



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FINAL DE GRADO

Aplicación interna para gestionar grupos de trabajo

Autor:
Francisco ROGLÁ IRÚN

Supervisor:
Marc RAMOS VIDAL
Tutor académico:
Guillermo PERIS RIPOLLÉS

Fecha de lectura: 22 de Junio de 2023
Curso académico 2022/2023

Resumen

El presente documento recoge el Trabajo de Fin de Grado llevado a cabo durante la estancia en prácticas en la empresa *CYE Energía*.

Durante la estancia de prácticas se ha realizado un proyecto cuyo objetivo consiste en desarrollar una aplicación web para gestionar diferentes grupos de trabajo. Dentro de los grupos se pueden gestionar a su vez diferentes proyectos, así como todas las tareas que los formen. La aplicación también permite al usuario realizar un fichaje tanto de la entrada como de la salida del trabajo y consultar un calendario con todas las tareas realizadas durante una semana.

El proyecto se ha desarrollado siguiendo una metodología secuencial, según la cual la empresa indicaba únicamente la siguiente parte a realizar de la aplicación. Una vez desarrollada, la empresa revisaba el trabajo realizado, indicando posibles errores o mejoras y, una vez finalizado este proceso, indicaba la siguiente tarea a realizar.

Algunas de las tecnologías utilizadas durante el desarrollo del proyecto han sido: .NET Core y C# como lenguajes de programación de la parte de *back-end*, Visual Studio Code como entorno de programación de la parte de *front-end* y MySQL para gestionar la base de datos de la aplicación.

Palabras clave

Aplicación web, gestión de proyectos, desarrollo web, .NET Core.

Keywords

Web application, project management, web development, .NET Core.

Índice general

1. Introducción	15
1.1. Contexto.....	15
1.2. Descripción del proyecto	16
1.3. Objetivos del proyecto.....	16
1.4. Alcance del proyecto	18
1.4.1. Alcance funcional	18
1.4.2. Alcance organizativo	19
1.4.3. Alcance informático.....	19
1.5. Estructura de la memoria	19
2. Planificación del proyecto	21
2.1. Metodología.....	21
2.2. Planificación temporal	22
2.2.1. Planificación inicial	22
2.2.2. Planificación final.....	26
2.3. Tecnología.....	29
2.4. Estimación de costes.....	33
2.4.1. Costes humanos	33
2.4.2. Costes tecnológicos	34
2.4.3. Costes indirectos	35
2.4.4. Coste total	35

2.5. Análisis de riesgos.....	36
2.6. Seguimiento del proyecto.....	37
2.6.1. Primera quincena	37
2.6.2. Segunda quincena	39
2.6.3. Tercera quincena.....	40
2.6.4. Cuarta quincena	41
2.6.5. Quinta quincena	43
3. Análisis del sistema	47
3.1. Actores.....	47
3.2. Diagrama de casos de uso.	48
3.3. Requisitos de datos.....	62
4. Diseño de la base de datos	65
4.1. Diseño lógico.....	65
4.2. Diseño físico.....	69
5. Diseño e implementación del sistema	75
5.1. Detalles de la implementación	75
5.2. Diseño de las páginas.....	84
5.2.1. Página de registro.....	84
5.2.2. Página de inicio de sesión	85
5.2.3. Páginas de cambio de contraseña.....	86
5.2.4. Página de inicio.....	88
5.2.5. Página de perfil de usuario.....	90
5.2.6. Página de grupos	92

5.2.7. Página de proyectos	94
5.2.8. Página de tareas	96
5.2.9. Página de calendario	103
5.2.10. Página de fichaje	105
5.3. Verificación y validación	107
6. Conclusiones	111
6.1. Posibles mejoras	111
6.2. Valoración personal	112
Bibliografía	113
A. Correos automáticos enviados	115

Índice de figuras

1.1	Logo de Cye Energía	15
2.1	Esquema tradicional de una metodología secuencial.	22
2.2	Diagrama de Gantt inicial.....	25
2.3	Diagrama de Gantt final	28
2.4	Logotipos de Visual Studio y Visual Studio Code	30
2.5	Muestra del repositorio del back-end desde SourceTree.	30
2.6	Muestra de la base de datos junto a todas sus tablas desde SQLyog.	31
2.7	Muestra del tablón de Trello utilizado en el proyecto.	32
3.1	Diagrama de casos de uso del proyecto.	49
4.1	Diseño lógico de la base de datos.	66
5.1	Estructura de ficheros del proyecto de front-end.	76
5.2	Estructura de la carpeta vendor.....	77
5.3	Estructura de la carpeta js.....	77
5.4	Estructura del back-end [I].....	78
5.5	Estructura del back-end [II].....	78
5.6	Estructura del back-end [III].....	78
5.7	Página de registro.	85
5.8	Página de inicio de sesión.	86

5.9	Página de olvido de contraseña.....	87
5.10	Página de cambio de contraseña.....	87
5.11	Página de inicio.....	88
5.12	Barra lateral completa.....	89
5.13	Barra lateral contraída.....	89
5.14	Barra superior.....	89
5.15	Desplegable de la barra superior.....	90
5.16	Pie de página.....	90
5.17	Perfil de usuario.....	91
5.18	Perfil de usuario con la edición de datos habilitada.....	91
5.19	Página de grupos.....	92
5.20	Pestaña para crear grupos.....	92
5.21	Botones con las acciones de cada grupo.....	93
5.22	Pestaña para gestionar miembros del grupo.....	93
5.23	Pestaña para editar grupos.....	94
5.24	Página de proyectos en caso de ser Administrador.....	95
5.25	Página de proyectos en caso de ser Desarrollador o Usuario.....	96
5.26	Página de tareas en caso de ser Administrador o Desarrollador.....	97
5.27	Pestaña para crear tareas.....	97
5.28	Pestaña de la tarea en caso de ser Administrador o Desarrollador.....	98
5.29	Pestaña para registrar horas realizadas en una tarea.....	99
5.30	Pestaña para gestionar etiquetas.....	99
5.31	Pestaña para crear etiquetas.....	100
5.32	Pestaña para gestionar responsables.....	100

5.33	Pestaña para gestionar documentos adjuntos.	100
5.34	Pestaña para adjuntar un documento.	101
5.35	Página de tareas en caso de ser Usuario.	102
5.36	Pestaña de la tarea en caso de ser Usuario.	102
5.37	Botones de la pestaña de tareas.	103
5.38	Página de calendario.	103
5.39	Muestra de muchos registros en un calendario.	104
5.40	Pestaña para registrar horas realizadas en una tarea desde el calendario.	105
5.41	Página de fichaje.	106
5.42	Validadores de campos obligatorios vacíos.	107
5.43	Validadores de formato de correo erróneo.	108
5.44	Aviso de cambio realizado con éxito.	108
5.45	Confirmación para borrar un elemento.	108
5.46	Confirmación para borrar un grupo o proyecto.	109
5.47	Confirmación para cerrar sesión.	109
A.1	Correo de bienvenida a la aplicación.	115
A.2	Correo necesario para el cambio de contraseña.	116
A.3	Correo de invitación a un grupo.	116

Índice de tablas

2.1. Planificación inicial del proyecto.....	24
2.2. Planificación final del proyecto	27
2.3. Costes humanos del proyecto.....	34
2.4. Costes tecnológicos del proyecto.	35
2.5. Coste total del proyecto.....	36
2.6. Riesgos del proyecto.	37
3.1. Relaciones entre actores y casos de uso.....	50
3.2. Caso de uso CU01 - Iniciar o cerrar sesión en la aplicación.....	51
3.3. Caso de uso CU02 - Registrarse en la aplicación.....	52
3.4. Caso de uso CU03 - Cambiar de contraseña.	52
3.5. Caso de uso CU04 - Editar el perfil del usuario.....	53
3.6. Caso de uso CU05 - Crear un grupo de trabajo.....	53
3.7. Caso de uso CU06 - Editar o borrar un grupo.	54
3.8. Caso de uso CU07 - Invitar usuarios a un grupo.	55
3.9. Caso de uso CU08 - Eliminar un miembro del grupo.	55
3.10. Caso de uso CU09 - Crear, editar o borrar un proyecto.	56
3.11. Caso de uso CU10 - Crear, editar, borrar, mover o archivar una tarea.	57
3.12. Caso de uso CU11 - Crear, seleccionar o borrar etiquetas.	58
3.13. Caso de uso CU12 - Seleccionar responsables de una tarea.	58
3.14. Caso de uso CU13 - Subir, cargar o eliminar un adjunto.	59

3.15. Caso de uso CU14 - Registrar horas trabajadas en una tarea.....	60
3.16. Caso de uso CU15 - Fichar entrada o salida al trabajo.	60
3.17. Caso de uso CU16 - Descargar el registro de fichajes.....	61
3.18. Caso de uso CU17 - Consultar fichajes, pudiendo filtrar los fichajes mostrados.	61
3.19. Caso de uso CU18 - Consultar calendario, pudiendo buscar el calendario de una fecha concreta.....	62
3.20. Requisito de datos RD01 – Iniciar sesión y cambio de contraseña.	62
3.21. Requisito de datos RD02 – Registrarse en la aplicación.	63
3.22. Requisito de datos RD03 – Editar el perfil de usuario.	63
3.23. Requisito de datos RD04 – Crear o editar un grupo.	63
3.24. Requisito de datos RD05 – Invitar usuarios.	63
3.25. Requisito de datos RD06 – Crear o editar un proyecto.	63
3.26. Requisito de datos RD07 – Crear o editar una tarea.	64
3.27. Requisito de datos RD08 – Crear etiquetas.	64
3.28. Requisito de datos RD09 – Subir un adjunto.	64
3.29. Requisito de datos RD10 – Registrar horas desde la página de tareas.	64
3.30. Requisito de datos RD11 – Registrar horas desde la página de calendario.	64

Índice de códigos fuente

4.1	Diseño Físico - Grupo	69
4.2	Diseño físico – Proyecto	70
4.3	Diseño físico – Tarea	70
4.4	Diseño físico – Usuario	70
4.5	Diseño físico – Usuario_grupo	71
4.6	Diseño físico – Usuario_tarea	71
4.7	Diseño físico – Etiqueta.....	71
4.8	Diseño físico – Etiqueta_tarea	72
4.9	Diseño físico – Registro_horas_tarea	72
4.10	Diseño físico – Registro_fichaje	72
4.11	Diseño físico – Documento_adjunto.....	73
4.12	Diseño físico – Vacaciones	73
5.1	Encriptado de la contraseña.	79
5.2	Comparación de la contraseña introducida con la encriptada.....	79
5.3	Construcción de un correo automático.	80
5.4	Generación de los tokens al iniciar sesión.	81
5.5	Métodos utilizados para generar los tokens.....	81
5.6	Authorize para asegurar poder ejecutar la operación.....	82
5.7	Tratamiento del refreshToken.	83
5.8	Llamada al refreshToken.....	84

Capítulo 1

Introducción

1.1. Contexto

Cye Energía [1] es una empresa proveedora de energía eléctrica que se encarga de comprar la energía a las grandes distribuidoras para vendérsela a los clientes, tanto a particulares como a empresas. Uno de los principios de la empresa es dedicarse principalmente a las energías renovables y energías verdes. En la Figura 1.1 se muestra su logo.



Figura 1.1: Logo de Cye Energía

La empresa nació en 2006 como *Ensoval* y se dedicaba principalmente a crear instalaciones de energías renovables. En 2009, la empresa se expandió y fundó una filial en Italia. Años después, en 2013, nació *Cye Energía*, centrándose en la comercialización de energía eléctrica.

Cye Energía tiene su sede principal en Valencia y, además, cuenta con otra sede en Castellón de la Plana. El equipo de trabajo está formado por alrededor de una veintena de trabajadores, repartidos entre las diferentes sedes, y su ámbito de trabajo esencialmente es nacional, aunque también realizan operaciones en Italia.

1.2. Descripción del proyecto

El proyecto descrito en esta memoria consiste en el desarrollo de una aplicación web propia que permite gestionar diferentes grupos de trabajo.

La finalidad principal que se quería conseguir con el desarrollo de la aplicación era poder facilitar la gestión de los proyectos que tenga un grupo de trabajo, agrupando en una misma aplicación muchos aspectos que la empresa consideraba importantes para el desarrollo de un proyecto.

Esta aplicación está pensada para que cualquier persona se pueda registrar en ella y pueda utilizarla. Los usuarios pueden crear grupos de trabajo o unirse a grupos ya existentes. Los grupos de trabajo pueden tener diferentes proyectos, los cuales a su vez están formados por tareas. Los usuarios de la aplicación pueden gestionar tanto los grupos como sus diferentes proyectos y todas las tareas que los formen.

Otro aspecto que permite realizar la aplicación es la gestión temporal de los proyectos realizados. Los miembros del grupo pueden registrar el tiempo dedicado a realizar una tarea. Estos registros se ven reflejados en un calendario que permite ver todo el trabajo realizado por un usuario.

Además, se ha realizado también un sistema de fichaje con el cual cada miembro del grupo puede fichar tanto la entrada como la salida del trabajo.

Para poder llevar a cabo el desarrollo de este proyecto, se ha debido diseñar una base de datos que permite cargar y guardar todos los datos de la aplicación, así como diseñar todas las páginas de la aplicación junto con todas las funcionalidades que éstas incluyen.

1.3. Objetivos del proyecto

En este apartado se van a detallar todos los objetivos del proyecto.

El objetivo principal de este proyecto es desarrollar una aplicación web que permita a grupos de trabajo gestionar diferentes proyectos.

Este objetivo se puede descomponer a su vez en los siguientes subobjetivos:

- Diseñar y crear una base de datos para almacenar y gestionar todos los datos de la aplicación.
- Diseñar e implementar las diferentes interfaces de la aplicación.
- Diseñar e implementar los diferentes módulos y formularios requeridos en la aplicación. Gracias a éstos se deben cumplir los siguientes objetivos:
 - Permitir registrar nuevos usuarios en la aplicación e iniciar sesión a los ya registrados.
 - Los usuarios han de poder crear grupos de trabajo o unirse a grupos ya existentes mediante una invitación del administrador del grupo.
 - Definir los proyectos de trabajo que realiza un grupo, así como las tareas que los forman y quién realiza cada tarea.
 - Poder adjuntar documentos a cada tarea.
 - Permitir registrar las horas dedicadas a cada tarea en un día de trabajo.
 - Definir un periodo de vacaciones para un trabajador del grupo.
 - Garantizar que los usuarios puedan realizar todas las operaciones para las que tengan permiso e impedir que hagan aquellas para las que no tengan el permiso necesario.
 - Garantizar la seguridad de las contraseñas de los usuarios realizando un proceso de encriptado.
- Realizar pruebas para comprobar el correcto funcionamiento de la aplicación desarrollada.
- Conseguir un sistema propio de fichaje con el que se puedan tramitar todas las entradas y salidas de los trabajadores sin depender de ninguna entidad externa para ello.

1.4. Alcance del proyecto

1.4.1. Alcance funcional

El proyecto a desarrollar debe cumplir las siguientes funcionalidades:

- La aplicación debe permitir registrar nuevos usuarios a través de un correo electrónico y, a su vez, permitir a los ya registrados iniciar y cerrar sesión en su cuenta.
- Cualquier usuario debe poder ver los grupos de los que forma parte y crear grupos de trabajo nuevos. En el caso de haber creado un grupo, el usuario debe poder editarlo, borrarlo e invitar a cualquier otro usuario a dicho grupo.
- Cualquier usuario debe poder ver los proyectos de un grupo de trabajo. En el caso de haber creado el grupo, también debe poder añadir nuevos proyectos y editar o borrar los ya existentes.
- Desde un proyecto se debe poder acceder a la lista de tareas que lo componen. Se debe poder diferenciar claramente si las tareas están pendientes de desarrollo, en desarrollo o ya finalizadas.
- Cada tarea debe poder tener etiquetas, que se gestionan desde la misma tarea; responsables, que se eligen de entre los miembros del grupo de la tarea; y debe permitir adjuntar documentos.
- La aplicación debe tener una página con un calendario semanal en el que se vean reflejados todos los registros realizados por el usuario. Se debe poder cambiar de semana si el usuario lo desea.
- Los usuarios deben poder registrar el número de horas dedicadas a una tarea desde la misma página de la tarea o desde la página del calendario, indicando la tarea de la que se quiere realizar el registro.
- Los usuarios deben poder fichar la entrada y la salida al trabajo desde la aplicación desarrollada.

1.4.2. Alcance organizativo

En cuanto al alcance organizativo, la aplicación está pensada para que sea utilizada tanto a nivel interno de la empresa como a nivel externo. Cualquiera debe poder registrarse y utilizar la aplicación, por lo que se espera que el número de usuarios sea elevado.

1.4.3. Alcance informático

Desde el punto de vista informático, para alojar todos los datos de la aplicación se debe diseñar e implementar primero una base de datos. Esta base de datos estará en un servidor, el cual se alojará en un *Docker* [3].

Para almacenar y gestionar los documentos adjuntos que los usuarios incluyan en las tareas de la aplicación, se va a utilizar el servicio de *Amazon Web Services* (AWS) [4], el cual ya es utilizado por la empresa habitualmente.

Se implementarán algoritmos de cifrado para el paso de las contraseñas de los usuarios desde la aplicación al servidor. También se implementará un sistema de *tokens* para garantizar que los usuarios sólo puedan realizar las operaciones para las que tengan permiso.

1.5. Estructura de la memoria

El contenido del presente documento está organizado en una serie de capítulos, los cuales se describen a continuación.

El primer capítulo, el actual, es la introducción. Contiene una breve introducción a la empresa y el proyecto a desarrollar, junto con sus objetivos y su alcance.

El segundo capítulo explica la planificación del proyecto. Incluye la metodología y las tecnologías empleadas, la planificación temporal del proyecto, los costes y riesgos

de éste y, además, un seguimiento de todo el desarrollo de la aplicación desde el inicio hasta el final de la estancia.

El tercer capítulo contiene el análisis del sistema, donde se explican los diferentes roles de usuario que hay en la aplicación y el diagrama de casos de uso.

El cuarto capítulo incluye el diseño de la base de datos, que se compone de diseño conceptual, lógico y físico.

El quinto capítulo detalla el diseño y la implementación del sistema, explicando en detalle cada una de las páginas de la aplicación desarrolladas en el proyecto. También se detallan en este capítulo las medidas tomadas en cuanto a la validación y verificación de los datos.

En el sexto capítulo se encuentra la conclusión, la cual incluye una valoración personal del trabajo realizado y posibles mejoras del proyecto.

Posteriormente a los capítulos se incluye la bibliografía y un anexo, que muestra los diferentes tipos de correos automáticos enviados desde la aplicación.

Capítulo 2

Planificación del proyecto

2.1. Metodología

La metodología que se ha seguido durante el proyecto es una metodología secuencial. Según esta metodología, el supervisor de la empresa indicaba la tarea a realizar y, hasta que ésta no se finalizase, no indicaba cuál era la siguiente.

A partir del momento en el que se indicaba una tarea, comenzaba el desarrollo de ésta hasta completarla con éxito. Una vez completada la tarea, era revisada por el supervisor. En caso de estar correcta, indicaba la siguiente tarea y, en caso de haber algún error o que considerara que había que modificar o añadir algún detalle, lo indicaba y hasta que no estuviera todo corregido no se avanzaba a la siguiente tarea.

En todo momento durante el desarrollo de la tarea se podía testear el trabajo realizado para poder comprobar el correcto funcionamiento de éste.

En caso de haber cualquier problema o duda, el supervisor de la empresa estaba presente para poder ayudar, ya fuera para explicar el funcionamiento de alguna tecnología que debía utilizar o para explicar la causa del problema.

Se escogió esta metodología porque es la metodología que utilizaba la empresa en todos sus proyectos.

En la Figura 2.1 se muestra un esquema básico tradicional siguiendo esta metodología.

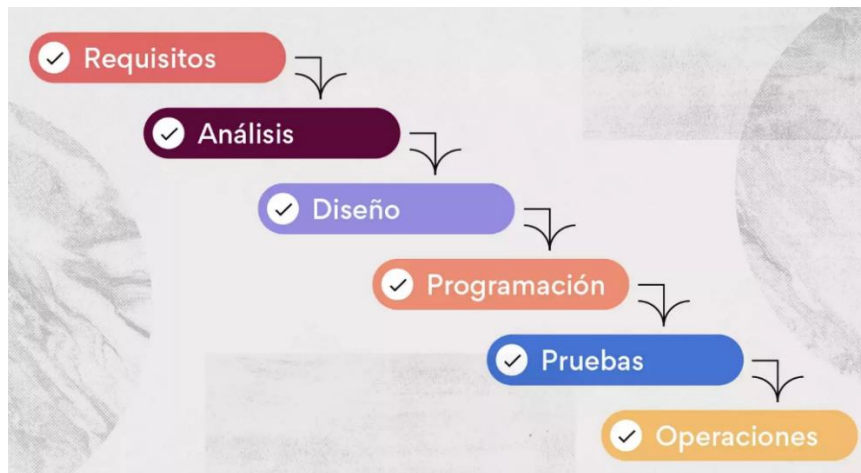


Figura 2.1: Esquema tradicional de una metodología secuencial. [16]

2.2. Planificación temporal

Esta sección muestra la planificación temporal del proyecto y se divide en dos secciones: la planificación inicial, realizada al comenzar la estancia de prácticas, y la planificación final, realizada al finalizar el proyecto.

2.2.1. Planificación inicial

En el proyecto no se realizaron reuniones como tal de ningún tipo y, debido a ser secuencial, no se pudo tener inicialmente una planificación temporal muy detallada. No obstante, se realizó una planificación genérica de lo que sería, en teoría, la estructura del proyecto, la cual se presenta en la Tabla 2.1.

Las tareas definidas inicialmente fueron las siguientes:

- **Explicación de las tecnologías a utilizar, creación de cuentas e inicialización de diferentes aplicaciones:** Esta primera tarea estaba centrada en la explicación de todas las tecnologías que se iban a utilizar durante la estancia, así como la creación de cuentas en algunas aplicaciones que se iban a emplear.
- **Diseño de la base de datos y creación de documentos de diseño:** Esta tarea estaba centrada en realizar el diseño de la base de datos y su correspondiente documentación.

- **Creación de los proyectos de *back-end* y *front-end*:** El desarrollo del código de la parte de *back-end* y de *front-end* se iba a realizar en dos proyectos por separado. Esta tarea estaba dedicada a la creación de cada uno de esos proyectos en sus diferentes entornos de desarrollo.
- **Creación del servidor y de la base de datos:** Tarea dedicada a la creación de la base de datos a partir del diseño realizado. Esta base de datos debía alojarse en un servidor, el cual se debía crear primero.
- **Implementación de la página de inicio de sesión de los usuarios, creación de los *tokens* y encriptado de las contraseñas:** Tarea enfocada al desarrollo de la página de inicio de sesión.
- **Implementación de la página de registro de usuarios y de olvido de contraseña:** Tarea enfocada al desarrollo de las páginas de registro y de olvido de contraseña.
- **Implementación de las páginas de los grupos, proyectos y tareas:** Tarea enfocada al desarrollo de las páginas de grupos, proyectos y tareas.
- **Implementar la página del calendario:** Tarea enfocada al desarrollo de la página del calendario.
- **Implementar la página de las vacaciones:** Tarea enfocada al desarrollo de la página para registrar las vacaciones.
- **Implementar la página de rellenar horas trabajadas:** Tarea enfocada al desarrollo de la página para registrar las horas trabajadas.
- **Implementar la página para controlar el fichaje:** Tarea enfocada al desarrollo de la página para realizar los fichajes.
- **Implementar la página de inicio de la aplicación:** Tarea enfocada al desarrollo de la página inicial de la aplicación.
- **Finalizar el diseño de la aplicación:** Tarea dedicada a finalizar todo el diseño de la aplicación
- **Realizar pruebas finales para comprobar el correcto funcionamiento de todo lo desarrollado:** Esta tarea consistía en realizar pruebas para detectar posibles errores y corregirlos.

Explicación de las tecnologías a utilizar, creación de cuentas e inicialización de diferentes aplicaciones	10 horas
Diseño de la base de datos y creación de documentos de diseño	15 horas
Creación de los proyectos de back-end y front-end	8 horas
Creación del servidor y de la base de datos	20 horas
Implementación de la página de inicio de sesión de los usuarios, creación de los tokens y encriptado de las contraseñas	40 horas
Implementación de la página de registro de usuarios y de olvido de contraseña	15 horas
Implementación de las páginas de los grupos, proyectos y tareas	60 horas
Implementar la página del calendario	15 horas
Implementar la página de las vacaciones	25 horas
Implementar la página de rellenar horas trabajadas	35 horas
Implementar la página para controlar el fichaje	30 horas
Implementar la página de inicio de la aplicación	7 horas
Finalizar el diseño de la aplicación	15 horas
Realizar pruebas finales para comprobar el correcto funcionamiento de todo lo desarrollado	5 horas
Total	300 horas

Tabla 2.1. Planificación inicial del proyecto

A partir de esta planificación se desarrolló el diagrama de Gantt que se muestra en la Figura 2.2.

