



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Alejandro Burgos Martínez

Tutor: María Luz Gordo Monzó

Curso 2020-2021

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas



Resumen

Con la actual reforma de los trabajadores, las empresas están obligadas a registrar la jornada laboral de sus empleados durante al menos un periodo de 4 años.

El proyecto que se ha desarrollado permite dar una solución a esta necesidad, para ello se ha realizado una plataforma web para las empresas, la cual permite llevar a cabo la gestión de la presencia de sus trabajadores mediante un sistema de fichajes.

Los usuarios son capaces de crear o unirse a su empresa y de este modo empezar a llevar un registro de su jornada laboral, al igual que llevar un seguimiento y control de la misma.

El desarrollo de la aplicación web se ha llevado a cabo mediante el uso de varias tecnologías, se ha elegido PHP en su versión 7.4 como principal lenguaje de programación con el uso de Symfony como framework de desarrollo, JavaScript es el siguiente lenguaje más usado, mediante el uso de su librería de JQuery y el framework de Vue.js, se ha buscado ofrecer una interfaz dinámica y agradable para el usuario final, por ultimo como sistema de gestión de bases de datos se ha utilizado MySQL.

Palabras clave: PHP, Symfony, JavaScript, JQuery, Vue.js, interfaz, MySQL.

Abstract

With the current labor reform, companies are required to record the working hours of their employees for at least a 4-year period.

The project that has been developed allows to solve this need, for which a web platform has been created for companies, which allows them to manage the presence of their workers through a transfer system.

Users can create or join your company and thus begin to track their working hours, as well as track and control them.

The development of the web application has been carried out through the use of several technologies, PHP version 7.4 has been chosen as the main programming language with the use of Symfony as a development framework, JavaScript is the next most used language, by using of its JQuery library and the Vue.js framework have sought to offer a dynamic and pleasant interface for the end user. Lastly, MySQL has been used as a database management system.

Keywords: PHP, Symfony, JavaScript, JQuery, Vue.js, interface, MySQL



Tabla de contenidos

1.	Introducción	11
1.1	Motivación	12
1.2	Objetivos.....	12
1.3	Objetivos de desarrollo sostenible	13
1.4	Competencias Transversales	14
1.5	Metodología.....	15
1.6	Estructura	17
2.	Estado del arte	18
2.1	Aplicaciones del mercado	18
2.1.1	Woffu.....	18
2.1.2	CHECKINGplan	19
2.1.3	VisualTime	20
2.1.4	Sesame.....	21
2.2	Tabla Resumen	23
2.3	Crítica al estado del arte	23
2.4	Propuesta	24
3.	Análisis del problema	25
3.1	Análisis del marco legal y ético	25
3.2	Análisis de requisitos	27
3.2.1	Requisitos funcionales	28
3.2.2	Requisitos no funcionales.....	31
3.3	Casos de Uso	32
3.4	Identificación y análisis de soluciones posibles.....	33
3.5	Solución propuesta	33
4.	Diseño de la solución.....	36
4.1	Arquitectura del Sistema.....	36
4.2	Diseño Detallado	37
4.2.1	Estructura de la aplicación Symfony.....	37
4.2.2	Modelo de datos.....	39
4.3	Tecnología Utilizada	41
4.3.2	PHP	41
4.3.3	Symfony	41

4.3.4 MySQL.....	45
4.3.5 Javascript.....	45
4.3.6 Vue.js.....	45
4.3.7 Git.....	46
4.3.8 Gestores de dependencias.....	47
4.4 Herramientas.....	48
4.4.1 PHPStorm.....	48
4.4.2 MySQL Workbench.....	48
5. Desarrollo de la solución propuesta.....	49
5.1 Desarrollo de la aplicación web.....	49
5.1.1 Assets.....	49
5.1.2 Config.....	51
5.1.3 Migrations.....	52
5.1.4 Public.....	53
5.1.5 Scripts.....	53
5.1.6 Src.....	54
5.1.7 Translations.....	65
5.2 Aplicación Web.....	66
5.2.1 Página de presentación.....	66
5.2.2 Registro.....	67
5.2.3 Login.....	67
5.2.4 Pantalla de Inicio en la plataforma.....	68
5.2.5 Registro de empresa.....	68
5.2.6 Perfil del usuario.....	69
5.2.7 Panel de Inicio del empleado.....	69
5.2.8 Mis Fichajes.....	70
5.2.9 Mis Permisos.....	70
5.2.10 Fichaje Remoto.....	71
5.2.11 Panel de control de Administración.....	71
5.2.12 Administración de empleados.....	72
5.2.13 Ficha del empleado.....	72
5.2.14 Administración de Grupos de trabajo.....	74
5.2.15 Administración de Fichajes.....	74
5.2.16 Administración de Permisos.....	75
5.2.17 Administración de horarios.....	75
5.2.18 Administración de eventos de la empresa.....	76

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

5.2.19 Administración de la plataforma	77
6. Implantación	78
7. Pruebas	80
7.1 PHPUnit.....	80
7.2 GitLab CI/CD	81
8. Conclusiones	83
8.1 Relación del trabajo realizado con los estudios cursados	84
9. Trabajos futuros.....	85
10. Referencias	86

Tabla de ilustraciones

Ilustración 1. Listado de Tareas	16
Ilustración 2. Tablero Kanban.....	16
Ilustración 3. Aplicación de Woffu	18
Ilustración 4. Solicitudes de Woffu	19
Ilustración 5. CHECKINGplan	20
Ilustración 6. VisualTime	21
Ilustración 7. Sesame.....	22
Ilustración 8. Estructura de la aplicación	37
Ilustración 9. Diagrama de la base de datos.....	39
Ilustración 10. Uso de Symfony a lo largo del tiempo	42
Ilustración 11. Plantilla Twig del perfil del empleado	43
Ilustración 12. Configuración del lenguaje.....	43
Ilustración 13. Archivos de traducciones	44
Ilustración 14. Traducciones de las validaciones.....	44
Ilustración 15. Flujo de trabajo de Gitflow.....	46
Ilustración 16. Estructura de la carpeta Assets.....	49
Ilustración 17. Template del componente Vue de fichajes.....	50
Ilustración 18. Script del componente Vue de fichajes	50
Ilustración 19. Estructura de la carpeta Config.....	51
Ilustración 20. Configuración del control de accesos.....	51
Ilustración 21. Configuración de los eventos de las entidades.....	52
Ilustración 22. Migración inicial.....	52
Ilustración 23. Estructura de la carpeta public.....	53
Ilustración 24. Script de testing	53
Ilustración 25. Estructura de la carpeta src.....	54
Ilustración 26. Comprobación de permisos de acceso	54
Ilustración 27. Controladores de la aplicación	55
Ilustración 28. Controlador de los usuarios de la empresa	56
Ilustración 29. Fixtures para crear a los usuarios de una empresa.....	57
Ilustración 30. Entidades del proyecto.....	58
Ilustración 31. Entidad User	58
Ilustración 32. Escuchador de eventos de la entidad Usuario.....	59
Ilustración 33. Método que solicita un permiso o ausencia.....	59
Ilustración 34. Eventos asociados a los permisos y ausencias.....	60
Ilustración 35. Clases suscrita a los eventos de los permisos y ausencias..	60
Ilustración 36. Estructura del directorio de formularios	60
Ilustración 37. Creación y validación de un formulario en el controlador	61
Ilustración 38. Creación de la vista y del formulario.....	61
Ilustración 39. Formulario de Usuarios de la empresa.....	61
Ilustración 40. Controlador pidiendo los permisos de un empleado al repositorio.....	62
Ilustración 41. Método del repositorio encargado de obtener los permisos entre 2 fechas.....	62
Ilustración 42. Método de control en el controlador	63
Ilustración 43. Voter encargado de los accesos sobre las ausencias	63

Ilustración 44. Servicio encargado de los fichajes de los usuarios	64
Ilustración 45. Servicio administrador de ausencias	64
Ilustración 46. Traducciones de las validaciones	65
Ilustración 47. Página de la presentación	66
Ilustración 48. Página de registro	67
Ilustración 49. Página de Login	67
Ilustración 50. Página de inicio	68
Ilustración 51. Página de Registro de Empresas	68
Ilustración 52. Página de perfil de usuario	69
Ilustración 53. Página de Panel de inicio	69
Ilustración 54. Página Mis fichajes	70
Ilustración 55. Página Mis permisos	70
Ilustración 56. Página de Fichaje remoto	71
Ilustración 57. Página de Panel de control	71
Ilustración 58. Página de Administración de empleados	72
Ilustración 59. Página Ficha del empleado	72
Ilustración 60. Página Fichajes del empleado	73
Ilustración 61. Página Permisos del empleado	73
Ilustración 62. Página de Grupo de trabajo	74
Ilustración 63. Página Administración de fichajes	74
Ilustración 64. Página administración de Permisos y ausencias	75
Ilustración 65. Página de Creación de Horarios	75
Ilustración 66. Página de creación de un Horario Estricto	76
Ilustración 67. Página de Administración de eventos de la empresa	76
Ilustración 68. Página de Administración de la plataforma	77
Ilustración 69. Página de Administración de un Usuario de la plataforma	77
Ilustración 70. Archivo deploy.php	78
Ilustración 71. Test de Registro	80
Ilustración 72. Listado de pasos automatizados al introducir cambios en el código	81
Ilustración 73. Validaciones realizadas en la integración continua	82
Ilustración 74. Archivo de configuración gitlab-ci.yml	82

Palabras clave

Interfaz¹: Una interfaz es un medio común para que los objetos se comuniquen entre sí, estableciendo una interfaz sobre un objeto estas indicando que va a implementar una funcionalidad establecida por la misma.

Librería²: Conjunto de implementaciones funcionales codificadas en un lenguaje de programación, ofreciendo una funcionalidad bien definida mediante una interfaz, una librería no se espera que se ejecute de forma autónoma si no ser utilizada por otros programas.

Framework³: Un entorno de trabajo es una estructura conceptual y tecnológica, ofreciendo módulos de software o librerías que pueden servir como base para establecer la organización de un proyecto.

SQL⁴: Lenguaje de consulta estructurada utilizado en programación diseñado para administrar y acceder a información almacenada en bases de datos relacionales.

API: Application program interface, La interfaz de programación de aplicaciones es un conjunto de rutinas que provee acceso a funciones, subrutinas y procedimientos de un determinado software.

¹ [https://es.wikipedia.org/wiki/Interfaz_\(programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Interfaz_(programaci%C3%B3n))

² [https://es.wikipedia.org/wiki/Biblioteca_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica))

³ <https://es.wikipedia.org/wiki/Framework>

⁴ [https://es.wikipedia.org/wiki/SQL#:~:text=SQL%20\(por%20sus%20siglas%20en,de%20bases%20de%20datos%20relacionales](https://es.wikipedia.org/wiki/SQL#:~:text=SQL%20(por%20sus%20siglas%20en,de%20bases%20de%20datos%20relacionales)

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas



1. Introducción

La actual situación de pandemia que afronta el mundo ha traído consigo una serie de consecuencias que afectan de forma drástica en el mundo laboral, como es el incremento del número de personas que han comenzado a trabajar desde casa.

A raíz de esta nueva situación, se deben de afrontar una serie de nuevos retos, la actual reforma de la ley del estatuto de los trabajadores. Según indica el artículo 34:

“La empresa garantizará el registro diario de jornada, que deberá incluir el horario concreto de inicio y finalización de la jornada de trabajo de cada persona trabajadora, sin perjuicio de la flexibilidad horaria que se establece en este artículo.

Mediante negociación colectiva o acuerdo de empresa o, en su defecto, decisión del empresario previa consulta con los representantes legales de los trabajadores en la empresa, se organizará y documentará este registro de jornada.

La empresa conservará los registros a que se refiere este precepto durante cuatro años y permanecerán a disposición de las personas trabajadoras, de sus representantes legales y de la Inspección de Trabajo y Seguridad Social.”

(1)

En definitiva, es requerido por parte de todas las empresas, un sistema de control del horario que sea capaz de registrar la jornada de sus empleados de forma diaria. Así como, de almacenar estos registros durante un periodo de al menos cuatro años.

La herramienta que se desea desarrollar debe de permitir dar una solución a esta necesidad mediante una plataforma web pensada y diseñada para empresas, la cual permita llevar a cabo la gestión de la presencia de sus trabajadores.

La solución propuesta en este TFG pretende brindar una respuesta a los principales obstáculos con los que se puede encontrar una empresa. Como son el cumplimiento y la gestión de los diferentes horarios que pueden tener los trabajadores, así como proporcionar una visión clara de los permisos y las posibles ausencias que puedan ocurrir a lo largo del periodo laboral.

Para ello la plataforma consta principalmente de dos partes:

- La primera, está enfocada a la gestión y al seguimiento, mediante la cual los usuarios que den de alta a su empresa podrán llevar un control de los distintos trabajadores, siendo capaces de ver y analizar los registros de la jornada de sus empleados. Esta parte debe ser capaz de tener en cuenta las posibles variaciones que puedan ocurrir como son las bajas, las horas extras y las vacaciones de los diferentes trabajadores.

- La segunda parte, está destinada a permitir a los usuarios mediante un sistema fácil e intuitivo poder registrar en el momento la hora en la que inicia y finaliza su jornada laboral.

El presente TFG pretende mostrar el proceso de diseño, desarrollo e implantación de una herramienta profesional que busca ser competitiva en el actual mercado laboral y simule, de una forma fiel, todo el recorrido necesario para la puesta en marcha de una solución software.

1.1 Motivación

A raíz de la nueva situación sanitaria y la actual reforma del estatuto de los trabajadores, son muchas las empresas que aún no han tenido tiempo de adaptarse, no han encontrado o no disponen de una herramienta capaz de cubrir sus necesidades.

La principal motivación que se pretende alcanzar con la puesta en marcha de este proyecto es proporcionar una herramienta útil en especial para la pequeña y mediana empresa, la cual sea capaz de cubrir sus necesidades frente a la nueva ley, además de permitir mejorar su capacidad de análisis y gestión de sus recursos humanos, de una forma clara, simple y medible.

Una de las principales motivaciones ha sido el interés por construir una solución web con el uso de tecnologías modernas en el sector, permitiendo conocer y aprender nuevas soluciones para posibles desarrollos futuros, de esta forma, se busca enriquecer el conocimiento personal de cara a afrontar retos en situaciones laborales reales.

Otra de las principales razones para elegir este proyecto ha sido la motivación personal de realizar una aplicación web capaz de adaptarse a las necesidades actuales del mercado. Con una estructura y tecnologías modernas de forma que, el proceso de desarrollo suponga un reto. Una forma de mejorar y aprender el comportamiento de una empresa brindándole una herramienta actualizada, escalable y capaz de afrontar nuevos cambios en el futuro.

