



TRABAJO FINAL DE CICLO

**CICLO FORMATIVO DE GRADO SUPERIOR DE
DESARROLLO DE APLICACIONES MULTIPLATAFORMA
(DAM)**

IES RODANAS

PLATAFORMA DE GUARDIAS

AUTORES:

**Mecles Cris Dumitru, Leonardo Darius Anton,
Javier Santos Rodríguez, Alex Díaz Subías**

CURSO 2023/2024

ÍNDICE

1. INTRODUCCIÓN.....	5
1.1 ESTADO DE LA CUESTIÓN.....	5
1.2 OBJETIVOS.....	6
1.3 METODOLOGÍAS.....	6
2. GESTIÓN DEL PROYECTO.....	7
2.1 TECNOLOGÍAS.....	7
Principales Características:.....	9
2.2 PLANIFICACIÓN.....	10
2.2.1 RECOPIACIÓN DE LA INFORMACIÓN.....	10
2.2.2 DISTRIBUCIÓN DE ROLES.....	11
2.2.3 ASIGNACIÓN DE TAREAS.....	12
3 Requisitos.....	12
3.1 Requisitos funcionales.....	12
3.2 Requisitos no funcionales.....	13
4 ESTUDIO DE LOS ESCENARIOS.....	15
4.1 ESCENARIOS.....	15
4.1.1 Caso de uso: "Iniciar sesión".....	15
4.1.2 Caso de uso: "Acceso al portal de inicio".....	16
4.1.3 Caso de uso: "Acceso al portal de inicio, rol invitado".....	16
4.1.4 Caso de uso: "Crear ausencias".....	16
4.1.5 Caso de uso: "Interactuar con las ausencias".....	17
4.1.6 Caso de uso: "Interactuar con las ausencias, rol jefatura".....	17
4.1.7 Caso de uso: "Gestionar las clases".....	17
4.1.8 Caso de uso: "Gestionar usuarios".....	18
4.1.9 Caso de uso: "Cerrar sesión".....	18
4.2 FLUJO DE LA APLICACIÓN.....	19
5 Arquitectura.....	20
5.1 Funcionalidades.....	21
5.1.1 Users.....	21
5.1.2 Formulario de ausencias.....	22
5.1.3 Formulario de clases.....	22
5.2 Esquema UML.....	23
6.2 Conexión a la base de datos.....	24
6.2.1 Acceso a PHPMyAdmin.....	24
6.2.2 Acceso a PHPMyAdmin con PHP.....	24
7 Diseño.....	26
7.1 Interfaz de identificación.....	26
7.2 Portal de inicio.....	27

7.2.1 Barra de navegación.....	28
7.2.2 Cards.....	28
7.2.3 Hoja de estilo (CSS).....	29
7.3 Pestaña de ausencias.....	30
7.4 Pestaña de clases.....	33
7.5 Pestaña de profesores.....	34
8 Identificación usando google.....	35
8.1 FUNCIONAMIENTO.....	37
9. CONCLUSIÓN.....	39
9.1 ANÁLISIS DE LOS OBJETIVOS.....	39
9.2. LECCIONES APRENDIDAS.....	40
9.3 RECURSOS ÚTILES NO USADOS.....	41
9.4 POSIBLES FUNCIONALIDADES.....	41
10. LICENCIA DE CONTRATO.....	43
11. BIBLIOGRAFÍA.....	44
12 MANUAL DE DESARROLLADOR.....	46
12.1 INTRODUCCIÓN.....	46
12.2 MONTAJE DEL ENTORNO.....	46
12.3 CONSIDERACIONES.....	47
13 MANUAL DE USUARIO.....	47
13.1 REQUISITOS.....	47
13.2 USO DE LA PLATAFORMA WEB.....	48
13.2.1 ROL JEFATURA.....	48
13.2.2 ROL PROFESOR.....	48
14 CONCLUSIÓN.....	49

ÍNDICE FIGURAS

4.1 Esquema UML Casos de uso.....	19
5 Tablas de la base de datos.....	21
5.1.1 Tabla users.....	22
5.1.2 Relaciones del formulario de ausencias.....	22
5.1.3 Relaciones del formulario de clases.....	23
5.2 Esquema UML de la base de datos.....	24
6.2.1 Imagen de la interfaz de phpMyAdmin.....	24
6.2.2 Código de la conexión con la base de datos.....	25
7.1 Interfaz de identificación usando google.....	26
7.1 (1) Interfaz de logueado con google.....	27
7.2 Diseño del portal de inicio.....	27
7.2.1 Diseño de la barra de navegación.....	28
7.2.2 Diseño de la tarjeta del Ies Rodanas.....	28
7.2.3 Código de la hoja de estilos menu.....	29
7.3 Diseño del modal de ausencias.....	30
7.3 (1) Código del formulario de ausencias.....	31
7.3 (2) Código de ajax tabla motivos.....	32
7.3 (3) Código de ajax tabla horas.....	32
7.3 (4) Diseño tabla de ausencias.....	33
7.4 Diseño del formulario de clases.....	33
7.4 (1) Botón de eliminar clases.....	33
7.4 (2) Diseño de la tabla de aulas.....	34
7.5 Diseño formulario de profesores.....	34
7.5 (1) Diseño formulario de profesores.....	35
8 Código composer.....	35
8 (1) Estructura vendor.....	36
8 (2) Carpeta auth.....	36

PARTE I

PRÓLOGO

1. INTRODUCCIÓN

El objetivo de este proyecto es desarrollar una plataforma de gestión para el profesorado desde cero en PHP/MySQL, como parte del proyecto de fin de ciclo formativo de grado superior de Desarrollo de Aplicaciones Multiplataforma (DAM). Este proyecto se ha creado con el objetivo de unificar en una interfaz intuitiva y amigable, todo los problemas de gestión que tiene actualmente el profesorado del centro Ies Rodanas.

1.1 ESTADO DE LA CUESTIÓN

En la actualidad, las plataformas de gestión se han vuelto esenciales para cualquier empresa o sector, ya que facilitan la organización, el almacenamiento y el control de la información. En el ámbito educativo, estas herramientas son igualmente cruciales para mejorar la eficiencia y efectividad de las actividades académicas, permitiendo una mejor gestión de los recursos educativos.

Existen diversas plataformas y aplicaciones móviles diseñadas específicamente para la gestión educativa, entre las cuales destacan Google Classroom, Edmodo y Microsoft Teams. Estas herramientas ofrecen múltiples funcionalidades, tales como la administración de cursos, la comunicación entre alumnos y profesores, y la gestión de tareas. No obstante, muchas de ellas presentan limitaciones significativas. Carecen de herramientas personalizadas de gestión y organización para el profesorado.

En respuesta a estas deficiencias, este proyecto de fin de grado (TFG) se propone a cubrir las brechas existentes desarrollando una plataforma que sea funcional, atractiva y que se adapte de manera óptima a las necesidades específicas de los profesores. El objetivo principal es crear una herramienta que permita una gestión eficiente y personalizada de la labor educativa, facilitando la organización del profesorado y optimizando el uso del tiempo.

1.2 OBJETIVOS

El principal objetivo de este proyecto es crear el motor de lo que será la plataforma web que permita al profesorado gestionar la administración y organización del centro educativo Ies Rodanas. Los objetivos específicos incluyen:

- ♦ Establecer una base de datos para almacenar todos los datos relevantes de manera segura y organizada.

- ♦ Implementar un sistema de autenticación utilizando Google para mejorar la seguridad del acceso.
- ♦ Diseñar un portal inicial que sirva como punto de entrada intuitivo y funcional para todo tipo de usuarios.
- ♦ Subir información a la base de datos mediante un botón desde una interfaz.
- ♦ Desarrollar una interfaz que permita al personal docente y directivo registrar y visualizar sus ausencias de forma clara y rápida.
- ♦ Desarrollar una interfaz donde el equipo directivo pueda administrar los usuarios y roles de estos.
- ♦ Crear una interfaz que muestre un calendario donde se recojan automáticamente las guardias creadas.
- ♦ Desarrollar una interfaz para que el equipo directivo pueda administrar los clases de estudiantes y asignarles puntuaciones según criterios establecidos.
- ♦ Generación de informes a partir de los datos de los formularios.

1.3 METODOLOGÍAS

Se han empleado metodologías ágiles para garantizar la adaptabilidad y la mejora continua basada en el feedback de los usuarios. Esto permite ajustar el desarrollo de la plataforma de manera ágil y eficiente a medida que se van identificando necesidades y sugerencias por parte de los usuarios.

Se ha aplicado la metodología Scrum, que es un marco de trabajo ágil que facilita la gestión y organización del equipo de desarrollo. Con Scrum, se establecen iteraciones cortas de trabajo llamadas sprints, durante las cuales se planifica, desarrolla y entrega un conjunto de funcionalidades. Esto permite una mayor transparencia, colaboración y rapidez en el desarrollo del proyecto.

Se ha aplicado la metodología incremental, la cual consiste en dividir el proceso de desarrollo de software en incrementos más pequeños y manejables. Se comienza con un incremento básico que cubre las funciones esenciales del programa y luego se agregan características adicionales de manera gradual hasta completar el producto final. Este enfoque nos permite adaptar el proyecto desde las etapas iniciales y lanzar una versión que los usuarios pueden utilizar sin tener que esperar al producto final.

2. GESTIÓN DEL PROYECTO

Para conseguir el resultado esperado, es crucial llevar a cabo una efectiva gestión del proyecto, ya que garantiza la eficiente utilización de los recursos, optimizando el tiempo. Además, promueve una comunicación clara y efectiva entre todos los miembros del equipo y las partes interesadas, lo que reduce la posibilidad de malentendidos y asegura un compromiso alineado con los objetivos del proyecto. Asimismo, permite identificar y mitigar riesgos de manera proactiva, minimizando posibles impactos negativos y maximizando oportunidades.

2.1 TECNOLOGÍAS

En este apartado, se detallan las tecnologías y software usados. Los cuales fueron seleccionados considerando las necesidades específicas del centro educativo IES Rodanas y con el propósito de abordar los desafíos que han surgido a lo largo del desarrollo.

PHP también conocido como Personal Home Page o Hypertext Preprocessor.

Fue creado por Rasmus Lerdorf en 1994, con su primera versión pública, PHP/FI, lanzada en 1995. La última versión estable es PHP 8.1.0, lanzada en noviembre de 2021.

Es un lenguaje de programación de código abierto diseñado para el desarrollo web. Principalmente se utiliza para crear sitios web dinámicos e interactivos, ejecutándose en el lado del servidor para generar contenido web antes de enviarlo al navegador del usuario.

Principales características:

- Compatible, veloz y soportado por una amplia gama de servidores web y sistemas operativos.
- Ofrece una gran variedad de funciones predefinidas y cuenta con una amplia comunidad de desarrolladores que contribuyen con bibliotecas y frameworks.

MySQL desarrollado por MySQL AB, que luego fue adquirida por Sun Microsystems y posteriormente por Oracle Corporation. La última versión estable de MySQL es la 8.0, lanzada en abril de 2018.

Se utiliza para almacenar, organizar y gestionar datos, y es especialmente popular para aplicaciones web que requieren bases de datos relacionales.

MySQL utiliza el lenguaje de consulta estructurado (SQL) para realizar consultas y manipular datos en la base de datos.

Principales características:

- Es de código abierto y gratuito.
- Ofrece alta velocidad, escalabilidad y confiabilidad.
- Es compatible con una amplia variedad de plataformas y lenguajes de programación.
- Proporciona mecanismos de recuperación y replicación para asegurar la integridad y disponibilidad de los datos.

Git es uno de los sistemas de control de versiones más populares y poderosos, permite a los desarrolladores trabajar de forma colaborativa en proyectos de software, mantener un historial de cambios, revertir cambios no deseados y coordinar el trabajo entre equipos de desarrollo.