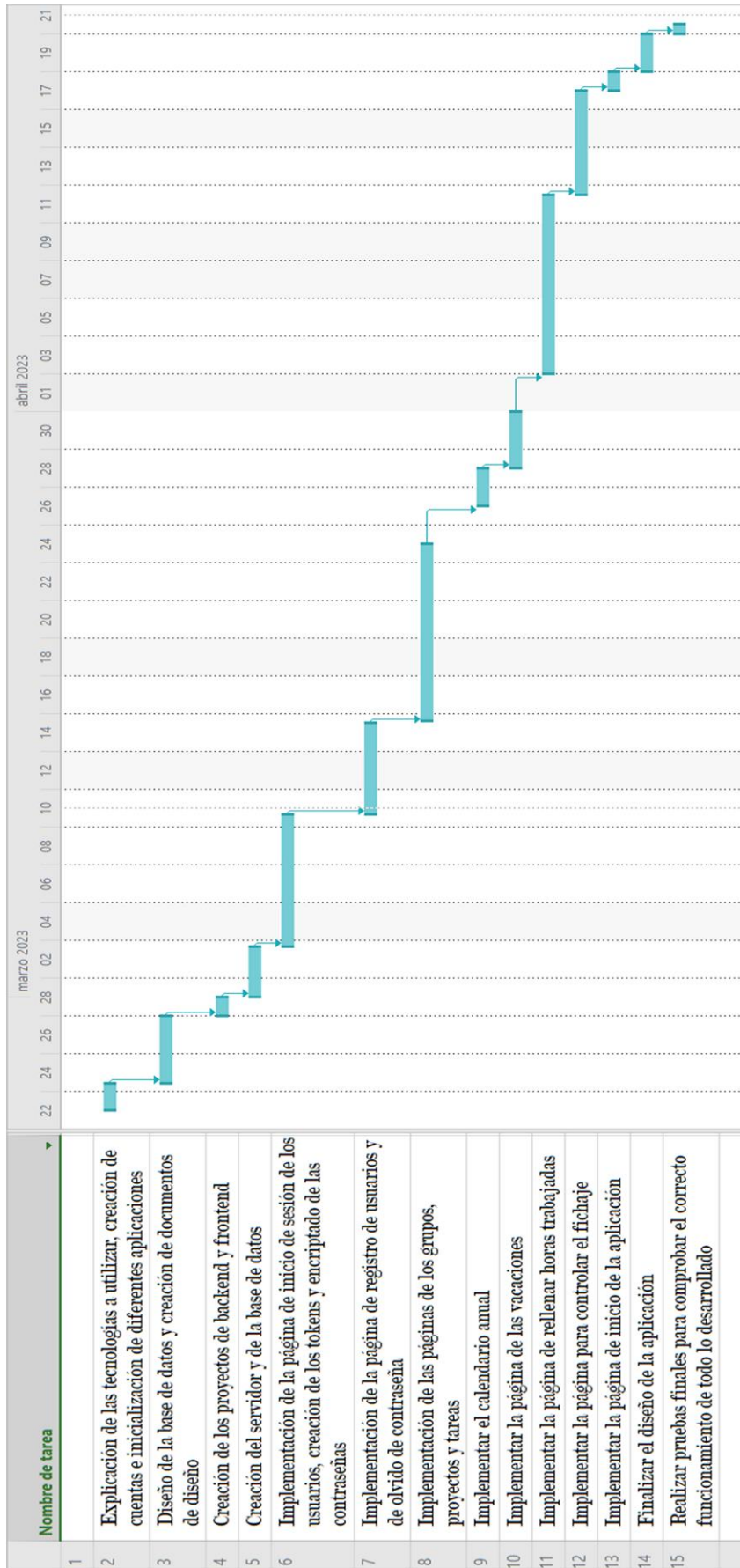


Figura 2.2: Diagrama de Gantt inicial

2.2.2. Planificación final

En esta sección se muestra cómo fue la planificación real del proyecto, una vez finalizada la estancia de prácticas.

Como se puede observar en la Tabla 2.2, hubo cambios durante la estancia a la hora de realizar las tareas planificadas inicialmente. La duración de algunas tareas varió un poco respecto a lo planificado, no obstante, también hubo otros cambios más destacables, los cuales son los siguientes:

- Inicialmente se había previsto más horas para la creación de los proyectos, el servidor y la base de datos, pero se consiguió realizar en mucho menos tiempo.
- La implementación de las páginas de los grupos, proyectos y tareas costó finalmente el doble de lo previsto. Este aumento se debió principalmente a la página de tareas, la cual tuvo un diseño mucho más elaborado del previsto inicialmente y contó con más funcionalidades de las planificadas.
- Finalmente no se creó ninguna página para rellenar las horas trabajadas, sino que esto se añadió como una funcionalidad que se podía realizar desde la página de tareas. Se podría decir que la tarea de *“Implementar la página de rellenar horas trabajadas”* fue absorbida por la tarea *“Implementación de las páginas de los grupos, proyectos y tareas”*.
- Tampoco se desarrolló la página de las vacaciones, pero en este caso fue por falta de tiempo. Esta página se consideraba más prescindible que las demás y se dejó para el final. Una vez finalizada la página de fichaje y el perfil del usuario, se decidió dedicar el tiempo restante de la estancia a añadir nuevas funcionalidades a las páginas ya desarrolladas y solucionar errores.
- El perfil del usuario y la funcionalidad de cerrar sesión no se contaron en la planificación inicial, pero eran aspectos que no podían faltar en la aplicación y se acabaron desarrollando.

Explicación de las tecnologías a utilizar, creación de cuentas e inicialización de diferentes aplicaciones	10 horas
Diseño de la base de datos y creación de documentos de diseño	10 horas
Creación de los proyectos de back-end y front-end	2 horas
Creación del servidor y de la base de datos	3 horas
Implementación de la página de inicio de sesión de los usuarios, creación de los tokens y encriptado de las contraseñas	30 horas
Implementación de la página de registro de usuarios y de olvido de contraseña	30 horas
Implementación de las páginas de los grupos, proyectos y tareas	120 horas
Implementar la página del calendario	30 horas
Implementar la página para controlar el fichaje	25 horas
Implementar página de inicio, perfil de usuario y cerrar sesión	10 horas
Adición de nuevas funcionalidades y corrección de errores	30 horas
Total	300 horas

Tabla 2.2. Planificación final del proyecto

A partir de esta nueva planificación se obtuvo el diagrama de Gantt mostrado en la Figura 2.3.

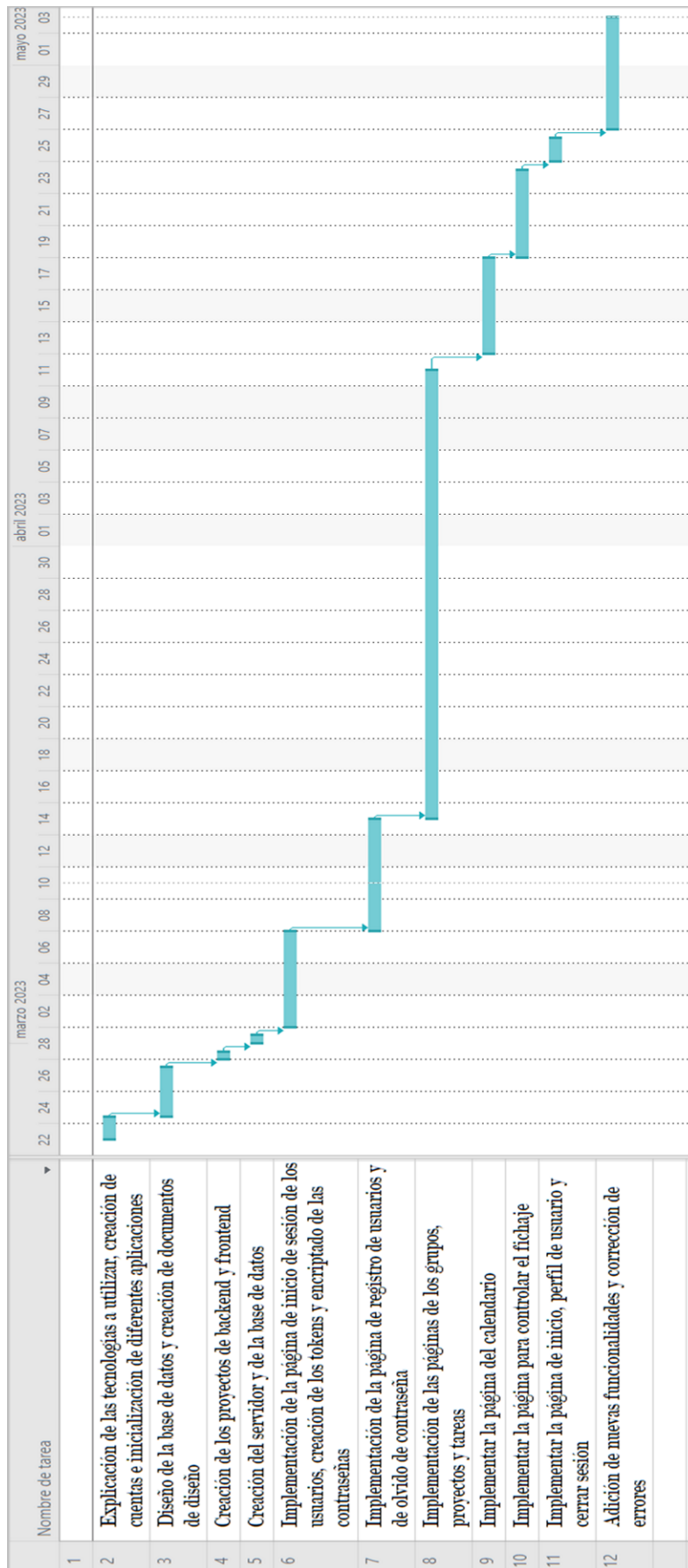


Figura 2.3: Diagrama de Gantt final

2.3. Tecnología

Durante el desarrollo del proyecto se utilizaron diferentes lenguajes y tecnologías. A continuación, se ofrece un listado de todas ellas:

Visual Studio

Se usó para desarrollar todo el código del *back-end*, utilizando los lenguajes *.NET Core* y *C#*.

Visual Studio (Figura 2.4) es un entorno de desarrollo diseñado por Microsoft compatible con lenguajes como C++, C#, Java, Python, PHP y .NET. Se puede utilizar para editar, depurar y compilar código y se suele utilizar al desarrollar aplicaciones.

.NET Core es una framework gratuito de código abierto diseñado por Microsoft para compilar muchos tipos de aplicaciones diferentes.

C# es un lenguaje de programación diseñado como parte de .NET. Su sintaxis deriva de los lenguajes C y C++.

Visual Studio Code

Se utilizó para desarrollar todo el código de la parte del *front-end*, utilizando los lenguajes *HTML* y *JavaScript*.

Visual Studio Code (Figura 2.4) es un entorno similar a Visual Studio pero dedicado principalmente para desarrolladores web, por lo que utiliza lenguajes como HTML y JavaScript.

HTML (*HyperText Markup Language*) es el lenguaje de marcas estándar para el desarrollo de páginas web y ha sido desarrollado por el *World Wide Web Consortium* (W3C). [6]

JavaScript es un lenguaje de programación interpretado, orientado a objetos y basado en el estándar ECMAScript. Permite desarrollar scripts para páginas web.

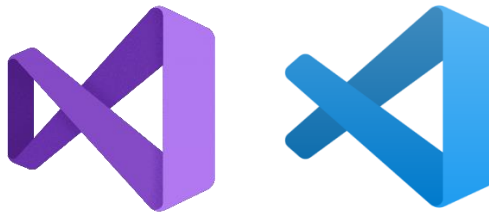


Figura 2.4: Logotipos de Visual Studio y Visual Studio Code. [5]

GitLab

Para alojar el código tanto de la parte de *front-end* como de la parte de *back-end* se empleó GitLab. Se utilizó un repositorio distinto para almacenar cada una de las partes.

GitLab es una plataforma que permite alojar y compartir código y utiliza el sistema de control de versiones Git [7].

SourceTree

Para actualizar el código almacenado en GitLab y realizar las operaciones Git de *Pull*, *Commit* y *Push*, se empleó la aplicación de SourceTree (Figura 2.5).

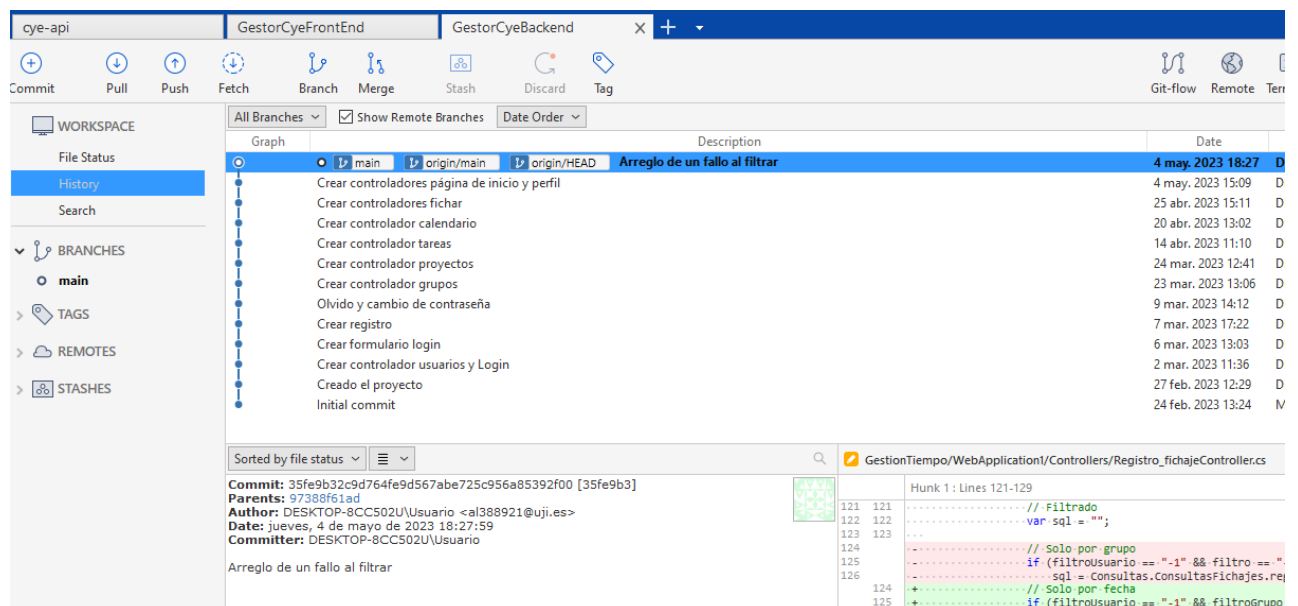


Figura 2.5: Muestra del repositorio del back-end desde SourceTree.

Docker Desktop

Para tener un servidor en el que poder almacenar la base de datos de la aplicación, se utilizó un *Docker*. Éste se manejaba e inicializaba desde la aplicación Docker Desktop.

Un *Docker* [3] es una tecnología que permite ejecutar un proceso dentro de lo que se conoce como un contenedor de software. Estos procesos se ejecutan de manera aislada, manteniendo todas sus dependencias en dicho contenedor.

SQLyog

Para manejar la base de datos, crear y manipular las tablas, así como ver sus contenidos, se utilizaba la aplicación SQLyog (Figura 2.6). La base de datos empleada utilizaba MySQL.

MySQL [8] es uno de los sistemas de gestión de bases de datos más populares y está desarrollado por Oracle. Las bases de datos que utilizan MySQL son relacionales, es decir, almacenan todos sus datos en diferentes tablas, las cuales se pueden relacionar entre sí. [9]

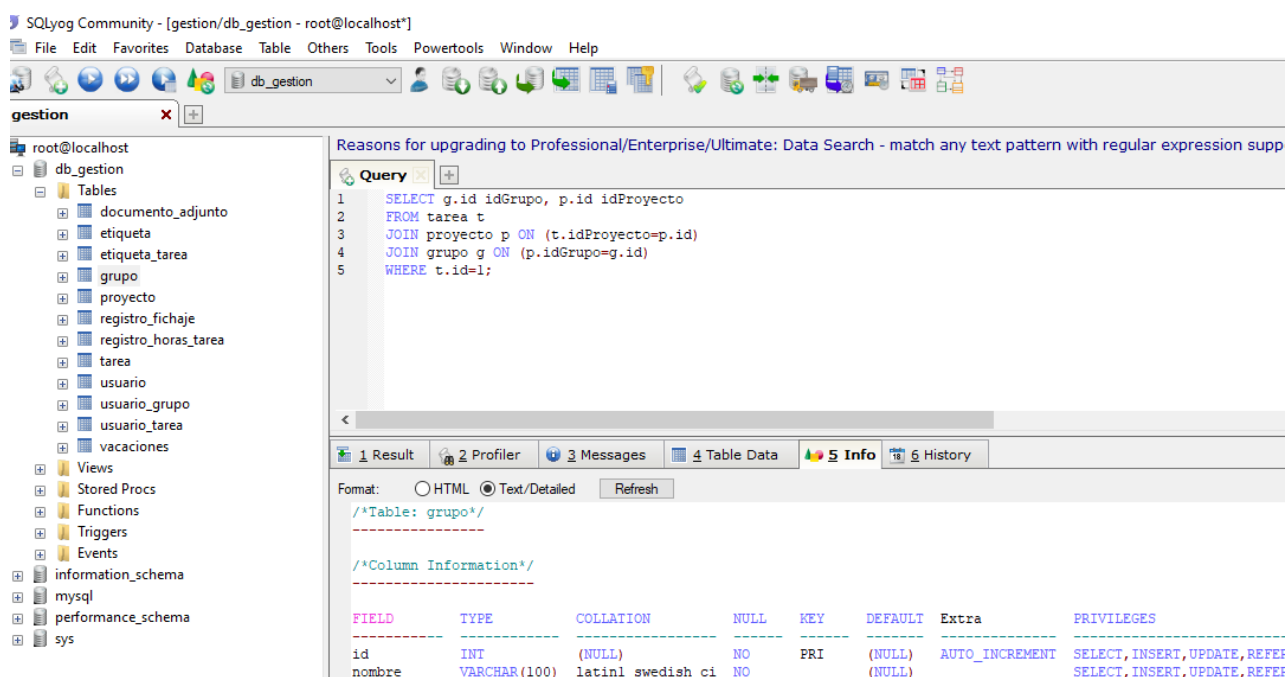


Figura 2.6: Muestra de la base de datos junto a todas sus tablas desde SQLyog.

Postman

Para poder simular la interacción entre el cliente y el servidor de la aplicación y poder ver el correcto funcionamiento de todas las operaciones *GET*, *PUT*, *POST* y *DELETE*, se utilizó la aplicación Postman.

Swagger

Para poder comprobar el correcto funcionamiento de los métodos y las operaciones desarrolladas en la parte del *back-end*, se implementó un Swagger.

Trello

Para realizar la planificación del proyecto e indicar los pasos que se estaban desarrollando en cada momento, así como los ya completados y los siguientes pasos a realizar, se utilizó la aplicación Trello. Esta aplicación muestra todas las tareas en un tablón tal como se muestra en la Figura 2.7.

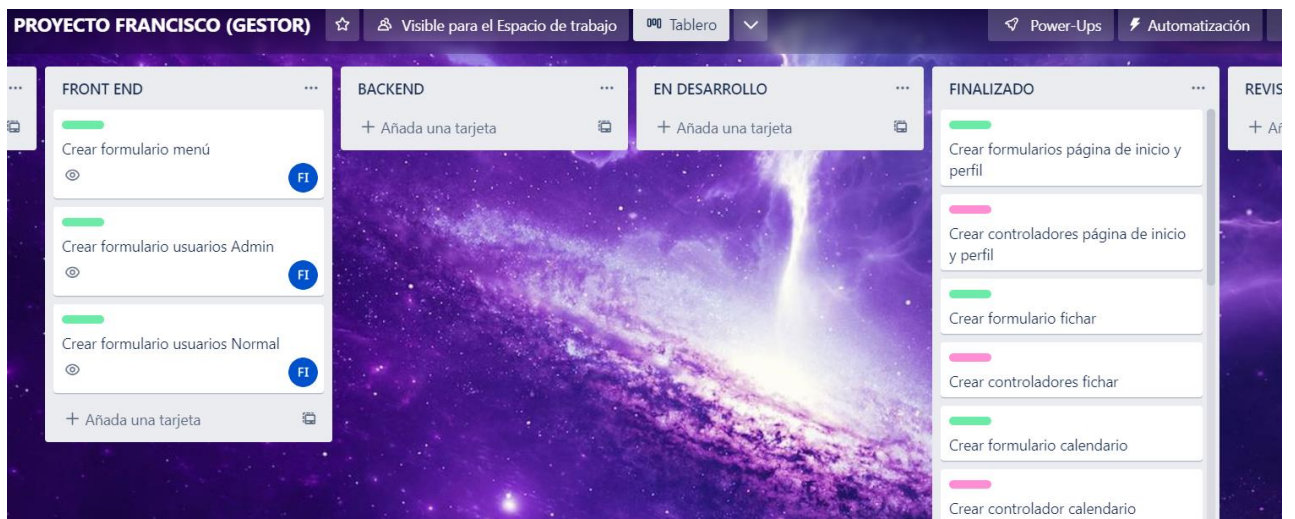


Figura 2.7: Muestra del tablón de Trello utilizado en el proyecto.

Slack

Para facilitar la comunicación entre el equipo de la empresa y mandar mensajes se empleó la aplicación Slack, aplicación que utiliza habitualmente la empresa.

Amazon Web Services (AWS)

Para gestionar todos los documentos que se adjuntasen en la aplicación se empleaba Amazon Web Services. Este servicio de Amazon permite almacenar y cargar todos los documentos adjuntos, además de otras funcionalidades.

2.4. Estimación de costes

Para realizar la correcta estimación del coste total del proyecto, se van a diferenciar tres tipos de costes: costes humanos, costes tecnológicos y costes indirectos.

2.4.1. Costes humanos

Los costes humanos se calculan a partir de todos los trabajadores implicados en el proyecto, teniendo en cuenta el número de horas que ha dedicado cada uno y su puesto de trabajo.

En el proyecto que recoge la presente memoria únicamente han participado dos trabajadores: el estudiante de prácticas y el supervisor de la empresa.

Por un lado, el estudiante ha realizado 300 horas de trabajo, mientras que el supervisor, considerando que ha dedicado una hora diaria de su jornada laboral a este proyecto, que se ha realizado en 43 días, ha realizado 43 horas de trabajo.

Según Glassdoor, el salario medio de un programador en España a fecha de Mayo de 2023 es de 25.400€ anuales [10] y el de un programador júnior en España para esa misma fecha es de 19.000€ anuales [11].

Teniendo en cuenta que el año 2023 tiene 52 semanas y que se trabaje una media de 8 horas al día, 5 días a la semana, se puede calcular el salario por hora a partir del salario anual de la siguiente manera:

$$\text{Salario por hora} = \text{Salario anual} / (52 \text{ semanas} * 5 \text{ días/semana} * 8 \text{ horas/día})$$

Teniendo en cuenta los anteriores datos se obtienen los costes humanos del proyecto, mostrados en la Tabla 2.3, los cuales ascienden a 3.264 €.

Puesto de trabajo	Coste anual	Coste/Hora	Horas	Coste total
Programador	25.400 €	12,21 €	43 h	525 €
Programador júnior	19.000 €	9,13 €	300 h	2.739 €
			Total	3.264 €

Tabla 2.3. Costes humanos del proyecto.

2.4.2. Costes tecnológicos

Los costes tecnológicos engloban los costes de todos los recursos de *software* y de *hardware* que se han empleado durante el desarrollo del proyecto.

De los recursos *software*, únicamente se puede contabilizar el coste de *Amazon Web Services (AWS)*, pues el resto de recursos utilizados fueron gratuitos.

En cuanto a recursos *hardware*, el único recurso empleado fue un ordenador portátil con un teclado, un ratón y una pantalla extra conectados. Como este recurso era ya de la empresa, se debe prorratear su coste para los meses que duró la estancia.

En la Tabla 2.4 se muestran los costes de los recursos tecnológicos, los cuales tienen el valor de 134,29 €.

Recurso	Coste	Vida útil	Coste/mes	Meses	Coste total
AWS			25 €	3	75 €
Portátil	1.000 €	60 meses	16,67 €	3	50 €
Teclado extra	15 €	72 meses	0,21 €	3	0,62 €
Ratón extra	8 €	72 meses	0,11 €	3	0,33 €
Pantalla extra	200 €	72 meses	2,78 €	3	8,34 €
				Total	134,29 €

Tabla 2.4. Costes tecnológicos del proyecto.

2.4.3. Costes indirectos

Los costes indirectos son todos aquellos que la empresa debe asumir en su día a día, aunque no estén directamente relacionados con el desarrollo de este proyecto. Por ejemplo, los costes de la luz, los costes de internet o los costes del alquiler de la oficina son costes indirectos.

Se calcula que en torno al 20% del coste total del proyecto son costes indirectos.

2.4.4. Coste total

El coste total del proyecto se ha obtenido al sumar los costes humanos y tecnológicos y, posteriormente, incrementar un 20% ese coste con motivo de gastos indirectos.

Tal como se muestra en la Tabla 2.5, el coste total del proyecto ha sido de 4.077,95 €.

Recursos	Coste
Humanos	3.264 €
Tecnológicos	134,29 €
	3.398,29 €
Indirectos (20%)	679,66 €
Total	4.077,95 €

Tabla 2.5. Coste total del proyecto.

2.5. Análisis de riesgos

A lo largo del desarrollo del proyecto podían llegar a ocurrir diferentes riesgos. Al inicio de la estancia de prácticas, se identificaron todos estos riesgos para poder tomar medidas de prevención y evitar que éstos ocurriesen o para reducir su impacto, en el caso de que se produjera alguno de ellos.

A continuación, se muestra la Tabla 2.6 con los diferentes riesgos que se detectaron, así como la manera en que se podrían tratar para solucionarlos.

Código	Riesgo	Solución
R01	Desconocimiento de la tecnología a utilizar	Buscar información sobre la tecnología o recibir ayuda por parte de la empresa.
R02	Falta de tiempo o problemas con las fechas de entrega	Avanzar a buen ritmo, no perder el tiempo en algo que no salga a la primera e ir revisando el avance del desarrollo del proyecto.

R03	Falta de comunicación con la empresa	Comunicarse con el tutor de prácticas.
R04	Que se produzcan cambios en las tareas a realizar durante el desarrollo y se tenga que descartar o rehacer algo que ya se haya finalizado	Comunicarse continuamente con la empresa y estar seguros de lo que hay que hacer antes de empezar con la implementación.
R05	Caer enfermo y no poder avanzar en el proyecto	Tratarse bien y recuperarse lo antes posible. Además, se debería gestionar el proyecto formativo para prolongar la estancia en la empresa.
R06	Que alguna de las aplicaciones a utilizar tenga algún fallo, dé algún error y no se pueda utilizar	Intentar arreglar el problema o buscar posibles alternativas y utilizar otra aplicación para esa tarea.

Tabla 2.6. Riesgos del proyecto.

2.6. Seguimiento del proyecto

A lo largo de la estancia se realizaron una serie de informes para recopilar el avance del desarrollo del proyecto. Cada informe de seguimiento se realizó cada dos semanas dejando un total de cinco informes, que se presentan íntegros en esta sección.

2.6.1. Primera quincena

Los primeros días estuvieron centrados en la planificación y definición del proyecto. Se determinaron todas las partes que iba a tener la aplicación a desarrollar, qué iba a tener cada una y cómo se iban a conectar. Una vez todo definido, se realizó un documento de diseño en el que se explicaba todo lo que se había dicho y se realizó también un modelo lógico de la base de datos. En éste se indicaban las tablas que se

iban a necesitar, los diferentes campos que tendría cada una y las relaciones entre ellas.

Los siguientes días estuvieron más centrados en realizar la configuración de las diferentes aplicaciones que se iban a utilizar durante el periodo de prácticas. En el caso de Gitlab, se crearon dos repositorios para almacenar todo el código desarrollado, tanto de la parte de *back-end* como de la parte de *front-end*.

El siguiente paso fue crear la base de datos. Para tener un servidor en el que poder almacenarla, se decidió utilizar un *Docker* de MySQL. Una vez creada, se crearon todas las tablas tal como estaban definidas en el modelo lógico.

