geometría TB Aula Magna II 14:30	Estadística TB Aula Magna II	
16-17		Programación I PB1 Aula Informática 3			
17-18		Programación I PB1 Aula Informática 3			

TERCER CURSO - 1º Semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
9-10	Control de Procesos TA Aula C-2	Ingeniería del Software II TA Aula C-3	Ingeniería del Software II TA Aula Magna I	Ingeniería del Software II TA Aula C-3	
10-11	Programación Avanzada TA Aula C-2	Interfaces Gráficas de Usuario TA Aula C-3	Programación Avanzada TA Aula Magna I	Interfaces Gráficas de Usuario TA Aula C-3	
11-12	Redes de Computadores I TA Aula C-2		Redes de Computadores I TA Aula Magna I		
12-13			Modelado y Simulación TA Aula C-2		
13-14			Modelado y Simulación TA Aula C-2	Control de Procesos TA Aula C-2	

Ilustración 50. Horario general filtrado

8.1.5. Horario aula

En el apartado de “Horario aula” se muestra el horario de una determinada aula, además de los exámenes y reservas que tengan lugar en ella:

The screenshot shows the 'Horario aula' interface. It features two dropdown menus for selecting a classroom:

- The first dropdown menu is labeled 'Aula:' and has a list of options: Aula A-1 (A1600), Aula A-3 (A3600), Aula B-2 (B2600), Aula Informática 3 (F2202), Aula 0-01 (M1300), Aula Magna II (F2500), Aula Magna I (F1800), aula inform, and Aula Informática 2 (F2201).
- The second dropdown menu is labeled 'Aula:' and has the text 'infor' entered. Its list of options includes: Aula Informática 3 (F2202), Aula Informática 0 (T1311), Aula Informática 1 (F2200), Aula Informática 2 (F2201), and Laboratorio de Informática (LABINF).

Ilustración 51. Horario aula seleccionar aula

UNIVERSIDAD DE SALAMANCA

Personal Horario general Horario aula

Aula: Aula Magna II (F2500)

Primer semestre

	Lunes	Martes	Miércoles	Jueves	Viernes
8-9	Álgebra Lineal y Geometría TA Inicio 8:30	Álgebra Lineal y Geometría TA			
9-10	Álgebra Lineal y Geometría TA Inicio 8:30	Álgebra Lineal y Geometría TA	Estadística TA	Estadística TA	
10-11	Computadores I TA	Programación I TA	Computadores I TA	Programación I TA	
11-12	Fundamentos Físicos TA	Fundamentos Físicos TB	Fundamentos Físicos TA	Fundamentos Físicos TB	
12-13	Computadores I TB	Programación I TB	Computadores I TB	Programación I TB	
13-14	Álgebra Lineal y Geometría TB	Estadística TB	Álgebra Lineal y Geometría TB 14:30	Estadística TB	

Segundo semestre

	Lunes	Martes	Miércoles	Jueves	Viernes

Exámenes

Asignatura	Tipo	Fecha	Hora	Aula
Programación I	Primer Semestre	2023-12-12	16:00	Aula Magna II

Reservas

Asignatura	Fecha	Hora inicio	Hora fin
Interfaces Gráficas de Usuario	2023-06-28	14:00	16:00
Álgebra Lineal y Geometría	2023-10-18	16:00	19:00

Ilustración 52. Horario aula

8.1.6. Perfil

En el apartado “Perfil” se muestran los datos del usuario:

UNIVERSIDAD DE SALAMANCA

Personal Horario general Horario aula Informes

Datos personales

Nombre	Diego Gutiérrez Martín
Correo	admin@fe.es
Categoría	Titular de Universidad
Área de conocimiento	Ingeniería de Sistemas y Automática
Capacidad docente	0

Ilustración 53. Perfil

8.2. Administrador

En la vista del administrador se activará la opción de modificar horarios y aparecen 4 secciones nuevas, “Informes”, “Cargar datos”, “Eliminar datos” y “Incompatibilidades”.