1.2 Objetivos

El objetivo de este proyecto es desarrollar una aplicación web con el uso de tecnologías modernas y demandadas en el actual mundo laboral, capaz de proporcionar una solución eficaz a los distintos retos que enfrentan las empresas en la actualidad debido a la actual situación con la gestión del personal.

El objetivo principal, es el desarrollo de un sistema que permita a las empresas llevar un seguimiento y verificar el cumplimiento de las jornadas de sus empleados.

La herramienta debe de constar de dos partes principales: la que se encarga de proporcionar un método para el empleado de realizar los registros de entradas y salidas de su puesto laboral, mientras que la otra parte está enfocada a los usuarios encargados de la administración de la empresa los cuales deben de poder llevar un seguimiento de los datos registrados en la plataforma.

Desde el punto de vista académico, este proyecto tiene como objetivos:

- La mejora de habilidades en el desarrollo e implantación de aplicaciones web.
- Aprendizaje de tecnologías y frameworks actuales demandados en el ámbito laboral como son PHP, Symfony⁵ y Vue.js⁶.
- Adquirir conocimientos de análisis y resolución de necesidades dentro de las empresas.
- Mejorar los conocimientos en torno al almacenamiento y explotación de datos mediante sistemas de gestión de bases de datos relaciones como es MySQL
- Aumentar habilidades en el diseño de interfaces para proporcionar una experiencia de usuario capaz de permitir a cualquier usuario usar y sentirse familiarizado con la herramienta.

1.3 Objetivos de desarrollo sostenible

Como punto a tener en cuenta a la hora de la elección de este TFG ha sido el análisis y la valoración de una serie de Objetivos de desarrollo sostenible, los cuales han influenciado de gran medida, ya que la herramienta busca aportar soluciones a ciertas empresas o personas que, por la situación actual, han visto en cierta medida modificada su situación laboral y su sitio de trabajo.

Con la puesta en marcha de la herramienta, se pretende fomentar y mejorar ciertos objetivos de desarrollo sostenible como pueden ser:

Asegurar acceso a herramientas para todos

Actualmente, las empresas se han visto obligadas a cambiar su método de control y gestión de su personal, por lo que este punto ha sido un factor muy a tener en cuenta; ya que con este sistema se pretende que ninguna empresa se quede atrás a la hora de poder hacer frente al cumplimiento de la ley.

Se ha diseñado un sistema por el cual las empresas con un número reducido personal sean capaces de utilizar esta herramienta sin ningún coste, lo que permita en cierta medida adaptarse mejor a las circunstancias.

⁵ <https://symfony.com/>

⁶ <https://vuejs.org/>

Igualdad de Género

En la actualidad, existe aún una brecha en la igualdad de género y en la asignación de ciertos permisos a los trabajadores. Como pueden ser: el permiso de paternidad o maternidad. Por lo que la aplicación busca hacer frente a este problema mediante un sistema actualizado de los permisos retribuidos de los trabajadores.

Además, la herramienta no realizará distinciones en el uso en relación con el género de los trabajadores, por lo que registros de las horas (tanto como de las horas extras) se deben adaptar a lo que marca la ley de igualdad.

Trabajo decente y crecimiento económico

Actualmente, en España se realiza un gran número de horas extra de las cuales un 44% no son retribuidas, como indica en el documento (2), esto supone un incumplimiento de la ley además de un abuso para los trabajadores.

Con el uso correcto de la herramienta se pretende aportar un conocimiento que permita mejorar los recursos de personal, lo que repercutirá en una mejor gestión, la cual se traducirá en un crecimiento económico para la empresa.

Acción por el clima

Con el uso de la aplicación, los empleados serán conscientes de sus entradas y salidas, al poder realizar esta acción de forma conjunta por varios miembros de la empresa, el uso de esta herramienta es capaz de proporcionar una motivación de buscar alternativas de transporte para llegar a ciertas horas y con la misma regularidad al entorno de trabajo por lo que puede fomentar el uso del mismo medio de transporte por varias personas del mismo grupo, de esta forma contribuir a un menor uso del transporte privado y mejorando las contaminaciones producidas por estos medios de transporte.

1.4 Competencias Transversales

Mediante la realización de este proyecto, se busca potenciar una serie de rasgos diferenciadores altamente valorados en el ámbito laboral, por lo tanto, las principales competencias transversales que se desean abordar son:

- **Análisis y resolución de problemas:** Para la realización de este trabajo ha sido necesario analizar los requisitos de diferentes empresas, así como de sus trabajadores para cubrir las necesidades a la hora de llevar a cabo un registro y control de las jornadas de los empleados, mejorando o dando solución a sus necesidades.
- **Diseño y proyecto:** Permitirá transformar una idea a partir de una necesidad real en un proyecto factible, teniendo en cuenta los recursos, objetivos y riesgos.
- **Planificación y gestión del tiempo:** Mediante un sistema de tareas se ha de ser capaz de llegar a los objetivos deseados y el plazo de tiempo deseado.
- **Conocimiento de problemas contemporáneos:** Debido a las necesidades del entorno aprender a conocer sus necesidades, ser capaz de analizar y diseñar una solución capaz de satisfacerlas.

- Instrumental específico: Para el desarrollo de esta herramienta ha sido necesario recopilar información y elegir una serie de herramientas que permitan el diseño y la realización del proyecto de forma que se adapte a las nuevas y mejores tecnologías disponibles.

1.5 Metodología

La forma con la que se ha llevado a cabo el desarrollo de la plataforma ha sido mediante un desarrollo en espiral. Buscando dividir y abordar el desarrollo de las distintas funcionalidades en una serie de varias fases, permitiendo adaptar el desarrollo a los plazos para los que se ha preparado el proyecto.

GitLab⁷ ha sido la plataforma utilizada para llevar a cabo la gestión y la planificación de estas tareas, permitiendo llevar un control y un seguimiento del estado de todas ellas.

A continuación, se enunciarán una serie de procedimientos que se han llevado a cabo para cumplir el desarrollo de cada una de las funcionalidades:

- Establecer un listado de tareas, todas las funcionalidades o tareas que se han realizado han sido descritas y añadidas a un listado para su posterior evaluación y análisis como podemos ver en la ilustración 1.
- En segundo lugar, antes de comenzar con el desarrollo de la funcionalidad se ha de realizado un pequeño análisis. Buscando establecer que se pretende lograr con la nueva funcionalidad, una evaluación de su coste, los riesgos y el impacto de su desempeño, en este punto consideraremos que la tarea está en el estado “ToDo”.
- Una vez terminada la fase de análisis y comprobada su viabilidad, se pondrá en marcha su desarrollo o implementación. Lo que se busca en este momento es desarrollar o poner solución a la nueva funcionalidad analizada anteriormente, por lo que la tarea pasara al estado “Doing”.
- Comprobación de todos los requisitos necesarios mediante una serie de medidas de aceptación, calidad de código y pruebas de las diferentes funcionalidades la tarea pasa a estar finalizada e indicándolo en su estado como “Closed”.

⁷ <https://about.gitlab.com/>

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Mediante este sistema de trabajo se ha logrado establecer una forma rápida de desarrollo, permitiendo lograr resultados a corto plazo y comprobar si se cumplen todas las necesidades para posteriormente, mejorar o aplicar las funcionalidades existentes.

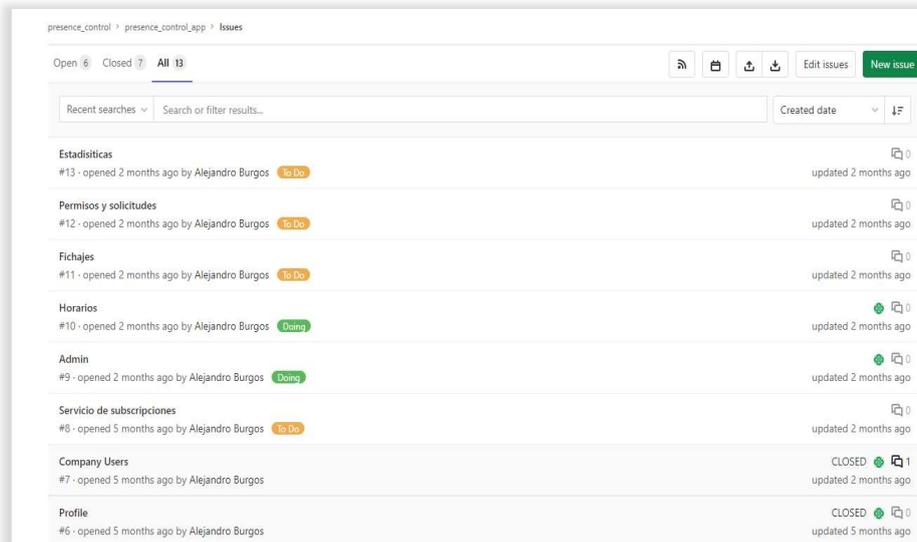


Ilustración 1. Listado de Tareas

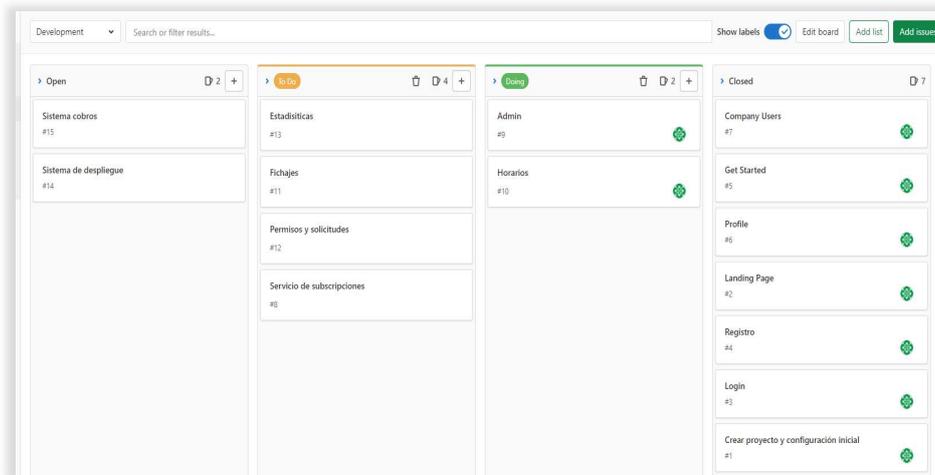


Ilustración 2. Tablero Kanban

1.6 Estructura

En este apartado se describe como se ha estructurado la memoria de este proyecto, con la intención de facilitar la lectura de la misma. La memoria está dividida en 9 capítulos, los cuales se detallan a continuación:

Capítulo 1: “Introducción” presentación del proyecto, definición y motivación del mismo, así como los objetivos a realizar y metodología de desarrollo utilizada.

Capítulo 2: “Estado del arte” se analiza el estado actual del mercado de las soluciones para el control horario en las empresas y se presentan varias aplicaciones con funcionalidades similares.

Capítulo 3: “Análisis del problema” se realiza el análisis para determinar cuál es la solución al problema, los requisitos necesarios y los casos de uso.

Capítulo 4: “Diseño de la solución” se determina la estructura del sistema, así como las tecnologías utilizadas para su desarrollo.

Capítulo 5: “Desarrollo de la solución propuesta” exposición de como se ha desarrollado el proyecto, sus partes y el resultado final.

Capítulo 6: “Implantación” pasos para la puesta en marcha de la aplicación.

Capítulo 7: “Pruebas” explicación de las pruebas realizadas durante el desarrollo.

Capítulo 8: “Conclusiones” reflexiones del alumno respecto al trabajo realizado.

Capítulo 9: “Trabajos futuros” se exponen futuras implementaciones que se deben de realizar al proyecto.

Capítulo 10: “Referencias” anexos a los estatutos del control horario de los trabajadores.

2. Estado del arte

En este apartado se analizará el estado del arte actual de una serie de aplicaciones que permiten realizar un control de la presencia de los trabajadores, además de sus puntos fuertes, sus debilidades y aspectos de mejora frente a las necesidades de la empresa.

Actualmente, existe una gran variedad de ofertas en el mercado que permiten llevar a cabo una gestión del personal de las empresas, ya que debido a la reforma en el estatuto de los trabajadores los empresarios se han visto obligados a buscar una solución a sus nuevas necesidades y muchas empresas han visto una gran oportunidad de mercado para desarrollar sus aplicaciones.

También existe una gran variedad de aplicaciones que permite llevar un control sobre los distintos proyectos que están realizando en la empresa y como invierten el tiempo sus empleados, pero en este caso solo se centrará la atención en una serie de herramientas que permiten controlar la jornada de sus usuarios.

Posteriormente se analizará como la solución desarrollada es capaz de hacerse un hueco entre todas estas aplicaciones, se comparará sus puntos fuertes y futuras mejoras e implementaciones para ser más competitiva en el mercado.

2.1 Aplicaciones del mercado

En este apartado se describirán una serie de aplicaciones que se engloban dentro del ámbito del control de la presencia de los empleados:

2.1.1 Woffu

Woffu es una herramienta que, además de llevar un control horario de los trabajadores, es capaz de proporcionar otros aspectos interesantes en relación con su funcionalidad principal.

Esta plataforma brinda una interfaz sencilla y amigable para los usuarios, lo que permite una mejor adaptación para los empleados como se puede apreciar en la ilustración 3.

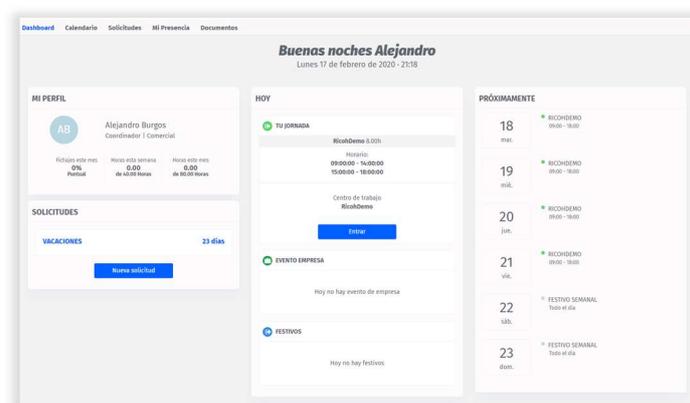


Ilustración 3. Aplicación de Woffu

Gestión de vacaciones y ausencias

La función más interesante a destacar es la facilidad que proporciona Woffu en la solicitud de vacaciones y permisos. Los trabajadores deben acceder al apartado de notificaciones e indicar en qué periodo desean disfrutar de un cierto permiso como se muestra en la ilustración 4.