Con la base de datos ya lista, se comenzó a desarrollar la aplicación. La primera parte que se quería desarrollar era el bloque formado por la pantalla de inicio de sesión de los usuarios de la aplicación, la de registro y la de olvido de contraseña. Durante el desarrollo de esta parte, destacan algunos aspectos que se tuvieron que utilizar y que costaron más, pues no se estaba familiarizado con ellos:

- El encriptado de las contraseñas. Las contraseñas de los usuarios de la aplicación se debían almacenar encriptadas en la base de datos. Además, una vez que el usuario introducía su contraseña al iniciar sesión, ésta se debía comparar con la almacenada en la base de datos, que estaba encriptada. Se tuvo que aprender la manera de encriptar las contraseñas y comparar una contraseña sin encriptar con una encriptada para determinar si eran la misma o no.
- El uso de *tokens*. Para controlar si los usuarios tenían la sesión iniciada y determinar qué operaciones podía hacer cada uno de ellos, se debían utilizar *tokens*. Costó entender el concepto y que funcionaran correctamente, pero se consiguió definirlos y crearlos correctamente.
- El envío automático de correos electrónicos. Tanto después de registrarse en la aplicación, como durante el proceso de olvido de contraseña y obtención de una nueva, era necesario mandar correos electrónicos automáticamente. El proceso fue algo costoso porque no dependía sólo del código a programar, también era necesario un correo desde donde enviarlos y tener una serie de permisos en ese correo.

Una vez finalizado todo ese bloque se comenzó ya con las páginas de la aplicación. Lo primero de todo que se hizo fue conseguir que todas las páginas utilizaran la misma barra lateral y la misma barra superior, de manera que todas

tuvieran una estructura similar. Para lograrlo, se decidió almacenar las barras de manera independiente a las páginas e insertarlas en cada una de ellas. De esta forma, si se modificaba una barra se podría apreciar el cambio en todas las páginas sin tener que revisar individualmente cada página.

La siguiente parte a desarrollar fue el bloque de páginas formado por la página de grupos de trabajo, la de proyectos y la de tareas. En esta quincena sólo se comenzó a plantear cómo sería el diseño de la página de grupos, dejando su desarrollo para la siguiente quincena.

2.6.2. Segunda quincena

La segunda quincena estuvo centrada principalmente en el desarrollo de la página de grupos de trabajo y la corrección de algunos errores.

Para poder trabajar con los grupos y los usuarios de cada grupo, se tuvo que crear primero todas las clases necesarias en el back-end, así como todas las llamadas HTTP que se iban a necesitar para la página.

La página de grupos tenía incorporada una tabla central en la que se mostraban todos los grupos a los que pertenecía el usuario. El primer aspecto en el que se centró el desarrollo fue conseguir que la tabla fuera una tabla dinámica que mostrase los datos almacenados en la base de datos y tuviera un filtro con el que mostrar más o menos resultados.

En la página de grupos también había un botón superior que permitía al usuario crear un grupo nuevo. Además, para cada grupo de la tabla había una serie de botones que permitían realizar las siguientes acciones:

- Ver los proyectos del grupo de trabajo.
- Invitar a otros usuarios al grupo.
- Editar el grupo.
- Eliminar el grupo.

Para los botones se decidió utilizar *pop-ups* que aparecían en la misma página en vez de redirigir al usuario a una página diferente con cada uno de ellos. Su implementación fue un proceso algo costoso al principio pues nunca se había utilizado ninguno, pero el resultado quedó muy vistoso y quedó muy bien estéticamente.

En esta quincena se consiguió implementar toda la página de grupos correctamente, pero también hubo un par de problemas imprevistos que hubo que solucionar.

El primero de ellos estuvo relacionado con los *tokens*. Cada usuario tenía dos *tokens*: el principal y uno de refresco llamado *refreshToken*. El uso del principal se implementó la primera quincena correctamente, permitiendo determinar qué operaciones podía realizar cada usuario. El problema que surgió esta quincena fue que los *tokens* principales tenían una caducidad y se debían actualizar automáticamente gracias al *refreshToken*, pero esta funcionalidad no se había implementado. Se tardó varios días en entender el proceso que se debía seguir y conseguir que se hiciera todo automático sin que el usuario se enterase de nada.

El otro problema que surgió estuvo relacionado con la aplicación en sí y las llamadas HTTP. Para hacer algunas llamadas entre las diferentes páginas en ocasiones era necesario enviar algunos valores que se habían almacenado en la aplicación. El problema fue que el protocolo HTTP solo permitía enviar valores entre páginas del mismo dominio. La aplicación hasta ese momento se estaba desarrollando localmente en un ordenador y HTTP entendía que cada fichero estaba en un dominio diferente y no enviaba los valores. La solución al problema fue *dockerizar* la aplicación y así conseguir que todas las páginas compartieran el mismo dominio.

Una vez finalizada la página de grupos se comenzó a implementar la página de proyectos. Esta implementación fue mucho más rápida y menos costosa pues la página era muy similar a la de grupos.

2.6.3. Tercera quincena

La tercera quincena estuvo centrada principalmente en el desarrollo de la página de tareas.

Esta página debía mostrar todas las tareas que formaban un proyecto de un grupo de trabajo y se podía considerar la página principal del bloque de páginas grupo-proyecto-tarea. Por este mismo motivo, para desarrollar esta página no se siguió la misma estructura que en las páginas de grupos y proyectos, sino que se eligió seguir un diseño de página más visual e interactivo.

La página constaba principalmente de tres columnas formadas por diferentes tarjetas interactivas. Cada una de estas tarjetas era una tarea, la cual se podía pulsar para ver información más detallada sobre la misma. Las columnas representaban los estados de las tareas, que podían ser:

- Pendientes.
- En Desarrollo.
- Finalizadas.

Las diferentes tarjetas eran interactivas y se podían mover por las columnas, de manera que se podía mover una tarjeta en una misma columna de una posición a otra, o bien, cambiarla de columna.

Con este diseño se podía observar claramente qué tareas formaban un proyecto de trabajo y cuáles de esas tareas estaban ya finalizadas, se estaban desarrollando en ese momento, o bien aún estaban pendientes de realizar.

Además de estos aspectos, las tarjetas también permitían realizar diferentes funciones. Al pulsar una tarjeta, además de la información de la tarea, se permitía poder editar la tarea, borrarla o archivarla, lo cual provocaría que la tarea ya no se mostrase en ninguna columna, pero no se borraba de la base de datos.

No obstante, aunque el diseño conseguido fue muy vistoso, su desarrollo fue bastante costoso. Hubo que buscar maneras de desarrollar tarjetas interactivas que se ajustasen a lo que se requería en la aplicación y que realizaran correctamente todas las funciones deseadas.

2.6.4. Cuarta quincena

La cuarta quincena estuvo centrada inicialmente en la finalización de la página de tareas.

El primer aspecto que se implementó fueron las etiquetas. En la página de tareas se debía poder crear y borrar etiquetas, las cuales serían comunes a todas las tareas de esa página. En cada una de las tareas, se debía poder observar todo el listado de etiquetas creadas y elegir aquellas que se querían asignar a la tarea seleccionada.

Para su desarrollo, se creó un botón en cada tarea que abría una pestaña nueva con el listado de etiquetas disponibles. Ahí se permitía seleccionar las etiquetas

deseadas, crear alguna nueva o borrar alguna ya existente. Además, se hizo que en la tarea salieran los nombres de las etiquetas seleccionadas.

Una vez finalizado ese aspecto, se desarrolló la parte de los responsables de las tareas. Cada tarea debía de tener algún responsable de entre los miembros del grupo. Este aspecto fue muy rápido de hacer pues se siguió el mismo método que con las etiquetas. En este caso, el listado que se mostraba eran todos los miembros del grupo y al seleccionar alguno saldría como responsable de la tarea.

El siguiente, y último, aspecto añadido en las tareas fue el registro de horas. En cada tarea, los usuarios podían indicar las horas que habían estado trabajando en esa tarea un día concreto. La cantidad de horas registradas se mostraba en la tarea, y se incrementaba cada vez que se añadía un nuevo registro.

Una vez desarrollados todos estos aspectos, la página de tareas estaba ya finalizada. Y con esto se daba por finalizado todo el bloque de páginas de grupos, proyectos y tareas.

El siguiente paso a realizar era una página nueva, la página del calendario. Esta página estaba formada por un calendario semanal central, que mostraba todos los registros de tareas que hubiese realizado el usuario en cuestión a lo largo de la semana. Los registros mostrados podían ser de cualquier tarea de cualquier proyecto o grupo, siempre que lo hubiera realizado ese usuario. Para diferenciarlas mejor, se hizo que los registros de cada tarea salieran con un color diferente cada uno.

Por defecto, al cargar la página se mostraba la semana actual, pero se incluyó un botón con un calendario para poder elegir cualquier fecha. De esta manera, al seleccionar una fecha concreta, se cargaba en el calendario la semana de esa fecha. Además, también había dos botones más en la página con los que poder avanzar a la siguiente semana o a la anterior, siempre respecto a la semana que estuviera mostrando el calendario en ese momento.

El desarrollo del calendario fue algo costoso al principio. El mayor problema fue trabajar con las fechas y las horas, ya que se tuvo que conseguir que se mostraran todos los registros en las casillas en las que debían aparecer, así como conseguir que al seleccionar una fecha en el botón del calendario, se mostraran los datos de esa semana correctamente.

Una vez finalizada la página del calendario, se inició el desarrollo de la página de fichaje. Esta página debía permitir al usuario fichar la entrada y la salida en el

trabajo. La estructura de la página consistía en dos botones en la parte superior para poder fichar la entrada y la salida, y una tabla central que mostraba los registros de los fichajes ya realizados.

El aspecto más destacable del desarrollo de esta página fueron los filtros de la tabla. La tabla mostraba todos los registros del usuario y debía de haber una manera para poder filtrarlos por fechas. Se decidió añadir un selector con el criterio de filtrado, que podía ser por año, mes o día. Al lado del selector se permitía seleccionar una fecha y, según el criterio seleccionado, se mostraban los datos filtrados.

2.6.5. Quinta quincena

La quinta quincena fue la última de la estancia de prácticas. Por este motivo, esta quincena se centró en finalizar muchos detalles de las páginas desarrolladas, así como corregir algunos aspectos de éstas.

Para comenzar, se finalizó la página de fichajes.

Se decidió que, además de los registros de fichaje del usuario, también se mostrasen los registros de los usuarios de aquellos grupos en los que el usuario fuera administrador. Por este motivo, se añadieron más filtros además del de la fecha. El primer filtro añadido fue el filtro de grupos, para mostrar los registros de todos los miembros del grupo seleccionado. El segundo fue un filtro para elegir un usuario concreto del grupo seleccionado. De esta manera, los registros mostrados se podían filtrar por fecha, grupo o usuario, con todas sus combinaciones posibles.

También se añadió un botón para poder descargar los registros mostrados en un documento CSV. Con este botón, se dio por terminada la página de fichajes.

La siguiente página que se realizó fue el perfil del usuario. En éste, se mostraban todos los datos del usuario. De inicio estos datos no se podían editar, pero existía un botón en un lateral que, al pulsarlo, habilitaba la edición de los datos a excepción del correo del usuario. En la parte inferior del perfil había otro botón que permitía guardar los cambios y volver a deshabilitar la edición de los datos.

Finalizado el perfil, se desarrolló la página de inicio de la aplicación, que hasta este momento estaba en blanco. El cuerpo de esta página consistió en cuatro tarjetas centrales, puestas de dos en dos, sobre las diferentes páginas de la aplicación. Las

cuatro páginas que aparecían en esas tarjetas fueron: la página de grupos, la del calendario, la del fichaje y el perfil del usuario. Cada una de las tarjetas incluía un icono identificativo, una breve descripción de la página y un enlace a la misma.

Los siguientes aspectos que se desarrollaron fueron realizar el *log-out* o cierre de sesión y realizar el diseño final de la barra superior y de la barra lateral de la aplicación.

A partir de este momento, debido a que se acercaba el final de la estancia, se decidió centrar más el esfuerzo en añadir algunos aspectos más a las páginas ya desarrolladas y no crear ninguna página nueva. Mientras se añadían estos nuevos aspectos se fueron observando algunos fallos menores, los cuales se fueron solucionando a medida que se detectaban.

Muchos de los detalles añadidos fueron detalles estéticos y no añadían ni modificaban ninguna funcionalidad de las páginas. No obstante, sí se añadieron algunas funcionalidades nuevas a destacar.

Una de ellas fue en la página de grupos. En el listado de grupos de esta página había un botón para invitar usuarios a ese grupo. Se decidió cambiar el botón para que además de invitar usuarios, se pudieran ver los miembros actuales y poder eliminar miembros del grupo. De esta manera, el botón pasó a ser “Gestionar usuarios” y no “Invitar usuario”.

Otro aspecto añadido fue en la página del calendario. En esta página se mostraban los registros del usuario en una semana determinada, pero los registros se realizaban en la página de tareas. Se decidió añadir un botón en la página del calendario para poder realizar un registro también desde esta página. El registro se realizaba de la misma manera que en la página de tareas con la excepción de que en el calendario se debía indicar la tarea en la que se quería realizar el registro. También se añadieron validadores tanto en esta página como en la de tareas, para asegurar que el usuario introduzca datos correctos.

El último aspecto que se añadió fue el de adjuntar documentos en las tareas. Este aspecto estaba planificado desde un principio, pero cuando se desarrolló la página de tareas, se decidió dejarlo para más adelante pues conllevaba utilizar otras herramientas que podían llevar mucho tiempo.

De esta manera, en la página de tareas, al pulsar en una tarea en concreto, se mostraba también la opción de documentos adjuntos. Al pulsar en el botón se

mostraba un listado con los documentos ya adjuntados y en un lateral había una opción para adjuntar un nuevo documento. Además, para cada uno de ellos se permitía cargar el documento y eliminar el adjunto.

En este último aspecto se tuvo que tener en cuenta a *Amazon Web Services*, una plataforma en la nube donde la empresa almacenaba los documentos. De este modo, todas las funcionalidades relacionadas con los adjuntos debían pasar a través de esta plataforma.

Finalizados estos aspectos, se dio por terminada la aplicación.

Capítulo 3

Análisis del sistema

3.1. Actores

En la aplicación desarrollada se diferenciaron varios roles de usuario dentro de un grupo de trabajo. A continuación se da una explicación detallada de cada uno de estos roles:

- **Administrador.** Es el creador del grupo y actúa como jefe de éste. Cuenta con todos los permisos y puede realizar todas las funcionalidades de los grupos, proyectos y tareas.
- **Desarrollador.** Este actor cuenta con algunos permisos para realizar ciertas funcionalidades de la aplicación, pero no todas. Concretamente, un desarrollador puede realizar todas las funcionalidades de las tareas pero, de los grupos y de los proyectos, sólo puede ver la información y no puede modificar nada.
- **Usuario.** El usuario es aquel miembro del grupo que sólo tiene permisos de lectura. Únicamente puede ver la información de los grupos, proyectos y tareas, pero no realizar ninguna de sus funcionalidades, salvo registrar las horas trabajadas en una tarea.

Además de estos actores, también se debe tener en cuenta el **servidor SMTP**. Éste se encarga de enviar correos automáticos en diferentes puntos de la aplicación.

3.2. Diagrama de casos de uso

El diagrama de casos de uso es un tipo de diagrama UML (*Unified Modeling Language* o *Lenguaje de Modelado Unificado*) de comportamiento [12] que permite relacionar los diferentes actores del sistema con las funcionalidades que puede realizar cada uno de ellos. En la Figura 3.1 se muestra el diagrama de casos de uso del proyecto desarrollado.

Para facilitar la visibilidad de dicho diagrama, se han agrupado los casos de uso en función de la página de la aplicación en la que se encuentran. Como se puede observar en el diagrama, el caso CU14 se puede realizar desde dos páginas distintas por lo que se ha colocado entre los dos paquetes correspondientes.

En la Tabla 3.1 se muestra un resumen con las relaciones entre los actores y los casos de uso identificados. Cabe destacar que la relación entre los actores en el diagrama se muestra con una generalización. En este diagrama, la generalización indica que el desarrollador también está relacionado con los casos de uso del usuario y, al mismo tiempo, el administrador también está relacionado con los casos de uso del desarrollador. Esto se debe a las diferentes funcionalidades que puede realizar cada uno de estos actores en la aplicación, tal como se describió en la sección anterior.

En las posteriores Tablas 3.2-3.19 se detallan los diferentes casos de uso.

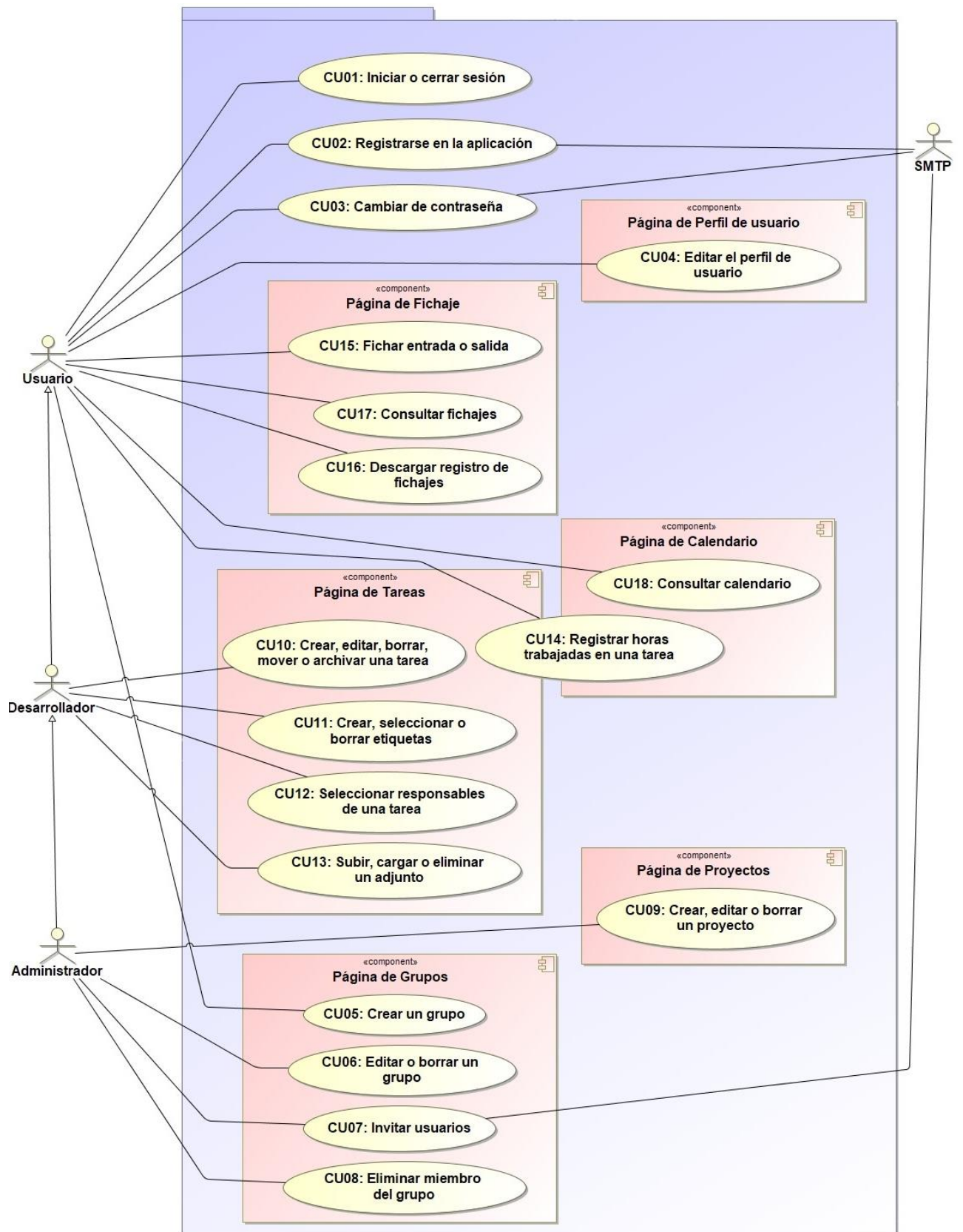


Figura 3.1: Diagrama de casos de uso del proyecto.

Actor	Identificador del caso de uso	Descripción del caso de uso
Usuario	CU01	Iniciar o cerrar sesión en la aplicación.
	CU04	Editar el perfil del usuario.
	CU05	Crear un grupo de trabajo.
	CU14	Registrar horas trabajadas en una tarea.
	CU15	Fichar entrada o salida al trabajo.
	CU16	Descargar el registro de fichajes.
	CU17	Consultar fichajes, pudiendo filtrar los fichajes mostrados.
	CU18	Consultar calendario, pudiendo buscar el calendario de una fecha concreta.
Usuario + Servidor SMTP	CU02	Registrarse en la aplicación.
	CU03	Cambiar de contraseña.
Desarrollador	CU10	Crear, editar, borrar, mover o archivar una tarea.
	CU11	Crear, seleccionar o borrar etiquetas.
	CU12	Seleccionar responsables de una tarea.
	CU13	Subir, cargar o eliminar un adjunto.
Administrador	CU06	Editar o borrar un grupo.
	CU08	Eliminar un miembro del grupo.
	CU09	Crear, editar o borrar un proyecto.
Administrador + Servidor SMTP	CU07	Invitar usuarios a un grupo.

Tabla 3.1. Relaciones entre actores y casos de uso.

CU01 - Iniciar o cerrar sesión en la aplicación.
<p>ID : CU01 Nombre : Iniciar o cerrar sesión en la aplicación. Actor : Usuario Fuente : Petición de un usuario Versión : 1</p>
<p>Descripción : Permite a un usuario iniciar sesión y entrar en la aplicación, o cerrar la sesión para salir de ésta.</p>
<p>Secuencia de pasos :</p> <ul style="list-style-type: none"> • Para iniciar sesión: <ul style="list-style-type: none"> - Acceder a la página de inicio de sesión. - Introducir correo electrónico y contraseña. - Pulsar en Iniciar Sesión. - Acceder a la aplicación. • Para cerrar sesión: <ul style="list-style-type: none"> - Pulsar en el icono del usuario en la esquina superior derecha de cualquier página de la aplicación. - Seleccionar la opción de Cerrar Sesión. - Se abre una pestaña para confirmar la operación. - Confirmar el cierre de sesión.
<p>Pre-requisitos :</p> <ul style="list-style-type: none"> • Para iniciar sesión: El usuario debe estar registrado en el sistema. • Para cerrar sesión: El usuario debe tener la sesión iniciada.
<p>Excepciones :</p> <ul style="list-style-type: none"> • Si el usuario introduce mal sus datos al iniciar sesión, se muestran validadores que indican el error cometido.

Tabla 3.2. Caso de uso CU01 - Iniciar o cerrar sesión en la aplicación.

CU02 - Registrarse en la aplicación.
ID : CU02 Nombre : Registrarse en la aplicación. Actor : Usuario y Servidor SMTP. Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario externo de la aplicación registrarse en ésta.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de registro. - Introducir los datos requeridos. - El sistema envía al usuario un correo electrónico en el que se le da la bienvenida a la aplicación. - El usuario es redirigido a la página de inicio de sesión.
Pre-requisitos : El usuario no debe estar registrado ya en el sistema.
Excepciones : <ul style="list-style-type: none"> • Si el usuario introduce mal sus datos al registrarse, se muestran validadores que indican el error cometido.

Tabla 3.3. Caso de uso CU02 - Registrarse en la aplicación.

CU03 - Cambiar de contraseña.
ID : CU03 Nombre : Cambiar de contraseña. Actor : Usuario y Servidor SMTP. Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario cambiar de contraseña.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de cambio de contraseña. - Introducir el correo electrónico. - El sistema envía al usuario un correo electrónico con un enlace a una página temporal para poder cambiar de contraseña. - Pulsar en el enlace del correo y acceder a la nueva página. - Introducir la nueva contraseña por duplicado. - Pulsar en Cambiar Contraseña para guardar el cambio.
Pre-requisitos : El usuario debe estar registrado en el sistema.
Excepciones : <ul style="list-style-type: none"> • Si el usuario accede al enlace del correo una vez ha expirado la página, al pulsar en Cambiar Contraseña: <ul style="list-style-type: none"> - El sistema muestra un mensaje de error porque la página ha expirado. - El usuario es redirigido a la página original de cambio de contraseña.

Tabla 3.4. Caso de uso CU03 - Cambiar de contraseña.

CU04 – Editar el perfil del usuario.
ID : CU04 Nombre : Editar el perfil del usuario. Actor : Usuario Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario cambiar sus datos personales en su perfil.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de perfil de usuario. - Pulsar en el botón lateral de edición. - Realizar los cambios que se deseen. - Pulsar en Guardar.
Pre-requisitos : El usuario debe estar registrado en el sistema.
Excepciones : <ul style="list-style-type: none"> • Si el usuario desea cancelar los cambios, volver a pulsar el botón de edición.

Tabla 3.5. Caso de uso CU04 - Editar el perfil del usuario.

CU05 - Crear un grupo de trabajo.
ID : CU05 Nombre : Crear un grupo de trabajo. Actor : Usuario Fuente : Petición de un usuario Versión : 1
Descripción : Permite a cualquier usuario crear un grupo de trabajo.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de grupos. - Pulsar en el botón verde Crear Grupo - Introducir datos necesarios. - Pulsar en Crear Grupo. - Se actualiza la lista de grupos.
Pre-requisitos : El usuario debe tener la sesión iniciada.
Excepciones : <ul style="list-style-type: none"> • Si el usuario introduce mal los datos, se muestran validadores que indican el error cometido.

Tabla 3.6. Caso de uso CU05 - Crear un grupo de trabajo.

CU06 - Editar o borrar un grupo.
ID : CU06 Nombre : Editar o borrar un grupo. Actor : Administrador Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario editar o borrar un grupo de trabajo ya existente.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de grupos. • Para editar un grupo: <ul style="list-style-type: none"> - Pulsar en el botón amarillo de un grupo. - Cambiar los datos del grupo. - Pulsar en Editar Grupo para guardar los cambios. - Se actualiza la lista de grupos. • Para borrar un grupo: <ul style="list-style-type: none"> - Pulsar en el botón rojo de un grupo. - Se abre una pestaña para confirmar la operación. - Confirmar el borrado de grupo. - Se actualiza la lista de grupos.
Pre-requisitos : <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. - El usuario debe tener el rol de Administrador.
Excepciones : <ul style="list-style-type: none"> • Si el usuario introduce mal los datos al editar el grupo, se muestran validadores que indican el error cometido.

Tabla 3.7. Caso de uso CU06 - Editar o borrar un grupo.

CU07 - Invitar usuarios a un grupo.
ID : CU07 Nombre : Invitar usuarios a un grupo. Actor : Administrador y Servidor SMTP Fuente : Petición de un usuario Versión : 1
Descripción : Permite al administrador del grupo invitar a un usuario al grupo.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de grupos. - Pulsar en el botón gris de un grupo. - Se abre una pestaña con el listado de miembros del grupo. - Introducir el correo electrónico del usuario a invitar. - Indicar el rol del nuevo usuario. - Pulsar en Invitar. - El sistema envía al usuario un correo electrónico indicando que ha sido invitado a un grupo. - Se actualiza la lista de miembros del grupo.
Pre-requisitos : <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. - El usuario debe tener el rol de Administrador.
Excepciones : <ul style="list-style-type: none"> • Si el administrador introduce mal el correo del invitado, se muestran validadores que indican el error cometido.