Ilustración 54. Nuevas secciones Administrador

8.2.1. Modificar horario

Para modificar un horario simplemente habrá que ir a la tabla con el horario, ya sea el del propio usuario o el de cualquier otro y hacer clic en la clase que se quiere modificar. Esta acción abrirá una ventana modal:

Primer cuatrimestre

	Lunes	Martes	Miércoles	Jueves	Viernes
10-11	Diseño de Bases de Datos Aula C-1 TA	Programación I Aula Magna II TA	Diseño de Bases de Datos Aula C-1 TA	Programación III Aula C-1 TA	
12-13		Programación I Aula Magna II TB		Programación I Aula Magna II TB	
16-17		Programación I Aula Informática 3 PB1			
17-18		Programación I Aula Informática 3 PB1			

Ilustración 55. Modificar horario Inicio

The modal window contains the following form fields:

- Asignatura: Programación I (101102)
- Grupo: TA, N° alumnos: 90
- Aula: Aula Magna II (F2500)
- Capacidad: 270
- Día: Martes
- Hora inicio: 10:00, Hora fin: 11:00
- Comentario: (empty)

Buttons: Cancelar, Guardar

Aula	Capacidad	Edificio	Sector	Planta
Aula A-3	160	Ciencias	A	2
Aula A-1	160	Ciencias	A	P.B.
Aula C-1	160	Ciencias	C	P.B.
Aula C-3	160	Ciencias	C	2
Aula Magna I	240	Ciencias	F	P.B.

Ilustración 56. Modificar horario

En esta ventana se podrá modificar todos los valores que se muestran a la izquierda. La aplicación actualiza automáticamente las aulas permitidas según el número de alumnos del

grupo. En cuanto a los cambios de hora, solo permite elegir las horas libres de la asignatura en el día seleccionado:

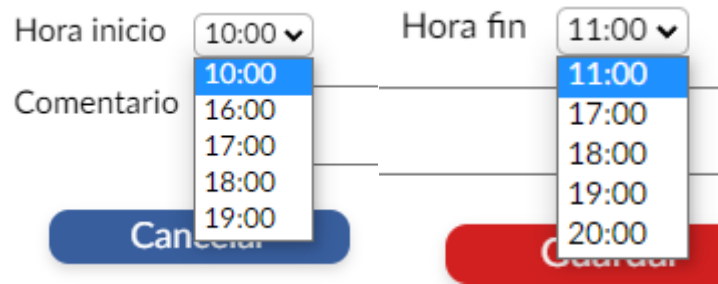


Ilustración 57. Opciones Hora inicio y Hora fin

Esto se puede comprobar visualizando el horario del aula seleccionada. Para ello hay que hacer clic en el botón con el reloj que está al lado del aula:

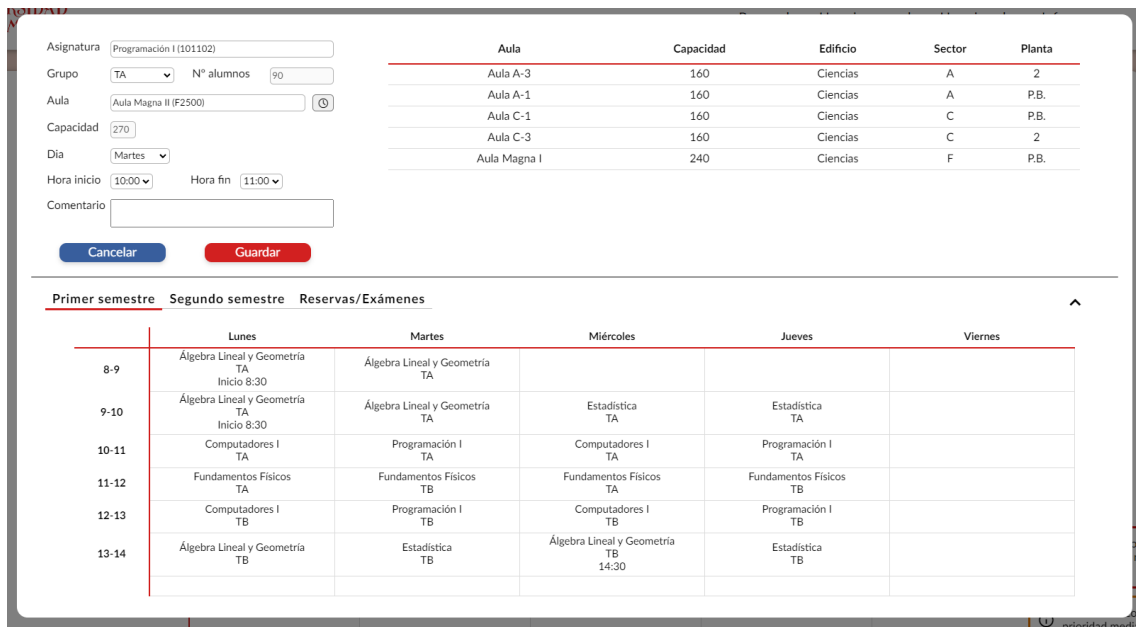


Ilustración 58. Modificar horario con el horario del aula

Como se puede observar en la imagen anterior, la aplicación indica como libre el aula de 10 a 11 en los selectores de hora de inicio y fin, debido a que se trata de la misma clase que se está queriendo modificar, mientras que el resto de horas que no muestra, efectivamente están ocupadas.

También se puede ver cómo el nuevo apartado inferior, permite ver el horario del primer y segundo semestre y las reservas y exámenes:

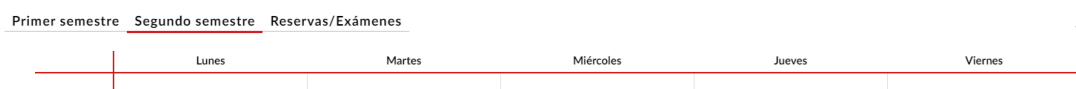


Ilustración 59. Modificar horario con el horario del aula Segundo semestre

Reservas				Exámenes			
Asignatura	Fecha	Hora inicio	Hora fin	Asignatura	Tipo	Fecha	Hora
Álgebra Lineal y Geometría	2023-10-18	16:00	19:00	Programación I	Primer Semestre	2023-12-12	16:00
Interfaces Gráficas de Usuario	2023-06-28	14:00	16:00				

Ilustración 60. Modificar horario con el horario del aula Reservas/Exámenes

También existe la opción de ver el horario, reservas y exámenes de las aulas permitidas haciendo clic en el nombre del aula.

8.2.2. Informes

En el apartado de “Informes”, se generan los informes de ocupación de un aula, para ello hay que seleccionar el aula y la semana (simplemente hay que seleccionar cualquier día de la semana y la aplicación obtendrá la semana):

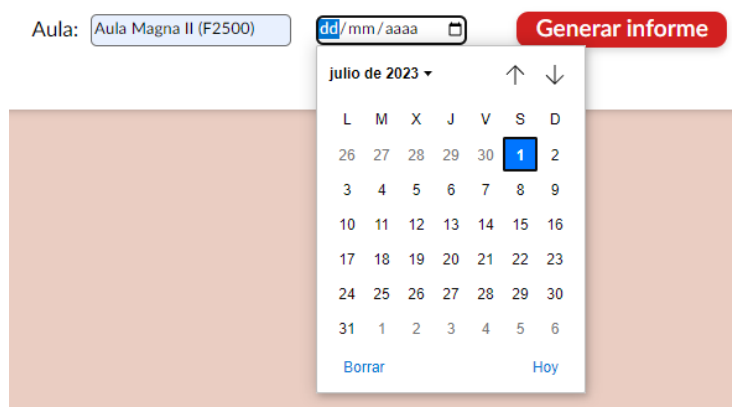


Ilustración 61. Informes seleccionar semana

Una vez generado el informe, en el primer gráfico se puede elegir si mostrar las horas que está ocupada al día y seleccionar entre estas tres opciones:

- Número total de alumnos
- % de alumnos sobre la capacidad
- Media de alumnos por hora

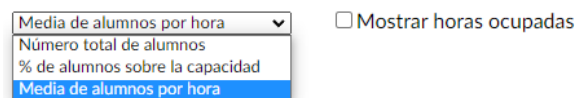


Ilustración 62. Opciones primer gráfico

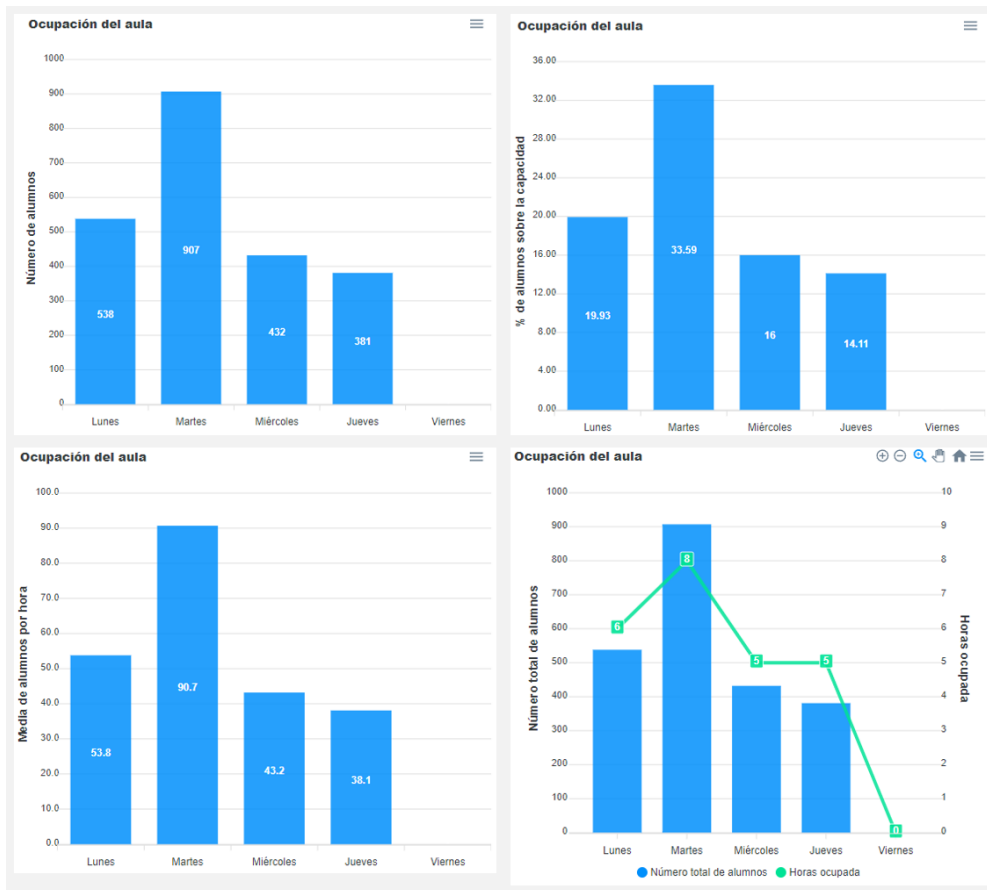


Ilustración 63. Diferentes opciones de visualización del primer gráfico



Ilustración 64. Informes

Todos los gráficos tienen la opción de ser descargados en SVG, PNG o CSV.