Principales características:

- Permite el control de versiones distribuidas, lo que significa que cada desarrollador tiene una copia completa del repositorio de código.
- Proporciona un historial completo de cambios en el código fuente, lo que facilita la colaboración y la resolución de conflictos.
- Es rápido, eficiente y escalable, lo que lo hace adecuado para proyectos de cualquier tamaño y complejidad.
- La naturaleza distribuida de Git permite a los desarrolladores trabajar offline y sincronizar sus cambios cuando sea necesario.

OAuth es un protocolo de autorización estándar que permite a las aplicaciones acceder a los recursos de un usuario en servicios en línea sin que el usuario comparta sus credenciales.

En lugar de compartir contraseñas, los usuarios autorizan a las aplicaciones para acceder a sus datos mediante OAuth.

Este protocolo es ampliamente utilizado en el desarrollo de aplicaciones web y móviles para integraciones seguras con servicios como Google, Facebook y Twitter.

Principales Características:

- Permite a las aplicaciones acceder de manera segura a los recursos de un usuario en servicios en línea.
- Los usuarios pueden autorizar el acceso a sus datos sin compartir sus credenciales.

XAMPP es un software gratuito y de código abierto que facilita la instalación y configuración de un entorno de desarrollo web local.

Incluye Apache, MySQL, PHP y Perl, proporcionando un servidor web completo que permite a los desarrolladores crear y probar sitios web dinámicos en sus propias computadoras antes de publicarlos en un servidor en vivo.

phpMyAdmin es una aplicación de software de administración de bases de datos MySQL basada en web.

Se utiliza para administrar bases de datos MySQL a través de una interfaz web intuitiva y fácil de usar, permitiendo tareas como la creación, eliminación y modificación de bases de datos y tablas, así como la ejecución de consultas SQL y la gestión de privilegios de usuario.

HTML también conocido como HyperText Markup Language, es el lenguaje estándar utilizado para crear y diseñar páginas web.

Define la estructura y el contenido de una página web utilizando una serie de etiquetas y elementos. HTML se utiliza junto con CSS y JavaScript para crear sitios web interactivos y visualmente atractivos.

CSS también conocido como Cascading Style Sheets, es un lenguaje utilizado para dar estilo y formato a páginas web escritas en HTML. Permite controlar aspectos como el diseño, el color, la tipografía y la disposición de los elementos en una página web.

Separa la estructura del contenido (HTML) del diseño y la presentación (CSS) permite un mayor control y flexibilidad en el diseño de sitios web.

Bootstrap es un framework de código abierto para el desarrollo de sitios web y aplicaciones web responsivas y móviles. Permiten a los desarrolladores crear sitios web visualmente atractivos y funcionales de manera rápida y eficiente.

Bootstrap utiliza HTML, CSS y JavaScript y es ampliamente utilizado en la industria para crear sitios web modernos y adaptativos.

AJAX que significa Asynchronous JavaScript and XML, es una técnica de desarrollo web que permite actualizar partes específicas de una página web sin necesidad de recargar toda la página.

Se basa en el uso de JavaScript para enviar y recibir datos del servidor de forma asíncrona, lo que permite una experiencia de usuario más fluida e interactiva.

jQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones que simplifica la manipulación del DOM (Document Object Model) y la realización de animaciones, manejo de eventos, y desarrollo de aplicaciones web interactivas.

Simplifica la escritura de código JavaScript al proporcionar una API fácil de usar que funciona de manera consistente en diferentes navegadores.

2.2 PLANIFICACIÓN

1. Recopilación de la información junto al equipo directivo y profesorado.
2. Creación del boceto de la interfaz y funcionalidad del proyecto (1 semana).
3. División del grupo en parejas con sus respectivas tareas (1 semana).
4. Desarrollo (2 semanas).
5. Aislar de las parejas entre ellas para conseguir asegurar la creación de las tareas (1 semana).
6. Desarrollo (2 semanas) .
7. Mantenimiento (1 semana).

2.2.1 RECOPIACIÓN DE LA INFORMACIÓN

La recopilación de la información es un punto vital para el correcto desarrollo de un proyecto. Realizar una mala recolección de la información puede causar días,semanas e incluso meses de producción en vano.

Para la recopilación de la información, los desarrolladores responsables del proyecto de Guardias se juntaron junto a un miembro del equipo directivo y un miembro del profesorado con conocimientos avanzados en programación, ambos miembros pertenecientes al centro educativo Ies Rodanas.

Durante la reunión, el miembro del equipo directivo (nuestro cliente) expuso todas las necesidades y problemas que tenían. Tomamos nota y empezamos a plantear soluciones al respecto.

2.2.2 DISTRIBUCIÓN DE ROLES

Para asegurar el correcto desarrollo y funcionamiento del proyecto, hemos conformado un equipo de seis integrantes, cada uno con diferentes roles, responsabilidades y conocimientos específicos.

A lo largo del flujo del proyecto, se pueden destacar los siguientes roles:

- **Dueño del producto:** el centro educativo Ies Rodanas, responsable de todo lo relacionado con el producto.

- **Cliente comercial:** la figura visible del IES Rodanas, quien comunicó los problemas y necesidades del proyecto, actuando como intermediario entre el equipo de desarrollo y el centro educativo.
- **Analistas comerciales:** dos integrantes desempeñaron este rol, que implica comprender los datos y defender las decisiones basadas en ellos. Estos analistas se encargaron de cumplir las necesidades del cliente, definir metas claras y convertir los objetivos en soluciones de software viables. Trabajaron estrechamente con el dueño del producto.
- **Project manager:** además de sus roles como analistas comerciales, estas dos figuras gestionaron el proyecto, asegurando el cumplimiento de los plazos, la coordinación del equipo y la gestión de riesgos y cambios.
- **Desarrolladores de software:** formado por un equipo de cuatro integrantes con el objetivo común de desarrollar y garantizar el correcto funcionamiento de la plataforma web del proyecto de guardias para el IES Rodanas. Cada miembro tenía responsabilidades y tareas específicas asignadas. Dentro del equipo, se identificaron roles como:
 - **Team lead:** en varios momentos del desarrollo, dos integrantes asumieron el rol de líder para asegurar la organización y la comunicación efectiva dentro del equipo.
 - **UX/UI:** un integrante se encargó de todo lo relacionado con la estética y la experiencia de usuario de la aplicación, asegurando que fuera intuitiva y visualmente atractiva.

Con esta estructura y distribución de roles, logramos mantener un flujo de trabajo eficiente y orientado a satisfacer las necesidades del cliente y los objetivos del proyecto.

2.2.3 ASIGNACIÓN DE TAREAS

Los integrantes han realizado distintas tareas y se han organizado y apoyado para la realización de estas.

Las tareas engloban estas necesidades/requisitos:

- Escoger las tecnologías y entornos adecuados.
- Definir la estructura del proyecto.
- Sistema de identificación segura mediante google.

- Desarrollar un modelo de datos para posteriormente crear la base de datos.
- Diseñar las distintas interfaces del proyecto.
- Definir las reglas de negocio para cada uno de los formularios.

3 Requisitos

La idea principal del proyecto era desarrollar una plataforma de gestión de guardias de los profesores, con un calendario y horario interactivo donde, en tiempo real, se mostrarán las nuevas guardias que iban surgiendo a lo largo de la jornada estudiantil.

Como se ha comentado anteriormente, a causa de la mala organización interna en el equipo de desarrollo, se tuvieron que modificar las necesidades y tareas, dando lugar a la creación de los actuales formularios integrados en el proyecto.

3.1 Requisitos funcionales

Para garantizar que la plataforma funcione correctamente, es necesario cumplir con una serie de funciones esenciales que especifican los requisitos funcionales. Estos requisitos determinan el comportamiento que debe tener el sistema para satisfacer las necesidades y expectativas del usuario. Los requisitos funcionales se pueden considerar como características percibidas por el usuario.

Identificación sencilla. Los usuarios tendrán la opción de iniciar sesión de manera conveniente utilizando sus cuentas de Google.

Interacción con formularios. Los usuarios podrán realizar diversas acciones (añadir, editar, borrar, visualizar) en tiempo real respecto a los formularios:

Gestión de usuarios. Los usuarios tendrán la capacidad de administrar las credenciales y los privilegios de otros usuarios dentro del sistema.

Cierre de sesión. Los usuarios podrán cerrar sesión de manera sencilla y rápida cuando lo deseen. Este proceso permite a los usuarios finalizar su sesión actual para acceder con otra cuenta o simplemente salir del sistema de manera segura.

3.2 Requisitos no funcionales

Son aquellas características que un proyecto debe cumplir para asegurar su calidad y eficiencia en el uso. A diferencia de los requisitos funcionales, que definen qué debe hacer el sistema, los requisitos no funcionales establecen cómo debe hacerlo y las restricciones bajo las cuales debe operar.

Aunque los requisitos no funcionales no afectan directamente a las funciones y características visibles para el usuario, son cruciales para asegurar una experiencia óptima del usuario y el cumplimiento de estándares de calidad definidos en el diseño y desarrollo de aplicaciones y sistemas de software.

Interfaz de usuario. Debe ser intuitiva y fácil de usar, con un diseño minimalista que permite al usuario realizar operaciones de manera rápida y eficiente, enfocándose en los aspectos clave para garantizar un uso adecuado.

Fiabilidad. Es crucial para mantener un funcionamiento continuo y correcto, asegurando la confianza de los usuarios. Debe ser escalable para adaptarse a cambios sin introducir errores.

Seguridad. Fundamental para proteger los datos del usuario y la integridad del sitio web. Las medidas de seguridad deben abarcar desde la configuración inicial del servidor hasta la monitorización y respuesta ante incidentes, adoptando un enfoque integral para reducir riesgos de ataques.

Usabilidad. La interfaz debe ser intuitiva y fácil de usar, permitiendo al usuario realizar operaciones de forma rápida y eficaz, enfocándose en los aspectos relevantes para garantizar un uso adecuado.

Escalabilidad. Debe ser capaz de adaptarse a cambios futuros y necesidades emergentes, permitiendo la integración de nuevos elementos sin comprometer el rendimiento y la funcionalidad.

Rendimiento. La aplicación debe ser rápida y eficiente en el procesamiento y visualización de datos proporcionados por el usuario.

PARTE II

DESARROLLO

4 ESTUDIO DE LOS ESCENARIOS

En este apartado se presenta el análisis de los distintos escenarios posibles durante el uso y el flujo del proyecto, los cuales representan la interacción entre un usuario humano y el sistema. Existen diversos estudios sobre la creación y diseño de casos de uso. En mi caso, he optado por un desarrollo lineal, que es más sencillo de implementar pero no es la mejor práctica óptima.

4.1 ESCENARIOS

4.1.1 Caso de uso: "Iniciar sesión"

- Actores: Usuario.
- Rol: Jefatura, profesor, invitado.
- Tipo: Primario.
- Precondición: El usuario no ha iniciado sesión en el sistema.
- Postcondición: El usuario está autenticado y ha accedido al sistema.
- Propósito: Permitir que un usuario registrado ingrese al sistema.
- Resumen: El usuario se identifica mediante un login de google, el sistema le permite el acceso.

4.1.2 Caso de uso: "Acceso al portal de inicio"

- Actores: Usuario.
- Rol: Jefatura, profesor, invitado.
- Tipo: Primario.
- Precondición:
 - El usuario se ha autenticado y ha accedido al sistema.
 - El usuario tiene rol de jefatura o profesor.
- Postcondición: El sistema le ha redirigido al portal de inicio.
- Propósito: Agrupar en un sitio común a los usuarios y ofrecerles libertad de uso.
- Resumen: El sistema permite al usuario identifica acceder al portal de inicio, donde pueden acceder a diferentes funcionalidades de la plataforma.