Tabla 3.8. Caso de uso CU07 - Invitar usuarios a un grupo.

CU08 - Eliminar un miembro del grupo.
ID : CU08 Nombre : Eliminar un miembro del grupo. Actor : Administrador Fuente : Petición de un usuario Versión : 1
Descripción : Permite al administrador del grupo eliminar a un usuario al grupo.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de grupos. - Pulsar en el botón gris de un grupo. - Se abre una pestaña con el listado de miembros del grupo. - Pulsar en el botón rojo al lado de un miembro del grupo. - Se abre una pestaña para confirmar la operación. - Confirmar la eliminación del miembro del grupo. - Se actualiza la lista de miembros del grupo.
Pre-requisitos : <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. - El usuario debe tener el rol de Administrador.

Tabla 3.9. Caso de uso CU08 - Eliminar un miembro del grupo.

CU09 - Crear, editar o borrar un proyecto.
<p>ID : CU09 Nombre : Crear, editar o borrar un proyecto. Actor : Administrador Fuente : Petición de un usuario Versión : 1</p>
<p>Descripción : Permite a un usuario crear, editar o borrar un proyecto.</p>
<p>Secuencia de pasos :</p> <ul style="list-style-type: none"> - Acceder a la página de proyectos. • Para crear un proyecto: <ul style="list-style-type: none"> - Pulsar en el botón verde Crear Proyecto. - Introducir datos necesarios. - Pulsar en Crear Proyecto. - Se actualiza la lista de proyectos. • Para editar un proyecto: <ul style="list-style-type: none"> - Pulsar en el botón amarillo de un proyecto. - Cambiar los datos del proyecto. - Pulsar en Editar Proyecto para guardar los cambios. - Se actualiza la lista de proyectos. • Para borrar un proyecto: <ul style="list-style-type: none"> - Pulsar en el botón rojo de un proyecto. - Se abre una pestaña para confirmar la operación. - Confirmar el borrado del proyecto. - Se actualiza la lista de proyectos.
<p>Pre-requisitos :</p> <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. - El usuario debe tener el rol de Administrador.
<p>Excepciones :</p> <ul style="list-style-type: none"> • Si el usuario introduce mal los datos al crear o editar un proyecto, se muestran validadores que indican el error cometido.

Tabla 3.10. Caso de uso CU09 - Crear, editar o borrar un proyecto.

CU10 - Crear, editar, borrar, mover o archivar una tarea.
<p>ID : CU10 Nombre : Crear, editar, borrar, mover o archivar una tarea. Actor : Desarrollador. Fuente : Petición de un usuario Versión : 1</p>
<p>Descripción : Permite a un usuario crear, editar, borrar, mover o archivar una tarea.</p>
<p>Secuencia de pasos :</p> <ul style="list-style-type: none"> - Acceder a la página de tareas. • Para crear una tarea: <ul style="list-style-type: none"> - Pulsar en el botón con el símbolo + de una columna. - Introducir datos necesarios. - Pulsar en Crear Tarea. - Se actualiza la lista de tareas. - Se abre la pestaña de la nueva tarea. • Para mover una tarea: <ul style="list-style-type: none"> - Seleccionar una tarea. - Arrastrar la tarea a la posición deseada. • Para editar una tarea: <ul style="list-style-type: none"> - Pulsar en una tarea. - Se abre la pestaña de la tarea. - Seleccionar el campo que se desea editar. - Cambiar los datos del campo. - Pulsar en Guardar para guardar los cambios. • Para archivar una tarea: <ul style="list-style-type: none"> - Pulsar en una tarea. - Se abre la pestaña de la tarea. - Pulsar en el botón gris para archivar la tarea. - Se abre una pestaña para confirmar la operación. - Confirmar la operación. - Se actualiza la lista de tareas. • Para borrar un proyecto: <ul style="list-style-type: none"> - Pulsar en una tarea. - Se abre la pestaña de la tarea. - Pulsar en el botón rojo para borrar la tarea. - Se abre una pestaña para confirmar la operación. - Confirmar el borrado de la tarea. - Se actualiza la lista de tareas.
<p>Pre-requisitos :</p> <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. • Para crear, editar, archivar o borrar una tarea: <ul style="list-style-type: none"> - El usuario debe tener el rol de Administrador o Desarrollador.
<p>Excepciones :</p> <ul style="list-style-type: none"> • Si el usuario introduce mal los datos al crear o editar una tarea, se muestran validadores que indican el error cometido.

Tabla 3.11. Caso de uso CU10 - Crear, editar, borrar, mover o archivar una tarea.

CU11 - Crear, seleccionar o borrar etiquetas.
ID : CU11 Nombre : Crear, seleccionar o borrar etiquetas. Actor : Desarrollador Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario crear nuevas etiquetas, seleccionar etiquetas para una tarea concreta o borrar etiquetas ya existentes.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de tareas. - Pulsar en una tarea. - Se abre la pestaña de la tarea. - Pulsar en el botón verde de etiquetas. - Se abre una pestaña con el listado de etiquetas • Para crear una etiqueta: <ul style="list-style-type: none"> - Pulsar en el botón con el símbolo +. - Se abre una pestaña para crear una etiqueta. - Introducir datos necesarios. - Pulsar en Crear. - Se actualiza la lista de etiquetas. • Para seleccionar una etiqueta: <ul style="list-style-type: none"> - Pulsar en el campo seleccionable a la izquierda de cada etiqueta. • Para borrar una etiqueta: <ul style="list-style-type: none"> - Pulsar en el botón rojo a la derecha de la etiqueta a eliminar. - Se abre una pestaña para confirmar la operación. - Confirmar el borrado de la etiqueta. - Se actualiza la lista de etiquetas.
Pre-requisitos : <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. - El usuario debe tener el rol de Administrador o Desarrollador.

Tabla 3.12. Caso de uso CU11 - Crear, seleccionar o borrar etiquetas.

CU12 - Seleccionar responsables de una tarea.
ID : CU12 Nombre : Seleccionar responsables de una tarea. Actor : Desarrollador Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario seleccionar los responsables de una tarea.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de tareas. - Pulsar en una tarea. - Se abre la pestaña de la tarea. - Pulsar en el botón azul de responsables. - Se abre una pestaña con el listado de usuarios del grupo. - Pulsar en el campo seleccionable a la izquierda de cada usuario.
Pre-requisitos : <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. - El usuario debe tener el rol de Administrador o Desarrollador.

Tabla 3.13. Caso de uso CU12 - Seleccionar responsables de una tarea.

CU13 - Subir, cargar o eliminar un adjunto.
<p>ID : CU13 Nombre : Subir, cargar o eliminar un adjunto. Actor : Desarrollador Fuente : Petición de un usuario Versión : 1</p>
<p>Descripción : Permite a un usuario subir, cargar o eliminar un documento adjunto de una tarea.</p>
<p>Secuencia de pasos :</p> <ul style="list-style-type: none"> - Acceder a la página de tareas. - Pulsar en una tarea. - Se abre la pestaña de la tarea. - Pulsar en el botón gris de adjuntos. - Se abre una pestaña con el listado de adjuntos. • Para subir un adjunto: <ul style="list-style-type: none"> - Pulsar en el botón con el símbolo +. - Se abre una pestaña para adjuntar un documento. - Seleccionar el documento. - Pulsar en Adjuntar. - Se actualiza la lista de adjuntos. • Para cargar un adjunto: <ul style="list-style-type: none"> - Pulsar en el botón azul a la derecha del adjunto. - En función del tipo de documento, éste se abre o se descarga. • Para eliminar un adjunto: <ul style="list-style-type: none"> - Pulsar en el botón rojo a la derecha del adjunto. - Se abre una pestaña para confirmar la operación. - Confirmar el borrado del adjunto. - Se actualiza la lista de adjuntos.
<p>Pre-requisitos :</p> <ul style="list-style-type: none"> - El usuario debe tener la sesión iniciada. - El usuario debe tener el rol de Administrador o Desarrollador.

Tabla 3.14. Caso de uso CU13 - Subir, cargar o eliminar un adjunto.

CU14 - Registrar horas trabajadas en una tarea.
ID : CU14 Nombre : Registrar horas trabajadas en una tarea. Actor : Usuario Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario registrar las horas trabajadas en una tarea.
Secuencia de pasos : <ul style="list-style-type: none"> • Desde la página de tareas: <ul style="list-style-type: none"> - Acceder a la página de tareas. - Pulsar en una tarea. - Se abre la pestaña de la tarea. - Pulsar en el botón azul inferior del apartado Horas. - Se abre una pestaña para realizar el registro. - Introducir datos necesarios. - Pulsar en Registrar Horas. - Se actualizan las horas de la tarea. • Desde la página del calendario: <ul style="list-style-type: none"> - Acceder a la página del calendario. - Pulsar en Realizar Registro. - Se abre una pestaña para realizar el registro. - Introducir datos necesarios. - Pulsar en Registrar Horas. - Se actualiza el calendario mostrado.
Pre-requisitos : El usuario debe tener la sesión iniciada.
Excepciones : <ul style="list-style-type: none"> • Si el usuario introduce mal los datos, se muestran validadores que indican el error cometido.

Tabla 3.15. Caso de uso CU14 - Registrar horas trabajadas en una tarea.

CU15 - Fichar entrada o salida al trabajo.
ID : CU15 Nombre : Fichar entrada o salida al trabajo. Actor : Usuario Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario fichar la entrada o la salida al trabajo.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de fichaje. - Pulsar en Fichar Entrada o Fichar Salida. - Se actualiza el listado de fichajes.
Pre-requisitos : El usuario debe tener la sesión iniciada.

Tabla 3.16. Caso de uso CU15 - Fichar entrada o salida al trabajo.

CU16 - Descargar el registro de fichajes.
ID : CU16 Nombre : Descargar el registro de fichajes. Actor : Usuario Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario descargar el registro de fichajes.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de fichaje. - Pulsar en Exportar. - Se descarga un documento CSV con el registro de fichajes.
Pre-requisitos : El usuario debe tener la sesión iniciada.

Tabla 3.17. Caso de uso CU16 - Descargar el registro de fichajes.

CU17 - Consultar fichajes, pudiendo filtrar los fichajes mostrados.
ID : CU17 Nombre : Consultar fichajes, pudiendo filtrar los fichajes mostrados. Actor : Usuario Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario consultar todos los fichajes, pudiendo filtrar los fichajes mostrados.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de fichaje. - Se muestra el listado de fichajes. • Para mostrar el calendario de otra semana: <ul style="list-style-type: none"> - Seleccionar los filtros a aplicar. - Pulsar en Aplicar. - Se actualiza el listado de fichajes.
Pre-requisitos : El usuario debe tener la sesión iniciada.

Tabla 3.18. Caso de uso CU17 - Consultar fichajes, pudiendo filtrar los fichajes mostrados.

CU18 - Consultar calendario, pudiendo buscar el calendario de una fecha concreta.
ID : CU18 Nombre : Consultar calendario, pudiendo buscar el calendario de una fecha concreta. Actor : Usuario Fuente : Petición de un usuario Versión : 1
Descripción : Permite a un usuario consultar el calendario, pudiendo buscar el calendario de una fecha concreta.
Secuencia de pasos : <ul style="list-style-type: none"> - Acceder a la página de calendario. - Se muestra el calendario de la semana actual. • Para mostrar el calendario de otra semana: <ul style="list-style-type: none"> - Pulsar en el botón de Semana Anterior, Semana Posterior o cambiar la fecha seleccionada. - Se actualiza el calendario mostrado.
Pre-requisitos : El usuario debe tener la sesión iniciada.

Tabla 3.19. Caso de uso CU18 - Consultar calendario, pudiendo buscar el calendario de una fecha concreta.

3.3. Requisitos de datos

En esta sección se definen los requisitos de datos, que son aquellos datos que necesita el sistema para llevar a cabo los casos de uso definidos en el apartado anterior.

Estos requisitos se muestran detalladamente en las Tablas 3.20-3.30.

RD01 – Iniciar sesión y cambio de contraseña.
ID : RD01 Nombre : Iniciar sesión y cambio de contraseña. Fuente : CU01 y CU03
Tipo : Datos de entrada/ salida
Descripción : Datos sobre el usuario que va a iniciar sesión o pedir cambio de contraseña.
Datos : Correo electrónico y contraseña.

Tabla 3.20. Requisito de datos RD01 – Iniciar sesión y cambio de contraseña.

RD02 – Registrarse en la aplicación.
ID : RD02
Nombre : Registrarse en la aplicación.
Fuente : CU02
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para registrar a un usuario en la aplicación.
Datos : Nombre, correo electrónico y contraseña.

Tabla 3.21. Requisito de datos RD02 – Registrarse en la aplicación.

RD03 – Editar el perfil de usuario.
ID : RD03
Nombre : Editar el perfil de usuario.
Fuente : CU04
Tipo : Datos de entrada/ salida
Descripción : Datos sobre el usuario que se van a modificar desde su perfil.
Datos : Nombre del usuario.

Tabla 3.22. Requisito de datos RD03 – Editar el perfil de usuario.

RD04 – Crear o editar un grupo.
ID : RD04
Nombre : Crear o editar un grupo.
Fuente : CU05 y CU06
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para crear o editar un grupo.
Datos : Nombre y descripción del grupo.

Tabla 3.23. Requisito de datos RD04 – Crear o editar un grupo.

RD05 – Invitar usuarios.
ID : RD05
Nombre : Invitar usuarios.
Fuente : CU07
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para invitar a un usuario a un grupo.
Datos : Correo electrónico y rol.

Tabla 3.24. Requisito de datos RD05 – Invitar usuarios.

RD06 – Crear o editar un proyecto.
ID : RD06
Nombre : Crear o editar un proyecto.
Fuente : CU09
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para crear o editar un proyecto.
Datos : Nombre y descripción del proyecto.

Tabla 3.25. Requisito de datos RD06 – Crear o editar un proyecto.

RD07 – Crear o editar una tarea.
ID : RD07
Nombre : Crear o editar una tarea.
Fuente : CU10
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para crear o editar una tarea.
Datos : Nombre, descripción y número de horas esperadas de la tarea.

Tabla 3.26. Requisito de datos RD07 – Crear o editar una tarea.

RD08 – Crear etiquetas.
ID : RD08
Nombre : Crear etiquetas.
Fuente : CU11
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para crear una etiqueta.
Datos : Nombre de la etiqueta.

Tabla 3.27. Requisito de datos RD08 – Crear etiquetas.

RD09 – Subir un adjunto.
ID : RD09
Nombre : Subir un adjunto.
Fuente : CU13
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para adjuntar un documento.
Datos : Documento y descripción.

Tabla 3.28. Requisito de datos RD09 – Subir un adjunto.

RD10 – Registrar horas desde la página de tareas.
ID : RD10
Nombre : Registrar horas desde la página de tareas.
Fuente : CU14
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para registrar horas desde la página de tareas.
Datos : Día, hora de inicio y hora de fin.

Tabla 3.29. Requisito de datos RD10 – Registrar horas desde la página de tareas.

RD11 – Registrar horas desde la página de calendario.
ID : RD11
Nombre : Registrar horas desde la página de calendario.
Fuente : CU11
Tipo : Datos de entrada/ salida
Descripción : Datos necesarios para registrar horas desde la página de calendario.
Datos : nombre del grupo, nombre del proyecto, nombre de la tarea, día, hora de inicio y hora de fin.

Tabla 3.30. Requisito de datos RD11 – Registrar horas desde la página de calendario.

Capítulo 4

Diseño de la base de datos

4.1. Diseño lógico

En esta sección se presenta el diseño lógico de la base de datos diseñada durante el proyecto. El diseño lógico muestra un modelo de base de datos independiente del SGBD (Sistema de Gestión de Bases de Datos) utilizado, y sirve como fuente de información para realizar el posterior diseño físico.

En la Figura 4.1 se muestra un esquema con todo el diseño lógico de la base de datos. Posteriormente, se detalla la definición de este diseño, indicando aquellos campos que acepten nulos, las claves ajenas presentes en cada tabla y sus respectivas reglas de modificación y de borrado.

Cabe puntualizar que en el diseño de la base de datos existe una tabla llamada *vacaciones*, que finalmente no se llegó a utilizar en la aplicación. El motivo es que se tenía pensado desarrollar una página más en la aplicación que emplease dicha tabla, pero finalmente, por falta de tiempo, no se llegó a implementar. Como la tabla sí se llegó a crear en la base de datos, se ha mantenido en el diseño presentado en esta sección.

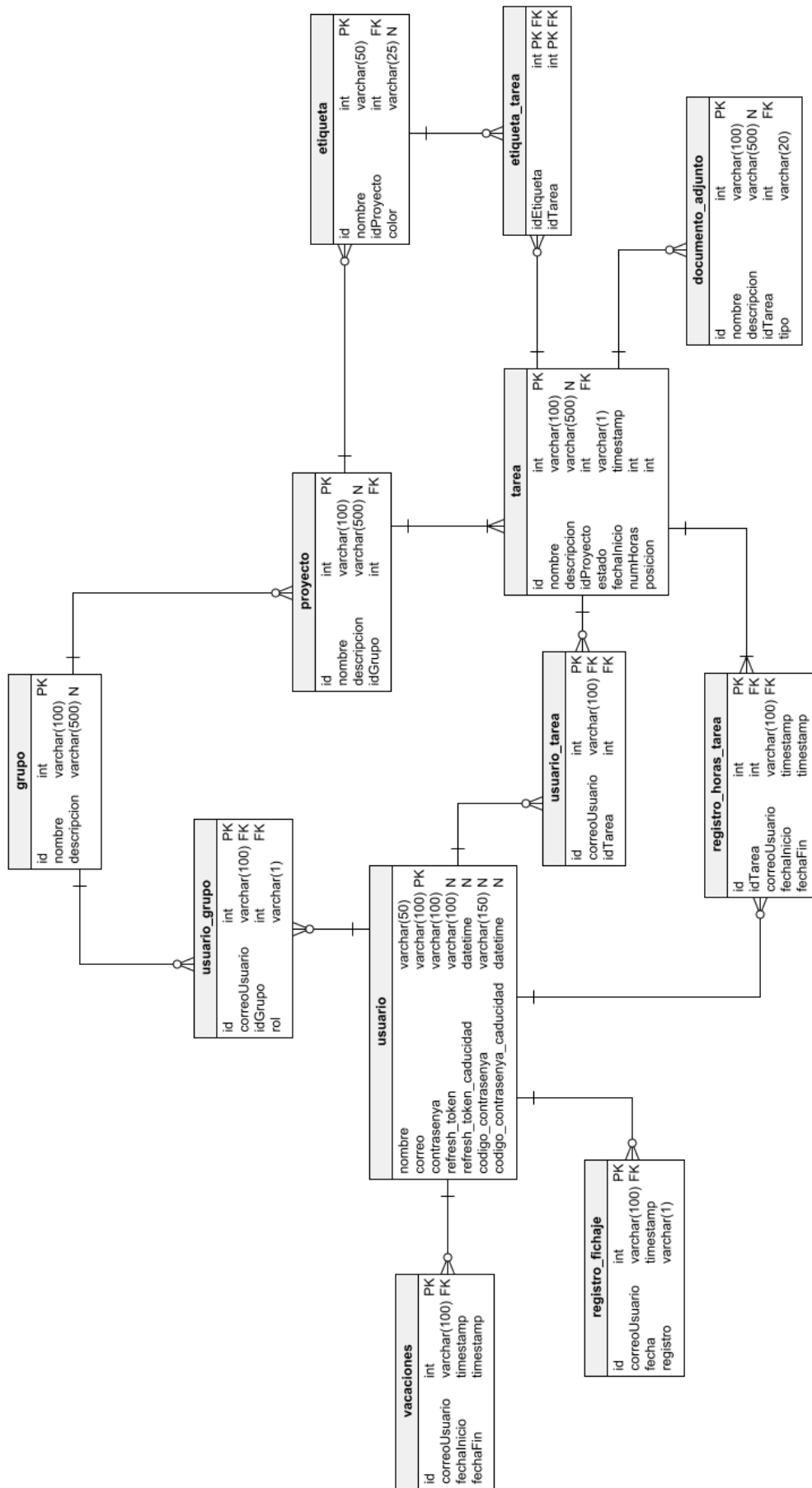


Figura 4.1: Diseño lógico de la base de datos.

A continuación se muestra la definición del diseño lógico:

GRUPO (id, nombre, descripcion)

descripcion acepta nulos

PROYECTO (id, nombre, descripcion, idGrupo)

descripcion acepta nulos

	idGrupo	nulos	modificación	borrado
PROYECTO	→	GRUPO	no propagar	restringir

TAREA (id, nombre, descripcion, idProyecto, estado, fechaInicio, numHoras, posicion)

descripcion acepta nulos

estado ∈ {D, P, F, A}

	idProyecto	nulos	modificación	borrado
TAREA	→	PROYECTO	no propagar	restringir

USUARIO (nombre, correo, contraseña, refresh_token, refresh_token_caducidad, codigo_contraseña, codigo_contraseña_caducidad)

refresh_token, refresh_token_caducidad, codigo_contraseña y codigo_contraseña_caducidad aceptan nulos

USUARIO_GRUPO (id, correoUsuario, idGrupo, rol)

rol ∈ {A, D, U}

	correoUsuario	nulos	modificación	borrado
USUARIO_GRUPO	→	USUARIO	no propagar	propagar

	idGrupo	nulos	modificación	borrado
USUARIO_GRUPO	→	GRUPO	no propagar	propagar

USUARIO_TAREA (id, correoUsuario, idTarea)

correoUsuario	nulos	modificación	borrado
USUARIO_TAREA → USUARIO	no	propagar	propagar
idTarea	nulos	modificación	borrado
USUARIO_TAREA → TAREA	no	propagar	propagar

ETIQUETA (id, nombre, idProyecto, color)

color acepta nulos

idProyecto	nulos	modificación	borrado
ETIQUETA → PROYECTO	no	propagar	propagar

ETIQUETA_TAREA (idEtiqueta, idTarea)

idEtiqueta	nulos	modificación	borrado
ETIQUETA_TAREA → ETIQUETA	no	propagar	propagar
idTarea	nulos	modificación	borrado
ETIQUETA_TAREA → TAREA	no	propagar	propagar

REGISTRO_HORAS_TAREA (id, idTarea, correoUsuario, fechaInicio, fechaFin)

fechaFin ha de ser mayor o igual a fechaInicio

idTarea	nulos	modificación	borrado
REGISTRO_HORAS_TAREA → TAREA	no	propagar	propagar
correoUsuario	nulos	modificación	borrado
REGISTRO_HORAS_TAREA → USUARIO	no	propagar	propagar

REGISTRO_FICHAJE (id, correoUsuario, fecha, registro)

correoUsuario	nulos	modificación	borrado
REGISTRO_FICHAJE → USUARIO	no	propagar	propagar

DOCUMENTO_ADJUNTO (id, nombre, descripcion, idTarea, tipo)

descripcion acepta nulos

	idTarea	nulos	modificación	borrado
DOCUMENTO_ADJUNTO	→ TAREA	no	propagar	propagar

VACACIONES (id, correoUsuario, fechaInicio, fechaFin)

fechaFin ha de ser mayor o igual a fechaInicio

	correoUsuario	nulos	modificación	borrado
VACACIONES	→ USUARIO	no	propagar	propagar

4.2. Diseño físico

Finalmente, en esta sección se muestra el diseño físico de la base de datos. Este diseño se obtiene al implementar el diseño lógico de la base de datos en el SGBD (Sistema de Gestión de Bases de Datos) escogido, en este caso, MySQL.