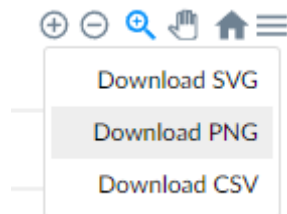


Ilustración 65. Opciones de descarga de los gráficos

8.2.3. Cargar datos

En el apartado de “Cargar datos”, se pueden cargar CSVs a la base de datos. En un principio, la página estará vacía, por lo que hay que añadir un apartado de carga en el que habrá que seleccionar la tabla y el archivo a cargar. Además, una vez cargado, muestra los datos del fichero en una tabla cuyo esquema de visualización se puede modificar.

También dispone de un botón que permite mostrar el esquema a seguir con un ejemplo.

The screenshot shows the 'Cargar datos' interface with three data loading windows. The first window is titled 'Tabla: Examen' and shows a table with columns: FK_Asignatura, FK_Aula, Fecha, Hora_inicio, Hora_fin, and Num_alumnos. The second window is titled 'Tabla: Aulas permitidas por asignatura' and shows a table with columns: FK_Asignatura, FK_Aula, and Curso_academico. The third window is titled 'Tabla: Grupo' and shows a table with columns: Nombre_Grupo, FK_Asignatura, Num_alumnos, FK_Tipo_Grupo, and Curso_academico.

Ilustración 66. Cargar datos

The screenshot shows the 'Cargar datos' interface with a data loading window. The table is collapsed, showing only the first few rows of data. The columns are: FK_Asignatura, FK_Aula, Fecha, Hora_inicio, Hora_fin, and Num_alumnos.

Ilustración 67. Cargar datos tabla contraída

The screenshot shows the 'Cargar datos' interface with a data loading window. The table is fully expanded, showing all rows of data. The columns are: FK_Asignatura, FK_Aula, Fecha, Hora_inicio, Hora_fin, and Num_alumnos.

Ilustración 68. Cargar datos tabla desplegada

Una vez se hayan cargado todos los CSVs, simplemente habría que hacer clic en subir todo.

8.2.4. Eliminar datos

En el apartado de eliminación de datos, hay cuatro tablas disponibles, dependiendo de la tabla que se seleccione se piden unos datos u otros para la eliminación de la fila:

The screenshot shows the 'Eliminar datos' interface for 'Aulas permitidas por asignatura'. On the left, there is a vertical menu with options: 'Aulas permitidas por asignatura', 'Examen', 'Horario', 'Grupo', and 'Personal por grupo'. The 'Aulas permitidas por asignatura' option is selected. The main area contains three input fields: 'Asignatura' (Nombre o ID), 'Aula' (Nombre o ID), and 'Curso académico' (2021-22). A red 'Eliminar' button is positioned below these fields.

Ilustración 69. Eliminar datos - Aulas permitidas por asignatura

The screenshot shows the 'Eliminar datos' interface for 'Examen'. The left menu has 'Examen' selected. The main area contains four input fields: 'Asignatura' (Nombre o ID), 'Fecha' (dd/mm/aaaa), 'Aula' (Nombre o ID), and 'Hora inicio' (8:00). A red 'Eliminar' button is positioned below these fields.

Ilustración 70. Eliminar datos - Examen

The screenshot shows the 'Eliminar datos' interface for 'Horario'. The left menu has 'Horario' selected. The main area contains four input fields: 'Día' (Lunes), 'Hora inicio' (8:00), 'Aula' (Nombre o ID), and 'Curso académico' (2022-23). A red 'Eliminar' button is positioned below these fields.

Ilustración 71. Eliminar datos - Horario

En el caso de la eliminación de un grupo, primero hay que seleccionar la asignatura y el curso, una vez seleccionados se mostrará el apartado con los grupos de esa asignatura en el curso seleccionado.

The screenshot shows the 'Eliminar datos' interface for 'Grupo'. The left menu has 'Grupo' selected. The main area contains three input fields: 'Asignatura' (Señales y Sistemas (101113)), 'Curso académico' (2022-23), and 'Grupo' (TB). A red 'Eliminar' button is positioned below these fields.

Ilustración 72. Eliminar datos - Grupo

En el caso de la eliminación de un docente de un grupo, hay que seguir el mismo proceso que en el anterior y al seleccionar el grupo aparecerán los docentes que lo imparten, por lo que simplemente habrá que seleccionar el docente.