4.1.3 Caso de uso: "Acceso al portal de inicio, rol invitado"

- Actores: Usuario.
- Rol: Invitado.
- Tipo: Primario.
- Precondición:

- El usuario se ha autenticado y ha accedido al sistema.
- El usuario tiene rol de invitado.
- Postcondición: El sistema le ha redirigido al portal de inicio.
- Propósito: Controlar a posibles intrusos en un portal de inicio modificado.
- Resumen: El sistema identifica que el usuario tiene un rol de invitado y lo retiene en un portal de inicio para asegurar la seguridad del resto de usuarios, dando posibilidad solo de volver al portal de inicio o cerrar la sesión.

4.1.4 Caso de uso: "Crear ausencias"

- Actores: Usuario.
- Rol: Jefatura, profesor
- Tipo: Primario.
- Precondición:
 - El usuario se ha autenticado y ha accedido al sistema.
 - El usuario tiene rol de jefatura o profesor
 - El sistema le ha redirigido al portal de inicio.
 - El usuario ha accedido al a la pestaña de ausencias
- Postcondición: El usuario pulsa el botón y completa el formulario.
- Propósito: Agregar una nueva ausencia.
- Resumen: El sistema permite al usuario añadir una nueva ausencia.

4.1.5 Caso de uso: "Interactuar con las ausencias"

- Actores: Usuario.
- Rol: Jefatura, profesor
- Tipo: Primario.
- Precondición:
 - El usuario se ha autenticado y ha accedido al sistema.
 - El usuario tiene rol de jefatura o profesor
 - El sistema le ha redirigido al portal de inicio.
 - El usuario ha accedido al a la pestaña de ausencias
 - El usuario ha añadido una ausencia.
- Postcondición: El usuario puede visualizar, editar y borrar solo sus ausencias.
- Propósito: Llevar un control de las ausencias.
- Resumen: El sistema permite al usuario interactuar con las ausencias.

4.1.6 Caso de uso: "Interactuar con las ausencias, rol jefatura"

- Actores: Usuario.
- Rol: Jefatura, profesor
- Tipo: Primario.
- Precondición:

- El usuario se ha autenticado y ha accedido al sistema.
- El usuario tiene rol de jefatura o profesor
- El sistema le ha redirigido al portal de inicio.
- El usuario ha accedido al a la pestaña de ausencias
- Algún usuario ha añadido una ausencia.
- Dispone del rol de jefatura
- Postcondición: El usuario puede visualizar, editar y borrar sus ausencias y las de los demás usuarios..
- Propósito: Llevar un control de las ausencias del profesorado.
- Resumen:El sistema permite al usuario con rol de jefatura interactuar y controlar las ausencias de todos los usuarios.

4.1.7 Caso de uso: "Gestionar las clases"

- Actores: Usuario.
- Rol: Jefatura.
- Tipo: Primario.
- Precondición:
 - El usuario se ha autenticado y ha accedido al sistema.
 - El usuario tiene rol de jefatura o profesor
 - El sistema le ha redirigido al portal de inicio.
 - El usuario ha accedido al a la pestaña de clases
 - Dispone del rol de jefatura
- Postcondición: El usuario puede visualizar, añadir, editar y borrar el nombre y puntuaciones de los clases .
- Propósito: Llevar un control de las clases .
- Resumen:El sistema permite al usuario con rol de jefatura interactuar y controlar los clases del centro educativo del Ies Rodanas, donde podrá modificar el nombre y puntuación de estos.

4.1.8 Caso de uso: "Gestionar usuarios"

- Actores: Usuario.
- Rol: Jefatura.
- Tipo: Primario.
- Precondición:
 - El usuario se ha autenticado y ha accedido al sistema.
 - El usuario tiene rol de jefatura o profesor
 - El sistema le ha redirigido al portal de inicio.
 - El usuario ha accedido al a la pestaña de gestión de usuarios.
 - Dispone del rol de jefatura
- Postcondición: El usuario puede visualizar, añadir, editar y borrar a los usuarios.
- Propósito: Llevar un control de los usuarios de la plataforma.

- Resumen: El sistema permite al usuario con rol de jefatura interactuar y controlar los usuarios del centro educativo controlando también el rol de estos.

4.1.9 Caso de uso: "Cerrar sesión".

- Actores: Usuario
- Rol: Jefatura, profesor, invitado.
- Tipo: Primario
- Precondición:
 - El usuario ha iniciado sesión en el sistema.
 - El usuario tiene rol de jefatura, profesor, invitado.
 - El sistema le ha redirigido al portal de inicio.
- Postcondición: El usuario ha cerrado sesión y no tiene acceso a las funciones protegidas del sistema.
- Propósito: Permitir que el usuario finalice su sesión activa en el sistema.
- Resumen: El usuario realiza la acción de cerrar sesión, lo que resulta en la eliminación de la sesión activa y la restricción de acceso a las funciones que requieren autenticación.

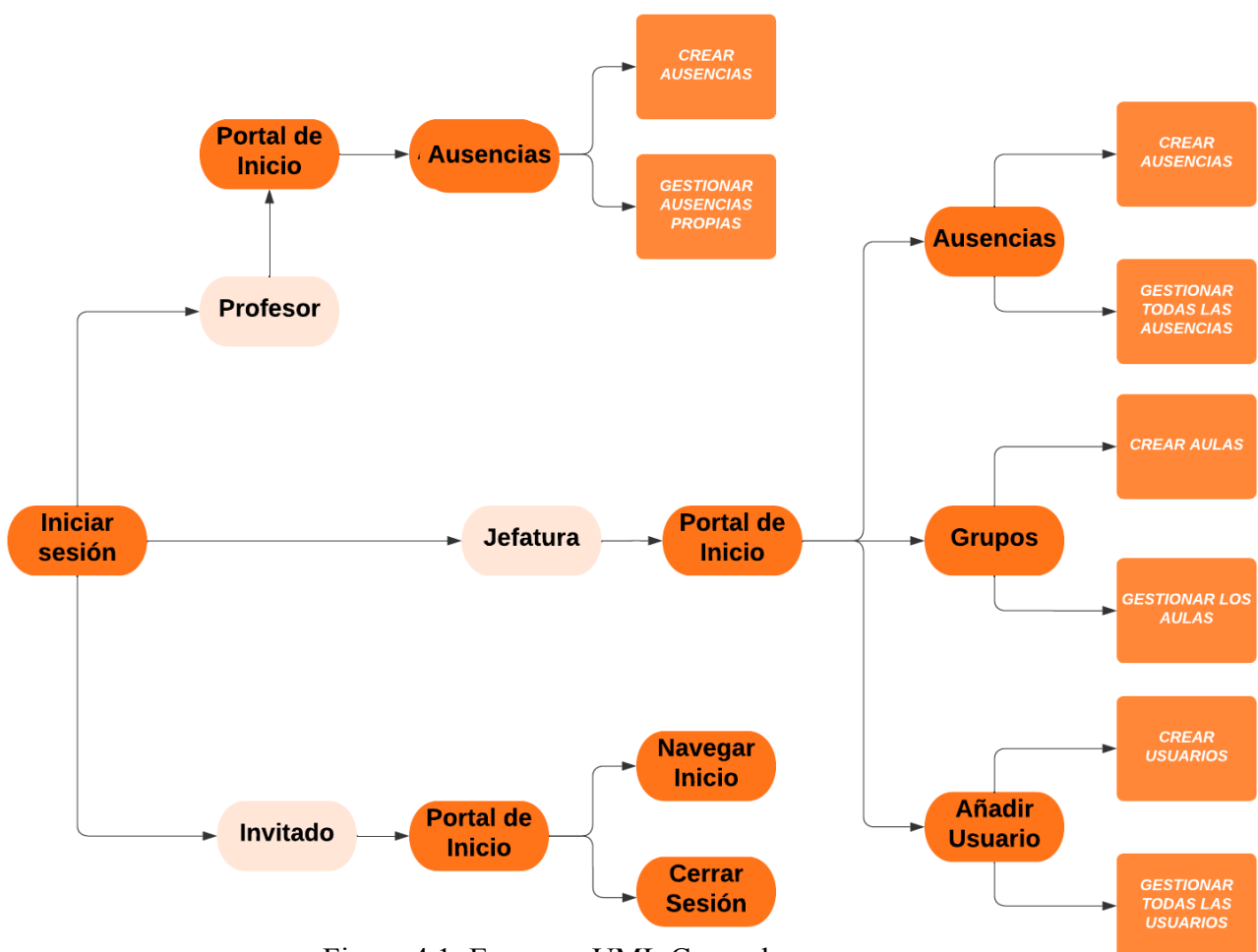


Figura 4.1: Esquema UML Casos de uso

4.2 FLUJO DE LA APLICACIÓN

En este apartado se detalla el flujo general de la aplicación, abarcando desde la identificación del usuario hasta la finalización de una tarea. Este flujo integra y contextualiza los casos de uso discutidos previamente, proporcionando una visión coherente y completa del funcionamiento de la aplicación

☐ Acceso a la plataforma

El usuario accede a la plataforma web.

☐ Inicio de sesión

El usuario se identifica mediante un sistema de identificación proporcionado por Google, el cual valida y gestiona las credenciales del usuario para acceder a servicios y plataformas mediante sus cuentas asociadas.

☐ Portal de inicio

Una vez identificado, el usuario tendrá acceso al portal de inicio donde tendrá posibilidad de interactuar con las diferentes tarjetas.

☐ Navegación entre casos de uso

Dependiendo del rol del usuario, se determinarán los niveles de acceso correspondientes a sus funciones específicas dentro del sistema o plataforma.

☐ Completado y gestión de formularios

1-El usuario rellena los campos necesarios y sigue las instrucciones especificadas en cada caso de uso.

2-El sistema verifica la información proporcionada y la envía a la base de datos para mostrarle al usuario al instante la información previamente agregada en una tabla.

☐ Cerrado de sesión

El usuario tiene la opción de cerrar su sesión al completar sus tareas o si desea iniciar sesión con una cuenta diferente.

5 Arquitectura

- **Base de datos**

Hemos optado por PHPMyAdmin como herramienta de gestión de bases de datos web, específicamente diseñada para administrar MySQL y MariaDB.

Para la base de datos, se ha documentado exhaustivamente cada una de las tablas junto con sus relaciones. Utilizamos un esquema UML para facilitar la comprensión del diseño. Seguimos los tipos de datos proporcionados por el usuario para estructurar estas tablas de manera que garanticen el funcionamiento óptimo de nuestro sitio web.

Se han creado siete tablas: "ausencia", "clases", "horas", "motivos", "profesor", "puntuacion" y "users". Cada una de estas tablas está diseñada con los datos correspondientes y los tipos de datos necesarios para asegurar la accesibilidad eficiente de la información.

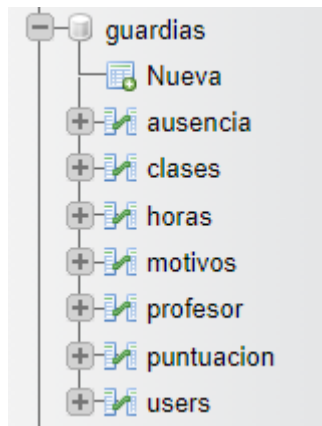


Figura 5: Tablas de la base de datos

5.1 Funcionalidades

En el presente apartado se realiza un análisis detallado de la manera en que se estructuran y recopilan los datos, así como del uso que se les ha dado a estos datos para crear y desarrollar las funcionalidades fundamentales de la plataforma.

5.1.1 Users

En nuestra base de datos, tenemos una tabla llamada users que guarda los correos electrónicos utilizados para iniciar sesión en nuestra página web. Implementamos la validación de Google para asegurar un acceso más seguro y eficiente.

guardias	users
id	int(11)
email	varchar(50)
first_name	varchar(50)
last_name	varchar(50)
gender	varchar(50)
full_name	varchar(50)
picture	varchar(255)
verifiedEmail	int(11)
token	varchar(255)

Figura 5.1.1: Tabla users

5.1.2 Formulario de ausencias

Para el correcto funcionamiento del formulario de las ausencias, hemos desarrollado 5 tablas: profesor, ausencia, horas y motivos. Estas tablas son esenciales para garantizar que cuando un profesor registre una ausencia, el sistema no presente errores.