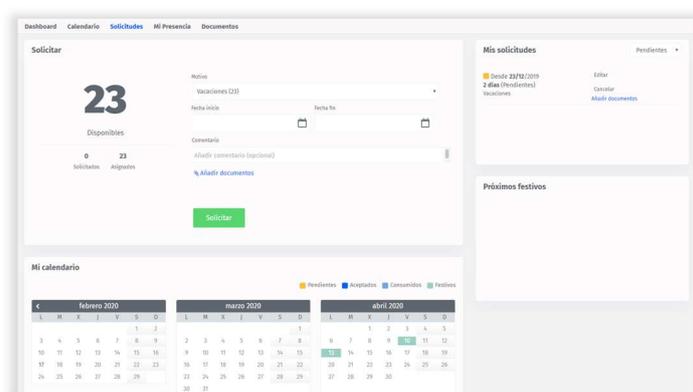


Ilustración 4. Solicitudes de Woffu

Turnos de trabajo

Woffu proporciona una funcionalidad muy útil. Como son los turnos de trabajo, lo que permite gestionar las horas de tus trabajadores de forma rápida y sencilla.

Esta funcionalidad es especialmente útil cuando la empresa cuenta con varios departamentos o turnos de trabajo.

Gestión documental

Una funcionalidad interesante con la que cuenta esta plataforma y es digna de destacar. Como puede ser la gestión de documentos en la nube. La plataforma permite tanto subir como compartir archivos, lo que puede ser interesante a la hora de adjuntar partes o justificantes.

2.1.2 CHECKINGplan

Aplicación Android diseñada para llevar un control de las entradas y salidas de los empleados mediante un dispositivo móvil como se puede observar en la ilustración 5.

CHECKINGplan ofrece una gran ventaja sobre la competencia ya que permite cotejar las coordenadas mediante geolocalización para determinar donde se encuentran los trabajadores a la hora de realizar sus registros de jornada.

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Ahora se destacará algunas de las características más relevantes de esta aplicación:

App

App de presencia para dispositivos Android, lo que permite realizar los registros mediante cualquier dispositivo con este sistema operativo, ya sea fijo o móvil. Es necesario instalar la aplicación.

Tele-Presencia

Control de presencia mediante llamadas telefónicas desde un teléfono fijo es una característica muy destacada en ciertos sectores, permitiendo registrar el código del empleado recogido por la centralita.

Reloj de Presencia

Esta característica permite al empleado registrar su jornada mediante una identificación con su huella dactilar o contraseña, lo que proporciona más seguridad y un registro más fiable por parte de los empleados.

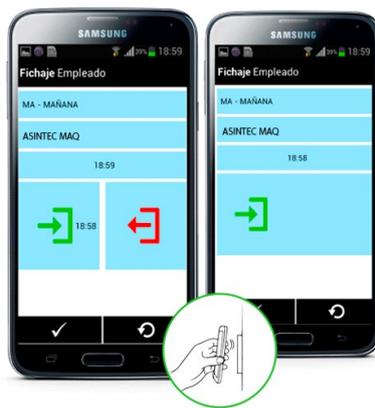


Ilustración 5. CHECKINGplan

2.1.3 VisualTime

Plataforma desarrollada por la empresa Robotics, Visualtime ofrece una opción muy completa, diseñada para adaptarse a las necesidades de la pequeña y mediana empresa.

Esta herramienta brinda la posibilidad de indicar en qué están empleando el tiempo nuestros empleados, además de ofrecer una mejor optimización de la gestión horaria y de esta forma generar un ahorro para la empresa.

Ahora se procederá a indicar varias de las características que ofrece esta aplicación:

Múltiples modos de fichaje

La plataforma ofrece varios entornos en los que el empleado es capaz de registrar su jornada laboral. Estos pueden ser tanto fichaje Web, como desde un

terminal con reconocimiento biométrico o desde su propia App, la cual cuenta con un diseño moderno y atractivo como se puede ver en la ilustración 6.

Turnos de trabajo

Diseñar los distintos turnos de trabajo adaptándolos a los diferentes horarios que puedan tener tus empleados es una de las características que ofrece esta plataforma.

Gestión de vacaciones y ausencias

De una forma ágil esta plataforma permite al empleado registrar tanto sus solicitudes en relación con las ausencias como vacaciones.

Informes y analítica de datos

VisualTime ofrece una forma de poder analizar y obtener informes personalizados. Este es un punto a tener en cuenta, ya que esta funcionalidad aporta un gran valor, permitiendo a la empresa tomar decisiones en relación a los KPIs aportados por estos informes.



Ilustración 6. VisualTime

2.1.4 Sesame

Sesame es una potente herramienta que permite a los empleados realizar sus registros de la jornada de forma eficiente, además de otorgar a los empresarios o usuarios de administración una herramienta en la que poder llevar una gestión de los trabajadores.

Esta aplicación ofrece la posibilidad de registrar de manera fácil y rápida tanto las entradas como las salidas, al igual que la posibilidad de indicar las pausas para las comidas o cualquier tipo de descanso. En la ilustración 7, se puede observar una interfaz clara en la que llevar el control de la jornada laboral.

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Sesame ofrece una gran cantidad de características que la vuelven una herramienta a tener en cuenta a la hora de analizar el mercado de las aplicaciones para el control horario. A continuación, se enumerarán:

App de Fichaje

Actualmente, existe una aplicación tanto para dispositivos Android como iOS, lo que aumenta de forma significativa la ventaja de adaptarse al mercado, siendo estos los dos sistemas operativos más utilizados en dispositivos móviles.

Crear puntos de acceso

Una característica muy destacable, es la posibilidad de convertir cualquier Tablet en un punto de acceso en el que se permita a los empleados realizar en estos su registro de la jornada.

Sistemas biométricos

Sesame permite integración con sistemas de reconocimiento facial o por huella dactilar.

Informes

La herramienta ofrece la posibilidad de realizar tanto informes personalizados como estadísticas de fichajes lo que permite la optimización y la detección de mejoras de gestión para la empresa.

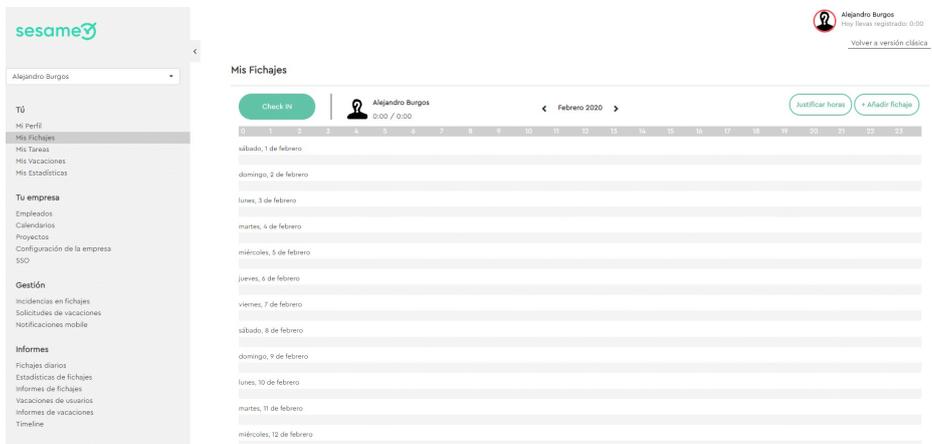


Ilustración 7. Sesame

2.2 Tabla Resumen

Por último, se ha realizado una tabla con la finalidad de realizar una comparativa entre las aplicaciones relacionadas con sus características más destacables, en que plataformas están disponibles y su precio de mercado:

APLICACIÓN	CARACTERÍSTICAS	VERSIÓN	PRECIOS
WOFFU	Turnos de trabajo Permisos y ausencias Gestión documental Informes/análisis	Web Móvil: Android Móvil: IOS	3.00 €/mes por empleado
CHECKINGPLAN	Turnos de trabajo Geolocalización Tele-presencia Gestión documental Gestión de calendarios	Web Móvil: Android Móvil: IOS Terminales de control de acceso	2,00 €/mes por empleado
VISUALTIME	Turnos de trabajo Planificación horaria Permisos y ausencias Informes/análisis	Web Móvil: Android Móvil: IOS Terminales de control de acceso	Demo gratis 3,00 €/mes por empleado
SESAME	Turnos de trabajo Permisos y ausencias Informes/análisis Gestión de calendarios	Web Móvil: Android Móvil: IOS	2,00€/mes por empleado 18,00€/año por empleado

2.3 Crítica al estado del arte

Actualmente, se puede encontrar una gran variedad de aplicaciones para realizar el control de la jornada de los usuarios, permisos y ausencias en el mercado. Todas ellas proporcionan una opción que cumple con los requisitos básicos para realizar los fichajes de los empleados y un registro de los mismos.

La herramienta desarrollada pretende dar una opción para cubrir esta necesidad, sobre todo a la pequeña y mediana empresa, la cual puede contar con un nivel bajo en sus ingresos. La aplicación web pretende ser gratuita para un reducido número de empleados, dando la opción a cualquier persona a crear y gestionar el control de su empresa de una forma fácil y eliminando las barreras iniciales con las que cuentan el principal número de soluciones del mercado.

Otro punto a destacar de la plataforma es la flexibilidad que ofrece a sus usuarios con su forma de interactuar con las organizaciones registradas en la plataforma, permitiendo ser ellos mismos quienes decidan. Ofreciendo la posibilidad de unirse, crear o salirse de las empresas que ellos deseen.

Mediante este enfoque se pretende que la solución realizada tenga una buena aceptación, permitiéndole hacerse un hueco en el mercado tan competitivo como el actual.

2.4 Propuesta

La propuesta del proyecto consiste en el desarrollo de una aplicación fácil de integrar para las empresas, en las que el primer contacto con la herramienta sea de la forma más sencilla e intuitiva posible para los usuarios.

Se busca ofrecer una aplicación potente capaz de incrementar el rendimiento y mejorar la gestión de los recursos humanos de la empresa.

La solución que se opta a desarrollar es una aplicación web ya que permite crear unas bases sólidas y construir todo un ecosistema a su alrededor a partir del desarrollo inicial.

Una solución web va a permitir una mejor implantación en las empresas ya que la herramienta estará disponible para cualquier dispositivo con conexión a internet y un navegador instalado.

3. Análisis del problema

En todo proyecto de software podemos encontrar la etapa de especificación de requisitos. En esta etapa se busca la descripción completa de los objetivos, comportamientos con los que cuenta el sistema, de esta forma se establece una relación entre los diferentes usuarios involucrados en el desarrollo, análisis y diseño de las partes que dispone la aplicación.

3.1 Análisis del marco legal y ético

Durante la etapa de análisis de la plataforma se ha buscado que esta cumpla las medidas establecidas en el marco legal que establece el Real Decreto-ley 8/2019, de 8 de marzo, de medidas de lucha contra la precariedad laboral en la jornada de trabajo.

“La empresa garantizará el registro diario de jornada, que deberá incluir el horario concreto de inicio y finalización de la jornada de trabajo de cada persona trabajadora, sin perjuicio de la flexibilidad horaria que se establece en este artículo.” (1)

Del texto citado se pueden extraer las siguientes apreciaciones:

- **El registro deberá ser diario:** Únicamente constarán como válidos los registros realizados de forma diaria, mientras que quedarán anulado cualquier registro de carácter semanal o mensual (3).
- **El registro tendrá que ser al inicio y al final de la jornada de trabajo:** No es necesario registrar las pausas en el trabajo, lo que implica que el trabajador salvo que lo indique el convenio colectivo indique otra cosa. El trabajador no está obligado a registrar sus pausas durante su jornada laboral, aunque es recomendable.

En el estatuto de los trabajadores ya se incluía la obligación de realizar el registro de determinadas horas de trabajo. En esta nueva modificación no se están eliminando las anteriores si no que, se complementan. Como podemos ver el artículo 12.4c):

“A efectos del cómputo de horas extraordinarias la jornada de cada trabajador se registrará día a día y se totalizará en el período fijado para el abono de las retribuciones, entregando copia del resumen al trabajador en el recibo correspondiente.” (4)

Por este motivo, se requiere proporcionar una forma de poder extraer el registro de todas las horas en un formato en el que se facilite la entrega al empleado.

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

La plataforma pretende almacenar información profesional y personal de sus usuarios por lo que es necesario realizar un correcto cumplimiento de La ley Orgánica 3/2018, de 5 de diciembre, de Protección de datos personales y garantía de los derechos digitales (5).

Para garantizar que se cumplen las medidas respecto a la ley de protección de datos se establecen las siguientes medidas (6):

- Necesidad por parte del usuario de aceptar las condiciones de uso de la plataforma antes de almacenar cualquier dato personal del mismo.
- Únicamente se almacenará la información personal estrictamente necesaria para el correcto uso de la plataforma.
- Los usuarios serán capaces de determinar con qué empresa comparten su información. Por lo que deben de ser capaces de restringir el acceso cuando ellos lo deseen.
- Medidas técnicas para la protección de la información de los usuarios como encriptación de contraseñas, sistemas de permisos y controles de accesos.

El siguiente aspecto analizado en este punto es el marco ético. En él se han tenido en cuenta principalmente los siguientes aspectos:

Sanciones a las empresas

Se considerará una infracción grave según la Ley sobre Infracciones en el Orden Social, en el artículo 7 en el caso de no realizar un correcto uso de las funcionalidades e intentar manipular los datos de los registros.

“La transgresión de las normas y los límites legales o pactados en materia de jornada, trabajo nocturno, horas extraordinarias, horas complementarias, descansos, vacaciones, permisos, registro de jornada y, en general, el tiempo de trabajo a que se refieren los artículos 12, 23 y 34 a 38 del Estatuto de los Trabajadores” (7)

Perjuicios a los trabajadores

El hecho de implantar esta medida no debe de suponer un impacto económico a los trabajadores, sin embargo, al realizar un uso de la herramienta es posible que la empresa tome medidas respecto a retrasos o ausencias no justificadas. Como se puede comprobar la resolución de una reciente sentencia de la Audiencia Nacional.

“por cuanto que la detracción de salarios obedece al lógico desarrollo dinámico de un contrato de naturaleza bilateral y sinalagmática como es el de trabajo como arriba se ha expuesto sin que implique el ejercicio de potestad disciplinaria alguna, mientras que las sanciones arriba mencionadas obedecen al legítimo ejercicio de la potestad disciplinaria previsto legal y convencionalmente” (8)

3.2 Análisis de requisitos

La especificación de requisitos tiene como objetivo describir de forma completa el proyecto, definiendo los requisitos los cuales son necesarios que la aplicación debe de cumplir.