Ilustración 73. Eliminar datos - Personal por grupo

8.2.5. Incompatibilidades

En el apartado de “Incompatibilidades”, al hacer clic en el botón se ejecuta un algoritmo que detecta asignaciones simultáneas de clases en la misma aula o a un mismo docente y muestra sus datos en pantalla.

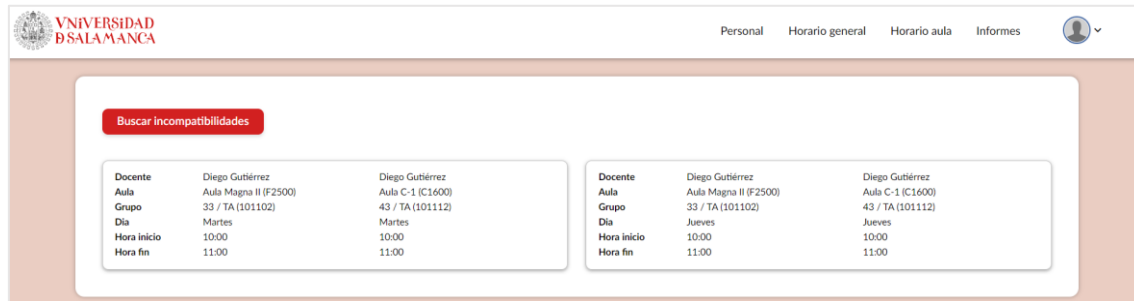


Ilustración 74. Buscar incompatibilidades

9. Conclusiones y líneas de trabajo futuro

9.1. Conclusiones

Se puede concluir que se han conseguido los siguientes objetivos:

- Desarrollo de una aplicación web, lo que permite que los usuarios solo requieran de un navegador web para acceder.
- Implementación de un sistema de autenticación que permite mantener la seguridad de la aplicación, tanto en el acceso a datos como en el acceso a zonas restringidas. Además, permite mantener sesiones.
- Sistema de visualización de los horarios de todos los docentes registrados en la plataforma, además de sus exámenes y asignaturas impartidas.
- Posibilidad de descargar los diferentes horarios en PDF.
- Creación de una sección desde la que se puede consultar toda la información de los docentes rápidamente, además de disponer de diferentes filtros de búsqueda.
- Pantalla para la visualización de los horarios completos de las diferentes titulaciones con la posibilidad de aplicar filtros.
- Pantalla para la visualización de los horarios de las aulas y los exámenes y las reservas que tenga asignada el aula.
- Generación de informes de ocupación de un aula en una determinada semana mediante gráficas, además, permite la modificación de algunos valores mostrados y su descarga en diferentes formatos como SVG, PNG y CSV.
- Acceso a un apartado con los datos del docente (el perfil del usuario) y posibilidad de consultar los datos de los perfiles de todos los usuarios de la aplicación.
- Posibilidad de carga y eliminación de datos de múltiples tablas a la base de datos mediante una interfaz sencilla.
- Algoritmo de búsqueda de incompatibilidades que detecta asignaciones simultáneas de clases en la misma aula o a un mismo docente.

El desarrollo de este proyecto me ha permitido iniciarme en el desarrollo web y adquirir conocimientos sobre su desarrollo y las múltiples tecnologías que se pueden utilizar. En este caso, he aprendido a desarrollar una web mediante Vue.js, que ha su vez ha conllevado la adquisición de conocimiento en JavaScript y en otros aspectos como la reactividad y los métodos asíncronos, además de mejorar mis habilidades con HTML y CSS.

En relación al backend, a través de una exhaustiva investigación para encontrar la forma más sencilla de implementarlo, he adquirido conocimientos sobre las diversas opciones disponibles. Específicamente, he desarrollado sólidas competencias en el uso de la plataforma Supabase, lo que a su vez me ha permitido aprender a utilizar PostgreSQL.