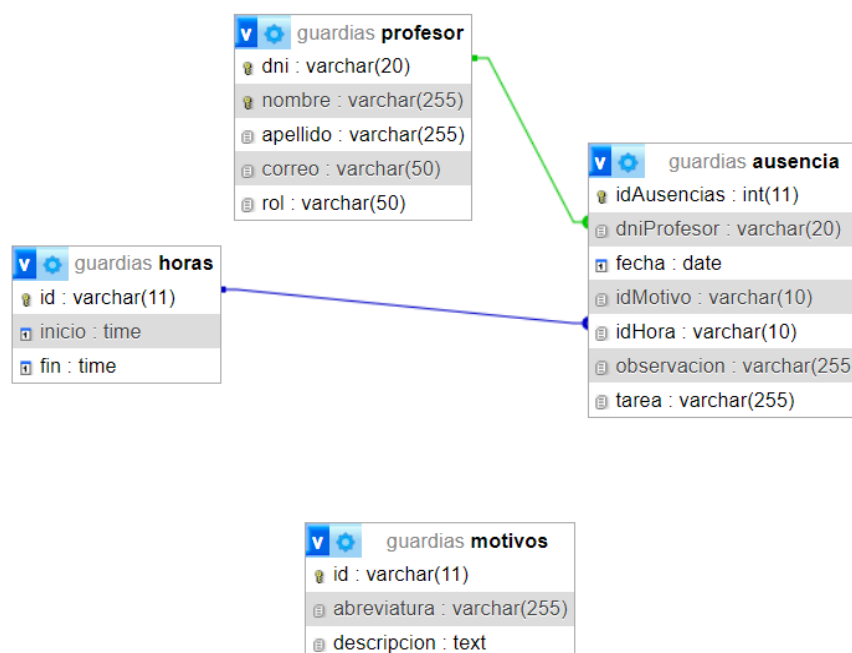


Figura 5.1.2: Relaciones del formulario de ausencias

La tabla "profesor" incluye todas las columnas necesarias para identificar quién ha registrado cada ausencia. Esta tabla se relaciona con la tabla "ausencia" a través del DNI, mientras que la tabla "ausencia" se vincula con la tabla "horas" mediante el ID de hora.

La tabla "motivos" agrupa todas las razones por las cuales un profesor puede estar ausente. En la interfaz, se proporcionará un menú desplegable al profesor para que pueda seleccionar el motivo de su ausencia. Podemos acceder a los datos de la tabla "ausencias" y así poder interactuar y gestionarlos.

5.1.3 Formulario de clases

Para el formulario de las clases, hemos utilizado dos tablas: una para las clases y otra para las puntuaciones, de modo que cada usuario pueda ver las puntuaciones asignadas.

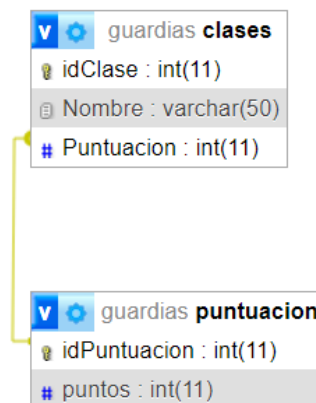


Figura 5.1.3: Relaciones del formulario de clases

Para el formulario de clases, la selección es más sencilla al no tener que controlar tantas tablas y columnas.

5.2 Esquema UML

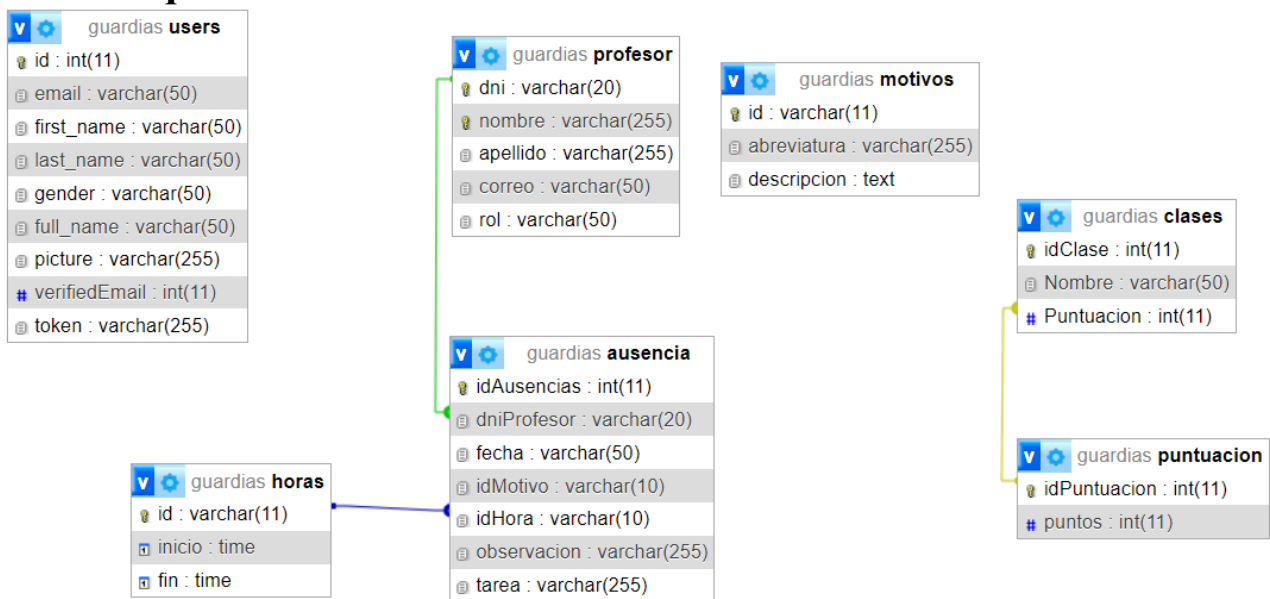


Figura 5.2: Esquema UML de la base de datos

6.2 Conexión a la base de datos.

6.2.1 Acceso a PHPMyAdmin.

Para acceder a PHPMyAdmin, abrimos nuestro navegador web e ingresamos la URL 'http://localhost/phpmyadmin'. Luego, introducimos el nombre de usuario y la contraseña de nuestra cuenta de MySQL. Tras iniciar sesión, se mostrará la pantalla principal de PHPMyAdmin.

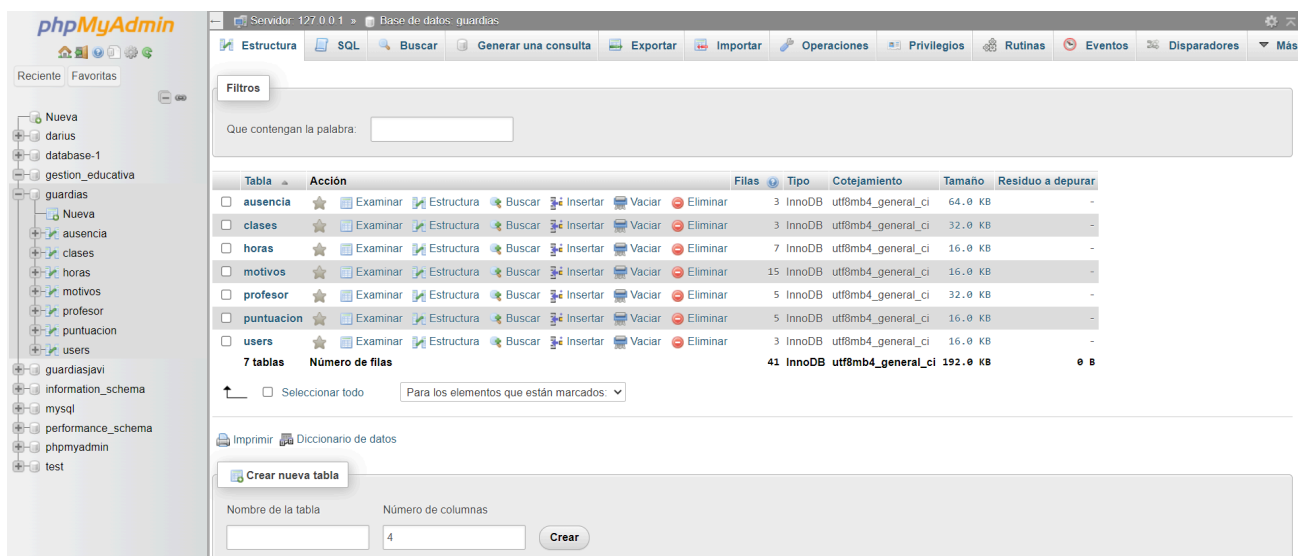


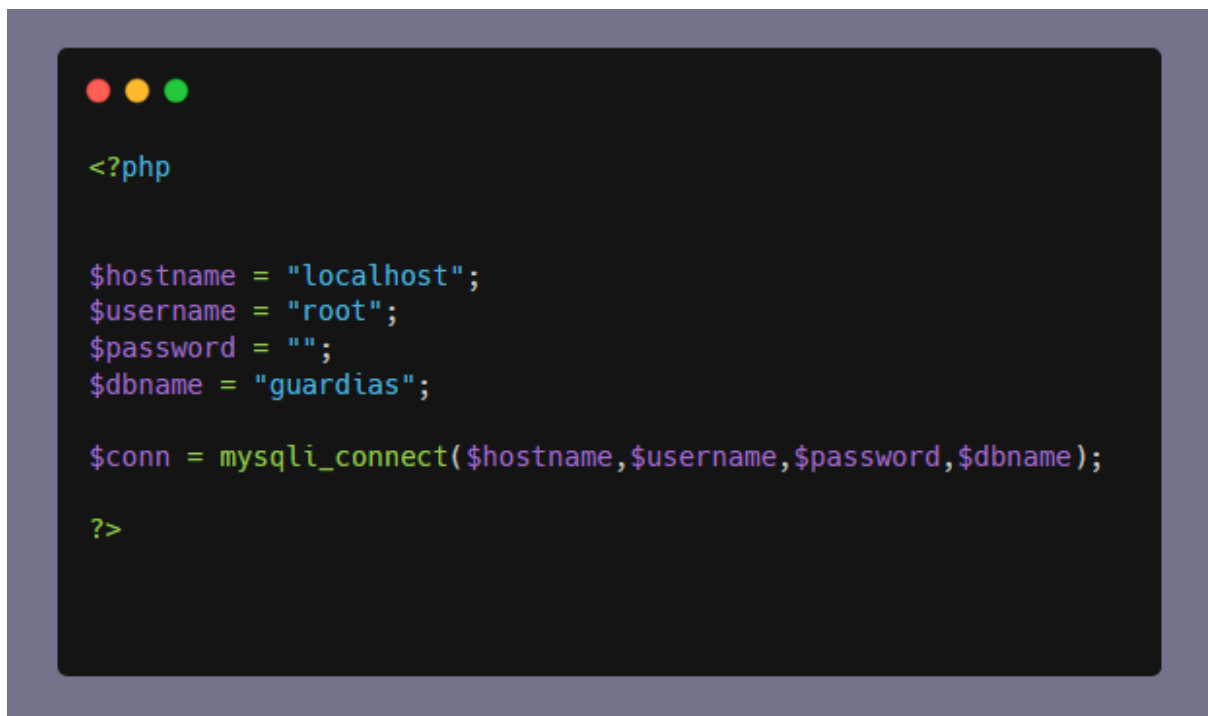
Figura 6.2.1: Imagen de la interfaz de phpMyAdmin

Una vez que inicies sesión, verás una lista de bases de datos en el panel izquierdo. Haz clic en la base de datos que deseas gestionar para crear, añadir, modificar y eliminar columnas o tablas.