La herramienta que se busca desarrollar está enfocada principalmente a cubrir las necesidades de la pequeña y mediana empresa, para ello se ha buscado garantizar que se cumplan los siguientes requisitos:

- Diseño de una herramienta de gestión y el control de presencia capaz de adaptarse a los principales dispositivos del mercado. Por esta razón se ha planteado una solución web, con esta primera versión de la herramienta se pretende que el máximo número de usuarios sean capaces de utilizarla.
- Los usuarios deben de ser capaces de dar de alta a su empresa y mediante un sistema de invitaciones poder añadir a sus trabajadores.
- Creación y personalización de grupos. En una empresa no todos los trabajadores tienen el mismo puesto de trabajo, por lo que la herramienta debe de dar la posibilidad de asignar a cada uno de los empleados a un grupo, lo que permita hacer distinciones en base a sus cargos y horarios.
- Solicitud y validación de permisos y vacaciones. Los trabajadores deben de ser capaces de solicitar sus permisos retribuidos y las vacaciones. De igual forma, los usuarios responsables tienen que ser capaces de validar y aceptar dichas solicitudes.
- Sistema de control de tiempos mediante el cual los trabajadores deben de indicar en el momento en el que inician y finalizan su jornada de forma fácil y rápida.
- Gestión de las jornadas y las horas extras. La herramienta debe de permitir tener varias jornadas en base al grupo en el que se encuentra el trabajador y registrar las horas extras en base a las jornadas que han realizado sus trabajadores.
- Métricas y seguimiento. Como parte fundamental de la aplicación, los usuarios deben de ser capaces de llevar un seguimiento que les aporte información y valor acerca de cómo están realizando sus trabajadores el registro de su jornada y alertando en caso de anomalías en el uso de la herramienta.

3.2.1 Requisitos funcionales

Registro de usuario

- **Entrada:** Los usuarios se podrán registrar añadiendo sus datos personales.
- **Comportamiento:** se comprobarán los datos introducidos y se aceptarán términos y condiciones.
- **Salida:** El usuario se almacenará en la base de datos y en caso de error se informará al usuario de los campos con fallo.

Inicio de sesión

- **Entrada:** Los usuarios podrán iniciar sesión mediante su usuario o email.
- **Comportamiento:** se deberá comprobar que los datos de usuario y contraseña son válidos.
- **Salida:** el usuario iniciará sesión en la plataforma o se le informara que los datos introducidos no son correctos.

Cerrar sesión

- **Entrada:** Los usuarios que hayan iniciado sesión deben de poder cerrarla.
- **Comportamiento:** se deberá cerrar la sesión del usuario identificado cuando este lo indique.
- **Salida:** El usuario deberá ser redirigido a la página de inicio de sesión al cerrar su sesión.

Editar Perfil

- **Entrada:** El usuario deberá de ser capaz de modificar sus datos personales y su imagen de perfil.
- **Comportamiento:** Se deberá de validar los datos introducidos y actualizarlos en caso de que estos se validen.
- **Salida:** Los datos del usuario deberán de actualizarse.

Registrar Empresa

- **Entrada:** Los usuarios deberán de poder dar de alta una empresa en la plataforma.
- **Comportamiento:** Validar los datos de la empresa introducidos por el usuario.
- **Salida:** En caso de ser validos se dará de alta la empresa en la plataforma.

Unirse a empresa

- **Entrada:** Los usuarios deberán de poder unirse a una empresa a la que hayan sido invitados.
- **Comportamiento:** Se deberá validar que el usuario haya sido invitado a una empresa.
- **Salida:** En el caso de que el usuario pertenezca a la empresa podrá acceder a ella.

Realizar un fichaje

- **Entrada:** El usuario deberá de ser capaz de realizar fichajes de entrada o de salida.
- **Comportamiento:** cuando el usuario seleccione que ha realizado el fichaje este se deberá de registrar en la plataforma.
- **Salida:** El usuario deberá de ser notificado que ha realizado un fichaje.

Fichar de forma remota

- **Entrada:** El usuario deberá de poder realizar un fichaje sin iniciar sesión.
- **Comportamiento:** Se deberá de validar el usuario y contraseña, en caso de ser correcto se registrará el fichaje en la base de datos.
- **Salida:** El usuario deberá ser notificado en caso de haber indicado los datos correctos.

Consultar los fichajes realizados

- **Entrada:** El usuario deberá de poder consultar los fichajes que ha realizado.
- **Comportamiento:** Se deberá de mostrar un listado de los fichajes que ha realizado el usuario.
- **Salida:** El usuario podrá ver el listado de los fichajes realizados.

Consultar Permisos/Ausencias

- **Entrada:** El usuario deberá de poder consultar los permisos y ausencias que ha solicitado.
- **Comportamiento:** Se deberá de mostrar un listado de los permisos solicitados por el usuario.
- **Salida:** El usuario podrá ver el listado de los permisos y ausencias solicitados.

Solicitar Permisos/Ausencias

- **Entrada:** El usuario podrá solicitar un permiso o una ausencia
- **Comportamiento:** En el caso de que cumpla los requisitos dependiendo del motivo, las fechas y los días de vacaciones disponibles el usuario podrá solicitar un permiso o una ausencia.
- **Salida:** El usuario deberá de ver que ha solicitado y las repercusiones en sus días de vacaciones.

Invitar Usuarios a la Empresa

- **Entrada:** Un usuario podrá invitar a otros usuarios a unirse a la empresa
- **Comportamiento:** Se deberá de comprobar que el usuario tenga permisos de administración dentro de la empresa, que el usuario invitado exista y no forme ya parte de la misma.
- **Salida:** El usuario invitado deberá de ser capaz de unirse a la empresa a la que ha sido invitado.



Editar Usuarios de la empresa

- **Entrada:** El usuario deberá de poder editar los datos relacionados con la empresa de un usuario.
- **Comportamiento:** El usuario deberá de contar con permisos de administración en la empresa, se validarán los datos introducidos y se actualizarán en la base de datos.
- **Salida:** se deberá de redirigir a la ficha del empleado y ver los datos actualizados.

Eliminar Usuarios de la empresa

- **Entrada:** El usuario deberá de poder dar de baja a alguien dentro de la empresa
- **Comportamiento:** Se deberá de validar que el usuario tenga permisos de administración dentro de la empresa.
- **Salida:** Se deberá de quitar el acceso y borrar los datos del empleado relacionados con la empresa.

Crear grupo de trabajo

- **Entrada:** El usuario deberá de poder crear grupos de usuarios.
- **Comportamiento:** Se deberá de validar que el usuario sea administrador en la empresa, los datos introducidos y los usuarios añadidos al grupo de trabajo.
- **Salida:** El usuario de trabajo podrá ver el grupo de trabajo y los miembros de este.

Editar fichaje de un usuario de la empresa

- **Entrada:** El usuario podrá editar los fichajes realizados.
- **Comportamiento:** Se deberá de validar que los datos no se solapen con otros fichajes y las fechas sean correctas, en caso de ser modificado por un usuario no administrador se debe marcar una alerta en el fichaje.
- **Salida:** El usuario deberá de ver el fichaje con los datos modificados.

Exportar fichajes realizados

- **Entrada:** El usuario deberá de poder exportar el listado de los fichajes.
- **Comportamiento:** Se deberá de generar un Excel con los datos de los fichajes del usuario.
- **Salida:** Se deberá descargar el archivo generado con los fichajes.

Gestionar permisos/ausencias

- **Entrada:** El usuario deberá de poder aceptar o rechazar los permisos o ausencias solicitado.
- **Comportamiento:** Se deberá de validar que el usuario tenga permisos de administración dentro de la empresa.
- **Salida:** Se deberá de mostrar los datos de la resolución del permiso o ausencia.

Crear horarios

- **Entrada:** El usuario deberá de poder crear un horario
- **Comportamiento:** Se deberá de comprobar que el usuario es administrador dentro de la empresa y validar los datos introducidos dependiendo del horario que se ha deseado crear.
- **Salida:** Se deberá mostrar el horario creado por el usuario

Crear eventos de la empresa

- **Entrada:** El usuario deberá de ser capaz de crear eventos de una empresa
- **Comportamiento:** Se deberá comprobar que el usuario sea administrador dentro de la empresa y validar los datos introducidos en el formulario.
- **Salida:** Se deberá de reflejar el evento creado en el calendario.

3.2.2 Requisitos no funcionales

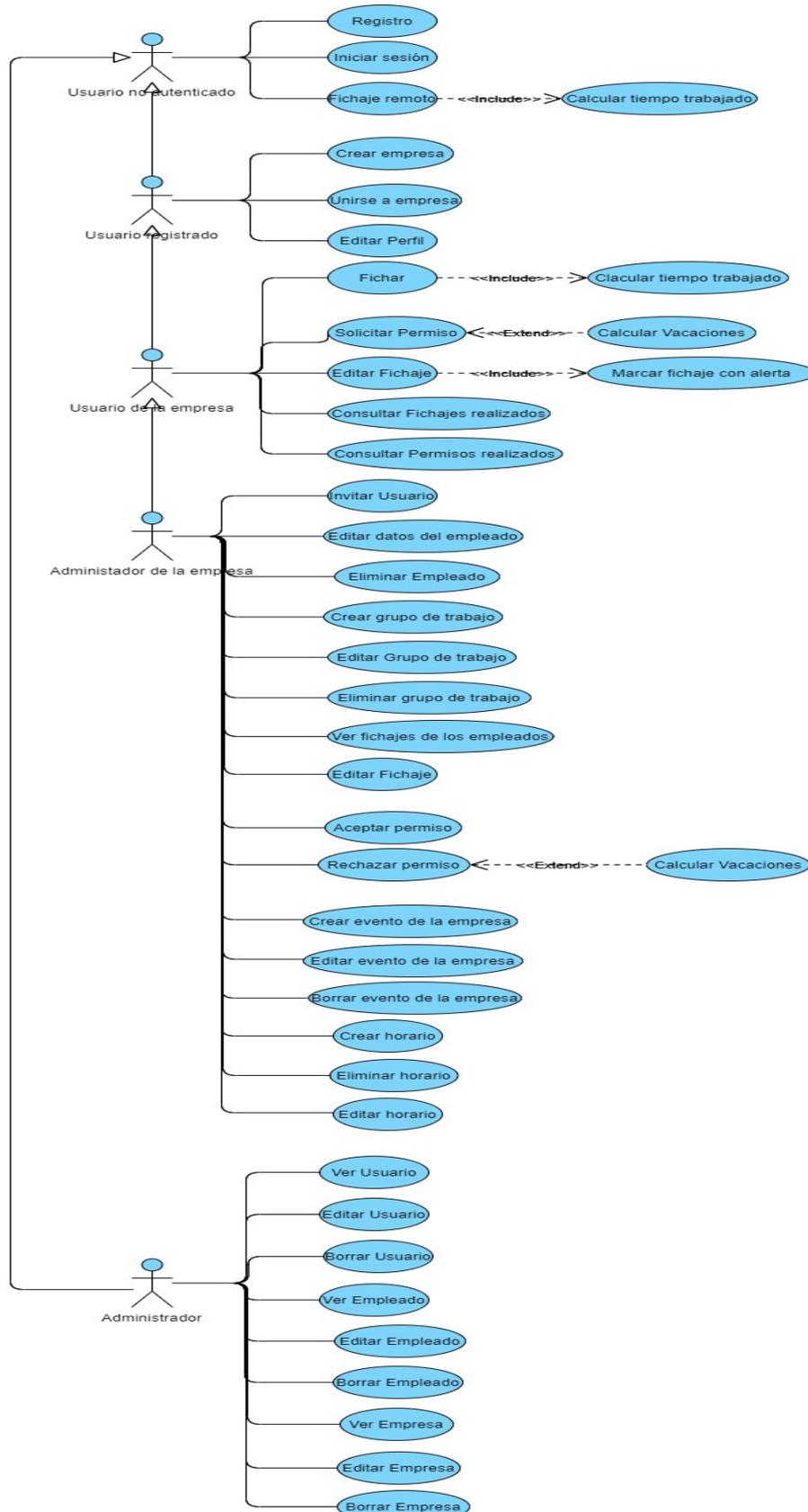
Los requisitos no funcionales que debe cumplir la aplicación son los siguientes:

- **Rendimiento:** Garantizar un funcionamiento fluido con tiempos de carga aceptables, independiente del volumen de datos de la base de datos.
- **Accesibilidad:** Se deberá poder garantizar que, desde cualquier dispositivo con un navegador y una conexión a internet, pueda acceder a la plataforma.
- **Mantenibilidad:** La aplicación deberá de estar actualizada y mantenida asegurando la actualización de las librerías de seguridad.
- **Escalabilidad:** La estructura de la aplicación se deberá de construir pensando en que pueda aumentar sus funcionalidades de forma sencilla de cara al desarrollo.
- **Usabilidad:** Cualquier persona de entre 18 y 65 años o comprendida en una edad laboral deberá de ser capaz de utilizarla.
- **Seguridad:** La aplicación deberá de contar con un sistema de autenticación, además de almacenar las contraseñas de forma encriptada. Se deberá de restringir los accesos en función del rol de los usuarios y sus permisos dentro de la plataforma.



3.3 Casos de Uso

A continuación, se presenta el diagrama de casos de uso con los distintos actores y las acciones que pueden llevar a cabo en el sistema.



3.4 Identificación y análisis de soluciones posibles

Durante el análisis para encontrar la mejor solución posible para desarrollar la herramienta se han planteado varias opciones:

Aplicación de escritorio

Una aplicación de escritorio utilizando como lenguaje de programación JAVA ha sido una de las opciones que se ha valorado, ya que los usuarios están familiarizados en instalar este tipo de programas.

Las aplicaciones de escritorio permiten a los empleados instalar el programa y usarlo de forma sencilla, en cambio, al establecer el objetivo de alcanzar el mayor número de terminales, ya sea ordenadores o puntos de acceso simplemente con un navegador, al requerir de la instalación de la máquina virtual de JAVA, se optó por descartar esta opción.

Aplicación móvil

Es una de las opciones valoradas de forma más positiva para el desarrollo, ya que es una de las opciones más cómodas para los usuarios y a las cuales están más familiarizados.

Esta solución se ha descartado por el motivo de la necesidad de proporcionar la aplicación como una herramienta para la administración de los usuarios por lo que una aplicación móvil dificultaría esta labor.

3.5 Solución propuesta

La solución que se propone para el desarrollo del proyecto ha sido la de una aplicación web.