En los Códigos Fuente 4.1 a 4.12 se recoge este diseño al completo.

```
CREATE TABLE grupo (  
  id INT NOT NULL AUTO_INCREMENT,  
  nombre VARCHAR(100) NOT NULL,  
  descripcion VARCHAR(500) DEFAULT NULL,  
  PRIMARY KEY (id)  
)
```

Código Fuente 4.1: Diseño Físico - Grupo

```

CREATE TABLE proyecto (
  id INT NOT NULL AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  descripcion VARCHAR(500) DEFAULT NULL,
  idGrupo INT NOT NULL,
  PRIMARY KEY (id),
  KEY proyecto_grupo (idGrupo),
  CONSTRAINT proyecto_grupo FOREIGN KEY (idGrupo)
REFERENCES grupo (id) ON DELETE RESTRICT ON UPDATE
CASCADE
)

```

Código Fuente 4.2: Diseño físico – Proyecto

```

CREATE TABLE tarea (
  id INT NOT NULL AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  descripcion VARCHAR(500) DEFAULT NULL,
  idProyecto INT NOT NULL,
  estado VARCHAR(1) NOT NULL,
  fechaInicio TIMESTAMP NOT NULL,
  numHoras INT NOT NULL,
  posicion INT NOT NULL,
  PRIMARY KEY (id),
  KEY tarea_proyecto (idProyecto),
  CONSTRAINT tarea_proyecto FOREIGN KEY (idProyecto)
REFERENCES proyecto (id) ON DELETE RESTRICT ON
UPDATE CASCADE,
  CONSTRAINT tarea_estado_chk_1 CHECK ((estado IN
('D','P','F','A')))
)

```

Código Fuente 4.3: Diseño físico – Tarea

```

CREATE TABLE usuario (
  nombre VARCHAR(50) NOT NULL,
  correo VARCHAR(100) NOT NULL,
  contrasena VARCHAR(100) NOT NULL,
  refresh_token VARCHAR(100) DEFAULT NULL,
  refresh_token_caducidad DATETIME DEFAULT NULL,
  codigo_contrasena VARCHAR(150) DEFAULT NULL,
  codigo_contrasena_caducidad DATETIME DEFAULT
NULL,
  PRIMARY KEY (correo)
)

```

Código Fuente 4.4: Diseño físico – Usuario

```

CREATE TABLE usuario_grupo (
  id INT NOT NULL AUTO_INCREMENT,
  correoUsuario VARCHAR(100) NOT NULL,
  idGrupo INT NOT NULL,
  rol VARCHAR(1) NOT NULL,
  PRIMARY KEY (id),
  KEY usuario_grupo_grupo (idGrupo),
  KEY usuario_grupo_usuario (correoUsuario),
  CONSTRAINT usuario_grupo_grupo FOREIGN KEY
(idGrupo) REFERENCES grupo (id) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT usuario_grupo_usuario FOREIGN KEY
(correoUsuario) REFERENCES usuario (correo) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT usuario_grupo_chk_1 CHECK ((rol IN
('A','D','U')))
)

```

Código Fuente 4.5: Diseño físico – Usuario_grupo

```

CREATE TABLE usuario_tarea (
  id INT NOT NULL AUTO_INCREMENT,
  correoUsuario VARCHAR(100) NOT NULL,
  idTarea INT NOT NULL,
  PRIMARY KEY (id),
  KEY usuario_tarea_tarea (idTarea),
  KEY usuario_tarea_usuario (correoUsuario),
  CONSTRAINT usuario_tarea_tarea FOREIGN KEY
(idTarea) REFERENCES tarea (id) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT usuario_tarea_usuario FOREIGN KEY
(correoUsuario) REFERENCES usuario (correo) ON
DELETE CASCADE ON UPDATE CASCADE
)

```

Código Fuente 4.6: Diseño físico – Usuario_tarea

```

CREATE TABLE etiqueta (
  id INT NOT NULL AUTO_INCREMENT,
  nombre VARCHAR(50) NOT NULL,
  idProyecto INT NOT NULL,
  color VARCHAR(25) DEFAULT NULL,
  PRIMARY KEY (id),
  KEY etiqueta_proyecto (idProyecto),
  CONSTRAINT etiqueta_proyecto FOREIGN KEY
(idProyecto) REFERENCES proyecto (id) ON DELETE
CASCADE ON UPDATE CASCADE
)

```

Código Fuente 4.7: Diseño físico – Etiqueta

```

CREATE TABLE etiqueta_tarea (
  idEtiqueta INT NOT NULL,
  idTarea INT NOT NULL,
  PRIMARY KEY (idEtiqueta,idTarea),
  KEY etiqueta_tarea_tarea (idTarea),
  CONSTRAINT etiqueta_tarea_etiqueta FOREIGN KEY
(idEtiqueta) REFERENCES etiqueta (id) ON DELETE
CASCADE ON UPDATE CASCADE,
  CONSTRAINT etiqueta_tarea_tarea FOREIGN KEY
(idTarea) REFERENCES tarea (id) ON DELETE CASCADE ON
UPDATE CASCADE
)

```

Código Fuente 4.8: Diseño físico – Etiqueta_tarea

```

CREATE TABLE registro_horas_tarea (
  id INT NOT NULL AUTO_INCREMENT,
  idTarea INT NOT NULL,
  correoUsuario VARCHAR(100) NOT NULL,
  fechaInicio TIMESTAMP NOT NULL,
  fechaFin TIMESTAMP NOT NULL,
  PRIMARY KEY (id),
  KEY registroHorasTarea_tarea (idTarea),
  KEY registroHorasTarea_usuario (correoUsuario),
  CONSTRAINT registroHorasTarea_tarea FOREIGN KEY
(idTarea) REFERENCES tarea (id) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT registroHorasTarea_usuario FOREIGN KEY
(correoUsuario) REFERENCES usuario (correo) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT registro_horas_tarea_chk_1 CHECK
((fechaFin >= fechaInicio))
)

```

Código Fuente 4.9: Diseño físico – Registro_horas_tarea

```

CREATE TABLE registro_fichaje (
  id INT NOT NULL AUTO_INCREMENT,
  correoUsuario VARCHAR(100) NOT NULL,
  fecha TIMESTAMP NOT NULL,
  registro VARCHAR(1) NOT NULL,
  PRIMARY KEY (id),
  KEY registro_fichaje_usuario (correoUsuario),
  CONSTRAINT registro_fichaje_usuario FOREIGN KEY
(correoUsuario) REFERENCES usuario (correo) ON
DELETE CASCADE ON UPDATE CASCADE
)

```

Código Fuente 4.10: Diseño físico – Registro_fichaje


```

CREATE TABLE documento_adjunto (
  id INT NOT NULL AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  descripcion VARCHAR(500) DEFAULT NULL,
  idTarea INT NOT NULL,
  tipo VARCHAR(20) NOT NULL,
  PRIMARY KEY (id),
  KEY documento_adjunto_tarea (idTarea),
  CONSTRAINT documento_adjunto_tarea FOREIGN KEY
(idTarea) REFERENCES tarea (id) ON DELETE CASCADE ON
UPDATE CASCADE
)

```

Código Fuente 4.11: Diseño físico – Documento_adjunto

```

CREATE TABLE vacaciones (
  id INT NOT NULL AUTO_INCREMENT,
  correoUsuario VARCHAR(100) NOT NULL,
  fechaInicio TIMESTAMP NOT NULL,
  fechaFin TIMESTAMP NOT NULL,
  PRIMARY KEY (id),
  KEY vacaciones_usuario (correoUsuario),
  CONSTRAINT vacaciones_usuario FOREIGN KEY
(correoUsuario) REFERENCES usuario (correo) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT vacaciones_chk_1 CHECK ((fechaFin >=
fechaInicio))
)

```

Código Fuente 4.12: Diseño físico – Vacaciones

Capítulo 5

Diseño e implementación del sistema

5.1. Detalles de la implementación

En esta sección se van a detallar algunos aspectos relacionados con la implementación del proyecto desarrollado.

Para comenzar, el desarrollo de la aplicación se dividió en dos proyectos diferenciados: el de *front-end* y el de *back-end*. Cada uno de ellos se trató con las tecnologías descritas anteriormente en la sección 2.3.

En cuanto al de **front-end**, se utilizaron los lenguajes HTML, CSS y JavaScript y, finalmente, se obtuvo la estructura de ficheros mostrada en la Figura 5.1.

Como se puede observar en la mencionada figura, dentro del proyecto se encontraban todos los documentos HTML del proyecto, que incluían todos los formularios con el diseño de la aplicación. Además de esos documentos, también se encontraba el documento Dockerfile, el cual era necesario para inicializar el Docker con la aplicación.

Junto a los documentos se encontraban una serie de carpetas, de entre las cuales destacan tres:

- **css.** CSS (*Cascading Style Sheets*) [13] es un lenguaje que permite aplicar estilos a elementos de diferentes documentos de manera automática. Esta carpeta incluye los dos documentos css con los estilos de los formularios utilizados.

- **vendor.** Esta carpeta incluía otras carpetas con algunas librerías que se utilizaban en la aplicación. Un ejemplo es *fontawesome-free*, que incluía todos los iconos utilizados en las diferentes páginas de la aplicación. Se puede ver la estructura de esta carpeta en la Figura 5.2.
- **js.** Esta carpeta recopilaba todos los documentos JavaScript. Estos documentos contenían todas las funcionalidades de las páginas desarrolladas. Algunos de ellos iban asociados a un documento HTML en concreto y, por ello, utilizaban su mismo nombre. La estructura de esta carpeta se puede observar en la Figura 5.3.

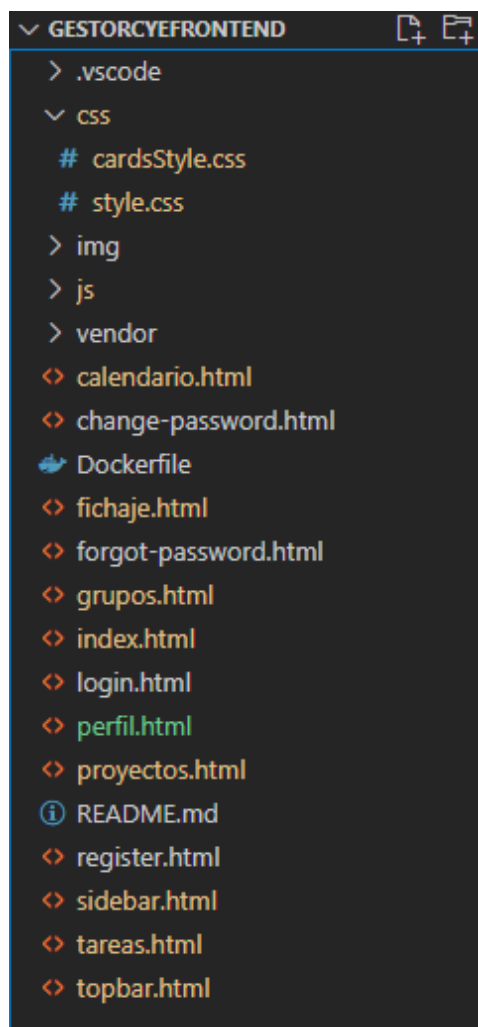


Figura 5.1: Estructura de ficheros del proyecto de front-end.

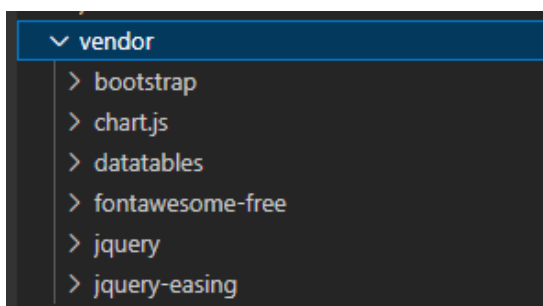


Figura 5.2: Estructura de la carpeta vendor.

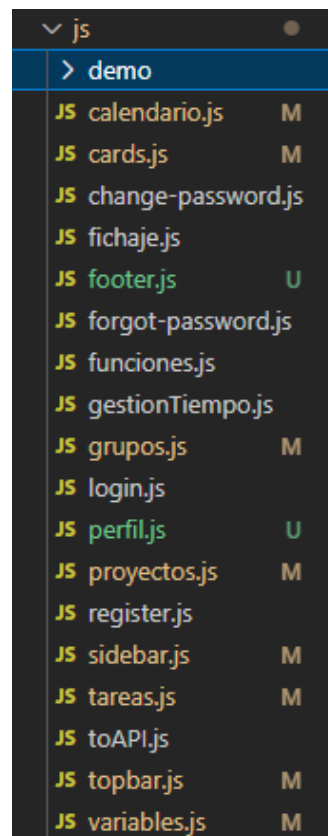


Figura 5.3: Estructura de la carpeta js.

En relación al proyecto de **back-end**, se utilizó .NET Core y C# para su desarrollo. La estructura de ficheros de este proyecto se recopila en las Figuras 5.4, 5.5 y 5.6. Esta estructura se basó en el patrón MVC (Modelo-Vista-Controlador) [14], el cual se suele utilizar en sistemas que requieren el uso de interfaces de usuario.

Según este modelo se puede diferenciar:

- **Modelo.** Es la parte donde se trabajan con los datos de la base de datos. Los documentos de la carpeta *Models* contienen representaciones de las tablas de la base de datos para poder trabajar con sus datos. La estructura de esta carpeta se puede ver en la Figura 5.6.
- **Vista.** La vista incluye todas las interfaces de usuario. En este caso, la parte de la vista se correspondería al proyecto de *front-end*.
- **Controlador.** El controlador es donde se implementan todas las llamadas HTTP que se realicen en la aplicación y sirve de enlace para conectar el modelo con la vista. Todos los controladores realizados se incluyen en la carpeta *Controllers*, cuya estructura se muestra en la Figura 5.5.

En el proyecto de *back-end* también se encuentra la carpeta *Consultas*, que incluye todas las consultas realizadas a la base de datos desde los controladores, *Context*, que conecta los modelos con los controladores, y otras clases con ficheros que incluyen ciertas funcionalidades de la aplicación.

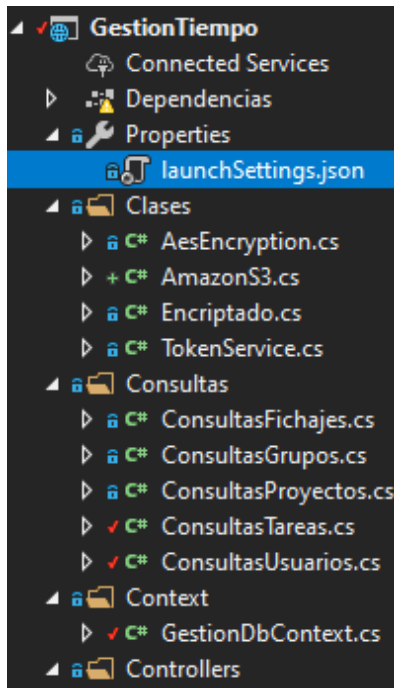


Figura 5.4: Estructura del back-end [I].

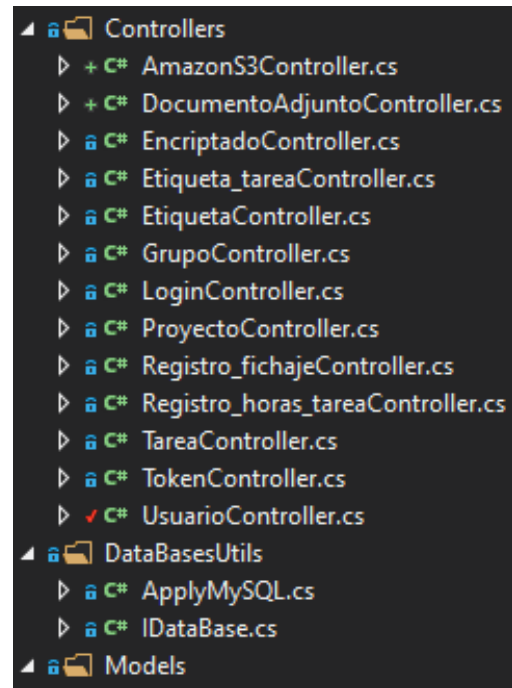


Figura 5.5: Estructura del back-end [II].

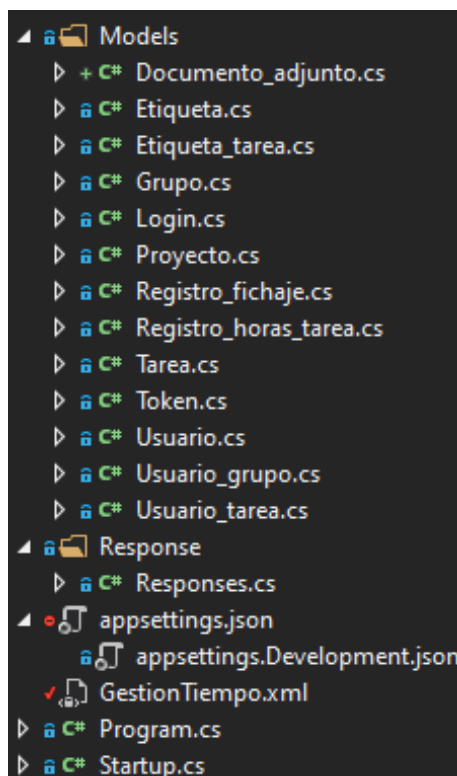


Figura 5.6: Estructura del back-end [III].

A lo largo de la realización de la aplicación se implementaron algunos componentes para realizar funcionalidades especiales.

Una de estas funcionalidades está relacionada con el **encriptado de las contraseñas**. Las contraseñas de los usuarios se almacenaban encriptadas en la base de datos por cuestiones de seguridad. Por este motivo, cuando un usuario se registraba en la aplicación se debía encriptar su contraseña. En el Código Fuente 5.1 aparece cómo se consiguieron encriptar las contraseñas utilizando la librería *BCrypt*.

```
if (UsuarioExists(usuario.correo))
{
    return Unauthorized();
}

string contrasenia = BCrypt.Net.BCrypt.HashPassword(usuario.contrasenia);

usuario.contrasenia = contrasenia;

_contexto.Usuario.Add(usuario);

// Guarda el usuario en la base de datos
await _contexto.SaveChangesAsync();
```

Código Fuente 5.1: Encriptado de la contraseña.

Además, cada vez que el usuario iniciaba sesión, se debía comparar la contraseña introducida con la almacenada que estaba encriptada para comprobar si era correcta. Esta comparación se puede observar en el Código Fuente 5.2.

```
// Recupera el usuario de BBDD
Usuario usuario = _contextoGestion.Usuario.FirstOrDefault(u => u.correo == userLogin.correo);

// Comprueba si el usuario y contraseña son válidos
// Compara la contraseña sin encriptar del frontend con la encriptada de la bbdd
if (usuario == null || !BCrypt.Net.BCrypt.Verify(userLogin.contrasenia, usuario.contrasenia))
{
    return Unauthorized();
}
```

Código Fuente 5.2: Comparación de la contraseña introducida con la encriptada.

Otra de las funcionalidades que hubo que desarrollar fue la del envío de **correos automáticos**. En determinados puntos de la aplicación, el sistema debía enviar correos electrónicos de manera automática. Para conseguirlo, se tuvo que

configurar una cuenta de correo electrónico desde la que mandar los demás correos y configurar el cuerpo de éstos.

En el Código Fuente 5.3 se puede ver la construcción de uno de los correos generados. Para que se realice el envío se ha de indicar la cuenta desde donde se manda el correo, el puerto desde donde se envía, así como el asunto y el mensaje a enviar.

```
// Manda un correo al usuario registrado
MailMessage mail = new MailMessage();
mail.From = new MailAddress(_config["Smtp:From"]);
mail.To.Add(new MailAddress(usuario.correo));

mail.Subject = "Registro GestionTiempo";
mail.Body = "Bienvenido, te has registrado en GestionTiempo correctamente";

var smtpClient = new SmtpClient(_config["Smtp:Host"])
{
    Port = int.Parse(_config["Smtp:Port"]),
    EnableSsl = true,
    UseDefaultCredentials = false,
    Credentials = new NetworkCredential(_config["Smtp:Username"], _config["Smtp:Password"]),
};

smtpClient.Send(mail);
```

Código Fuente 5.3: Construcción de un correo automático.

En el Anexo 1 se muestran los diferentes correos automáticos que se podían enviar desde la aplicación.

La última de las funcionalidades a destacar del sistema fueron los **tokens**. Éstos servían para controlar que los usuarios de la aplicación tuviesen la sesión iniciada y, también, para determinar qué operaciones podían hacer cada uno de ellos.

Cada usuario contaba con dos *tokens*, que se creaban al iniciar sesión: el *token* y el *refreshToken*. Ambos tenían un tiempo de validez, siendo el *refreshToken* el que tenía una duración mayor.

En el Código Fuente 5.4 se muestra el fragmento de código ejecutado al iniciar sesión con el que se crean ambos *tokens* y los métodos utilizados para ello se muestran en el Código Fuente 5.5.


```

// Añade rol al token
string rol = "1";

List<Claim> claims = new List<Claim>
{
    new Claim(ClaimTypes.Name, userLogin.correo),
    new Claim(ClaimTypes.Role, rol),
};

// Genera el token de acceso y el refresh token
string accessToken = _tokenService.GenerateAccessToken(_config, claims);
string refreshToken = _tokenService.GenerateRefreshToken();

// Guardo el Refresh token en la BDD
usuario.refresh_token = refreshToken;
usuario.refresh_token_caducidad = DateTime.Now.AddDays(7);

_contextoGestion.SaveChanges();

```

Código Fuente 5.4: Generación de los tokens al iniciar sesión.

```

// GENERA EL TOKEN DE ACCESO
public string GenerateAccessToken(IConfiguration configuration, IEnumerable<Claim> claims)
{
    var secretKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(configuration["JwtAuthentication:SecurityKey"]));
    var signinCredentials = new SigningCredentials(secretKey, SecurityAlgorithms.HmacSha256);

    var tokenOptions = new JwtSecurityToken(
        issuer: configuration["JwtAuthentication:ValidIssuer"],
        audience: configuration["JwtAuthentication:ValidAudience"],
        claims: claims,
        expires: DateTime.Now.AddMinutes(5), // Validez del token
        signingCredentials: signinCredentials
    );

    var tokenString = new JwtSecurityTokenHandler().WriteToken(tokenOptions);
    return tokenString;
}

// GENERA EL REFRESH TOKEN
public string GenerateRefreshToken()
{
    var randomNumber = new byte[32];
    using (var rng = RandomNumberGenerator.Create())
    {
        rng.GetBytes(randomNumber);
        return Convert.ToBase64String(randomNumber);
    }
}

```

Código Fuente 5.5: Métodos utilizados para generar los tokens.

Todas las operaciones que se realizaban con las llamadas HTTP en los controladores del *back-end* estaban protegidas de manera que, si no se disponía de un *token* activo, y por lo tanto válido, no se permitía realizar la operación. En el

Código Fuente 5.6 se observa el método *Authorize* en una llamada HTTP, el cual se encarga de proteger estas operaciones.

```
// GET
// api/v1/grupo

/// <summary>
/// Obtener todos los grupos.
/// </summary>
/// <returns>Lista de grupos.</returns>
/// <response code="200">Devuelve la lista de grupos.</response>
/// <response code="500">Si fue un error del servidor</response>
[HttpGet, Authorize(Roles = "1")]
[ProducesResponseType(200)]
[ProducesResponseType(500)]
[EnableCors("MiPolitica")]
public IActionResult GetGrupos()
```

Código Fuente 5.6: Authorize para asegurar poder ejecutar la operación.

Si un *token* caducaba se debía tratar utilizando el *refreshToken*. Cuando un usuario intentaba realizar una operación pero el *token* ya había caducado, la llamada HTTP devolvía un error. En ese momento, desde el *front-end* se realizaba otra llamada automáticamente para comprobar el *refreshToken*.

Si éste seguía siendo válido, se generaban nuevos *tokens* que eran asignados al usuario y se volvía a realizar la llamada HTTP original, la cual en esta ocasión ya se realizaba correctamente.

En el caso de que el *refreshToken* también hubiese caducado, se redirigía al usuario a la página de iniciar sesión porque se entendía que la sesión había caducado.

En el Código Fuente 5.7 se observa el tratamiento del *refreshToken* y la posterior generación de los nuevos *tokens*, en caso de ser válido.

```

public IActionResult Refresh(Token tokenApiModel)
{
    if (tokenApiModel is null)
    {
        return BadRequest("Invalid client request");
    }

    string accessToken = tokenApiModel.accessToken;
    string refreshToken = tokenApiModel.refreshToken;

    ClaimsPrincipal principal = _tokenService.GetPrincipalFromExpiredToken(_config, accessToken);
    string username = principal.Identity.Name; //Obtiene el email del usuario

    Usuario usuario = _contextoGestion.Usuario.SingleOrDefault(u => u.correo == username);

    if (usuario == null || usuario.refresh_token != refreshToken || usuario.refresh_token_caducidad <= DateTime.Now)
    {
        return BadRequest("Invalid client request");
    }

    string newAccessToken = _tokenService.GenerateAccessToken(_config, principal.Claims);
    string newRefreshToken = _tokenService.GenerateRefreshToken();

    // Guardo el Refresh token en la BDD
    usuario.refresh_token = newRefreshToken;
    usuario.refresh_token_caducidad = DateTime.Now.AddDays(7);

    _contextoGestion.SaveChanges();
}

```

Código Fuente 5.7: Tratamiento del refreshToken.

En el Código Fuente 5.8 aparece la llamada HTTP original, indicando que en caso de haber un error se compruebe el *refreshToken*. A su vez se indica que, si éste es válido actualice los tokens y vuelva a realizar la llamada original, y en caso contrario redirija a la página de iniciar sesión.

Toda esta funcionalidad era llevada a cabo por el sistema de manera automática sin que el usuario se enterase de estas operaciones.

```

function toApi(url, method, data, resultSuccess, resultError){
    token = localStorage.getItem("token");

    var errorDoc = document.getElementById('error')
    errorDoc.style.color = 'red'
    var mensajesError = [];
    switch (resultError){
        case "refresh": // Actualizar token expirado
            resError = function(xhr, status, error){
                if(error=="Unauthorized"){
                    //llamada post
                    var url_refresh = urlLocalHost + "/token/refresh"
                    var token = localStorage.getItem("token");
                    var refreshToken = localStorage.getItem("refreshToken");

                    var data_refresh = JSON.stringify({
                        "accessToken": token,
                        "refreshToken": refreshToken
                    })
                    resultSuccess_refresh = function(results, textStatus) {
                        localStorage.setItem("token", Object.values(results)[0])
                        localStorage.setItem("refreshToken", Object.values(results)[1])
                        toApi(url, method, data, resultSuccess, resultError);
                    }
                    resultError_refresh = function(xhr, status, error){
                        localStorage.removeItem("token");
                        localStorage.removeItem("refreshToken");
                        location.href="login.html" ;
                    }
                    toApi(url_refresh,"POST", data_refresh, resultSuccess_refresh, resultError_refresh);
                }
            }
    }
}

```

Código Fuente 5.8: Llamada al refreshToken.

5.2. Diseño de las páginas

5.2.1. Página de registro

La página de registro es aquella en la que cualquier usuario se puede registrar en la aplicación. Su diseño está centrado en un formulario donde el usuario puede introducir sus datos junto con un botón para realizar el registro. Los datos a introducir son el nombre del usuario, un correo electrónico y una contraseña.

Además, en la parte inferior hay un par de enlaces a las páginas de inicio de sesión y cambio de contraseña. El diseño completo de esta página se puede ver en la Figura 5.7.

Al realizar el registro, el usuario recibe un correo electrónico de bienvenida a la aplicación. Se puede ver un ejemplo de este tipo de correo en la Figura A.1 del Anexo A.

Para realizar esta página también hubo que tener en cuenta el encriptado de la contraseña del usuario, tal y como se explicó en la sección 5.1.

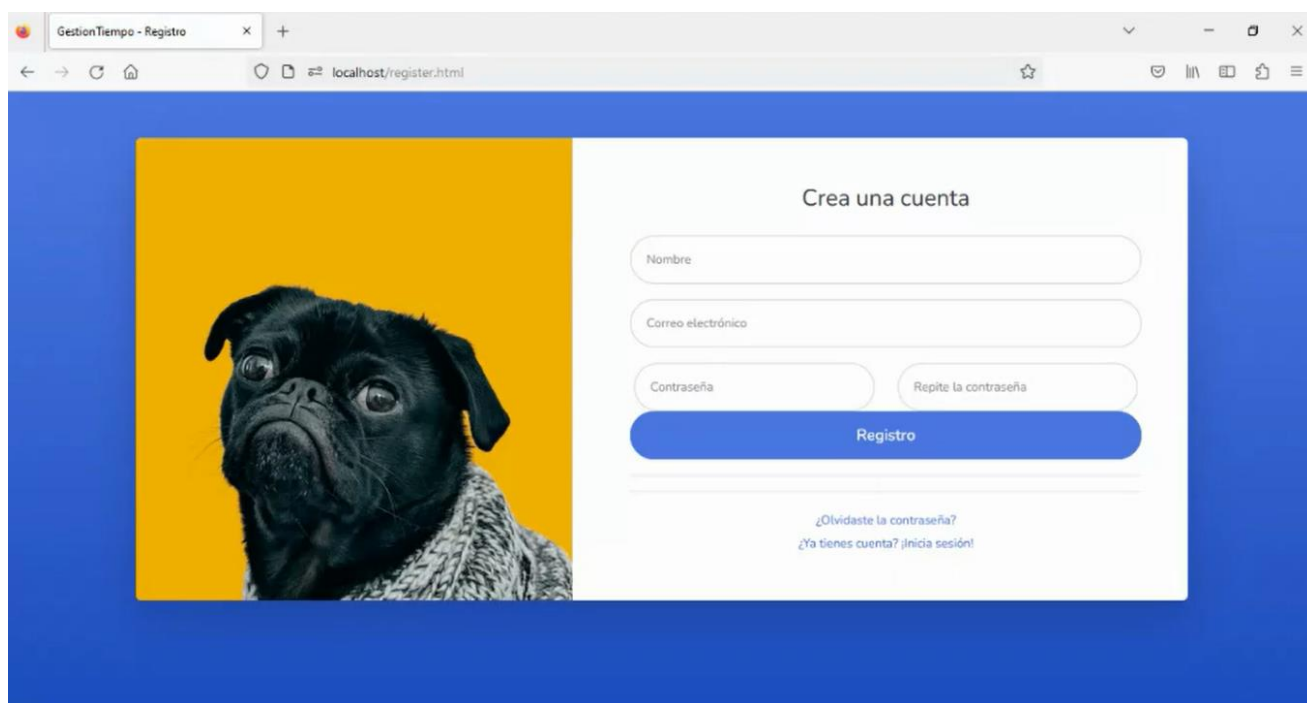


Figura 5.7: Página de registro.