6.2.2 Acceso a PHPMyAdmin con PHP.

Para establecer la conexión desde nuestro código, hemos creado una clase llamada `conex.php`. Esta clase se encarga únicamente de gestionar la conexión, permitiéndonos conectarnos desde todas las demás clases que desarrollemos.

Para lograrlo, utilizamos la extensión `mysqli` de PHP, que proporciona una interfaz para comunicarse con servidores de bases de datos MySQL. Esta extensión permite a los desarrolladores interactuar con MySQL de manera eficiente y segura desde sus scripts PHP.

A screenshot of a code editor with a dark background and light-colored text. The code is in PHP and defines variables for database connection: \$hostname, \$username, \$password, and \$dbname, followed by a call to mysqli_connect. The code is enclosed in PHP tags <?php and ?>. The editor has a light blue border and three colored window control buttons (red, yellow, green) in the top left corner.

```
<?php

$hostname = "localhost";
$username = "root";
$password = "";
$dbname = "guardias";

$conn = mysqli_connect($hostname,$username,$password,$dbname);

?>
```

Figura 6.2.2: Código de la conexión con la base de datos

- ♦ **\$hostname:** Es una variable que almacena el nombre del servidor local (localhost).
- ♦ **\$username:** Esta variable almacena el nombre del usuario.
- ♦ **\$password:** Es una variable que almacena la contraseña. En este caso, está vacía y por eso se escribe entre comillas.
- ♦ **\$dbname:** Es la variable que contiene el nombre de la base de datos. "localhost": Especifica la dirección del servidor de bases de datos, generalmente usado si MySQL y PHP están en el mismo servidor.
- ♦ **"root":** Es el nombre de usuario utilizado para acceder a MySQL.
- ♦ **"":** Representa la contraseña asociada al usuario, que está vacía en este caso.

- ♦ **"guardias"**: Es el nombre de la base de datos a la cual se desea conectar. Si la conexión es exitosa,
- ♦ **\$conn** contendrá un objeto mysqli que representa la conexión establecida.

7 Diseño

En este capítulo se aborda el diseño completo de la plataforma web del proyecto, incluyendo todos sus elementos y las funcionalidades que ofrece.

El diseño de la plataforma se ha realizado utilizando lenguajes de frontend como HTML y CSS, junto con la biblioteca Bootstrap. Esta herramienta ha sido fundamental gracias a su capacidad para dividir la página en 12 secciones, simplificando así la disposición de los distintos componentes.

Además, Bootstrap ofrece una amplia gama de recursos visuales y modernos que ahorran tiempo al no requerir diseñar todo desde cero.

7.1 Interfaz de identificación

La interfaz de inicio de sesión ahora es muy simple, al usar el sistema de identificación de google, no es necesario crear las interfaces de registro y logueado.

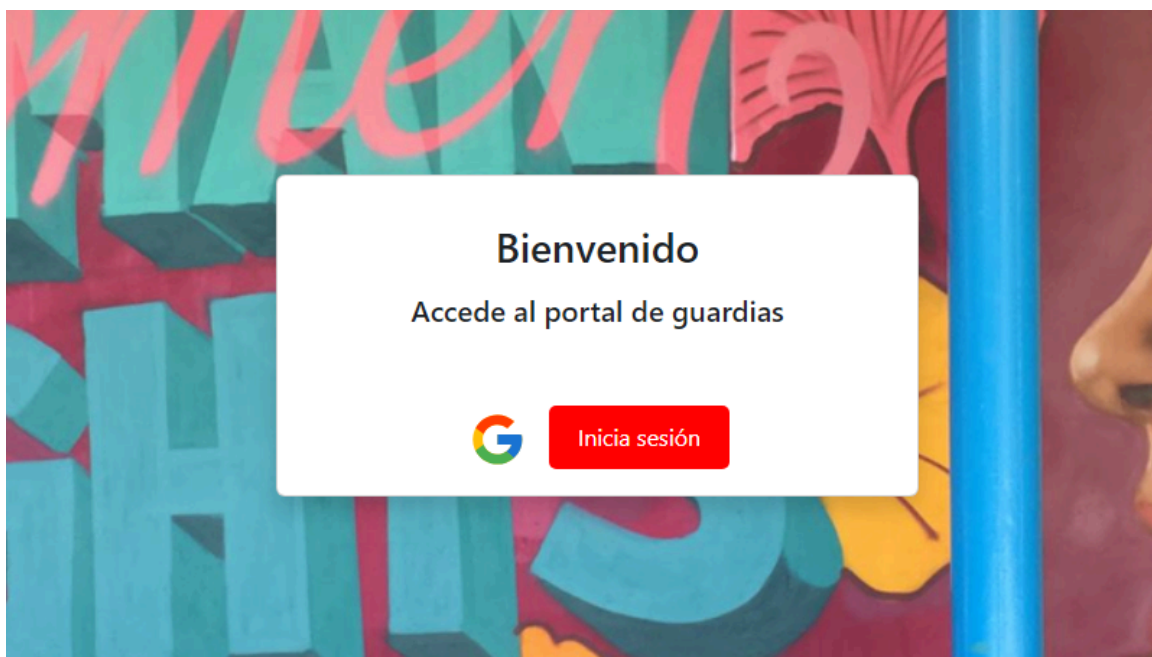


Figura 7.1: Interfaz de identificación usando google

Por lo tanto, al hacer clic en el botón "Iniciar sesión", será redirigido a una pestaña que le será familiar.

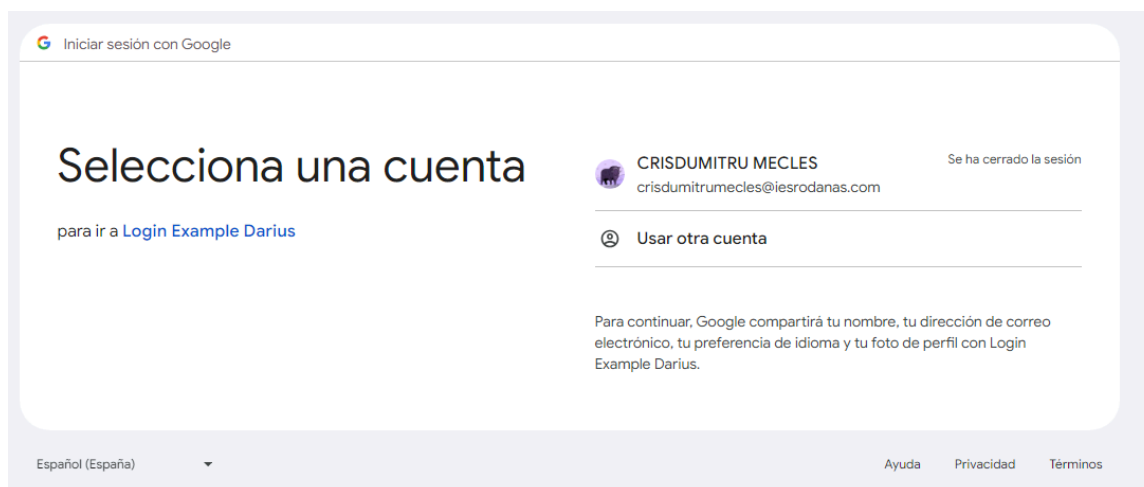


Figura 7.1 (1): Interfaz de logueado con google

Una vez que se haya identificado, el sistema de Google enviará un token. Si ese token coincide con el de nuestra base de datos, podrá acceder al portal de inicio.

7.2 Portal de inicio

En esta sección, tenemos la capacidad de gestionar a los usuarios según sus roles. Hemos desarrollado una interfaz interactiva y fácil de usar que se adapta a todos los usuarios, con un diseño estético similar al sitio web del centro IES Rodanas.

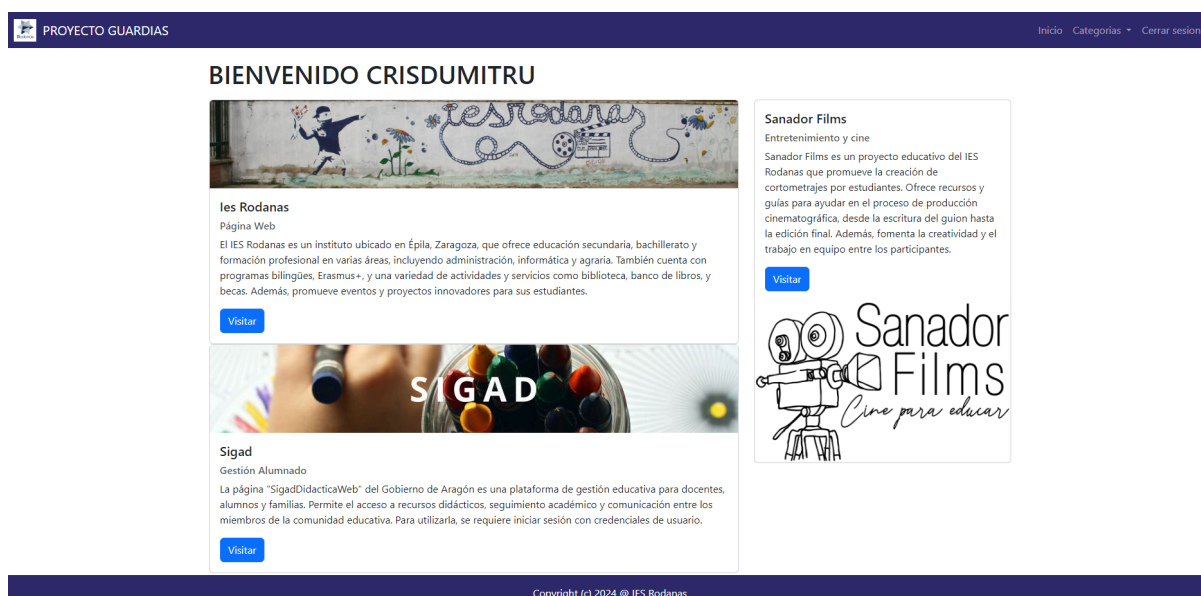


Figura 7.2: Diseño del portal de inicio.

7.2.1 Barra de navegación

La barra de navegación mejora la accesibilidad para los diferentes roles de usuarios al permitirles desplazarse fácilmente entre las diversas secciones del sitio web. Incluye enlaces a funciones clave como el cierre de sesión y el menú desplegable de categorías, que proporciona más opciones de navegación.

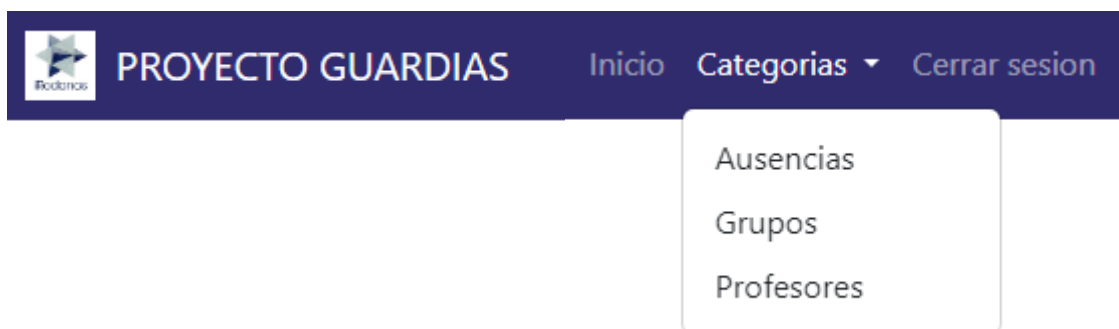


Figura 7.2.1: Diseño de la barra de navegación

7.2.2 Cards

Las cards o también conocidas como tarjetas, son componentes flexibles que permiten mostrar contenido relevante en una pequeña sección. Muestran información como textos, imágenes, encabezados y enlaces de una manera organizada y fácil de consumir.

Hemos creado tres cards donde mostramos páginas relacionadas con el centro educativo, tales como la propia web del “Ies Rodanas”, el canal de youtube de “Sanador Films” y la plataforma de gestión de “Sigad”.