Otro aspecto relevante, es mi familiarización con el uso de metodologías ágiles para planificación y gestión del proyecto como son Scrum o los tableros Kanban.

9.2. Líneas de trabajo futuro

Un aspecto interesante sería la implementación de un sistema parecido al de SIRA para la reserva de aulas con una sección para los administradores en la que puedan gestionar las peticiones y en caso de que la petición sea aceptada, se añada automáticamente a la base de datos.

Apertura de la aplicación a los alumnos con una interfaz de inicio algo diferente.

Creación de una sección con un calendario del docente que a su vez esté enlazado con Google para la gestión de reuniones e incluso, la sincronización de los recordatorios locales de la aplicación con los remotos de Google.

Gestión total de la base de datos desde la aplicación web, es decir, que no sea necesario que los administradores puedan cargar, modificar y eliminar datos o permisos fácilmente y sin tener que ser conocedores de conceptos avanzados de bases de datos.

La descarga de los horarios en formato PDF es bastante mejorable. La calidad de descarga no es muy alta y no tiene ninguna opción de personalización. Además, se podría implementar un sistema de muestreo de las horas más flexible y completo:

- Si hay dos horas de clase, juntar ambos recuadros
- Permitir clases que no comiencen o finalicen a en punto
- Utilización de diferentes colores para la diferenciación de los grupos, ya sea en el color del fondo, en el de las letras o en ambos.

Otro aspecto que podría ser útil es la gestión de los cambios de grupo, permitiendo a los alumnos solicitar cambios de grupo de teoría o prácticas en los plazos pertinentes desde la web. Para ello podría crearse un apartado para el alumno en el que adjuntar los documentos necesarios y otro apartado para el administrador en el que se pueda visualizar rápidamente los documentos y datos de los solicitantes, con la correspondiente actualización automática de la base de datos.

Convendría que la página se adaptara al formato móvil, ya que ahora mismo no hay cambios en la disposición cuando se utilizan anchos muy pequeños. También podría ser interesante la creación de una aplicación móvil.

Implementación de un sistema para cargar en la base de datos las horas que un docente puede impartir clase (solo por la mañana, solo por la tarde...) y un algoritmo para que, en la sección de búsqueda de incompatibilidades, detecte si hay algún docente con horas asignadas fuera de sus horas disponibles o que imparte más horas de las que puede.

Posibilidad de cargar fotos de perfil de los usuarios a la base de datos, es este caso Supabase Storage.

Mejora de la sección de los informes mediante la generación de más gráficas de otros tipos, tablas e incluso informes en PDF. También podría incluirse la creación de informes a nivel de sector o centro, titulación, asignatura, etc.

División de la sección de personal en diferentes páginas con un máximo de docentes para así evitar descargar los datos de todos los docentes y que se genere una tabla con cientos de filas.

Podría ser interesante incluir una parte con documentos de interés tanto para los administradores como para el personal, o incluso, los alumnos.

Permitir a los docentes escoger si quieren que su horario sea público.

Limitación de la capacidad de modificación de datos de los administradores dependiendo del rango del que dispongan (modificación solo de datos de una titulación, departamento, centro, etc.)