5.2.2. Página de inicio de sesión

La página de inicio de sesión permite a un usuario registrado poder entrar en la aplicación. Su diseño es muy sencillo, pues sólo contiene un formulario donde introducir su correo y su contraseña y un botón para iniciar sesión.

La estructura de la página es muy similar a la página de registro y también contiene un par de enlaces en la parte inferior, en esta ocasión a las páginas de registro y cambio de contraseña.

En la Figura 5.8 se muestra el diseño de esta página.

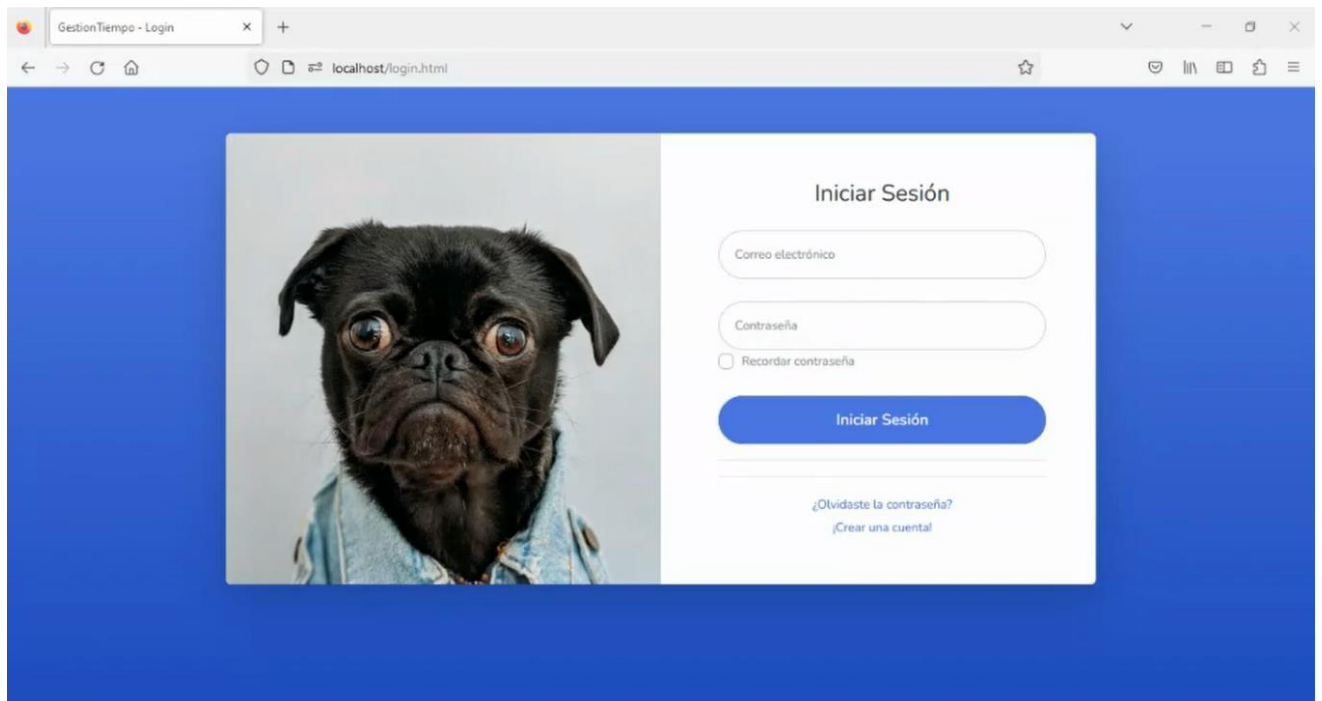


Figura 5.8: Página de inicio de sesión.

5.2.3. Páginas de cambio de contraseña

La página de olvido de contraseña tiene la particularidad de que realmente está formada por dos páginas distintas.

La página principal, a la que se va a nombrar como página de olvido de contraseña, tiene una estructura similar a las páginas de registro y de inicio de sesión, con un enlace a ambas en la parte inferior, tal y como se puede apreciar en la Figura 5.9.

En esta página el usuario ha de introducir su correo electrónico. Al pulsar en el botón de recordar contraseña se enviará un correo automático, como el que se puede observar en la Figura A.2 del Anexo A, a la dirección indicada.

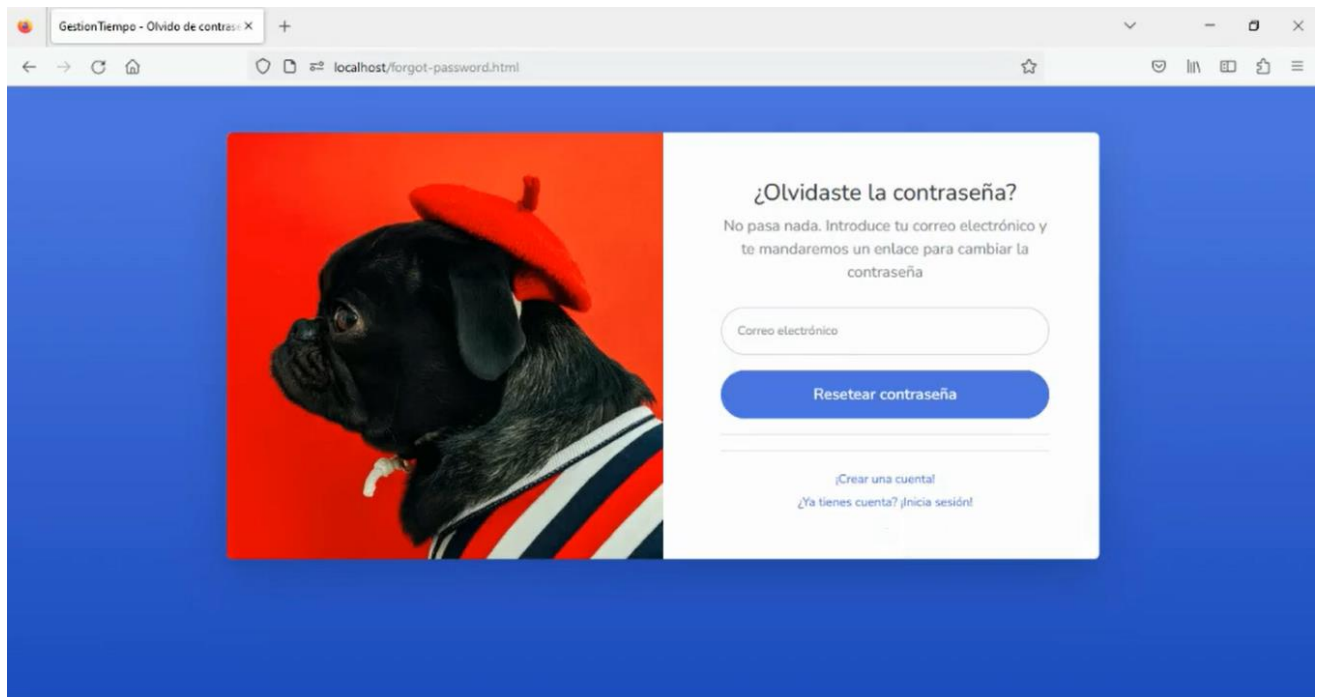


Figura 5.9: Página de olvido de contraseña.

En el correo enviado hay un enlace a la página de cambio de contraseña, mostrada en la Figura 5.10. Esta página es una página temporal, sólo accesible desde el correo, a la que se le ha dado un tiempo de validez de unos pocos minutos. De esta manera, pasado ese tiempo si el usuario intenta acceder de nuevo a esta página será redirigido a la página de olvido de contraseña.

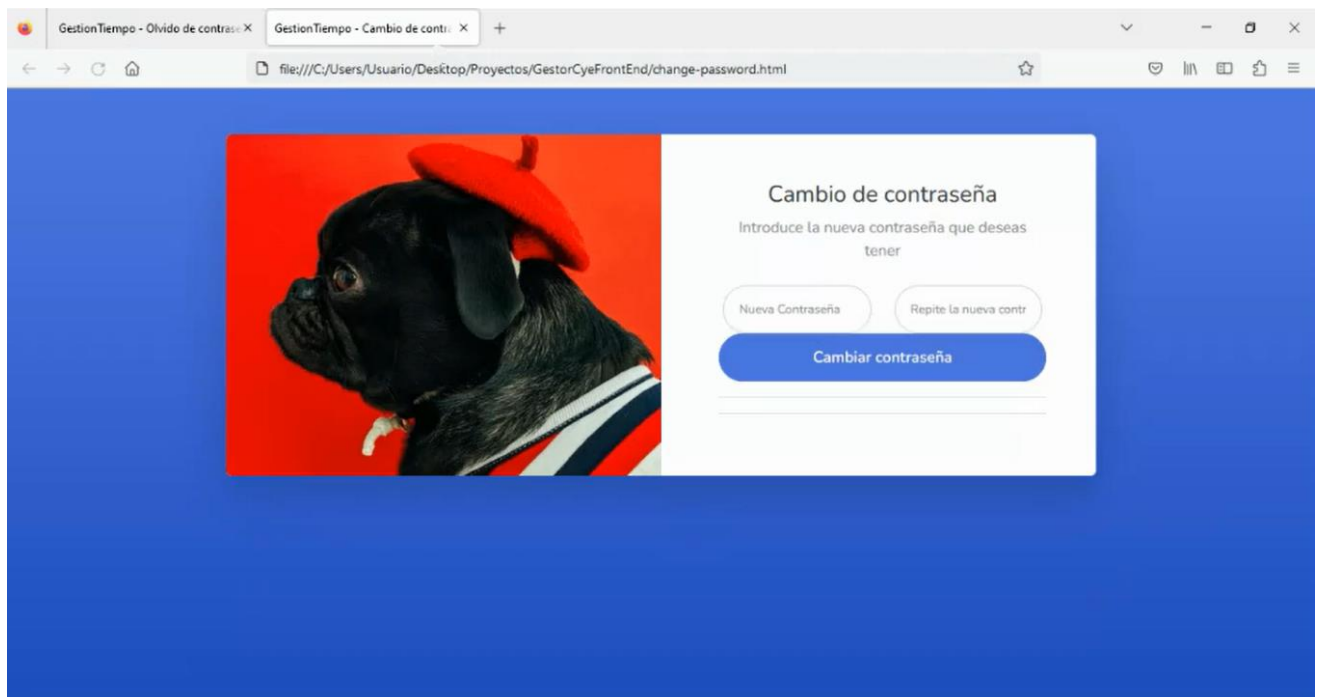


Figura 5.10: Página de cambio de contraseña.

5.2.4. Página de inicio

A continuación se describe la página de inicio de la aplicación.

Esta página cuenta con cuatro tarjetas centrales relacionadas con diferentes páginas de la aplicación: la página de grupos, la del calendario, la del fichaje y el perfil de usuario.

Cada tarjeta contiene el nombre de la página con la que se relaciona, una breve descripción de la misma y un icono característico para diferenciarse del resto de tarjetas. Pulsando en el nombre se puede acceder a la página correspondiente. La Figura 5.11 contiene el diseño completo de esta página.



Figura 5.11: Página de inicio.

En esta página se pueden destacar tres componentes que se van a mantener en las demás páginas de la aplicación.

El primero de ellos es la barra lateral, la cual se encuentra en el lateral izquierdo de la página y es de color azul. Esta barra contiene enlaces a las páginas de inicio, fichaje, calendario y grupos. Los iconos que acompañan a estas tres últimas páginas son los mismos que aparecen en las tarjetas descritas previamente.

El nombre de GestionTiempo es el nombre provisional que se le dio a la aplicación y redirige también a la página de inicio.

La barra lateral se muestra por defecto tal y como aparece en la Figura 5.12, pero cuenta con un botón inferior que permite contraer la barra como en la Figura 5.13. De esta manera, ocupa un menor espacio en pantalla.

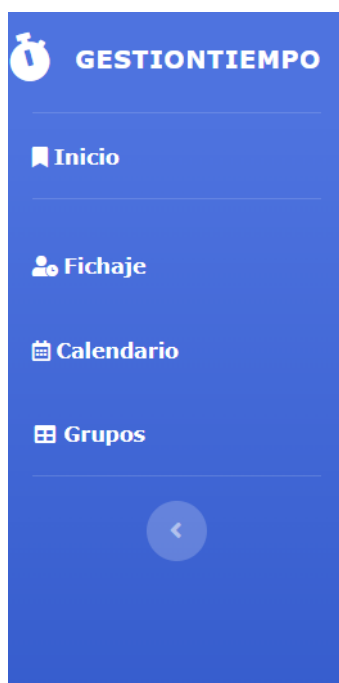


Figura 5.12: Barra lateral completa.

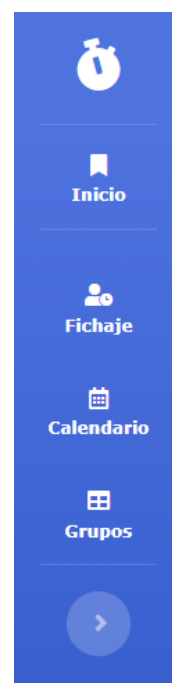


Figura 5.13: Barra lateral contraída.

Por otro lado, en la parte superior se encuentra una barra en cuyo extremo derecho aparece el correo del usuario que haya iniciado sesión (Figura 5.14). Si se pulsa en esta zona se muestra un desplegable, como el de la Figura 5.15, que permite acceder al perfil del usuario o cerrar sesión.

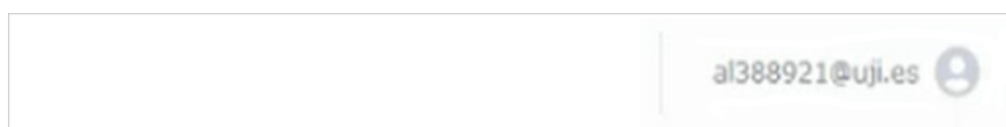


Figura 5.14: Barra superior.

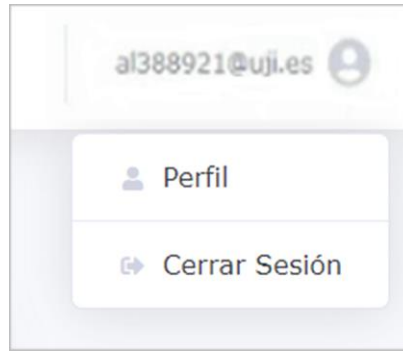


Figura 5.15: Desplegable de la barra superior.

Finalmente, en la parte inferior hay un pie de página que se mantiene durante toda la aplicación y se puede observar en la Figura 5.16.



Figura 5.16: Pie de página.

5.2.5. Página de perfil de usuario

La página del perfil contiene todos los datos del usuario que ha iniciado sesión. En el momento de finalizar la estancia, únicamente se mostraban el nombre y el correo electrónico, tal y como se puede ver en la Figura 5.17, aunque la empresa tenía intención de añadir más datos en un futuro.

Los datos del perfil no se pueden editar por defecto. Para habilitar su edición se ha de pulsar el botón azul de la parte derecha. Al realizar esto, se cambia el icono del botón al mostrado en la Figura 5.18 y se permite editar cualquier dato del usuario, a excepción del correo electrónico.

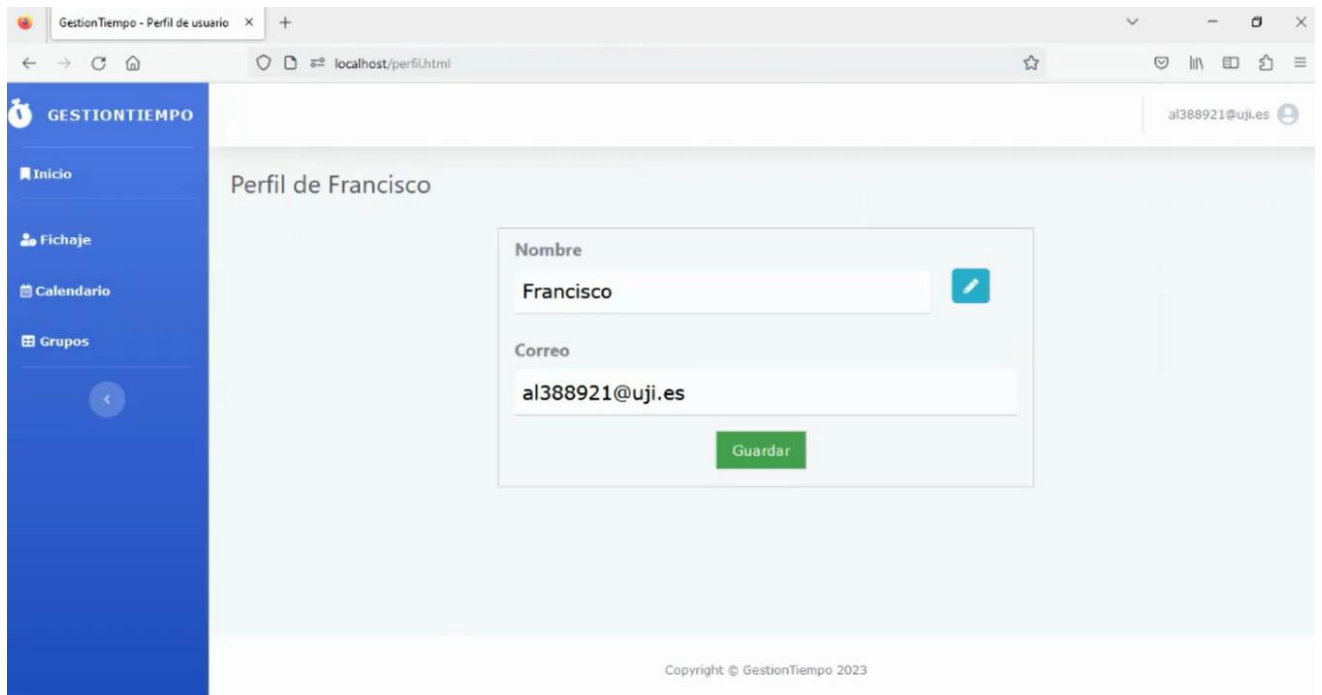


Figura 5.17: Perfil de usuario.

En el caso de querer guardar los cambios, se ha de pulsar en el botón inferior de guardar mientras que, si se desea cancelar esos cambios, se puede volver a pulsar el botón de edición para volver al estado inicial de la Figura 5.17.

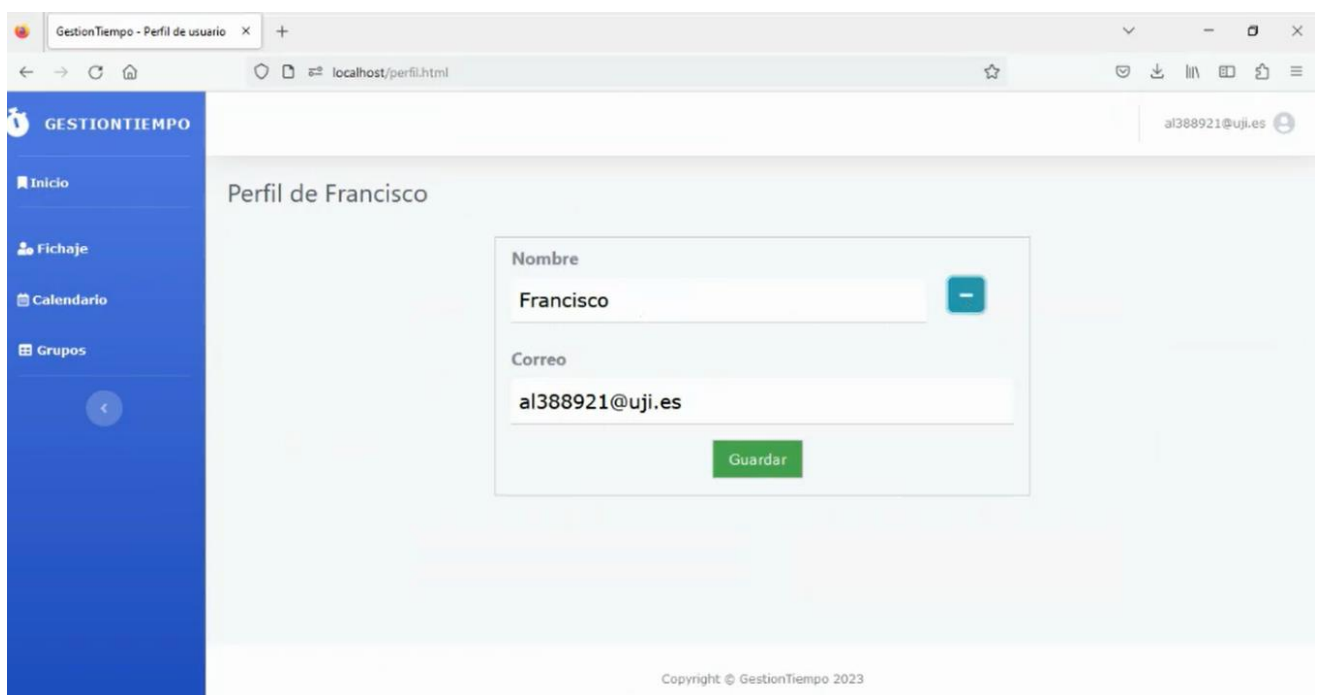


Figura 5.18: Perfil de usuario con la edición de datos habilitada.

5.2.6. Página de grupos

La página de grupos es una de las páginas principales de la aplicación. El diseño de la página está enfocado a mostrar un listado con todos los grupos a los que pertenece el usuario y se muestra en la Figura 5.19.

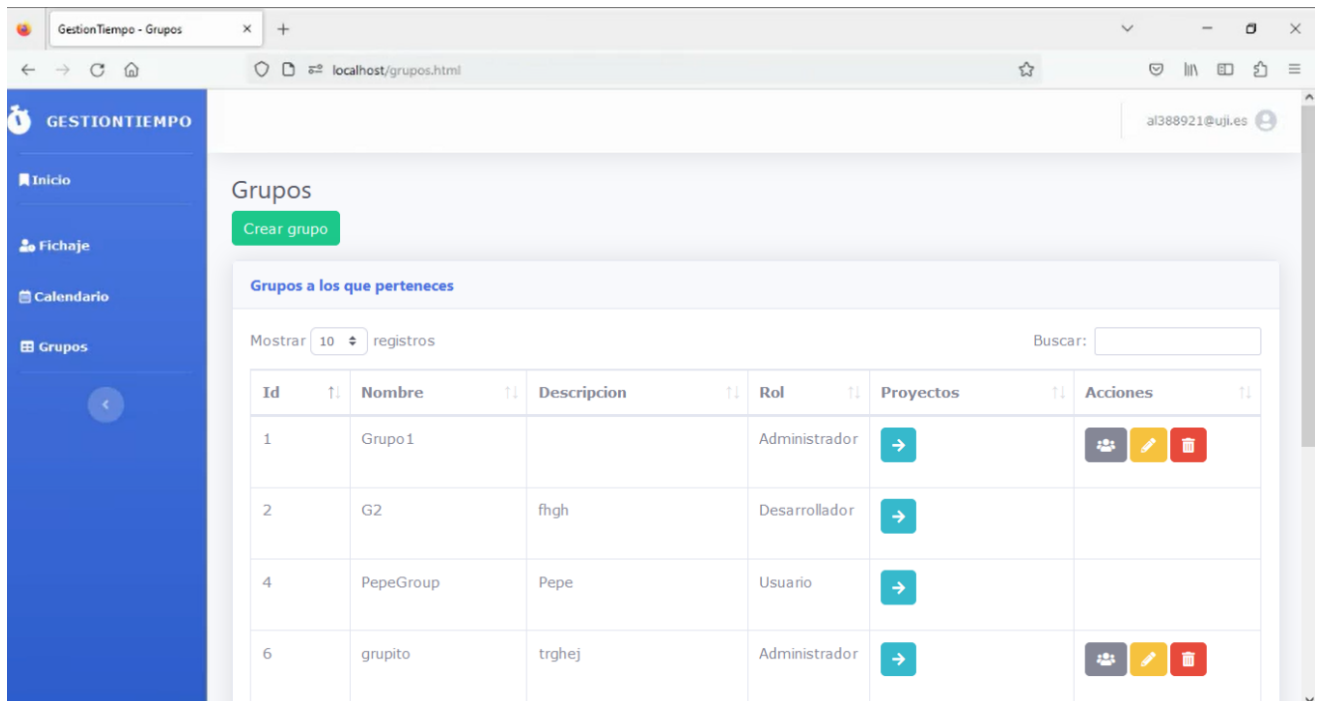


Figura 5.19: Página de grupos.

En la parte superior del listado hay un botón para crear grupos. Al pulsarlo, se muestra la pestaña de la Figura 5.20 para introducir los datos del nuevo grupo.

The modal form for creating a group has the following structure:

- Header: Nombre del grupo (with a close button 'X')
- Input field: Nombre de grupo
- Section: Descripción (Opcional)
- Input field: Descripción
- Button: Crear grupo

Figura 5.20: Pestaña para crear grupos.

En el listado, para cada uno de los grupos se muestra su identificador (id), su nombre y su descripción. Además, también se muestra el rol que tiene el usuario en cada grupo.

En la columna Proyectos hay un enlace a la página de los proyectos del grupo seleccionado.

En la columna Acciones aparecen tres botones con funcionalidades para gestionar los grupos. Estas funcionalidades sólo están disponibles para el administrador del grupo. Se pueden observar los botones con mayor detalle en la Figura 5.21.



Figura 5.21: Botones con las acciones de cada grupo.

El primer botón, de color gris, es el que permite gestionar los miembros del grupo. Al pulsarlo, se abre la pestaña de la Figura 5.22.

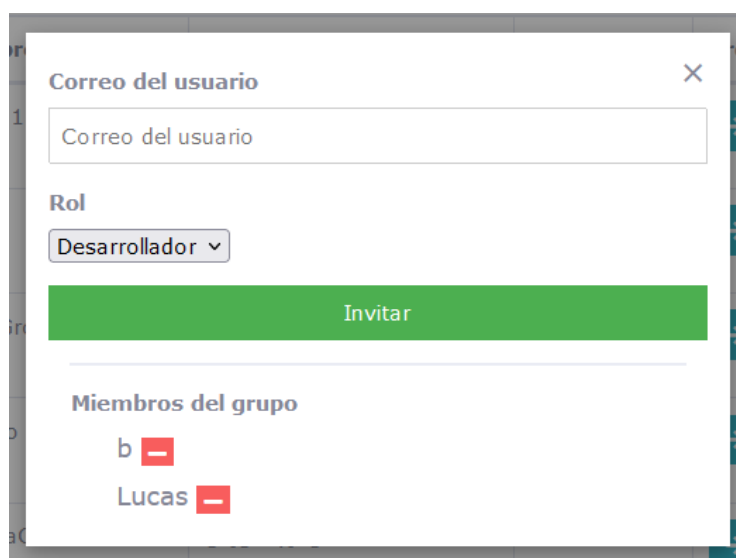


Figura 5.22: Pestaña para gestionar miembros del grupo.