Ies Rodanas

Página Web

El IES Rodanas es un instituto ubicado en Épila, Zaragoza, que ofrece educación secundaria, bachillerato y formación profesional en varias áreas, incluyendo administración, informática y agraria. También cuenta con programas bilingües, Erasmus+, y una variedad de actividades y servicios como biblioteca, banco de libros, y becas. Además, promueve eventos y proyectos innovadores para sus estudiantes.

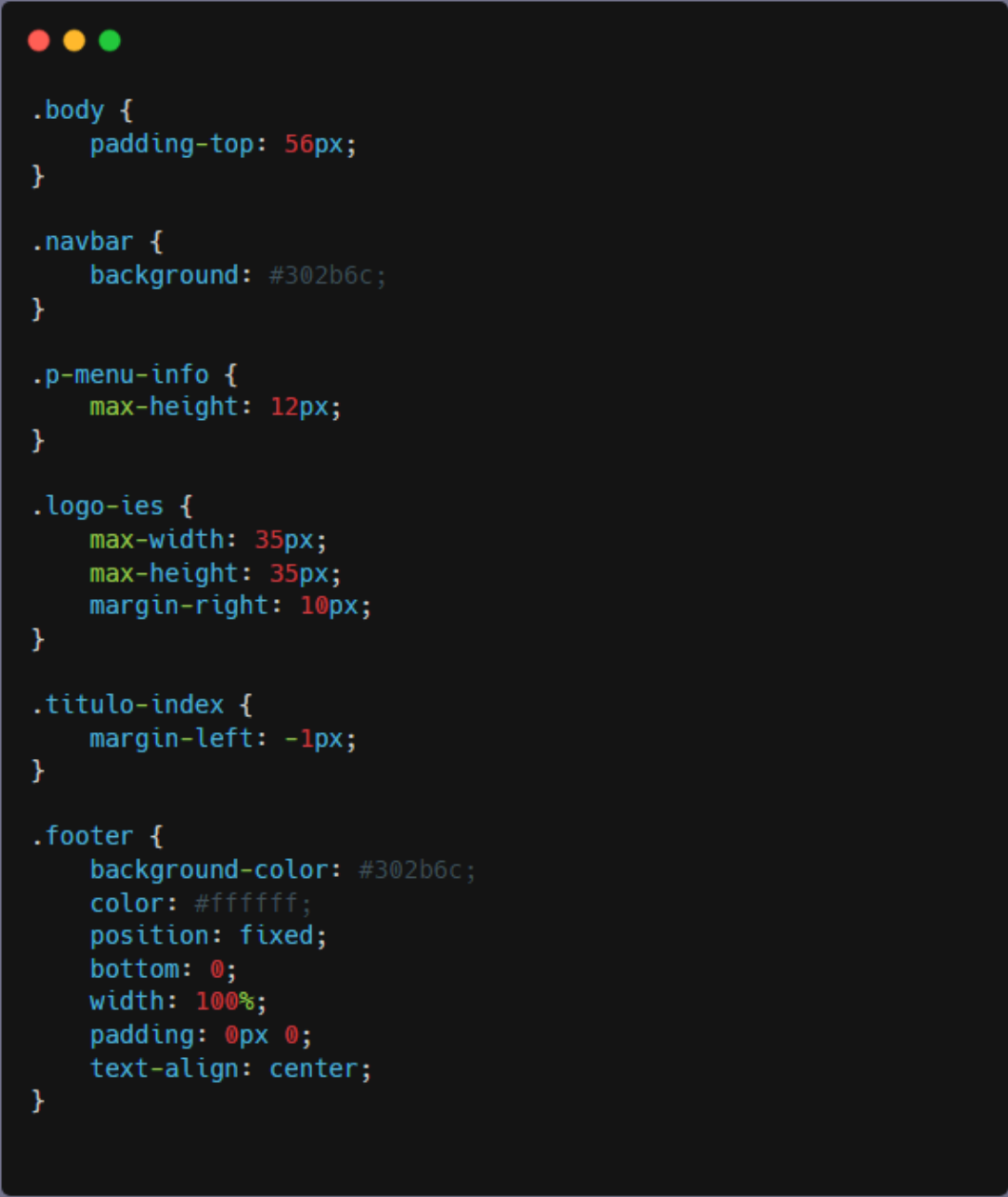
Visitar

Figura 7.2.2: Diseño de la tarjeta del Ies Rodanas.

7.2.3 Hoja de estilo (CSS)

Todo el diseño del portal está proporcionado por el siguiente archivo CSS, llamado menu.css. Donde encontramos elementos como:

- .navbar: controlamos el color de la barra de navegación.
- .logo-ies: ajustamos el logotipo para que cuadre con la barra de navegación.
- .footer: controlamos el diseño y posición del footer para que no interfiera con las cards.

A screenshot of a code editor with a dark background and light-colored text. The code is CSS for a menu. It includes rules for .body, .navbar, .p-menu-info, .logo-ies, .titulo-index, and .footer. The .body rule sets padding-top to 56px. The .navbar rule sets background to #302b6c. The .p-menu-info rule sets max-height to 12px. The .logo-ies rule sets max-width to 35px, max-height to 35px, and margin-right to 10px. The .titulo-index rule sets margin-left to -1px. The .footer rule sets background-color to #302b6c, color to #ffffff, position to fixed, bottom to 0, width to 100%, padding to 0px 0, and text-align to center.

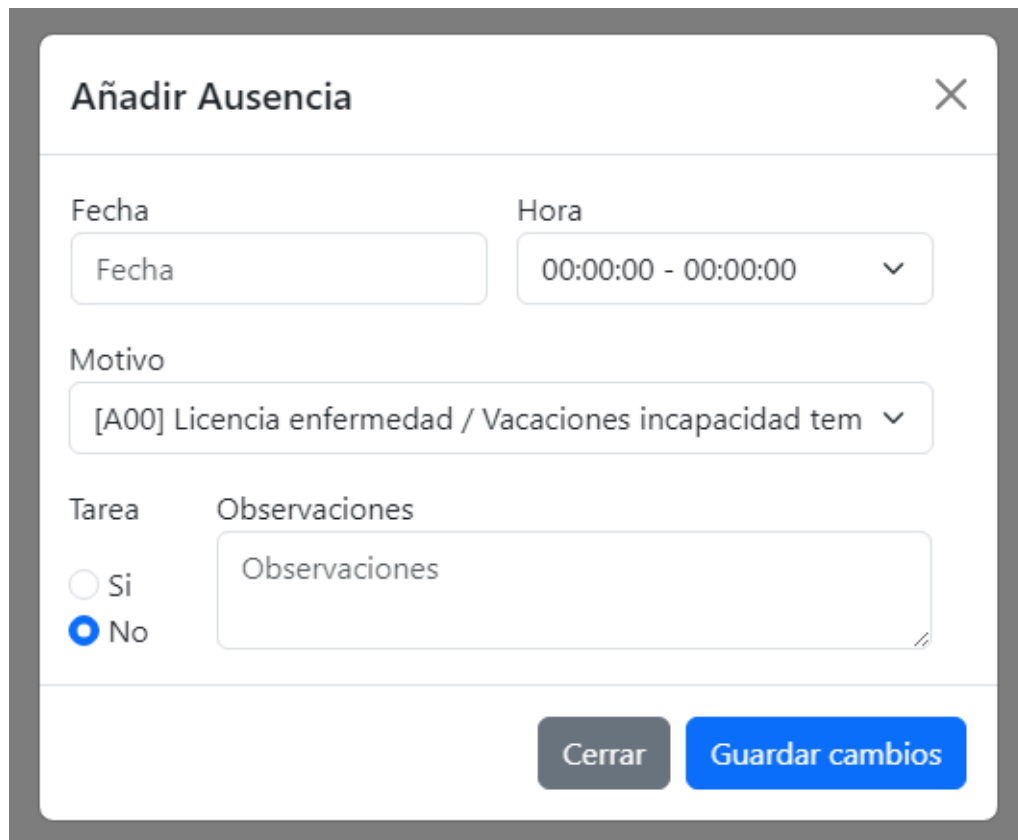
```
.body {  
  padding-top: 56px;  
}  
  
.navbar {  
  background: #302b6c;  
}  
  
.p-menu-info {  
  max-height: 12px;  
}  
  
.logo-ies {  
  max-width: 35px;  
  max-height: 35px;  
  margin-right: 10px;  
}  
  
.titulo-index {  
  margin-left: -1px;  
}  
  
.footer {  
  background-color: #302b6c;  
  color: #ffffff;  
  position: fixed;  
  bottom: 0;  
  width: 100%;  
  padding: 0px 0;  
  text-align: center;  
}
```

Figura 7.2.3: Código de la hoja de estilos menu

7.3 Pestaña de ausencias

En la sección siguiente, usted tiene la responsabilidad de mantener un control de los registros de las ausencias del profesorado. Tenemos disponible una interfaz interactiva y atractiva que le permite agregar nuevas ausencias con solo pulsar un botón.

Al hacer clic en este botón, se abrirá un modal, también llamado ventana de diálogo, que presenta un formulario estético con varios campos que deben completarse.



El modal 'Añadir Ausencia' presenta un formulario con los siguientes campos:

- Fecha:** Un campo de texto con el placeholder 'Fecha'.
- Hora:** Un selector de hora que muestra '00:00:00 - 00:00:00'.
- Motivo:** Un selector de motivo que muestra '[A00] Licencia enfermedad / Vacaciones incapacidad tem'.
- Tarea:** Un grupo de botones de radio con las opciones 'Si' y 'No'. El botón 'No' está seleccionado.
- Observaciones:** Un campo de texto con el placeholder 'Observaciones'.

En la parte inferior del modal hay dos botones: 'Cerrar' (gris) y 'Guardar cambios' (azul).

Figura 7.3: Diseño del modal de ausencias

Después de guardar los cambios, la nueva ausencia aparecerá en una tabla junto con las otras ausencias previamente creadas, si las hubiera.

Se han realizado una serie de consultas en el archivo ausencia.php, que nos permiten acceder a los datos de la tabla "ausencias" para interactuar y gestionarlos. También se accede a las tablas "horas" y "motivos", de modo que el modal pueda utilizar AJAX para acceder a la información de forma dinámica.

```
<?php
include("../properties/userinfo.php");
$sqlDni = "SELECT dni FROM profesor WHERE correo
='{$_userinfo['email']}'";
$resultado = mysqli_query($conn,$sqlDni);
$dni = mysqli_fetch_assoc($resultado);
$sqlRol = "SELECT rol FROM profesor WHERE correo =
'{$_userinfo['email']}'";
$resultadoRol = mysqli_query($conn, $sqlRol);
$rol = mysqli_fetch_assoc($resultadoRol)['rol'];

if($rol == 'profesor') {
    $sql = "SELECT a.*, h.inicio, h.fin, m.abreviatura,
p.nombre FROM
    ausencia a
    JOIN horas h ON a.idHora = h.id
    JOIN motivos m ON a.idMotivo = m.id
    JOIN profesor p ON a.dniProfesor = p.dni
    WHERE dniProfesor = '{$_dni['dni']}'";
} elseif ($rol == 'jefatura') {
    $sql = "SELECT a.*, h.inicio, h.fin, m.abreviatura,
p.nombre FROM
    ausencia a
    JOIN profesor p ON a.dniProfesor = p.dni -- Revisar si es
necesario esto
    JOIN horas h ON a.idHora = h.id
    JOIN motivos m ON a.idMotivo = m.id";
}

$resultado = mysqli_query($conn, $sql);

if (!$resultado) {
    die('Error no hay datos');
} else {
    $ausencias = array();
    while ($data = mysqli_fetch_assoc($resultado)) {
        $data['hora'] = $data['inicio'] . ' - ' .
        $data['fin'];
        $data['motivo'] = $data['abreviatura'];
        $ausencias[] = $data;
    }
    echo json_encode(['data' => $ausencias]);
}

mysqli_free_result($resultado);
mysqli_close($conn);
?>
```

Figura 7.3 (1): Código del formulario de ausencias

Para poder acceder a los motivos y a las horas, gracias a ajax usando un método get se ha podido acceder a la propia base de datos para obtener estos datos de forma dinámica.

```
$.ajax({
  url: '../PHP/TablaMotivos.php',
  method: 'GET',
  success: function(data) {
    var motivos = JSON.parse(data);
    var select = $("#motivosSelect");
    select.empty();
    motivos.forEach(function(motivo) {
      var optionText = motivo.abreviatura;
      var option = $("
<option>").val(motivo.id).text(optionText);
      select.append(option);
    });
  }
});
```

Figura 7.3 (2): Código de ajax tabla motivos







```
$.ajax({
  url: '../PHP/TablaHoras.php',
  method: 'GET',
  success: function(data) {
    var horas = JSON.parse(data);
    var select = $("#horaSelect");
    select.empty();
    horas.forEach(function(hora) {
      var optionText = hora.inicio + ' - ' +
hora.fin;
      var option = $("
<option>").val(hora.id).text(optionText);
      select.append(option);
    });
  }
});
```

Figura 7.3 (3): Código de ajax tabla horas

Por último, observamos la posibilidad de filtrar por búsqueda, editar y borrar estas ausencias.