La aplicación web cumplirá a las características siguientes:

- Accederá al mayor número de usuarios, ya que cualquier dispositivo con conexión a internet y un navegador es capaz de realizar acciones sobre la plataforma.
- Establecerá las bases para el desarrollo de otras aplicaciones que accedan a ella, mediante la realización de la parte servidora y una API otras aplicaciones ya sean de escritorio o móvil podrán realizar las mismas funciones que mediante la solución web.
- Permitirá una mayor escalabilidad ya que, incrementando los recursos de nuestro servidor, es posible cumplir con la demanda en caso de que la adopción de la herramienta aumente.



Plan de trabajo

Para la realización del TFG se ha definido un plan de trabajo, el cual se ha dividido en varias fases.

Primeramente, se ha realizado una lista de todos los apartados necesarios para su cumplimiento, realizando una estimación de los mismos.

Una vez finalizada cada fase se ha anotado el tiempo invertido en dicha fase para detectar las desviaciones y poder hacer en retrospectiva un análisis de cara a mejorar la fase de estimación del proyecto.

A continuación, se muestra la tabla resumen con las estimaciones realizadas a lo largo del desarrollo del presente TFG:

TAREA	TIEMPO ESTIMADO (HORAS)	TIEMPO INVERTIDO (HORAS)
PLANTEAMIENTO DE OBJETIVOS Y PLANIFICACIÓN	3	3
INTRODUCCIÓN	5	8
MOTIVACIÓN	3	2
OBJETIVOS	2	4
METODOLOGÍA	4	6
ESTADO DEL ARTE	6	12
PROPUESTA	2	1
ANÁLISIS DEL PROBLEMA	4	8
IDENTIFICACIÓN Y ANÁLISIS DE POSIBLES SOLUCIONES	4	6
SOLUCIÓN PROPUESTA	2	6
PLAN DE TRABAJO	2	3
PRESUPUESTO	2	2
DISEÑO DE LA SOLUCIÓN	6	8
DESARROLLO DE LA SOLUCIÓN	40	70
IMPLANTACIÓN	2	4
PRUEBAS	2	2
CONCLUSIONES	2	3
TOTAL	90	148

Una vez recopilada la estimación y el total del tiempo invertido se puede concluir que la estimación inicial de la mayor parte de los apartados del proyecto se ha estimado de una forma demasiado optimista, la desviación principalmente se ha debido a realizar correcciones y al proceso de recopilar información, por lo que en futuras estimaciones se debe de tener este factor en cuenta para no repetir los mismos errores.

Presupuesto

En este apartado se analizará el presupuesto planteado en relación al coste de los recursos empleados del desarrollo del proyecto. Para ello se asignará un precio hora del tiempo empleado del alumno en el desarrollo del proyecto, así como los recursos de servicios de terceros empleados para la puesta en marcha y mantenimiento de la plataforma.

El salario de un ingeniero informático que acaba de terminar su carrera o grado puede rondar los 18.000 y 22.000 euros brutos al año (9), basándonos en este dato, se establece un precio medio de 10 euros brutos por hora de trabajo del alumno.

A continuación, se muestra una tabla con el presupuesto realizado del proyecto:

COSTOS	TIEMPO	PRECIO
DESARROLLO	70 horas	700 €
DOMINIO		5€ / año
SERVIDOR		60€ / año
COSTOS TOTALES		765€

Con esta tabla podemos concluir que el costo principal ha sido el desarrollo del proyecto, poniendo en en valor el trabajo realizado por el alumno, el cual ha sido necesario para la realización del presente TFG.

4. Diseño de la solución

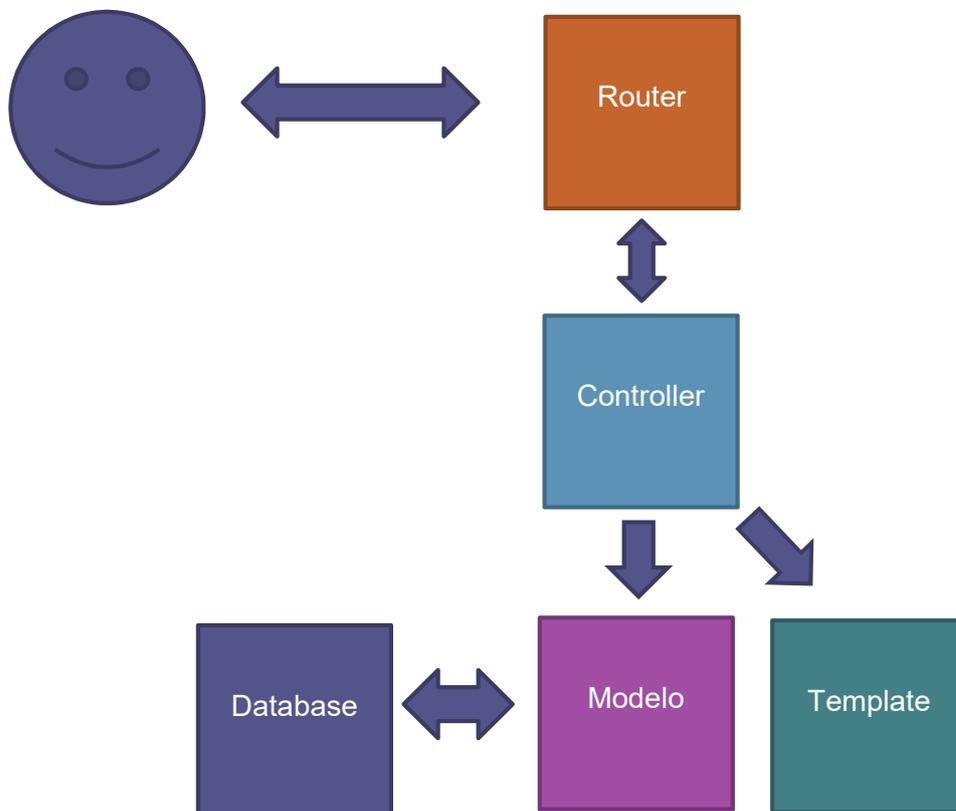
A continuación, se expondrán las distintas tecnologías, lenguajes de programación y entornos de desarrollo que se han utilizado para el desarrollo de la plataforma.

También se van a describirán las técnicas y métodos utilizados para el control de calidad, integración continua y el control de versiones.

4.1 Arquitectura del Sistema

El desarrollo de la aplicación se ha basado en el patrón clásico del diseño web conocido como MVC o Modelo vista controlador, el cual está compuesto por 3 niveles principalmente:

- **Modelo:** es la parte de la lógica de negocio o los servicios con los que trabaja la aplicación.
- **Vista:** Es el resultado de lo que se muestra al usuario, utilizando la lógica procesada del modelo se encarga de construir una respuesta con la que el usuario puede interactuar.
- **Controlador:** El controlador es la parte encargada de procesar las interacciones demandadas por el usuario, orquestando los servicios y el modelo para obtener el resultado de los diferentes procesos.



4.2 Diseño Detallado

En este apartado se verá de manera más detallada como se ha realizado el diseño y la organización de las distintas partes del proyecto.

4.2.1 Estructura de la aplicación Symfony

Siguiendo la estructura de un proyecto Symfony se ha establecido la siguiente estructura de directorios la cual procedemos a explicar paso a paso a continuación:

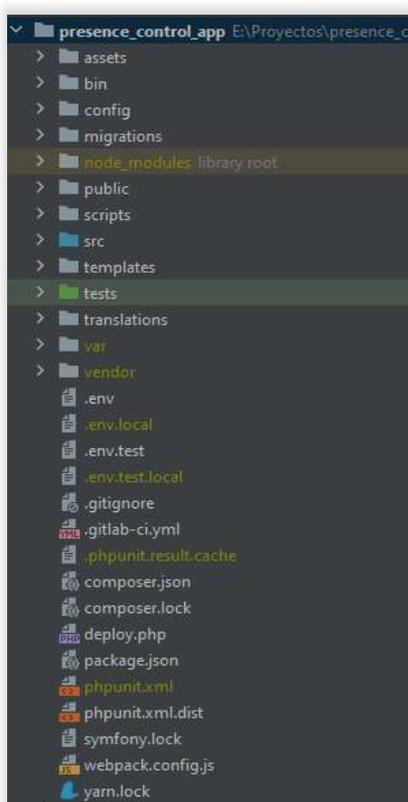


Ilustración 8. Estructura de la aplicación

Assets

Esta carpeta contiene los archivos web de la aplicación como son los css, javascript o imágenes.

Bin

En esta carpeta podemos observar los archivos ejecutables de la aplicación.

Dentro podemos encontrar la parte fundamental de un componente que nos proporciona symfony “Console”, el cual nos permite crear líneas de comando que pueden ser usadas para crear tareas, importaciones y otra serie de acciones.

Config

En este directorio se encuentra la configuración relacionada con los diferentes componentes de symfony y ésta puede ser configurada dependiendo del entorno.

Migrations

En esta carpeta se encuentran los archivos que modifican el modelo de datos.

Node_modules

Este directorio no está versionado ya que contiene las dependencias utilizadas en el proyecto.

Public

En public colocaremos todo lo que sea accesible por los usuarios que utilizan la plataforma.

Scripts

Esta carpeta contiene los archivos ejecutables que se han desarrollado.

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Src

En esta carpeta se encuentra el código PHP del proyecto que contiene la lógica de la aplicación.

Templates

Plantillas y fragmentos de plantillas forman la parte visual de la plataforma.

Tests

Pruebas funcionales que verifican el funcionamiento de los módulos de la aplicación.

Translations

Archivos utilizados para la traducción de los textos de la plataforma.

Var y vendor

Las carpetas de var y vendor están excluidas del control de versiones, ya que sirven para almacenar ficheros relacionados con las librerías externas, los cuales están gestionados por los gestores de dependencias.

Archivos de configuración

Dentro de la carpeta principal del proyecto podemos encontrar una serie de archivos encargados de configurar ciertas partes de la aplicación, los más importantes son:

- **env**: Establecen las diferentes variables de entorno del proyecto.
- **gitignore**: En este fichero se determinan qué archivos y directorios van a estar excluidos de ser versionados.
- **git-lab.ci**: Archivo en el que se encuentra la configuración de la integración continua.
- **composer**: Configuración de las dependencias instaladas en el proyecto.
- **deploy**: Archivo PHP que tiene la configuración necesaria para el despliegue del proyecto.
- **phpunit**: Archivo de configuración para la ejecución de los tests.
- **webpack**: Configuración del procesador de archivos web.
- **yarn**: Configuración para las dependencias y librerías del frontend.

4.2.2 Modelo de datos

La estructura que sigue a continuación muestra las entidades que forman las tablas de la base de datos como se muestra en la ilustración 9.

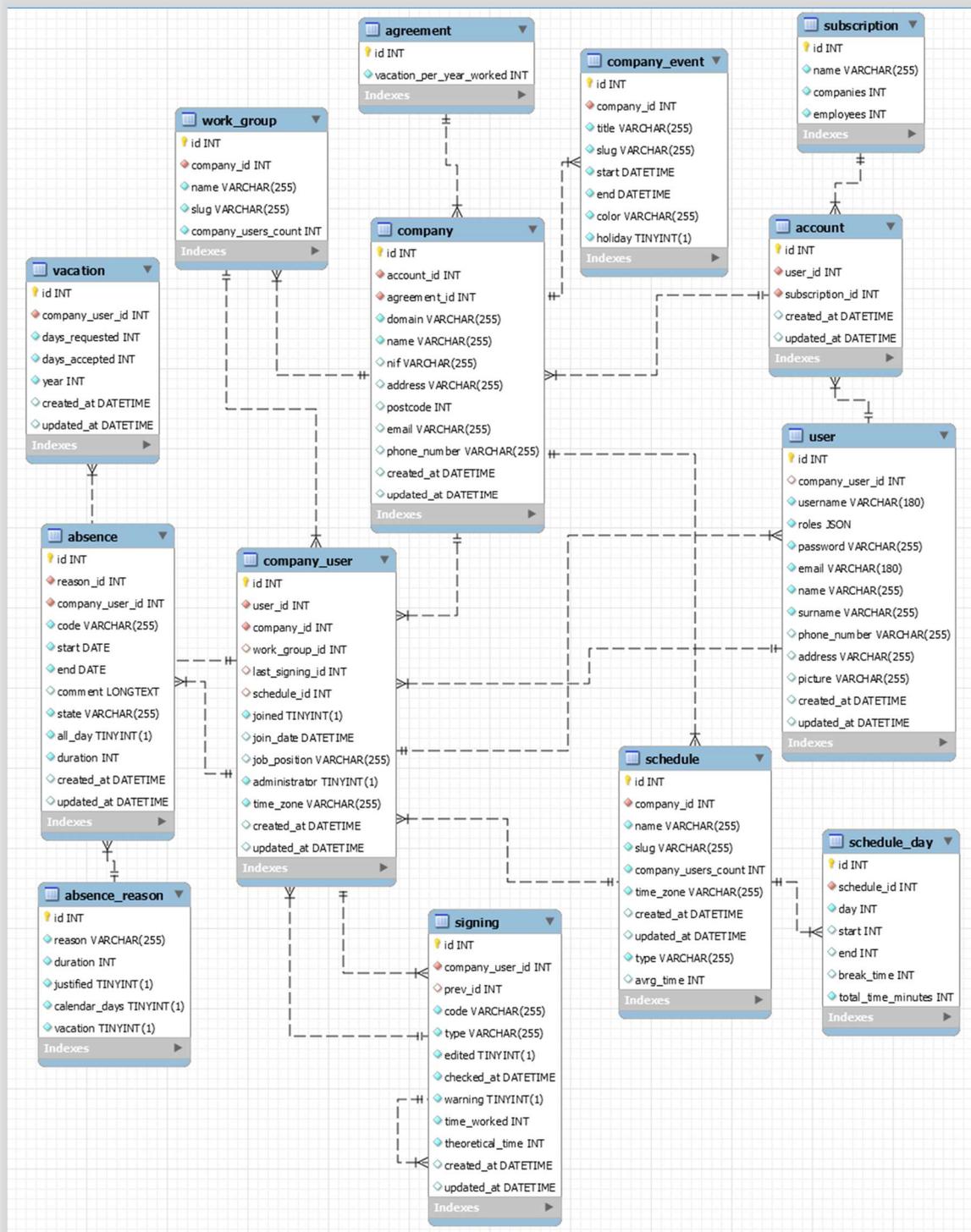


Ilustración 9. Diagrama de la base de datos

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

El diagrama planteado en el proyecto consta de 14 tablas, de las cuales podemos constatar que las principales son las siguientes:

- **User:** Almacena la información principal del usuario, ya que para acceder a las funcionalidades de la plataforma es necesario registrarse dentro de ella. Esta tabla guarda los datos personales, así como sus roles y el tipo de cuenta.
- **Company:** Guarda la información de las empresas que registran los usuarios, sobre la cual construyen el resto de la estructura de la organización.
- **CompanyUser:** Establece la relación entre el usuario y la empresa, guardando la información determinado el rol que constituye el usuario dentro de la misma.
- **Signing:** Fichaje de entrada o de salida que guarda la información de todos los registros que realiza un usuario dentro de la plataforma.
- **Schedule:** Horario que tiene un usuario dentro de la empresa, permitiendo indicar el tipo de horario incluyendo la información a nivel de día.
- **Absence:** Permiso o ausencia que recoge la información necesaria para determinar el tipo de permiso que ha solicitado o ha registrado un usuario.
- **WorkGroup:** Con el fin de organizar a los usuarios dentro de una empresa, se ha establecido una tabla para facilitar su posición, para eso se ha creado esta tabla llamada grupo de trabajo.