En la parte superior de la pestaña se puede invitar a un miembro, indicando su correo y el rol que tendrá en el grupo: desarrollador o usuario. Al realizar esta

acción, el nuevo miembro recibirá un correo con una invitación similar al de la Figura A.3 del Anexo A.

En la parte inferior aparecen listados los miembros actuales del grupo, a excepción del administrador. Al lado de cada miembro hay un botón para eliminarlo del grupo, si así se desea.

El segundo botón de la columna Acciones, de color amarillo, es el de editar el grupo. Este botón abre la pestaña de la Figura 5.23, que contiene los datos actuales del grupo. Si éstos se modifican y se pulsa en “Editar grupo”, se actualizan los datos y el listado de grupos.



Figura 5.23: Pestaña para editar grupos.

El botón rojo de la columna Acciones permite eliminar el grupo con todo su contenido.

5.2.7. Página de proyectos

La página de proyectos muestra el listado de proyectos de un grupo en concreto. Como se puede observar en la Figura 5.24, la estructura de la página es idéntica a la de la página de grupos.

Los cambios que se pueden apreciar son:

- En el título superior de la página se puede ver el nombre del grupo al que pertenecen los proyectos mostrados.

- En lugar de columna Proyectos, hay una columna Tareas. Su funcionalidad consiste en permitir acceder a la lista de tareas del proyecto seleccionado.
- En la columna Acciones sólo hay dos botones: el de editar el proyecto y el de eliminarlo junto a su contenido.
- Los botones de crear y editar proyectos abren unas pestañas prácticamente idénticas a las mostradas en las Figuras 5.20 y 5.23, respectivamente. El único cambio respecto a esas Figuras es la palabra “grupo” por “proyecto”.

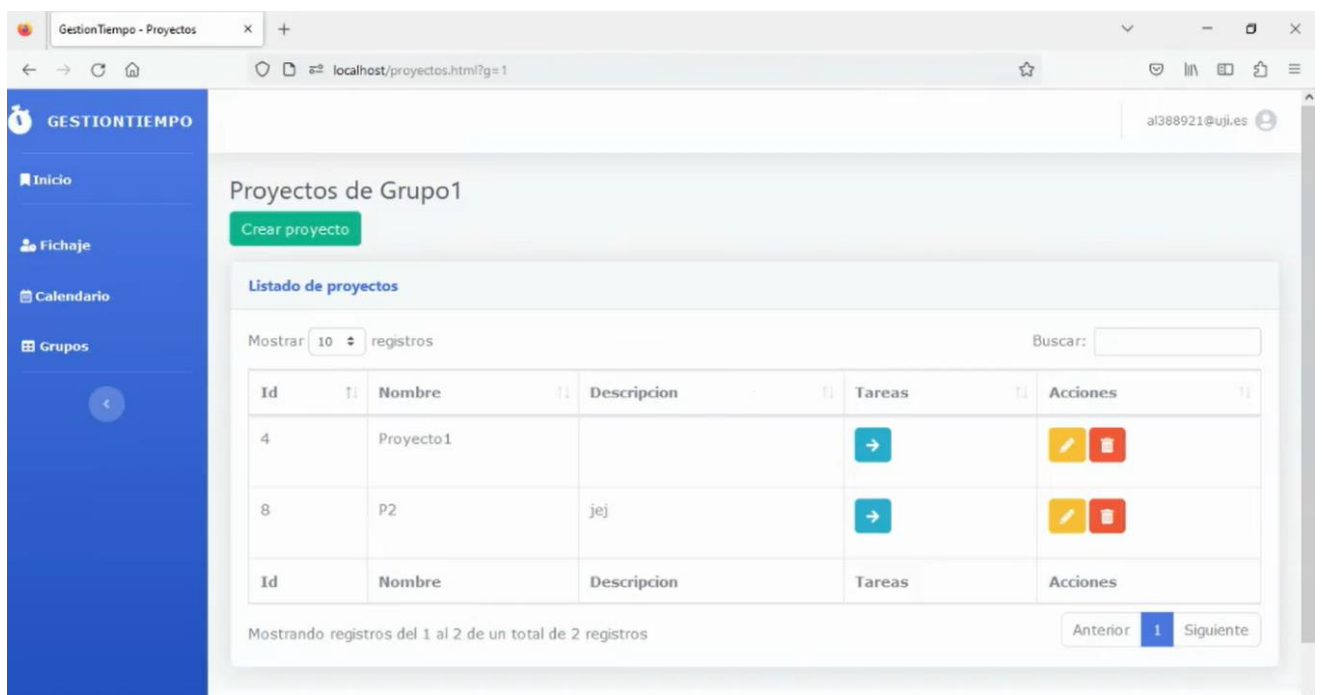


Figura 5.24: Página de proyectos en caso de ser Administrador.

Ahora bien, el diseño mostrado en la Figura 5.24 sólo se muestra en caso de ser el administrador del grupo. Si por lo contrario el usuario tiene el rol de Desarrollador o Usuario, el diseño mostrado es el de la Figura 5.25.

En este caso, no se permite gestionar los proyectos y por lo tanto, no hay botones para crear, editar o borrar proyectos.

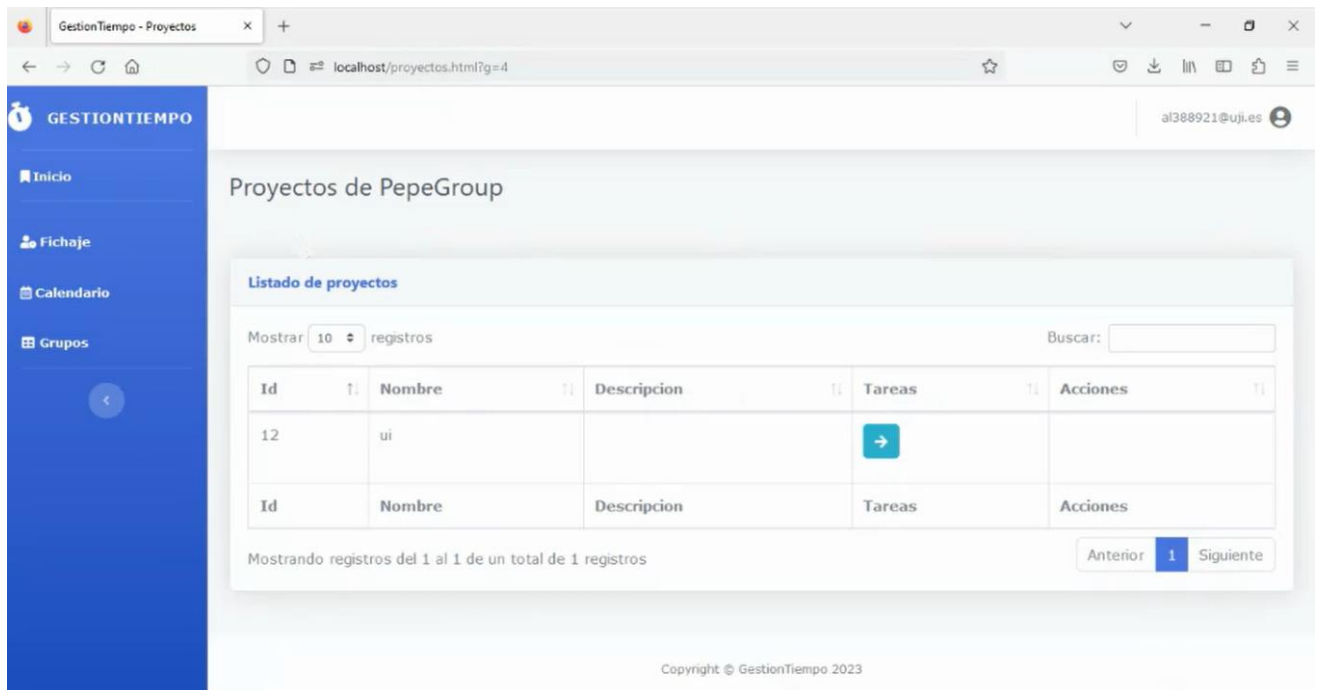


Figura 5.25: Página de proyectos en caso de ser Desarrollador o Usuario.

5.2.8. Página de tareas

La página de tareas contiene el listado de todas las tareas de un proyecto. Esta es considerada una de las páginas más importantes de la aplicación y por ello tiene un diseño único.

Tal y como se muestra en la Figura 5.26, la página está formada por tres columnas fácilmente diferenciadas por su color. Cada columna contiene tarjetas, que se corresponden con las tareas. Las diferentes tarjetas son totalmente interactivas, se pueden mover de posición dentro de una misma columna o mover de una columna a otra. También se pueden pulsar, abriendo la pestaña mostrada en la Figura 5.28, para poder consultar toda la información relacionada con la tarea.

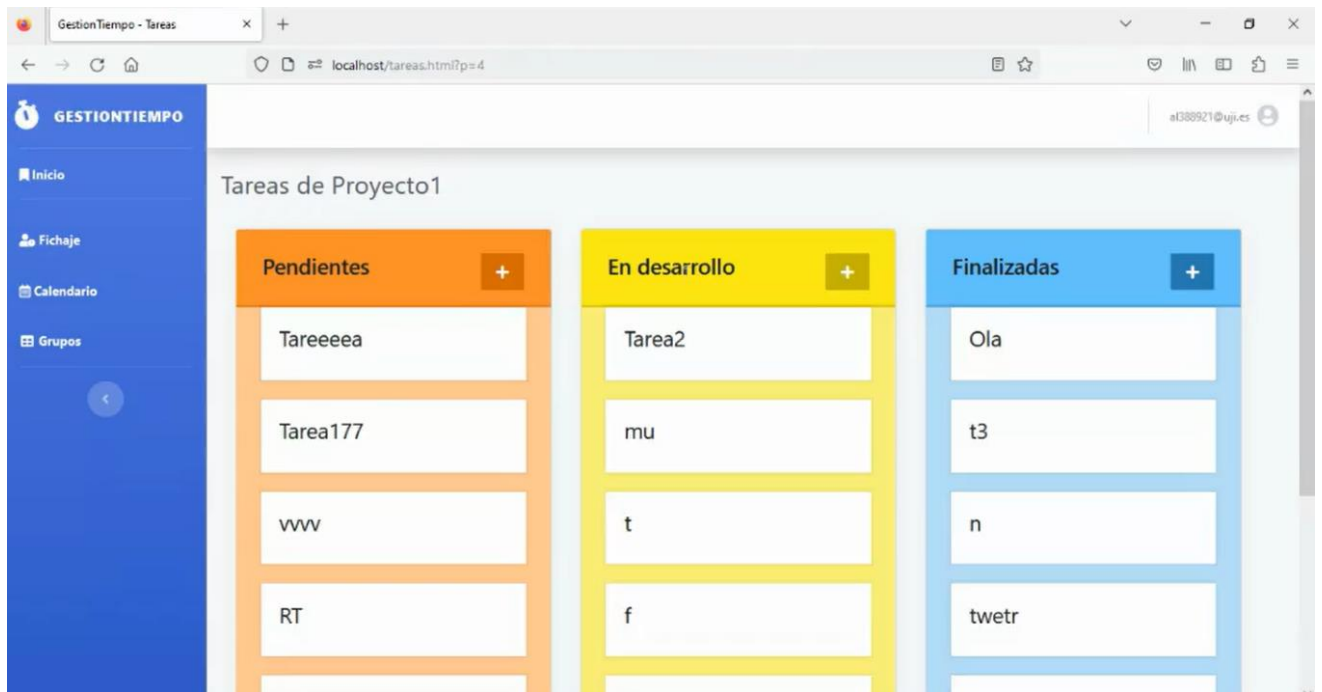


Figura 5.26: Página de tareas en caso de ser Administrador o Desarrollador.

Las tareas se agrupan en una columna u otra en función de su estado: pendientes, en desarrollo o finalizadas. Gracias al diseño de las columnas y la interactividad de las tarjetas es muy fácil gestionar las tareas de un proyecto, ver cuáles se están desarrollando en cada momento y todas las que están pendientes de hacer.

En la parte superior de cada columna existe un botón con un símbolo + que permite crear una nueva tarea en dicha columna. Al pulsar en este botón se abre la pestaña de la Figura 5.27 para introducir los datos necesarios para crear la nueva tarea.

Nombre de la tarea ×

Descripcion (Opcional)

Número de horas esperado

Crear tarea

Figura 5.27: Pestaña para crear tareas.

En relación a la información que contiene cada tarea, como se puede ver en la Figura 5.28, cada una tiene un nombre y una descripción de la misma. Estos dos campos se pueden editar, siempre y cuando se guarden los cambios al pulsar en el botón de Guardar. Además, las tareas tienen etiquetas y responsables, los cuales se muestran bajo el nombre de la tarea.

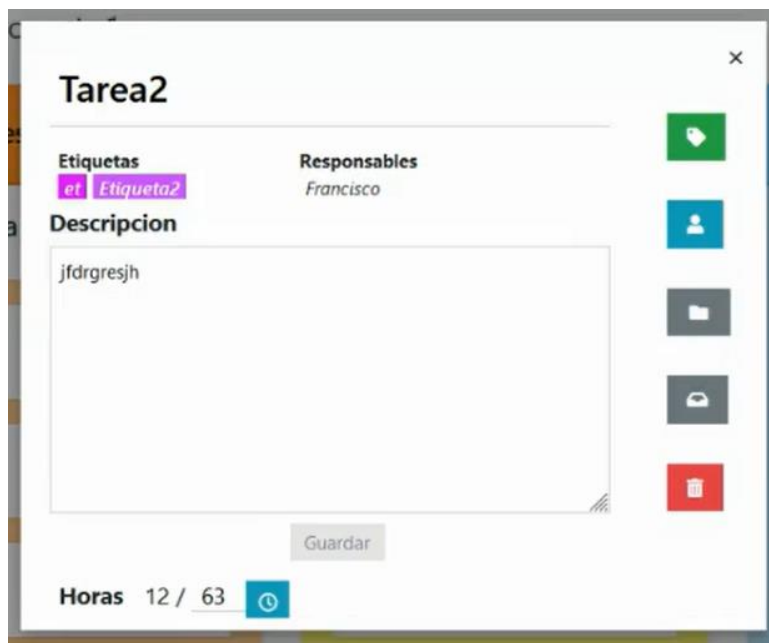


Figura 5.28: Pestaña de la tarea en caso de ser Administrador o Desarrollador.

En la parte inferior de la pestaña se muestran las horas de la tarea. El formato es muy sencillo y se corresponde a:

Número de horas realizadas / Número de horas esperadas

Al crear una tarea, se indica el número de horas esperadas para realizar esa tarea. Por su parte, el número de horas realizadas se obtiene al sumar los registros de horas realizadas en esa tarea por parte de todos los usuarios del grupo de trabajo.

Para realizar un nuevo registro de horas en una tarea basta con pulsar el icono azul de la parte inferior para abrir la pestaña de la Figura 5.29 y realizar el registro. Al realizar un nuevo registro se actualizarán las horas realizadas de la tarea.

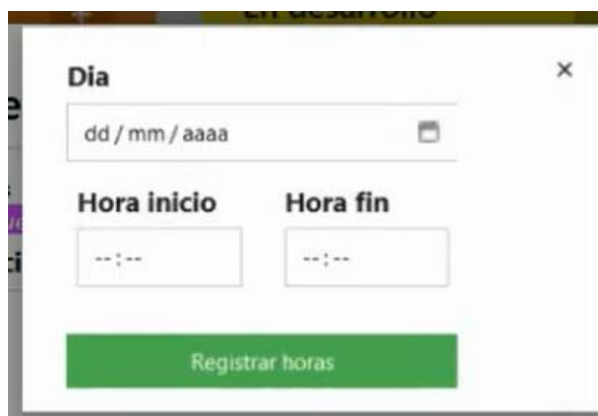


Figura 5.29: Pestaña para registrar horas realizadas en una tarea.

En el lado derecho de la tarea existen una serie de botones que permiten realizar diferentes funcionalidades. Todos ellos únicamente están disponibles para los usuarios con el rol de Administrador o Desarrollador. En la Figura 5.37 se puede ver el diseño de estos botones y el de registrar horas con más detalle.

La primera de las funcionalidades, correspondiente al botón verde, es la de gestionar las etiquetas. Al pulsar en el botón se abre un listado con todas las etiquetas del proyecto actual, como el mostrado en la Figura 5.30.

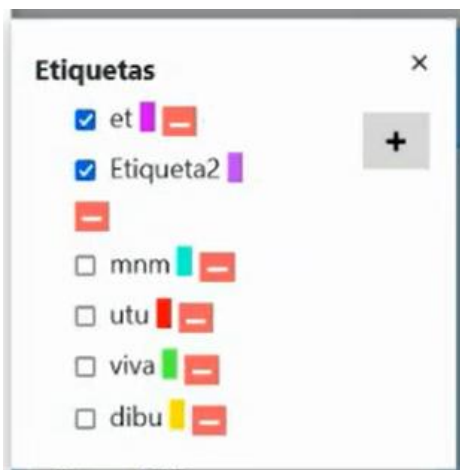


Figura 5.30: Pestaña para gestionar etiquetas.

En el listado de etiquetas, en el lateral existe un botón que abre la pestaña de la Figura 5.31 y permite crear una nueva etiqueta en el proyecto actual. Para borrar una etiqueta del proyecto, en el listado existe un botón rojo al lado de cada etiqueta. Aunque las etiquetas sean creadas o borradas en una tarea en concreto, estas acciones afectarán a todas las demás tareas del mismo proyecto.

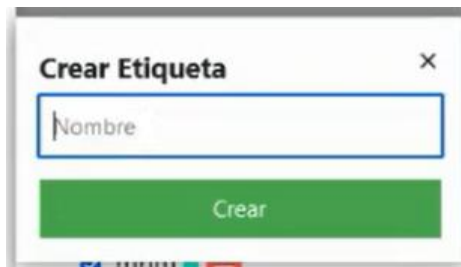


Figura 5.31: Pestaña para crear etiquetas.

Dentro del listado, se pueden seleccionar todas aquellas etiquetas que se deseen para la tarea actual. Las seleccionadas se mostrarán en la pestaña de la tarea (Figura 5.28), actualizando la sección Etiquetas.

La segunda funcionalidad de las tareas, correspondiente al botón azul, es la de gestionar los responsables de la tarea. Cada tarea puede tener uno o más responsables de entre todos los miembros del grupo de trabajo. Al pulsar el botón, aparece el listado de miembros del grupo, como el mostrado en la Figura 5.32. Aquellos responsables elegidos se mostrarán en la sección Responsables de la tarea (Figura 5.28).

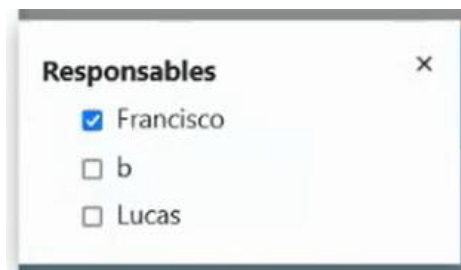


Figura 5.32: Pestaña para gestionar responsables.

La tercera funcionalidad es la de gestionar los documentos adjuntos y corresponde al primer botón de color gris. En una tarea se pueden adjuntar documentos. Al pulsar en este botón se puede ver el listado de los documentos adjuntos de la tarea actual, de aspecto similar al de la Figura 5.33.



Figura 5.33: Pestaña para gestionar documentos adjuntos.

Al lado del nombre de cada documento hay un botón azul, para ver el contenido del documento, y uno rojo, para eliminar el adjunto. Si se desea adjuntar un nuevo documento, se ha de pulsar el botón gris junto al listado y seleccionar el documento a adjuntar en la pestaña de la Figura 5.34.

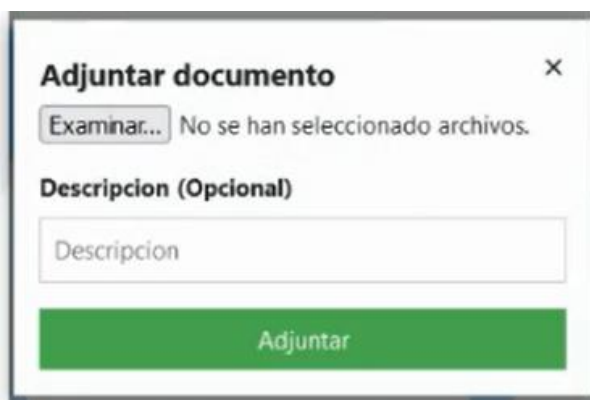


Figura 5.34: Pestaña para adjuntar un documento.

La siguiente funcionalidad de las tareas consiste en archivar la tarea seleccionada y corresponde al segundo botón de color gris. Al archivar la tarea, ésta desaparece de la página de tareas y ya no se muestra, pero no se borra de la base de datos de la aplicación.

La última funcionalidad, correspondiente al botón rojo, es la de borrar la tarea. Al realizar esta acción, la tarea es eliminada por completo, tanto de la página de tareas como de la base de datos.

Para terminar esta sección, a continuación en la Figura 5.35 se muestra el aspecto de la página de tareas en caso de tener el rol de Usuario.

A nivel de diseño, el único cambio apreciable es la falta del botón para crear tareas. No obstante, las tarjetas de las tareas no se pueden desplazar en esta ocasión, solamente se puede pulsar en ellas para ver la información de la tarea.

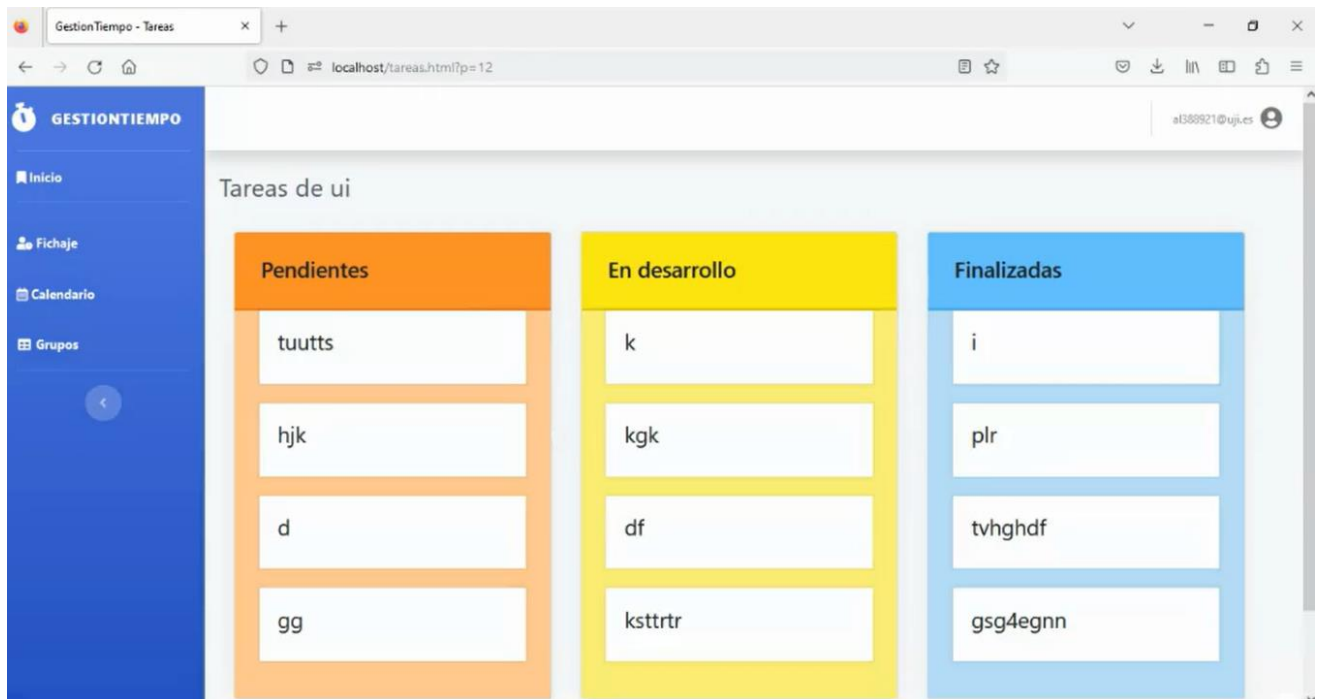


Figura 5.35: Página de tareas en caso de ser Usuario.

En cuanto al diseño de la pestaña de la tarea es idéntico al mostrado en la Figura 5.28 salvo por la falta de los botones laterales y el botón para guardar los cambios si se desea editar la tarea. Por lo demás, el usuario puede seguir viendo el nombre y la descripción de la tarea; las secciones Etiquetas, Responsables y Horas; y también puede realizar registros de horas.

Se puede ver un ejemplo de este diseño en la Figura 5.36.



Figura 5.36: Pestaña de la tarea en caso de ser Usuario.



Figura 5.37: Botones de la pestaña de tareas.

5.2.9. Página de calendario

La página de calendario tiene como principal característica un calendario semanal que muestra todos los registros de horas realizadas en una tarea que ha hecho el usuario. En la figura 5.38 se puede observar el diseño de esta página.