10 ▾ entries per page

Search:

Dni	Nombre	Fecha	Motivo	Hora	Tarea	Observacion		
12345678A	Cris Dumitru	06/19/2024	[A05] Traslado domicilio	10:20:00 - 11:10:00	si	Tarea lengua página 132 ejercicio 12-16.		
12345678A	Cris Dumitru	06/11/2024	[A00] Licencia enfermedad / Vacaciones incapacidad temporal	00:00:00 - 00:00:00	no			
12345678A	Cris Dumitru	06/19/2024	[A02] Permiso progenitor diferente de madre biológica	10:20:00 - 11:10:00	si	Buscar mando del proyector.		
45678901D	Luz Ibarz	06/20/2024	[A00] Licencia enfermedad / Vacaciones incapacidad temporal	08:30:00 - 09:20:00	no	Ninguna		
45678901D	Luz Ibarz	06/13/2024	[A00] Licencia enfermedad / Vacaciones incapacidad temporal	00:00:00 - 00:00:00	no	Cuidado con ciertos alumnos.		

Showing 1 to 5 of 5 entries

«
<
1
>
»

Figura 7.3 (4): Diseño tabla de ausencias

7.4 Pestaña de clases

En esta sección, tenemos el control de las aulas individuales del centro, además de la capacidad de administrar la puntuación que estas otorgan. Disponemos de una interfaz similar a la pestaña de ausencias.

- Tenemos la posibilidad de agregar nuevas clases con su respectivo nombre y puntuación.

Añade una clase

☐ Mostrar alerta al añadir clase

Figura 7.4: Diseño del formulario de clases

- La posibilidad de borrar todas las clases:

Eliminar todas las clases

Figura 7.4 (1): Botón de eliminar clases

- La posibilidad de gestionar dichas clases en tiempo real, se puede borrar y actualizar la puntuación pulsando en cada uno de los diferentes botones de puntuación.


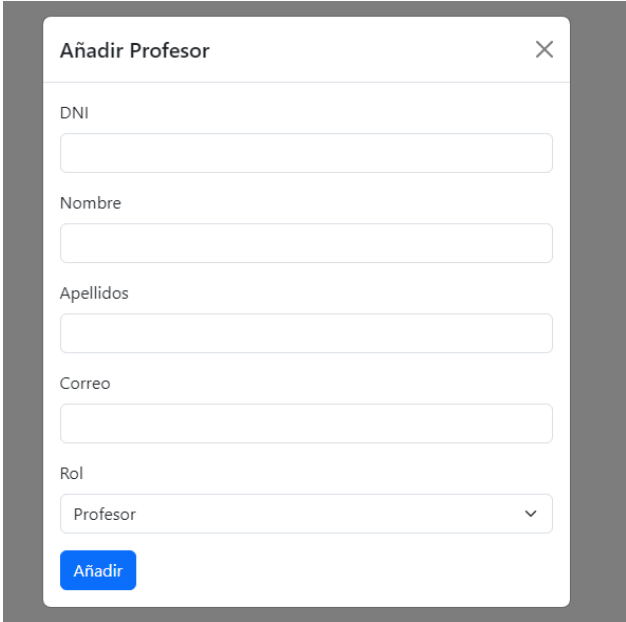
ID	Nombre	Puntuacion	Acciones / Editar Puntuación
31	1ESO	5	 1 2 3 4 5
32	1ESOB	5	 1 2 3 4 5
33	2ESOA	4	 1 2 3 4 5
34	2ESOB	4	 1 2 3 4 5
35	3ESOA	3	 1 2 3 4 5
36	3ESOB	3	 1 2 3 4 5
37	4ESOA	2	 1 2 3 4 5
38	4ESOB	2	 1 2 3 4 5
39	1BACHA	1	 1 2 3 4 5
40	1BACHB	1	 1 2 3 4 5

Figura 7.4 (2): Diseño de la tabla de aulas

7.5 Pestaña de profesores

En este último apartado disponemos de una pantalla que solo está accesible para el equipo de jefatura. Esta pantalla es fundamental para gestionar a los usuarios de la plataforma, ya que permite modificar roles en tiempo real, lo cual es crucial para evitar intrusos y mantener un control adecuado sobre los accesos.

De nuevo, disponemos de un botón que abre un modal, donde se podrán introducir las credenciales de los usuarios, tales como dni,nombre,apellido,correo,rol.



Añadir Profesor ✕

DNI

Nombre

Apellidos

Correo

Rol

Añadir

Figura 7.5: Diseño formulario de profesores

Donde se almacenará todo en una tabla la cual se podrá gestionar en tiempo real y se dispondrá de la posibilidad de editar y borrar dichos usuarios.

Dni	Nombre	Apellidos	Correo	Rol	Acciones
12345678A	Cris Dumitru	Mecles	crisdumitrumecles@iesrodanas.com	jefatura	<button>Editar</button> <button>Borrar</button>
45678901D	Luz Ibarz	Luz Ibarz Puyal	dariusanton@iesrodanas.com	jefatura	<button>Editar</button> <button>Borrar</button>
X25632FD	Silvia Baron	Ibañez	cafdawsda@gmail.com	jefatura	<button>Editar</button> <button>Borrar</button>

Figura 7.5 (1): Diseño formulario de profesores

8 Identificación usando google

La carpeta "vendor" en un proyecto web normalmente alberga las dependencias y bibliotecas de terceros instaladas a través de un gestor de dependencias, como Composer en PHP. Al integrar el inicio de sesión con Google, esta carpeta suele incluir bibliotecas que simplifican la autenticación y la interacción con las API de Google.

Esta carpeta se genera utilizando la terminal de nuestro IDE mediante un comando específico, y su contenido puede variar según el dispositivo utilizado.

```
composer require google/apiclient:"^2.0"
```

Figura 8: Código composer

Se ha utilizado la carpeta autoload, la cual posee un mecanismo que permite cargar automáticamente las clases de PHP cuando son necesarias, evitando la necesidad de incluir manualmente cada archivo que las contiene. Composer, el gestor de dependencias para PHP, proporciona un sistema de autoloading que simplifica este proceso.

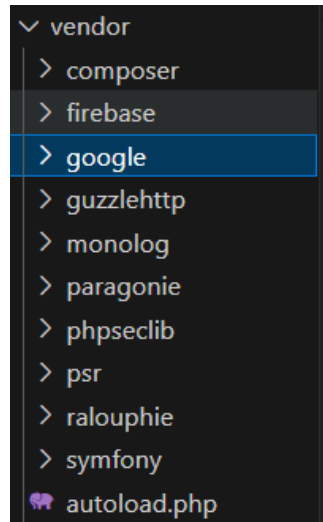


Figura 8 (1): Estructura vendor

Se ha utilizado OAuth, es un protocolo estándar para la autorización que permite a los usuarios compartir sus recursos almacenados en un sitio con otro sitio sin necesidad de revelar sus credenciales completas.

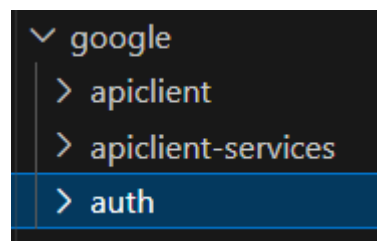


Figura 8 (2): Carpeta auth

8.1 FUNCIONAMIENTO

1. **Solicitud de autorización:** El usuario inicia el proceso de autorización en tu aplicación web, la cual redirige al usuario al proveedor de identidad (Google) para obtener el permiso necesario.
2. **Concesión de autorización:** El usuario otorga permiso a la aplicación para acceder a ciertos datos.
3. **Intercambio de tokens:** El proveedor de identidad emite un token de acceso que la aplicación puede utilizar para realizar solicitudes autorizadas en nombre del usuario.

4. **Uso del token:** La aplicación emplea el token de acceso para obtener los recursos del usuario desde el servidor de recursos (Google).

Un inconveniente de iniciar sesión mediante OAuth es que, si los servidores de Google se vieran comprometidos, no sería posible acceder a la aplicación.

PARTE III

EPÍLOGO

9. CONCLUSIÓN

En este último capítulo se habla sobre los objetivos logrados y no logrados, sobre las habilidades obtenidas y unos consejos / ideas para el futuro equipo que acoja el proyecto.

9.1 ANÁLISIS DE LOS OBJETIVOS

- ♦ Establecer una base de datos para almacenar todos los datos relevantes de manera segura y organizada.

Conseguido → Se ha construido lo que serán los cimientos para las futuras versiones del proyecto, la base de datos es funcional y está conectada correctamente.

- ♦ Implementar un sistema de autenticación utilizando Google para mejorar la seguridad del acceso.

Conseguido → La idea de usar la autenticación de google es genial para evitar llevar el almacenamiento y encriptado de las contraseñas de los usuarios.

- ♦ Diseñar un portal inicial que sirva como punto de entrada intuitivo y funcional para todo tipo de usuarios.

Conseguido → Se ha creado una interfaz común donde poder recoger a los usuarios, para que posteriormente estos naveguen donde deseen. La navegación es clara, simple y fácil de entender para todo tipo de usuarios.

- ♦ Subir información a la base de datos mediante un botón desde una interfaz.

No conseguido → Aunque al inicio del proyecto el equipo de desarrollo contaba con esta funcionalidad, a medida que las tareas fueron cambiando, su uso dejó de ser necesario y/o se complicó su implementación.

- ♦ Desarrollar una interfaz que permita al personal docente y directivo registrar y visualizar sus ausencias de forma clara y rápida.

Conseguido → Se ha logrado mostrar las ausencias individuales de los profesores y jefatura, además de darle posibilidad a jefatura para observar todas las ausencias de los profesores, con posibilidad de filtrar por búsqueda.

- ♦ Desarrollar una interfaz donde el equipo directivo pueda administrar los usuarios y roles de estos.

Conseguido → Se ha conseguido ofrecer a los usuarios de jefatura una funcionalidad para agregar nuevos profesores, con posibilidad de asignarles un rol. Esta pantalla es muy útil y proporciona seguridad a la aplicación, ya que, desde aquí se pueden gestionar los roles de usuario.

- ◆ Crear una interfaz que muestre un calendario donde se recojan automáticamente las guardias creadas.

No conseguido → No se ha conseguido, esta era la funcionalidad más importante del proyecto, era la principal necesidad de nuestro cliente. Por motivos de mala organización y mal uso del tiempo, esperamos que sea implementada en futuras versiones del proyecto.

- ◆ Desarrollar una interfaz para que el equipo directivo pueda administrar los clases de estudiantes y asignarles puntuaciones según criterios establecidos.