4.3 Tecnología Utilizada

En este apartado se pretende mostrar el conjunto de tecnologías utilizadas para el desarrollo de la solución.

4.3.2 PHP

PHP (acrónimo recursivo de PHP: Preprocesador de Hipertexto) es un lenguaje de programación orientado a objetos, de código abierto y especialmente popular para el desarrollo web (10).

Este lenguaje se ejecuta en la parte del servidor, manipulando la presentación la gestión de los datos y la información que posteriormente es enviada al cliente, siendo este utilizado para realización de sitios web con diseño dinámico permitiendo realizar búsquedas, consultas, inserción, modificación y eliminación de datos mediante peticiones desde la parte cliente.

Actualmente el 79% de todos los sitios web del mundo se crean utilizando PHP, desde proyectos únicamente utilizando este lenguaje como con la utilización de sus diferentes frameworks.

Las principales características por las que se ha elegido este lenguaje de programación son:

- Gran cantidad de documentación: PHP es utilizado por millones de usuarios en una gran cantidad de proyectos de manera que, para un programador nuevo supone una gran ventaja, ya que es de vital importancia que detrás de un lenguaje exista una gran comunidad.
- Ofertas de trabajo: Cada día se desarrollan webs más complejas que funcionan bajo PHP, así como una gran cantidad en su día realizaron su proyecto en este lenguaje y actualmente tienen la necesidad de actualizar su plataforma.
- Programación orientada a objetos: Esta característica dota al lenguaje de la posibilidad de abstraer y dividir los problemas en partes más pequeñas e independientes aumentando la reusabilidad, mantenibilidad y escalabilidad.
- Multiplataforma: PHP es un lenguaje totalmente libre y multiplataforma lo que puede funcionar en cualquier tipo de servidor.

4.3.3 Symfony

Symfony es un framework de desarrollo de aplicaciones web basado en el patrón Modelo Vista Controlador construido sobre componentes los cuales son desacoplados y reutilizables.

SensioLabs es la empresa creadora de este framework y actualmente es utilizado por miles de empresas de desarrollo en todo el mundo.

La primera versión de Symfony fue lanzada en octubre de 2005 y fue utilizado como framework de PHP por la empresa Yahoo, con el que desarrolló el proyecto Yahoo Bookmarks.

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Las principales ventajas y características por la que se ha elegido este framework de desarrollo han sido:

- Licencia MIT: licencia de software libre permisiva, lo que permite su modificación al no poseer Copyright.
- Gran aceptación y popularidad entre los programadores de PHP en Europa, lo que le proporciona una gran comunidad que ofrece documentación, formación, consultorías y desarrollo de proyectos.
- Escalabilidad: Symfony es un framework que permite la creación desde sitios web sencillos a grandes proyectos profesionales.
- Componentes: Gran cantidad de componentes reutilizables han sido creados por la comunidad lo que proporciona una forma de desarrollo rápida y confiable.

La popularidad de Symfony no ha hecho más que aumentar, aunque el framework de PHP Laravel está clasificado como el más usado, Symfony no se queda atrás, como el segundo y siendo cada vez más utilizado como podemos comprobar en ilustración 10.

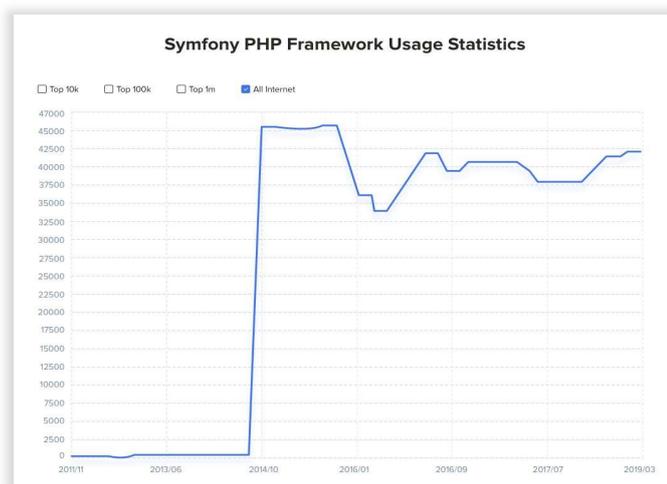


Ilustración 10. Uso de Symfony a lo largo del tiempo

Componentes de Symfony

Los principales componentes utilizados en el proyecto y por los que se ha tomado la decisión de usar este framework han sido los siguientes:

Twig plantillas

Motor de plantillas para el lenguaje de programación PHP. Nace con el objetivo de facilitar el desarrollo de aplicaciones web, pensado para proyectos que usan el patrón MVC facilitando el trabajo en la parte de la vista.

Cuando nos referimos a una plantilla de Twig se está hablando de un archivo de texto que soporta una serie de formatos como puede ser HTML, XML, etc. el cual permite añadir expresiones lógicas que son remplazadas por valores una vez la plantilla se evalúa.

A continuación, en la ilustración 11 se puede ver un ejemplo de la plantilla usada en el perfil del empleado.

```
{% block content %}
<div class="row">
  <div class="col-md-12 col-lg-6">
    <div class="widget-head-color-box navy-bg p-lg text-center">
      <div class="m-b-md">
        <h2 class="font-bold no-margins">
          {{ app.user.name | capitalize }} {{ app.user.surname | capitalize }}
        </h2>
        {% if app.user.companyUser is not null %}
        <h3>{{ app.user.companyUser.jobPosition }}</h3>
        {% endif %}
      </div>
      {% if app.user.picture %}
      
      {% else %}
      
      {% endif %}
    </div>
    <div class="widget-text-box">
      <h4 class="media-heading">{{ app.user.username }}</h4>
      <h4 class="media-heading">{{ app.user.name | capitalize }} {{ app.user.surname | capitalize }}</h4>
      <table class="table">
        <tbody>
          <tr>
            <th>Email</th>
            <td>{{ app.user.email }}</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

Ilustración 11. Plantilla Twig del perfil del empleado

Xml traducciones

Con el fin de aumentar la escalabilidad de la plataforma en términos de “internalización” se ha tomado la decisión de usar el componente de Symfony translations, consiguiendo abstraer una gran cantidad de mensajes y cadenas de textos según la configuración regional de los usuarios (idioma y país).

Para ello se ha configurado la plataforma para establecer como lenguaje principal el castellano como podemos comprobar en la ilustración 12. Para ello se ha trabajado desde el lenguaje predeterminado del framework, que es el inglés, para adaptar la mayor cantidad de mensajes establecidos que proporcionan otros componentes como puede ser el de las validaciones.

```
framework:
  default_locale: es
  translator:
    default_path: '%kernel.project_dir%/translations'
    fallbacks:
      - es
```

Ilustración 12. Configuración del lenguaje



Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Mediante este sistema, conseguimos mostrar respuestas al mercado objetivo inicial que es el español y permitiendo adelantar una gran parte del trabajo de cara a la internalización. Adaptando una gran parte del texto total de la plataforma al inglés ya que es el idioma más expandido a lo largo del mundo.

A continuación, se muestra como se han agrupado y establecido las traducciones dentro del proyecto: Ilustración13 E ilustración 14

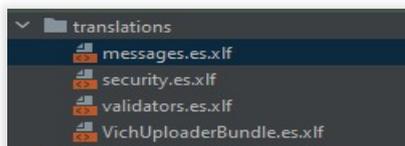


Ilustración 13. Archivos de traducciones

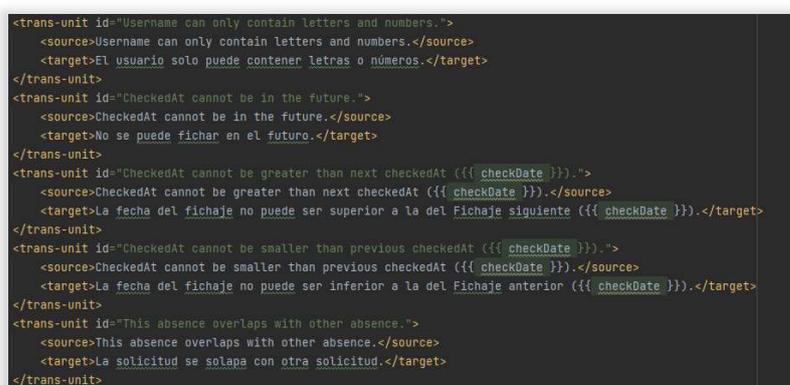


Ilustración 14. Traducciones de las validaciones

EasyAdmin

Componente que ha permitido crear un panel de administración muy completo en el cual los usuarios con permisos de administración pueden ser capaces de llevar un control de las principales clases de la plataforma.

EasyAdmin mejora la experiencia de desarrollo a la hora de implementar las funciones básicas que buscas de un CRUD, proporcionando de forma rápida sin tener que involucrarse tanto en las opciones visuales, proporcionando una interfaz visualmente agradable y permitiendo centrarte más en la funcionalidad necesaria a la hora de administrar una plataforma.

Migrations

Las migraciones dentro de las bases de datos son una forma segura y fácil de actualizar un esquema en cualquier entorno, tanto desarrollo como producción.

Symfony incorpora la posibilidad de crear estas migraciones y aplicarlas mediante la línea de comandos. Además, detecta cambios en la estructura del modelo en base a las anotaciones descritas en las entidades de cada clase que representa una tabla en la base de datos.

FlySystem

Librería que nos permite abstraer el sistema de almacenamiento de documentos de la aplicación, pudiendo estos ser almacenados donde deseemos, sin preocuparnos si estos se encuentran en un directorio local, S3 de Amazon o Blob Storage de Azure.

4.3.4 MySQL



Sistema de gestión de bases de datos relacional, desarrollado inicialmente por la empresa MySQLAB y con su lanzamiento en 1994, este popular proyecto se ha hecho un hueco hasta ser adquirido posteriormente y hasta día de hoy por la empresa Oracle Corporation.

MySQL es la base de datos de código abierto más utilizada y popular del mundo. Su uso va desde pequeños proyectos a los sitios web más grandes del mundo como Wikipedia, Google, Facebook, Twitter y Youtube,

Principalmente utilizado para la realización de aplicaciones web MySQL, forma parte de los stacks más populares como LAMP (Linux + Apache+ MySQL + PHP) y LEMP (Linux + Nginx+ MySQL + PHP).

4.3.5 Javascript

JavaScript es un lenguaje de programación, interpretado, ligero, utilizado principalmente en el lado del cliente.

Se ha utilizado en el proyecto para otorgar de dinamismo a la parte visual.

4.3.6 Vue.js

Vue es un framework progresivo, diseñado para la construcción de interfaces de usuario.

La principal característica por la que se ha decidió utilizar este framework es el modularidad que ofrece, a diferencia de otros los cuales están diseñados con una estructura monolítica, lo que le proporciona una adaptación e integración con Symfony perfecta, dotándolo de la posibilidad de reutilizar componentes en cualquier parte de la aplicación.

Vue es altamente utilizado y cada vez gana más terreno en el mundo laboral, por lo que lo convierte en la mejor opción a la hora de elegir un framework para la parte frontend del proyecto.

4.3.7 Git

Software de control de versiones, diseñado para mejorar la eficiencia, compatibilidad y confiabilidad en el mantenimiento de versiones que se realizan sobre los archivos compartidos principalmente en un repositorio.

Git otorga una gran libertad a la forma de trabajar sobre un proyecto. Permitiendo la coordinación entre grupos de trabajo en torno a una metodología de trabajo en grupo, mejorando el flujo de trabajo y aumentando la productividad.

Git-Flow

Durante la realización del proyecto se ha establecido como flujo de trabajo Git-Flow⁸, pensado para proyectos que utilizan metodologías ágiles y se basan en entregables.

Basado principalmente en la utilización de dos ramas principales (master y develop) y ramificando a partir de estas las otras ramas con tiempo finito que sirven de apoyo ya será para la solución de errores o el desarrollo de nuevas funcionalidades.

Durante el desarrollo del proyecto se ha seguido el siguiente flujo de trabajo:

- Master es la rama que se despliega en el entorno de producción
- Develop es la rama utilizada como punto de unión entre las diferentes funcionalidades convergiendo en esta rama al implementar una funcionalidad
- Las ramas de Feature deben de ser creadas a partir de la rama develop y servirán para desarrollar funcionalidades que aporten valor al proyecto
- Release consiste en una rama diseñada para realizar pruebas en entornos de pre producción permitiendo testear las nuevas funcionalidades y abordar problemas previos a la puesta en producción del proyecto.
- Hotfix son ramas creadas a partir de master, las cuales sirven para solucionar errores que hayan podido llegar a producción una vez finalizada la corrección la unión del desarrollo realizado en estas ramas se integrara tanto en master como en develop.

A continuación, en la ilustración 15 se muestra un esquema con el flujo utilizado:

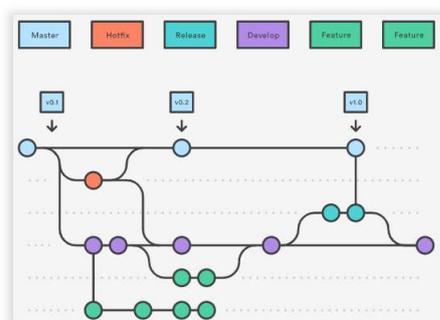


Ilustración 15. Flujo de trabajo de Gitflow

⁸ <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>

4.3.8 Gestores de dependencias

Este proyecto usa una gran cantidad de frameworks y librerías. Por lo que mantener el código actualizado sería imposible sin la utilización de un sistema que nos permita gestionar de forma eficiente todas las dependencias del proyecto.

Los gestores de dependencias utilizados permiten administrar, descargar e instalar tanto dependencias como librerías, lo que evita descargar cada dependencia de forma manual automatizando este tedioso proceso.

Composer

Manejador de paquetes PHP, permite declarar las librerías de las cuales depende el proyecto y las administra, instalándolas y actualizándolos de forma automática.

Yarn

Administrador de paquetes que se ha usado en el proyecto para la gestión de dependencias de las tecnológicas utilizadas en la parte de la interfaz de usuario.

Yarn, al igual que composer, permite: manejar, actualizar y gestionar las librerías de forma automática, aumentado la escalabilidad de la plataforma permitiendo adaptar las actualizaciones adaptando el proyecto a los cambios ya que se están utilizando tecnologías novedosas las cuales se encuentran en constante cambio.



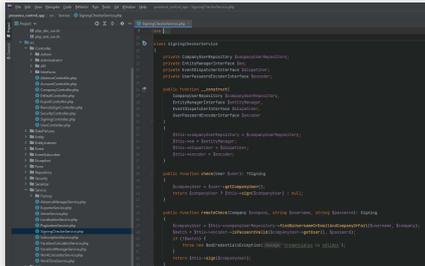
4.4 Herramientas

En este apartado se verán las principales herramientas utilizadas para el desarrollo de la aplicación, el manejo y la gestión de la base de datos.

4.4.1 PHPStorm



Para el desarrollo de la aplicación ha sido fundamental el uso del IDE PHPStorm, debido a su integración con los frameworks de Symfony y Vue, mediante una serie de plugins proporcionados por el mismo IDE.



PHPStorm ofrece compatibilidad con los principales frameworks de trabajo, ideal para trabajar con Symfony, Drupal, Laravel, además de las tecnologías front-end más utilizadas como HTML 5, CSS, Sass, JavaScript.

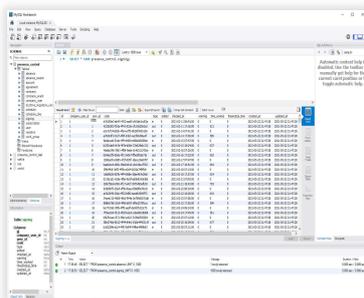
El plugin de Symfony proporciona una serie de facilidades en la navegación para desarrollar, ya que es capaz de detectar todas las tecnologías utilizadas ya sean twig, anotaciones, componentes de Vue o clases en PHP que es capaz de forma fácil permitirte acceder a los recursos que estás buscando durante el desarrollo.

Re factorizar de forma rápida y segura es otra de las principales características permitiéndonos una forma confiable de realizar operaciones como formatear nuestro código según el estándar de programación, eliminar librerías no utilizadas en una clase o extraer un pedazo de nuestro código en un método lo que agiliza y otorga de calidad al código.

4.4.2 MySQL Workbench



MySQL Workbench es una herramienta enfocada a desarrollares, arquitectos y administradores de bases de datos.



La herramienta proporciona una interfaz visual que permite modelar, generar y administrar bases de datos.

MySQL Workbench está disponible en los sistemas operativos de Windows, Linux y Mac OS de forma gratuita.

Las herramientas que nos ofrece MySQL Workbench nos permiten ejecutar, crear y optimizar consultas SQL mediante un editor de texto el cual nos resaltara la sintaxis. Además, no ofrece autocompletado y la reutilización de fragmentos de código.

5. Desarrollo de la solución propuesta

En este punto se realiza un recorrido a través de todas las partes de la aplicación web, como se ha desarrollado y estructurado para al final ver el resultado final de la misma.

5.1 Desarrollo de la aplicación web

En este apartado se explica en detalle que contienen los directorios del proyecto, además analizaremos y profundizaremos en detalle cómo está estructurado el código del proyecto.

5.1.1 Assets

En la ilustración 16 se puede ver la estructura de esta carpeta la cual contiene recursos utilizados en la parte de la vista, los cuales podemos catalogar principalmente de la siguiente forma:

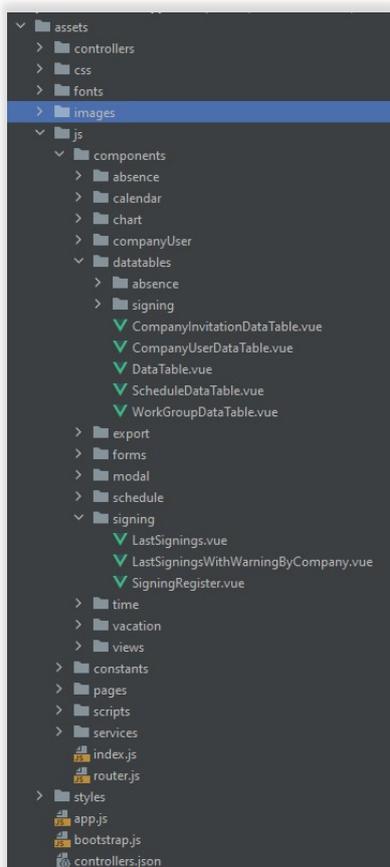


Ilustración 16. Estructura de la carpeta Assets

- **Estilos:** en la carpeta de css se encuentran los ficheros destinados a otorgar un diseño a los distintos elementos de la vista.
- **JavaScript:** dentro de la carpeta js se pueden encontrar recursos y componentes de Vue.js creados para otorgar reactividad y dinamismo a la capa de la vista
- **Imágenes:** las imágenes estáticas dentro de la plataforma.

Los archivos dentro de la carpeta se compilan mediante el uso de webpack⁹ el cual los transforman a elementos estáticos que se almacenan dentro de la carpeta public del proyecto para que sean accesibles para el usuario en la aplicación web.

Con el uso de webpack se ha podido integrar el uso Vue el cual ha proporcionado a la plataforma desarrollar componentes los cuales se han podido utilizar en diferentes vistas.

Para el desarrollo de la aplicación se han diseñado una gran variedad de componentes de Vue con la intención de otorgar reactividad y escalabilidad al proyecto.

⁹ <https://webpack.js.org/>

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

La estructura de un componente de Vue, consta de 3 partes que posteriormente webpack transformara en archivos estáticos, las partes de un componente son las siguientes:

- **Template:** Es la estructura que forma la parte visual del componente, la cual se actualizará dependiendo de las variables computadas y los eventos que sufra el mismo. En la ilustración 17 se puede visualizar el template del componente encargado de los fichajes.
- **Script:** Otorga la lógica al componente, donde se guardan los datos que almacena, las funciones, propiedades computadas, así como la posibilidad de sobrescribir las funciones de su ciclo de vida. En la ilustración 18 se puede ver como es la parte del script del componente de fichajes.
- **Style:** Esta parte permite establecer estilos que se van a aplicar exclusivamente dentro de nuestro componente.