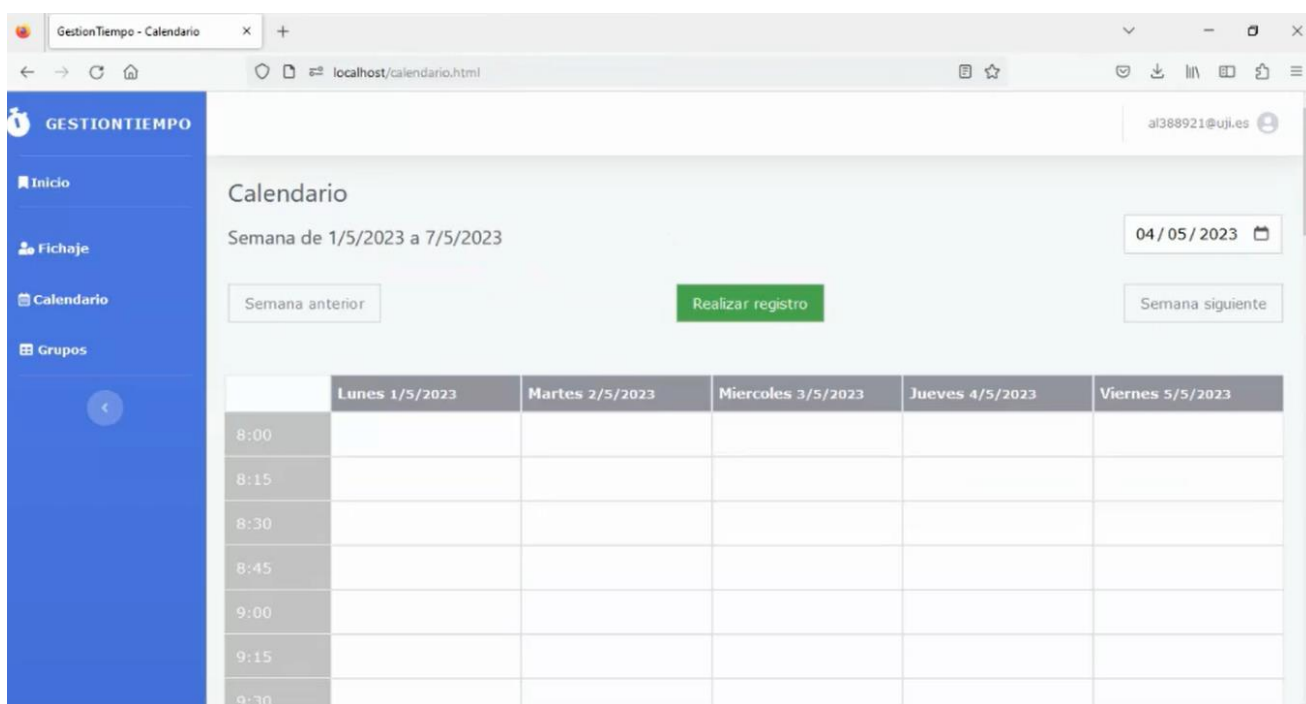


Figura 5.38: Página de calendario.

Por defecto, el calendario muestra la semana actual de cuando se consulte esta página. En la parte superior, bajo el título de la página, se indica la semana que se está mostrando en el calendario.

La tabla del calendario comprende el periodo desde las 08:00h hasta las 20:00h, con una separación entre filas de 15 minutos. Estos tres valores fueron establecidos por la empresa, pero si se decidiese cambiar cualquiera de ellos en un futuro, la tabla seguiría funcionando correctamente y se construiría según los valores indicados.

Al principio de cada columna se indica el día de la semana y la fecha exacta correspondiente a ese día. Cada uno de los registros de la semana se coloca en el día en el que fue realizado y a lo largo de todas las franjas horarias que durase dicho registro. Los registros mostrados incluyen el nombre de la tarea a la que pertenecen y su hora de inicio y de fin.

Para facilitar la diferenciación entre unos registros y otros, cada uno aparece con un color distinto. En el caso de mostrarse en el calendario varios registros de una misma tarea, éstos se mostrarán con el mismo color.

En la Figura 5.39 se puede ver un ejemplo de un calendario con muchos registros y cómo cada uno de ellos se muestra con un color diferente.

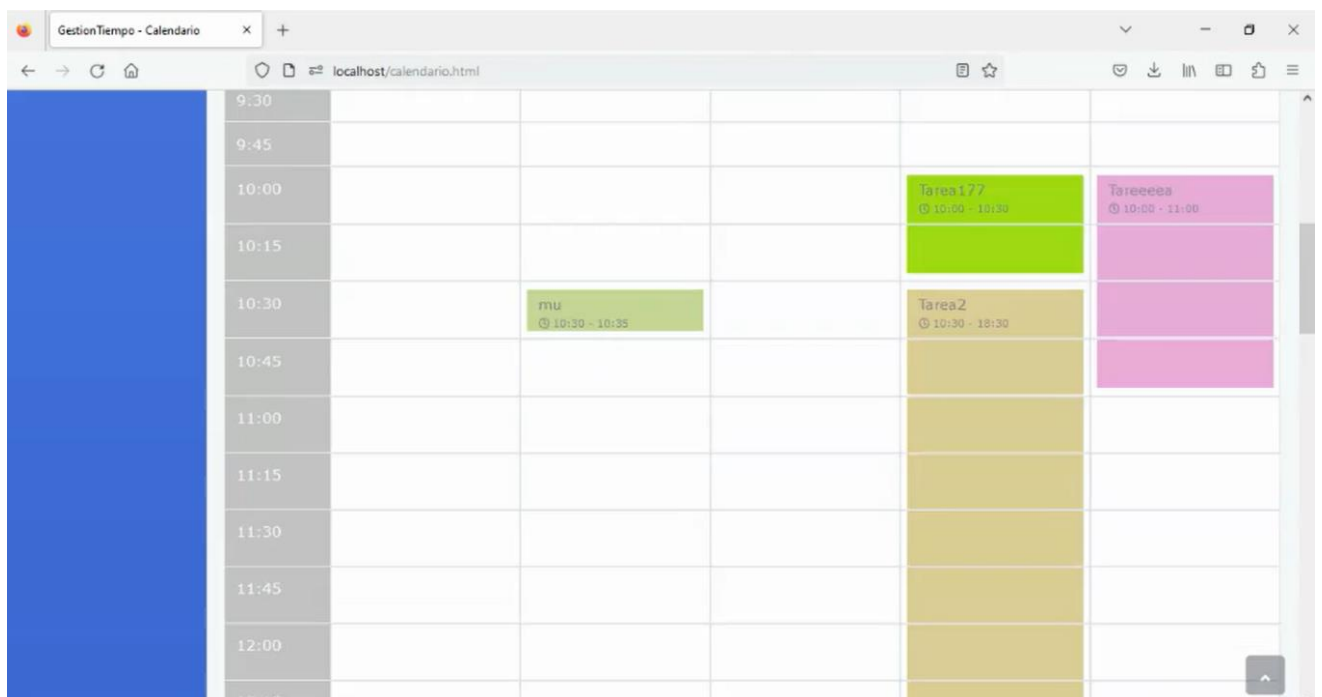


Figura 5.39: Muestra de muchos registros en un calendario.

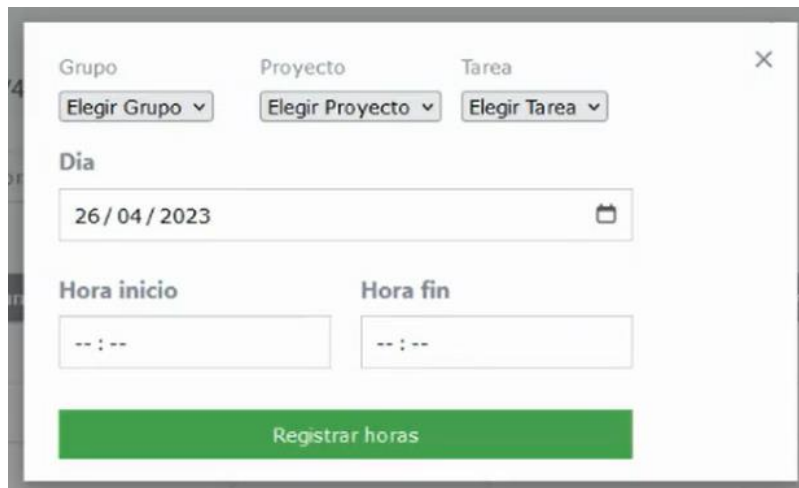
En la esquina superior derecha de la página se muestra la fecha exacta de la consulta de cuando se visite la página. Ese mismo icono es un selector que permite seleccionar cualquier fecha. Al cambiar la fecha, el calendario se adapta y muestra los registros de la semana correspondiente.

Sobre el calendario existen tres botones. Los botones laterales permiten cambiar a la semana anterior, en el caso del botón del lado izquierdo, o a la semana posterior, en el caso del botón del lado derecho, siempre respecto a la semana que esté mostrando en ese momento el calendario.

El nombre de la semana mostrado bajo el título se adapta tanto al seleccionar una nueva fecha como al cambiar a la semana anterior o posterior.

El botón central, de color verde, permite realizar un nuevo registro de horas en una tarea. Esta funcionalidad también se podía realizar desde la página de tareas, pero hay un aspecto que cambia. Allí el registro se realizaba desde una tarea en concreto y por tanto, iba asociado a esa tarea, mientras que aquí se ha de indicar la tarea a la que va asignada el nuevo registro.

Como se puede ver en la Figura 5.40, la pestaña que se muestra al pulsar el botón verde es algo diferente a la mostrada en la Figura 5.29 ya que incluye los selectores superiores para indicar la tarea. Para poder seleccionar la tarea, primero se ha de indicar su grupo y proyecto.



El formulario muestra tres selectores de menú desplegable en la parte superior: 'Grupo' con el texto 'Elegir Grupo', 'Proyecto' con 'Elegir Proyecto' y 'Tarea' con 'Elegir Tarea'. Debajo de ellos hay un campo de texto para la fecha con el valor '26 / 04 / 2023' y un ícono de calendario. En la siguiente línea, hay dos campos de hora: 'Hora inicio' y 'Hora fin', ambos con el formato '-- :--'. Al final del formulario hay un botón rectangular de color verde con el texto 'Registrar horas'.

Figura 5.40: Pestaña para registrar horas realizadas en una tarea desde el calendario.

5.2.10. Página de fichaje

La última página de la aplicación es la página de fichaje, cuyo diseño se puede ver en la Figura 5.41.

Esta página incluye una tabla central con todos los registros de los fichajes realizados por el usuario durante el año actual, mostrándose primero los más recientes. Además, si el usuario es administrador de algún grupo, también se muestran los registros de fichajes de todos los miembros de ese grupo.

Para cada registro se muestra si éste fue de entrada o de salida, el correo del usuario que lo realizó y la fecha y hora del registro.

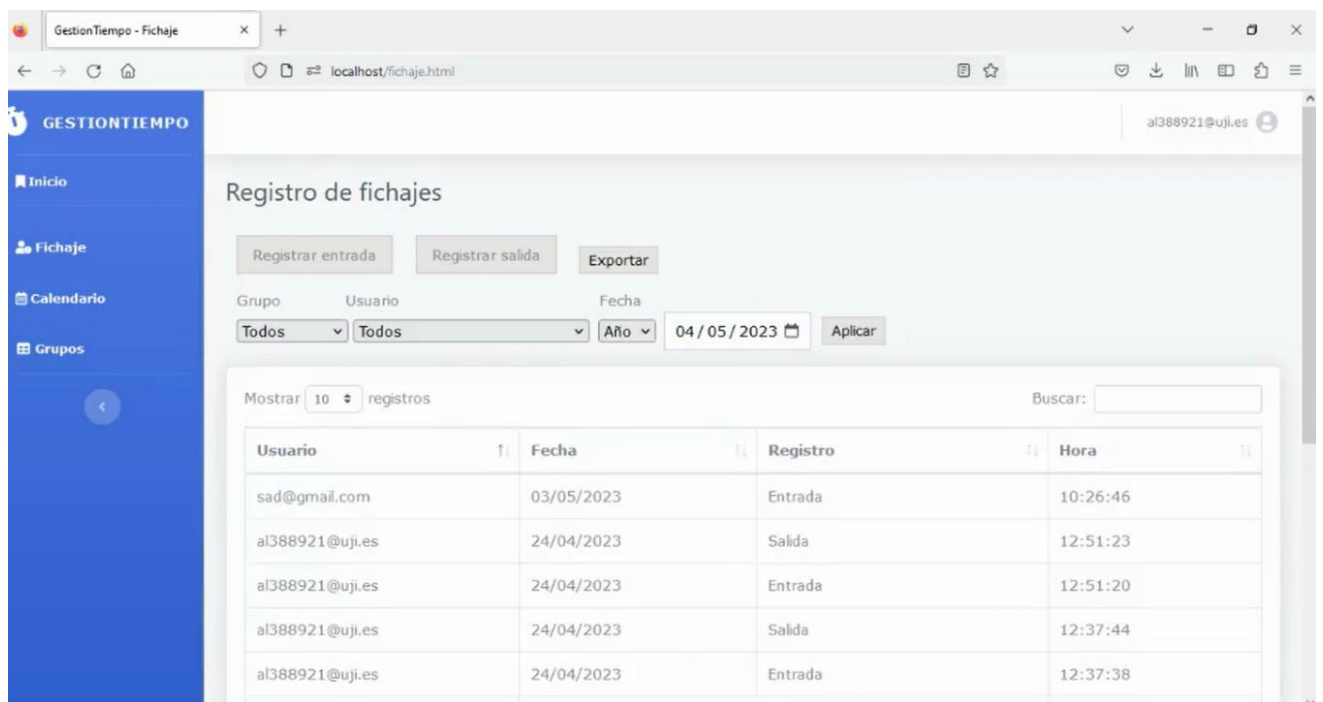


Figura 5.41: Página de fichaje.

Bajo el título de la página hay dos botones que permiten registrar la entrada y la salida, respectivamente. En cualquiera de los dos casos, el nuevo registro se añade a la tabla automáticamente.

A la derecha de esos botones hay un tercer botón que permite exportar toda la tabla de fichajes en un fichero CSV.

Bajo los botones existen una serie de filtros para modificar los registros que se muestran en la tabla.

El primero de ellos es un filtro de grupos que permite mostrar los registros de los miembros de un grupo concreto.

El segundo es un filtro de usuario, que permite mostrar sólo los registros de fichaje de un usuario en concreto.

El último es el filtro de fecha, el cual consta de dos partes. Primero tiene un selector que permite elegir el criterio de fecha entre Día, Mes y Año. Seguidamente hay otro selector que permite elegir una fecha. El filtrado se realiza por la fecha escogida y según el criterio elegido, de manera que se muestren los registros del día, mes o año de la fecha escogida.

Para aplicar el filtrado se ha de pulsar el botón de Aplicar situado a la derecha de los filtros. Todos ellos se pueden realizar de manera individual o combinándose entre sí.

5.3. Verificación y validación

En este apartado se van a explicar todas las medidas de verificación y de validación [15] llevadas a cabo en este proyecto.

La verificación consiste en comprobar el correcto funcionamiento del sistema, mientras que la validación se basa en comprobar que el trabajo realizado cumple con los objetivos establecidos del proyecto.

A lo largo del desarrollo del proyecto no se realizó ninguna prueba automática ni test unitario sobre la aplicación. Todos los métodos desarrollados se fueron comprobando mientras se implementaban realizando pruebas manuales. Las llamadas HTTP se probaron en la herramienta Postman para comprobar que se realizaban correctamente.

En todos los formularios en los que el usuario debía introducir información se añadieron validadores para mostrar avisos en caso de error. Por ejemplo, había avisos si faltaba algún campo obligatorio o si el formato del correo electrónico era incorrecto. Estos ejemplos aparecen en las Figuras 5.42 y 5.43, respectivamente, para el caso del formulario de registro.



El formulario muestra el título "Crea una cuenta" y cuatro campos de entrada: "Nombre", "Correo electrónico", "Contraseña" y "Repite la contraseña". Debajo de los campos hay un botón azul con el texto "Registro". En la parte inferior del formulario, un mensaje de error en rojo indica: "Escribe un nombre, Escribe un correo electrónico, Escribe una contraseña".

Figura 5.42: Validadores de campos obligatorios vacíos.

Formulario de creación de cuenta con el título "Crea una cuenta". Incluye tres campos de entrada: el primero contiene "Lucas", el segundo "sad", y el tercero está oculto con dos asteriscos. Debajo de los campos hay un botón azul con el texto "Registro". En la parte inferior del formulario, un mensaje de error en rojo indica: "Formato de correo erróneo. Ejemplo correcto: nombre@mail.com".

Figura 5.43: Validadores de formato de correo erróneo.

Además, para garantizar el correcto funcionamiento de las funcionalidades de la aplicación, cada vez que se realizaba un método se emitía un aviso como el de la Figura 5.44. Estos avisos indicaban que un objeto se había creado correctamente, se había realizado algún cambio con éxito o se había eliminado con éxito algún elemento.



Figura 5.44: Aviso de cambio realizado con éxito.

En el caso de eliminar elementos de la aplicación, siempre se comprobaba si el usuario estaba seguro de querer borrar el elemento mediante la señal de la Figura 5.45. En el caso de los grupos y los proyectos, al tener una gran cantidad de datos asociados a ellos, la señal era algo más grande e incluía un icono, como se muestra en la Figura 5.46.

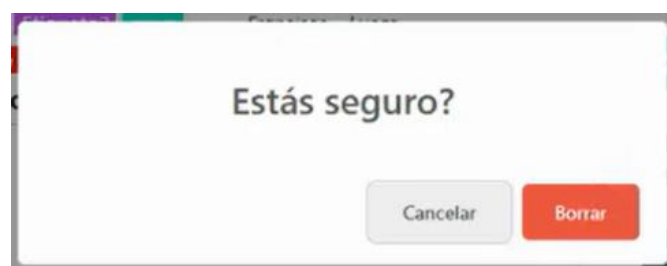


Figura 5.45: Confirmación para borrar un elemento.



Figura 5.46: Confirmación para borrar un grupo o proyecto.

Antes de cerrar sesión también se mostraba un aviso para comprobar si el usuario estaba seguro, pero en esta ocasión el diseño era distinto y se puede observar en la Figura 5.47.



Figura 5.47: Confirmación para cerrar sesión.

Capítulo 6

Conclusiones

6.1. Posibles mejoras

El proyecto se desarrolló de manera que se pudiera ampliar en un futuro sin ningún problema, ya sea añadiendo nuevas páginas o funcionalidades.

En cuanto a nuevas páginas, originalmente se planteó realizar una página para gestionar las vacaciones, pero por falta de tiempo no se llegó a desarrollar. Considero que se podría añadir en un futuro cercano ya que la empresa ya tenía una idea de cómo sería la estructura de esa página.

Por otro lado, hay algunas clases que tienen pocos datos. Durante el proyecto, el esfuerzo estuvo más focalizado en conseguir que la aplicación funcionase, pero una vez acabada, se podrían añadir más campos en algunas clases. Por ejemplo, en el perfil del usuario sólo se muestran el nombre y el correo y se podrían mostrar más datos ahí.

Para finalizar, considero que la aplicación tiene un problema que se debería solucionar. Este problema sólo se producía en ocasiones puntuales y está relacionado con los *tokens*.

Una de las características que debía cumplir la aplicación era que debía de ser asíncrona. Este aspecto se cumplió con éxito, pero en alguna ocasión generaba un pequeño problema con los *tokens*. En algunas páginas, al cargar la página se ejecutaban muchos métodos, todos ellos protegidos por los *tokens*. El problema venía si el *token* caducaba justo en ese momento, ya que podía ocurrir que la mayoría o todos los métodos fallasen. Esto provocaría que cada método intentase actualizar el *token* a la vez y al final volverían a fallar todos ya que cada uno habría

actualizado el *token*. Al final, cuando se producía el error algún método no se realizaba. En muchas ocasiones, se podía solucionar al volver a cargar la página, pero es cierto que alguna vez se tuvo que volver a iniciar sesión para inicializar los *tokens* de nuevo y así conseguir que volviera a funcionar todo correctamente.

Aunque sea un fallo muy puntual, es importante tratarlo y pienso que la actualización de los tokens debería ser síncrona para solucionarlo. Otra opción sería cambiar la mecánica de los tokens tal y como está implementada, aunque desconozco si esta opción sería mucho más costosa que la primera.

6.2. Valoración personal

Para concluir la memoria, debo destacar que el proyecto se realizó con éxito y en el tiempo establecido. Es cierto que hubo una página, la de vacaciones, que no se pudo desarrollar, pero a cambio se implementaron nuevas funcionalidades en las páginas ya desarrolladas que permitieron perfeccionar esas páginas.

Sobre el proyecto de prácticas, pienso que esta experiencia me ha ayudado enormemente ya que me ha permitido conocer cómo es el ámbito laboral desde dentro. He conseguido poner en práctica muchos conocimientos de la carrera y he adquirido otros nuevos que estoy seguro de que me serán útiles en un futuro. Ahora ya estoy familiarizado con un proyecto de desarrollo como éste y me siento capacitado para poder seguir trabajando en proyectos similares.

En cuanto a la empresa, el ambiente ha sido muy agradable y me han ayudado en todo momento cuando tenía alguna duda. Gracias a esto, he podido trabajar a gusto y pienso que he aprovechado al máximo las horas de prácticas. En definitiva, estoy muy satisfecho con ellos.

En conclusión, pienso que realizar este proyecto me ha aportado mucho y me ha sido de gran ayuda.

Bibliografía

- [1] Cye Energía. *Cye energía – Conócenos*.
<https://cye-energia.com/conocenos/>. [Consulta: 07 de Mayo de 2023].
- [2] Wikipedia. *Front end y back end - Wikipedia, la enciclopedia libre*.
https://es.wikipedia.org/wiki/Front_end_y_back_end. [Consulta: 08 de Mayo de 2023].
- [3] Bernal González, David. *¿Qué es Docker y para qué sirve?*
https://profile.es/blog/que_es_docker/ [Consulta: 08 de Mayo de 2023].
- [4] Amazon. *AWS - Cloud Computing - Servicios de informática en la nube*.
<https://aws.amazon.com/es/>. [Consulta: 08 de Mayo de 2023]
- [5] Microsoft. *Visual Studio: IDE y Editor de código para desarrolladores de software y Teams*. <https://visualstudio.microsoft.com/es/> [Consulta: 09 de Mayo de 2023]
- [6] W3C. *Standards – W3C* <https://www.w3.org/standards/>. [Consulta: 09 de Mayo de 2023]
- [7] Wikipedia. *Git – Wikipedia, la enciclopedia libre*
<https://es.wikipedia.org/wiki/Git>. [Consulta: 09 de Mayo de 2023]
- [8] Wikipedia. *MySQL - Wikipedia, la enciclopedia libre*
<https://es.wikipedia.org/wiki/MySQL>. [Consulta: 10 de Mayo de 2023]
- [9] IBM. *¿Qué es una base de datos relacional? | IBM*
<https://www.ibm.com/mx-es/topics/relational-databases> [Consulta: 10 de Mayo de 2023]

- [10] Glassdoor. *Sueldo: Programador (Mayo, 2023) | Glassdoor* https://www.glassdoor.es/Sueldos/programador-sueldo-SRCH_KO0,11.htm#:~:text=%C2%BFCu%C3%A1nto%20gana%20un%20Programador%3F%20En%20Espa%C3%B1a%2C%20el%20sueldo,forma%20an%C3%B3nima%20por%20empleados%20que%20trabajan%20de%20Programador. [Consulta: 10 de Mayo de 2023]
- [11] Glassdoor. *Sueldo: Programador Júnior en España en Mayo, 2023 | Glassdoor* https://www.glassdoor.es/Sueldos/programador-j%C3%BAnior-sueldo-SRCH_KO0,18.htm#:~:text=%C2%BFCu%C3%A1nto%20gana%20un%20Programador%20J%C3%BAnior%3F%20En%20Espa%C3%B1a%2C%20el,an%C3%B3nima%20por%20empleados%20que%20trabajan%20de%20Programador%20J%C3%BAnior. [Consulta: 10 de Mayo de 2023]
- [12] Creately. *Tutorial de diagramas de casos de uso (Guía con ejemplos)* <https://creately.com/blog/es/diagramas/tutorial-diagrama-caso-de-uso/#:~:text=El%20diagrama%20de%20caso%20de%20uso%20es%20un,y%20c%C3%B3mo%20esos%20roles%20interact%C3%BAan%20con%20el%20sistema.> [Consulta: 11 de Mayo de 2023]
- [13] Manz. *¿Qué es CSS? - CSS en español – lenguaje CSS* <https://lenguajecss.com/css/introduccion/que-es-css/> [Consulta: 13 de Mayo de 2023]
- [14] Álvarez, Miguel Ángel. *Qué es MVC* <https://desarrolloweb.com/articulos/que-es-mvc.html> [Consulta: 13 de Mayo de 2023]
- [15] PC Solución. *Diferencia entre validación y verificación.* <https://pc-solucion.es/tecnologia/diferencia-entre-validacion-y-verificacion/> [Consulta: 16 de Mayo de 2023]
- [16] Laoyan, Sarah. *Qué es la metodología waterfall y cuándo utilizarla.* <https://asana.com/es/resources/waterfall-project-management-methodology> [Consulta: 08 de Mayo de 2023]

Anexo A

Correos automáticos enviados

A lo largo de la aplicación se pueden enviar tres tipos de correos automáticos.

El primero de ellos, mostrado en la Figura A.1, se realiza desde la página de registro. Cuando un usuario se registra en la aplicación, se envía este correo para darle la bienvenida.

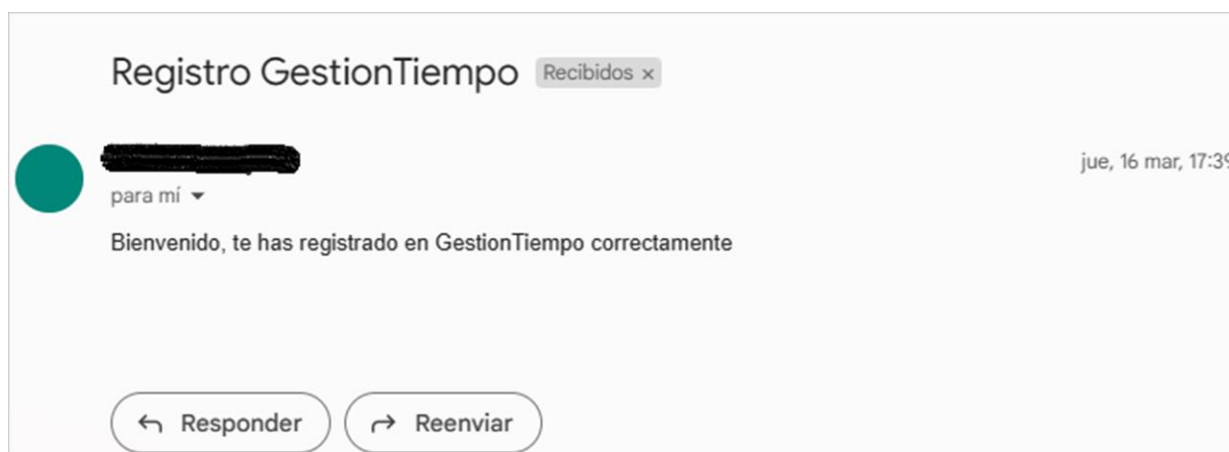


Figura A.1: Correo de bienvenida a la aplicación.

El correo mostrado en la Figura A.2 es enviado cuando el usuario pide el cambio de contraseña. En éste se encuentra un enlace a la página temporal donde el usuario puede cambiar su contraseña.

En el caso de la figura, el enlace mostrado es un enlace local que no tiene el diseño con el dominio final de la aplicación. Esto se debe a que durante la estancia, la aplicación aún no tenía ningún dominio y se decidió mostrar estos correos con este enlace.



Figura A.2: Correo necesario para el cambio de contraseña.

El último tipo de correo se envía cuando el administrador de un grupo de trabajo invita a un usuario al grupo. En la Figura A.3 se muestra un correo informativo que indica que el usuario ha sido invitado a un grupo llamado 'Grupo1'.



Figura A.3: Correo de invitación a un grupo.