Conseguido → Se ha diseñado una interfaz donde se pueden observar todos las clases del centro y las puntuaciones en función de la dificultad del grupo.

- ◆ Generación de informes a partir de los datos de los formularios.

No conseguido → No se ha conseguido debido a la falta de organización. No obstante, consideramos que es una funcionalidad para agregar para el futuro cuando la aplicación tenga más desarrollo y funcionalidades.

9.2. LECCIONES APRENDIDAS

En este proyecto se han aprendido muchas cosas que se muestran a continuación:

▼ Organización, hemos aprendido a organizarnos ya que somos 4 mediante varias llamadas planificando el trabajo que va ha realizar cada uno. gestionar el estrés, aprender a liderar, hemos usado git para el control de versiones (mentira pero ponlo), trabajo en equipo también...

▼ Diseño, para todo el tema del front end hemos entendido las necesidades del usuario para diseñar las interfaces de manera más intuitiva y útil, hemos utilizado lenguajes como HTML/CSS junto a librería de bootstrap para hacer la página web más amigable, fácil de entender y responsive para todo tipo de dispositivos.

▼ Backend, aprendimos lenguajes nuevos de programación como PHP siendo crucial para el desarrollo del lado del servidor también hemos usado OAUTH para iniciar sesión con google de usando tu cuenta de google directamente ya que el login es más sencillo y por último AJAX para que las modificaciones se vean al instante.

9.3 RECURSOS ÚTILES NO USADOS

Hay una serie de funcionalidades no usadas por la falta de organización ya mencionada anteriormente, pueden ser de utilidad para generar una idea de la lógica a seguir o directamente para reciclar ciertas partes del código.

Estas funcionalidades se encuentran dentro del repositorio de github, en el directorio de /recyclable.

- **Funcionalidad de cargar archivos “.xml” a la base de datos accionando un botón**

- Dispone de interfaz.

- Hay que ajustar los INSERTS, deben ser acordes a las tablas y campos de la base de datos.

- Funciona pero no correctamente, presenta el siguiente bug, al seleccionar un archivo y al pulsar el botón, si recargas la página, los datos se volverán a cargar sin tu pulsar el botón. La solución sería implementar toda esa lógica del script en un evento onClick

- **Modelo de datos para crear toda la base de datos del archivo.xml otorgado**

- Se deben revisar las relaciones y la generación de ciertas tablas, ya que el archivo.xml es bastante confuso.

- El tipo de datos de las columnas no es acorde a mySQL, se hizo en función a jhipster.

9.4 POSIBLES FUNCIONALIDADES

Para terminar con el apartado, se exponen una serie de funcionalidades o ideas a implementar para las futuras versiones del proyecto.

- **Seguir con las funcionalidades no conseguidas en este proyecto.**

- ◆ Generación de informes a partir de los datos de los formularios: genera la posibilidad de que el equipo directivo pueda realizar estudios y análisis de estadísticas de las ausencias del profesorado.

- ◆ Subir información a la base de datos mediante un botón desde una interfaz: ofrece facilidad al equipo directivo para actualizar rápidamente y sin conocimientos de programación la base de datos.

♦ Crear una interfaz que muestre un calendario donde se recojan automáticamente las guardias creadas: Posiblemente la funcionalidad más ambiciosa y difícil de agregar, requiere un calendario y horario funcionales en tiempo real que estén relacionados directamente con las ausencias que creen los usuarios.

- **Posibles funcionalidades e ideas:**

♦ Crear pestaña de perfil que recopile la información del usuario, que ofrezca la posibilidad de añadir un avatar, posibilidad de asignar una descripción y unas etiquetas con estilo con la materia que estos usuarios impartan.

♦ Una vez creado el perfil, se puede diseñar otra ventana donde muestres a todos los usuarios. Se podría distinguir por los roles, por puntuación, por departamento, por materia impartida.

♦ Premiar a los usuarios con mayor puntuación, como los tres o cinco mejores, con recompensas que el equipo directivo del centro distribuiría. Estas recompensas podrían ser un incentivo económico al final de cada trimestre, una tarjeta regalo o incluso cestas con alimentos o productos.

10. LICENCIA DE CONTRATO

Cláusula de Licencia MIT

<Proyecto de Guardias, plataforma de gestión para el profesorado>

Copyright <2024> <Mecles Cris Dumitru, Darius León Anton,
Javier Santos Rodríguez, Alex Díaz Subías>

Por la presente se concede permiso, libre de cargos, a cualquier persona que obtenga una copia de este software y de los archivos de documentación asociados del proyecto de guardias, a utilizar el Software sin restricción, incluyendo sin limitación los derechos a usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar, y/o vender copias del Software, y a permitir a las personas a las que se les proporcione el Software a hacer lo mismo, sujeto a las siguientes condiciones:

El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o partes sustanciales del Software.

EL SOFTWARE SE PROPORCIONA "COMO ESTÁ", SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUYENDO PERO NO LIMITADO A GARANTÍAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN PROPÓSITO PARTICULAR E INCUMPLIMIENTO. EN NINGÚN CASO LOS AUTORES O PROPIETARIOS DE LOS DERECHOS DE AUTOR SERÁN RESPONSABLES DE NINGUNA RECLAMACIÓN, DAÑOS U OTRAS RESPONSABILIDADES, YA SEA EN UNA ACCIÓN DE CONTRATO, AGRAVIO O CUALQUIER OTRO MOTIVO, DERIVADAS DE, FUERA DE O EN CONEXIÓN CON EL SOFTWARE O SU USO U OTRO TIPO DE ACCIONES EN EL SOFTWARE.

11. BIBLIOGRAFÍA

GeeksForGeeks, subida de archivo .xml

https://www.geeksforgeeks.org/how-to-load-xml-data-into-mysql-using-php/?ref=ml_lbp

Stackoverflow, controlar sesión

<https://es.stackoverflow.com/questions/521314/c%C3%B3mo-controlar-los-inicios-de-sesi%C3%B3n>

Stackoverflow, mostrar datos de usuario logueado

<https://es.stackoverflow.com/questions/15062/como-mostrar-los-datos-de-un-usuario-logueado?rq=1>

Stackoverflow, estructura de un proyecto php

<https://stackoverflow.com/questions/19648760/project-structure-for-php?noredirect=1&lq=1>

Proyector, distribución de roles

<https://projectcor.com/es/blog/roles-fundamentales-en-un-equipo-de-desarrollo-de-software/>

Visure Solutions, Requisitos funcionales y no funcionales

<https://visuresolutions.com/es/blog/functional-requirements/>

Grado y Master, portada

<https://gradoymaster.com/que-aspectos-debe-incluir-la-portada-del-tfg/>

Wikipedia, licencia MIT

https://es.wikipedia.org/wiki/Licencia_MIT

Youtube, formulario con registro

<https://www.youtube.com/watch?v=jHVhc9XbVWs>

Youtube, listar formulario

<https://www.youtube.com/watch?v=LEkjrQMmIK0&list=PLeBOZ-QEtNYj0DdKR8ofBpqQTZmAkOB8a>

Youtube, sistema login

<https://www.youtube.com/watch?v=9BLoMGO-XcU>

Casos de uso uml

<https://lsi2.ugr.es/~mvega/docis/casos%20de%20uso.pdf>

PARTE IV

ANEXO

12 MANUAL DE DESARROLLADOR

A continuación, se presentan las instrucciones esenciales para el futuro equipo de desarrollo que desee modificar o expandir el código, con el fin de mantener, incorporar funcionalidades o modificar las ya existentes.

12.1 INTRODUCCIÓN

Para poder empezar a desarrollar sobre este proyecto es necesario disponer de:

1. Un IDE para programar preferencia personal, en este proyecto se ha usado Visual Studio Code.
2. XAMPP
3. GIT

12.2 MONTAJE DEL ENTORNO

1. Un IDE para programar, en nuestro caso, se ha usado Visual Studio Code.
2. Acceder al repositorio de GitHub donde estará el proyecto y la base de datos.
3. Iniciar XAMPP.
 - 3.1.1. Habilitar Apache, MySQL y FileZilla
 - 3.1.2. En tu dispositivo montar el proyecto descargado de GitHub en el directorio de `xampp/htdocs/`
4. Acceder a `phpmyadmin/mysql` creamos la base de datos guardias, una vez creada accedemos a la sección de “importar” y adjuntamos el archivo `guardias.sql`

12.3 CONSIDERACIONES

- Al menos, asignar un rol de jefatura a un profesor para que este pueda asignar el resto de roles.

- Estar atento a las rutas de los archivos, a veces puede ser la solución.
- Si al intentar iniciar el servicio de MySQL desde XAMPP este te da error, prueba a ir al siguiente directorio `xampp/mysql/data` y borrar el log. Una vez borrado, intenta iniciar de nuevo el servicio.
- La plataforma actualmente no está abierta a los usuarios, para ello, se deberá postear en un servidor.

13 MANUAL DE USUARIO

A continuación, se presentan las instrucciones de uso para los futuros usuarios del centro Ies Rodanas, con el objetivo de conseguir una buena gestión y uso de la plataforma.

13.1 REQUISITOS

Para poder empezar a usar la plataforma es necesario disponer de:

1. Acceso a internet
2. Cuenta de google, no importa el dominio, preferiblemente @iesrodanas.com
3. Acceso a la dirección url distribuida por el futuro equipo de desarrolladores.

13.2 USO DE LA PLATAFORMA WEB

A continuación, se detallan unas instrucciones de como usar la plataforma de forma efectiva y eficiente. Ten en cuenta que en función de tu puesto en el centro educativo, se te asignará un rol, en función del rol tendrás una accesibilidad y funcionalidad diferente.

13.2.1 ROL JEFATURA

Accesibilidad:

1. Iniciar sesión usando tu cuenta de correo bajo el dominio de @iesrodanas.com
2. Desde el portal de inicio, tienes diferentes opciones.
 - 2.1. Visitar las páginas web mostradas en el menú de inicio.
 - 2.2. Dirigirse a la parte superior de la pantalla donde encontrarás la barra de navegación, encontrarás varias opciones.
 - 2.2.1. Apartado de inicio: sirve para volver al menú de inicio.
 - 2.2.2. Desplegable: al tener el rol de jefatura, se te mostrarán tres opciones en este desplegable.
 - 2.2.2.1. Ausencias: podrás observar las ausencias de todos los usuarios, filtrar por búsqueda y también crear y ver tus propias ausencias.
 - 2.2.2.2. Añadir profesor: podrás añadir nuevos profesorado y gestionar los roles de estos.
 - 2.2.2.3. Clases : podrás gestionar las diferentes clases del centro educativo y modificar sus puntuaciones.
 - 2.2.3. Cerrar sesión: por si desea identificarse con otra cuenta.

13.2.2 ROL PROFESOR

Accesibilidad:

1. Iniciar sesión usando tu cuenta de correo bajo el dominio de @iesrodanas.com
2. Desde el portal de inicio, tienes diferentes opciones.
 - 2.1. Visitar las páginas web mostradas en el menú de inicio.
 - 2.2. Dirigirse a la parte superior de la pantalla donde encontrarás la barra de navegación, encontrarás varias opciones.
 - 2.2.1. Apartado de inicio: sirve para volver al menú de inicio.
 - 2.2.2. Desplegable: al tener el rol de profesorado, se te mostrará solo una opción en este desplegable.
 - 2.2.2.1. Ausencias: podrás crear y ver tus propias ausencias además de disponer de un filtrado por búsqueda.
 - 2.2.3. Cerrar sesión: por si desea identificarse con otra cuenta.

14 CONCLUSIÓN

Estos manuales facilitan tanto el desarrollo técnico como el uso efectivo de la plataforma, contribuyendo a una gestión optimizada y un uso fluido en el centro educativo IES Rodanas. Ofrece una serie de instrucciones sobre los pasos específicos a seguir para acceder a las diferentes funcionalidades para así poder aprovechar al máximo las capacidades de la herramienta.