```
<template>
  <div>
    <div v-if="loading" class="text-center">
      <b-spinner label="Loading..."></b-spinner>
    </div>
    <div v-else-if="isWorking">
      <div class="text-center">
        <button @click="doCkeck" class="btn btn-outline btn-danger dim" type="button">
          <i class="fas fa-sign-out-alt"></i>
          Salida
        </button>
      </div>
      <div class="text-center">
        Tiempo Trabajado:
        <minutes-to-time :minutes="minutesWorked"></minutes-to-time>
      </div>
    </div>
    <div v-else class="text-center">
      <button @click="doCkeck" class="btn btn-outline btn-success dim" type="button">
        <i class="fas fa-sign-in-alt"></i>
        Entrada
      </button>
    </div>
  </div>
</template>
```

Ilustración 17. Template del componente Vue de fichajes

```
<script>
import ...

export default {
  name: "SigningRegister",
  components: {
    MinutesToTime,
  },
  data: () => ({...}),
  methods: {
    async doCkeck() {...},
    updateCurrentTime() {...},
    getLastSigning() {...},
    check() {...}
  },
  computed: {
    minutesWorked() {...},
    isWorking() {...}
  },
  created() {...},
  beforeMount() {...}
}
</script>
```

Ilustración 18. Script del componente Vue de fichajes

5.1.2 Config

En esta carpeta se encuentra la configuración necesaria para el funcionamiento de los componentes de Symfony utilizados dentro del proyecto, en la ilustración 19 podemos visualizar el contenido de la misma.



Ilustración 19. Estructura de la carpeta Config

Dentro de esta cantidad de archivos de configuración de los distintos componentes de Symfony podemos destacar los siguientes:

Security.yaml

Con este archivo de configuración podemos establecer que clase va a ser la predeterminada para acceder a la plataforma, en nuestro caso será User, además de los diferentes accesos y restricciones a las diferentes rutas teniendo en cuenta el rol del que dispondrá el usuario autenticado como podemos comprobar en la ilustración 20.

```
access_control:
- { path: ^/get-started, roles: ROLE_USER }
- { path: ^/user, roles: ROLE_USER }
- { path: ^/account, roles: ROLE_USER }
- { path: ^/company, roles: ROLE_USER }
- { path: ^/dashboard, roles: ROLE_USER }
- { path: ^/signing, roles: ROLE_USER }
- { path: ^/absence, roles: ROLE_USER }
- { path: ^/administrator, roles: ROLE_USER }
- { path: ^/api, roles: ROLE_USER }
- { path: ^/admin, roles: ROLE_ADMIN }
```

Ilustración 20. Configuración del control de accesos

Services.yaml

Symfony también cuenta con un sistema que se especifica dentro del archivo de configuración de services.yaml, el cual indica que todas los archivos dentro de la carpeta /src del proyecto van a ser un servicio, mediante esta funcionalidad podemos usar la inyección de dependencias de todas las clases que se encuentren dentro de esta carpeta.

Services.yaml permite configurar eventos que ocurren en el ORM indicando que clase capturara los eventos que sufra una determinada entidad como se aprecia en la ilustración 21.

```
App\Entity\Listener\UserEntityListener:
  tags:
    - { name: 'doctrine.orm.entity_listener', event: 'prePersist', entity: 'App\Entity\User' }
    - { name: 'doctrine.orm.entity_listener', event: 'preUpdate', entity: 'App\Entity\User' }
App\Entity\Listener\CompanyEntityListener:
  tags:
    - { name: 'doctrine.orm.entity_listener', event: 'prePersist', entity: 'App\Entity\Company' }
    - { name: 'doctrine.orm.entity_listener', event: 'preUpdate', entity: 'App\Entity\Company' }
App\Entity\Listener\CompanyEventEntityListener:
  tags:
    - { name: 'doctrine.orm.entity_listener', event: 'prePersist', entity: 'App\Entity\CompanyEvent' }
    - { name: 'doctrine.orm.entity_listener', event: 'preUpdate', entity: 'App\Entity\CompanyEvent' }
App\Entity\Listener\CompanyUserEntityListener:
  tags:
    - { name: 'doctrine.orm.entity_listener', event: 'prePersist', entity: 'App\Entity\CompanyUser' }
    - { name: 'doctrine.orm.entity_listener', event: 'preUpdate', entity: 'App\Entity\CompanyUser' }
App\Entity\Listener\WorkGroupEntityListener:
  tags:
    - { name: 'doctrine.orm.entity_listener', event: 'prePersist', entity: 'App\Entity\WorkGroup' }
    - { name: 'doctrine.orm.entity_listener', event: 'preUpdate', entity: 'App\Entity\WorkGroup' }
App\Entity\Listener\ScheduleEntityListener:
  tags:
    - { name: 'doctrine.orm.entity_listener', event: 'prePersist', entity: 'App\Entity\Schedule' }
    - { name: 'doctrine.orm.entity_listener', event: 'preUpdate', entity: 'App\Entity\Schedule' }
App\Entity\Listener\ScheduleDayEntityListener:
  tags:
    - { name: 'doctrine.orm.entity_listener', event: 'prePersist', entity: 'App\Entity\ScheduleDay' }
    - { name: 'doctrine.orm.entity_listener', event: 'preUpdate', entity: 'App\Entity\ScheduleDay' }
```

Ilustración 21. Configuración de los eventos de las entidades

5.1.3 Migrations

En la carpeta de migraciones se guardan todas las migraciones realizadas, las migraciones son ficheros PHP que ejecutan realizando modificaciones sobre la base de datos y posteriormente se almacena la versión de la migración ejecutada.

Cada migración consta de dos métodos principales, up y post, los cuales indican las sentencias SQL que se deben de ejecutar en el caso de aplicar la migración o revertirla, en la ilustración 22 podemos ver la migración inicial realizada en el proyecto.

Las migraciones también contemplan la posibilidad de realizar métodos una vez finalizadas las migraciones en el caso de que quieras incluir registros predeterminados o cualquier acción al realizar un paso de dicha migración.

```
final class Version20210922193051 extends AbstractMigration
{
    public function getDescription(): string
    {
        return 'Migración inicial';
    }

    public function up(Schema $schema): void
    {
        // This up() migration is auto-generated, please modify it to your needs
        $this->addSql('CREATE TABLE absence (id INT AUTO_INCREMENT NOT NULL, reason_id INT NOT NULL)');
        $this->addSql('CREATE TABLE absence_reason (id INT AUTO_INCREMENT NOT NULL, reason VARCHAR(255))');
        $this->addSql('CREATE TABLE account (id INT AUTO_INCREMENT NOT NULL, user_id INT NOT NULL)');
        $this->addSql('CREATE TABLE agreement (id INT AUTO_INCREMENT NOT NULL, vacation_per_year INT)');
        $this->addSql('CREATE TABLE company (id INT AUTO_INCREMENT NOT NULL, account_id INT NOT NULL)');
        $this->addSql('CREATE TABLE company_event (id INT AUTO_INCREMENT NOT NULL, company_id INT NOT NULL)');
        $this->addSql('CREATE TABLE company_user (id INT AUTO_INCREMENT NOT NULL, user_id INT NOT NULL)');
        $this->addSql('CREATE TABLE schedule (id INT AUTO_INCREMENT NOT NULL, company_id INT NOT NULL)');
        $this->addSql('CREATE TABLE schedule_day (id INT AUTO_INCREMENT NOT NULL, schedule_id INT NOT NULL)');
        $this->addSql('CREATE TABLE signing (id INT AUTO_INCREMENT NOT NULL, company_user_id INT NOT NULL)');
        $this->addSql('CREATE TABLE subscription (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255))');
        $this->addSql('CREATE TABLE user (id INT AUTO_INCREMENT NOT NULL, company_user_id INT NOT NULL)');
    }
}
```

Ilustración 22. Migración inicial

5.1.4 Public

La carpeta de build es el resultado de Webpack, en la que dispondremos de los recursos procesados de la carpeta assets, los cuales se encuentran en forma de archivos simples capaces de ser interpretados por los navegadores, como podemos ver en la ilustración 23.

También se guardarán los archivos subidos por los usuarios, estos recursos se guardan en la carpeta de upload que se encuentra ignorada en Git.

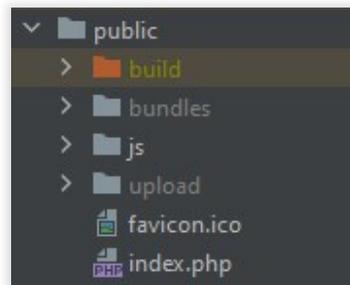


Ilustración 23. Estructura de la carpeta public

5.1.5 Scripts

En esta carpeta se guardan archivos que realizan varias ejecuciones a nivel de entorno, los scripts que disponemos en esta carpeta son los siguientes:

- ci-php-install.sh: Ejecuta y configura las instalaciones necesarias de php en la imagen de docker para la ejecución del proyecto.
- Php_dev_run.sh: Mediante esta ejecución podemos configurar el entorno de desarrollo que nos permitirá restablecer una configuración y base de datos, este comando está pensado para mejorar y agilizar el desarrollo del proyecto.
- Php_test_run: Este archivo de ejecución establecerá los procesos que se deben de cumplir para garantizar la confiabilidad del proyecto, en este archivo se establecen las mismas ejecuciones que se utilizan también en la integración continua, en la ilustración 24 podemos analizar su contenido.