10.Referencias

- [1] <<Diaweb>> <https://diaweb.usal.es/diaweb/index.jsp>
- [2] <<SIRA>> <https://sicpd.usal.es/display/LAZ/SIRA%3A+Sistema+Integral+de+Reserva+de+Aulas>
- [3] Miguel Ángel Álvarez, <<Qué es el DOM>> <https://desarrolloweb.com/articulos/que-es-el-dom.html>
- [4] Rockcontent, <<Qué son los stakeholders, qué tipos existen y de qué manera impactan a una empresa>> <https://rockcontent.com/es/blog/que-es-un-stakeholder/>
- [5] Edix, <<Framework>><https://www.edix.com/es/instituto/framework/>
- [6] <<Vue CLI>> <https://cli.vuejs.org/guide/>
- [7] <<Visual Studio Code>> <https://code.visualstudio.com>
- [8] <<Adobe XD>> <https://helpx.adobe.com/es/xd/help/adobe-xd-overview.html>
- [9] <<Visual paradigm>> <https://www.visual-paradigm.com>
- [10] <<Lucidchart>>, <https://www.lucidchart.com/pages/>
- [11] <<Vue>><https://vuejs.org>
- [12] <<Vue Router>> <https://router.vuejs.org>
- [13] <<Pinia>> <https://pinia.vuejs.org>
- [14] <<Supabase>> <https://supabase.com>
- [15] <<PostgreSQL>><https://www.postgresql.org/>
- [16] <<Firebase>> <https://firebase.google.com/?hl=es>
- [17] <<Supabase Database>> <https://supabase.com/docs/guides/database>
- [18] <<Supabase Realtime>><https://supabase.com/docs/guides/realtime>
- [19] <<Supabase Auth>> <https://supabase.com/docs/guides/auth>
- [20] <<Supabase Storage>><https://supabase.com/docs/guides/storage>
- [21] <<Supabase Edge Functions>><https://supabase.com/docs/guides/functions>
- [22] <<Deno>> <https://deno.com/>
- [23] <<Miro>> <https://miro.com/index/>
- [24] <<Html2pdf>> <https://github.com/eKoopmans/html2pdf.js>
- [25] <<Apexcharts>> <https://apexcharts.com>

- [26] <<React>> <https://es.react.dev>
- [27] <<Angular>> <https://angular.io>
- [28] <<Netlify>> <https://www.netlify.com>
- [29] Atlassian <<Qué es scrum y cómo empezar>>
<https://www.atlassian.com/es/agile/scrum#:~:text=¿Qué%20es%20scrum%3F,de%20valores%2C%20principios%20y%20prácticas.>
- [30] A. Durán Toro y B. Bernárdez Jiménez, «Metodología para la Elicitación de Requisitos de Sistemas Software».
- [31] <<Historias de usuario con ejemplos y plantilla>>
<https://www.atlassian.com/es/agile/project-management/user-stories>
- [32] <<Vue guide>> <https://012.vuejs.org/guide/>
- [33] Microsoft, <<Model-View-ViewModel>> <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>
- [34] GeeksForGeeks, <<How a View-model works in Vue.js>>,
<https://www.geeksforgeeks.org/how-a-view-model-works-in-vue-js/>
- [35] Built In, <<What is MVVM architecture?>>,
<https://builtin.com/software-engineering-perspectives/mvvm-architecture>
- [36] Supabase, <<Initializing>> <https://supabase.com/docs/reference/javascript/initializing>
- [37] Supabase, <<Fetch data>> <https://supabase.com/docs/reference/javascript/select>
- [38] Supabase, <<Auth>> <https://supabase.com/docs/guides/auth>
- [39] Vue Router, <<Navigation guards>> <https://router.vuejs.org/guide/advanced/navigation-guards.html>
- [40] Supabase, <<Row level security>> <https://supabase.com/docs/guides/auth#row-level-security>
- [41] Dave Parsons, <<Intro to Postgres Row-Level Security>> <https://dev.to/davepar/intro-to-postgres-row-level-security-3b3>
- [42] Nick-Mazuk, <<Using authentication policies with views>>
<https://github.com/orgs/supabase/discussions/901>
- [43] Supabase, <<Database Functions>>
<https://supabase.com/docs/guides/database/functions>
- [44] Supabase, <<plv8: JavaScript Language>>
<https://supabase.com/docs/guides/database/extensions/plv8>
- [45] Supabase, <<Cascade deletes>>
<https://supabase.com/docs/guides/database/postgres/cascade-deletes>

[46] Universidad de Salamanca, <<Manual Institucional de Identidad Corporativa>>
<https://identidadcorporativa.usal.es>

[47] <<Robin>> <https://robinpowered.com>