```
php bin/console cache:clear --env=test
php bin/console doctrine:database:drop --force --env=test
php bin/console doctrine:database:create --env=test
php bin/console doctrine:migrations:migrate --env=test --no-interaction
php bin/console doctrine:schema:validate --env=test
php bin/console doctrine:fixtures:load --env=test --no-interaction
php -d memory_limit=512M bin/phpunit tests
```

Ilustración 24. Script de testing

5.1.6 Src

En esta carpeta se encuentra la parte principal del proyecto donde está la parte de la lógica de la aplicación, en la ilustración 25 podemos comprobar su estructura de directorios.

Dentro de esta carpeta se encuentran los archivos agrupados dependiendo del papel que desempeñan en el proyecto, a continuación, Se procede a explicar cómo se ha estructurado esta parte del proyecto.

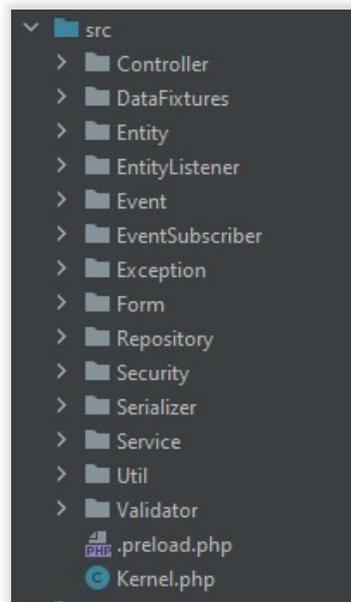


Ilustración 25. Estructura de la carpeta src

Controller

Los controladores tienen la función de capturar las peticiones y leer la información de las “Requests”, posteriormente son los encargados de ejecutar la lógica o llamar a los servicios necesarios para procesar la petición para después, devolver al usuario que realizó la petición una respuesta.

Symfony nos proporciona una clase controlador base, de la cual extenderán el resto de los controladores del proyecto, esta nos proporciona una serie de métodos auxiliares, como pueden ser, el control de accesos o la creación de respuestas entre otros muchos.

```
$this->denyAccessUnlessGranted(AbsenceVoter::ABSENCE_VIEW, $absence);  
return $this->render('absence/show.html.twig', [  
    'absence' => $absence,  
]);
```

Ilustración 26. Comprobación de permisos de acceso

A continuación, se va a explicar cómo se han estructurado los controladores dentro del proyecto, como se puede ver en la ilustración 27.

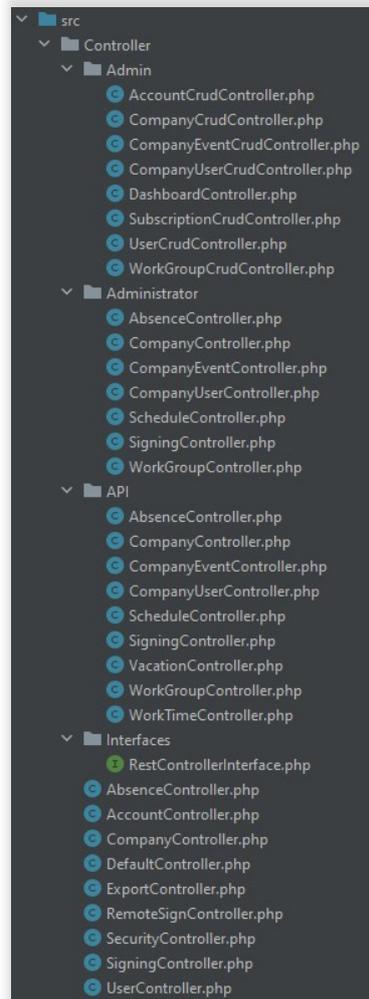


Ilustración 27. Controladores de la aplicación

- **Admin:** dentro de admin se encuentran los controladores pensados para la gestión de la plataforma, al cual solo los usuarios con rol administrador dentro de la plataforma podrán acceder.
- **Administrator:** Estos controladores están enfocados a la parte de la administración de la empresa, en el cual los usuarios con permisos de administración podrán llevar un control de la misma.
- **API:** En este directorio podemos encontrar controladores con métodos expuesto al exterior los cuales son accesibles mediante rutas fuera de la aplicación.
- **Interfaces:** En este directorio se encuentra una interfaz llamada RestControllerInterface de la cual todos los controladores del directorio API implementan, lo que permite al Kernel distinguir las peticiones que reciben estos controladores y transformarlas para adaptar las peticiones que se reciben.

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Los controladores que se encuentran en la raíz de la carpeta Controller son los que atienden las rutas básicas de la aplicación.

A continuación, en la ilustración 28 se muestra el controlador encargado de atender las rutas relacionadas con la administración de un empleado por parte de los administradores de la empresa.

```
/**
 * Class CompanyUserController
 * @package App\Controller
 * @Route("/administrator/company-user", name="administrator_company_user")
 */
class CompanyUserController extends AbstractController
{
    /** @Route("/", name="index") ...*/
    public function index(): Response{...}

    /** @Route("/new", name="new", methods={"GET","POST"}) ...*/
    public function new(Request $request, EntityManagerInterface $entityManager, SubscriptionService $subscriptionService){...}

    /** @Route("/{username}/show", name="show", methods={"GET"}, options={"expose"= true}) ...*/
    public function show(User $employee, CompanyUserRepository $companyUserRepository): Response{...}

    /** @Route("/{username}/signing", name="signing", methods={"GET"}, options={"expose"= true}) ...*/
    public function signing(User $employee, CompanyUserRepository $companyUserRepository, WorkTimeService $workTimeService): Response{...}

    /**
     * @Route("/{username}/absence", name="absence", methods={"GET"}, options={"expose"= true})
     * @param User $employee
     * @param CompanyUserRepository $companyUserRepository
     * @return Response
     */
    public function absence(User $employee, CompanyUserRepository $companyUserRepository): Response{...}

    /** @Route("/{username}/edit", name="edit", methods={"GET","POST"}, options={"expose"= true}) ...*/
    public function edit(
        User $employee,
        Request $request,
        CompanyUserRepository $companyUserRepository,
        EntityManagerInterface $entityManager
    ){...}

    /** @Route("/{username}/delete", name="delete", methods={"DELETE"}, options={"expose"= true}) ...*/
    public function delete(User $employee, Request $request, CompanyUserRepository $companyUserRepository){...}
}
```

Ilustración 28. Controlador de los usuarios de la empresa

DataFixtures

DoctrineFixturesBundle es un componente de Symfony, capaz de crear registros de prueba dentro de la base de datos, este componente está enfocado al desarrollo y al testing.

A continuación, en la ilustración 29 se muestra un ejemplo de cómo se han creado los usuarios de la empresa de pruebas:

```
class CompanyUserFixtures extends Fixture implements
DependentFixtureInterface
{
    public const COMPANY_USER_OWNER = "company-user-owner";
    public const COMPANY_USER = "company-user";

    public function load(ObjectManager $manager)
    {
        $this->loadOwner($manager);
        $this->loadEmployees($manager);

        $manager->flush();
    }

    public function loadOwner(ObjectManager $manager): void
    {
        /** @var User $owner */
        $owner = $this->getReference(UserFixtures::OWNER_REFERENCE);
        /** @var Company $company */
        $company = $this-
>getReference(CompanyFixtures::OWNER_COMPANY_REFERENCE);
        $companyUser = new CompanyUser();
        $company->addCompanyUser($companyUser);
        $owner->addCompanyUser($companyUser);
        $companyUser->setJoined(true);
        $companyUser->setJobPosition('CEO');
        $owner->setCompanyUser($companyUser);
        $manager->persist($companyUser);

        $this->addReference(self::COMPANY_USER_OWNER, $companyUser);
    }

    public function loadEmployees(ObjectManager $manager): void
    {
        /** @var Company $company */
        $company = $this-
>getReference(CompanyFixtures::OWNER_COMPANY_REFERENCE);
        for ($i = 0; $i < SubscriptionFixtures::FREE_EMPLOYEES; $i++)
        {
            $employee = $this-
>getReference(UserFixtures::EMPLOYEES_REFERENCE . " " . $i);
            $companyUser = new CompanyUser();
            $company->addCompanyUser($companyUser);
            $employee->addCompanyUser($companyUser);
            $companyUser->setJoined(true);
            $companyUser->setJobPosition('Desarrollador');
            $manager->persist($companyUser);
            $this->addReference(self::COMPANY_USER.$i, $companyUser);
        }
    }

    public function getDependencies(): array
    {
        return [
            CompanyFixtures::class,

```

Ilustración 29. Fixtures para crear a los usuarios de una empresa

Entity

Las entidades en Symfony son objetos de dominio, los cuales se persisten en la base de datos, las entidades representan una tabla en una base de datos relacional y cada instancia corresponde a una fila de esa misma tabla. Como se puede comprobar en la ilustración 30.

Los “traits” o rasgos un mecanismo de reutilización de código permitiendo una herencia simple, permitiendo reutilizar de forma fácil atributos y métodos dentro las entidades.

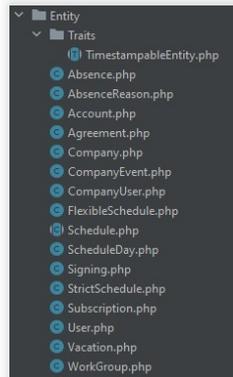


Ilustración 30. Entidades del proyecto

Dentro de cada entidad se pueden destacar varias peculiaridades asociadas a componentes de Symfony, las cuales destacan al aparecer como anotaciones, como podemos comprobar en la ilustración 31, estas son:

- **Assertios:** Las validaciones son una parte común en el desarrollo de aplicaciones web, Symfony provee de una forma de validar nuestras entidades estableciendo una serie de reglas para determinar si una entidad es capaz de persistirse en base de datos.
- **Serializer:** Permite la transformación de objetos o clases en objetos que se puedan trasladar mediante una respuesta en una API en un formato json.
- **ORM Mapping:** Anotaciones que establecen la estructura de la entidad y sus atributos a nivel de base de datos.

```
+ ORM\Entity(repositoryClass="App\Repository\UserRepository")
+ ORM\UniqueEntity(fields={"username"}, message="There is already an account with this username.")
+ ORM\UniqueEntity(fields={"email"}, message="There is already an account with this email.")
+ \Valid\Uploadable()
+
class User implements UserInterface
{
    use TimestampableEntity;

    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @Groups({"companyUser:List", "companyUser:item", "signing:admin:List", "signing:admin:item", "absence:List", "absence:item"})
     * @Assert\NotBlank()
     * @Assert\Length(
     *     min=4,
     *     max=180
     * )
     * @Assert\Regex(
     *     pattern="/^[a-z0-9]+$/i",
     *     message="Username can only contain letters and numbers."
     * )
     * @ORM\Column(type="string", length=180, unique=true)
     */
    private $username;
```

Ilustración 31. Entidad User

EntityListener

Como se ha podido ver en la parte de configuración en el archivo services.yaml se han asignado escuchadores.

A continuación, en la ilustración 32 se muestra el escuchador de la entidad usuario:

```
class UserEntityListener
{
    private AccountFactory $accountFactory;
    private UserPasswordEncoderInterface $passwordEncoder;

    public function __construct(AccountFactory $accountFactory, UserPasswordEncoderInterface $passwordEncoder)
    {
        $this->accountFactory = $accountFactory;
        $this->passwordEncoder = $passwordEncoder;
    }

    public function prePersist(User $user, LifecycleEventArgs $eventArgs): void
    {
        $this->changePassword($user);
        $this->createAccount($user);
    }

    public function preUpdate(User $user, LifecycleEventArgs $eventArgs): void
    {
        $this->changePassword($user);
        $this->createAccount($user);
    }

    private function changePassword(User $user): void
    {
        $plainPassword = $user->getPlainPassword();
        if ($plainPassword) {
            $user->setPassword(
                $this->passwordEncoder->encodePassword(
                    $user,
                    $plainPassword
                )
            );
        }
    }
}
```

Ilustración 32. Escuchador de eventos de la entidad Usuario

Events y EventSubscriber

Los eventos y los subscriptores son una herramienta para indicar que ha sucedido una acción, en caso de que este ocurra todas las clases suscritas a dicho evento ejecutarán las instrucciones que tengan programadas.

A continuación, vamos a explicar un caso de uso que se da al realizar alguna de las acciones permitidas sobre un permiso de un empleado.

Cuando algún empleado solicita un permiso, se ejecuta la función que la marca como solicitada, posteriormente se lanza el evento, de esta manera abstraemos el servicio y los subscriptores que estén escuchando procederán a realizar las acciones que sean necesarias.

Como se puede comprobar, en la ilustración 33, cuando un permiso o ausencia se establece como solicitado se lanza un evento.

```
public function request(Absence $absence): Absence
{
    $absence->setState(state: Absence::STATE_REQUESTED);
    $this->entityManager->persist($absence);
    $this->eventDispatcher->dispatch(new AbsenceRequestedEvent($absence), eventName: AbsenceRequestedEvent::NAME);
    return $absence;
}
```

Ilustración 33. Método que solicita un permiso o ausencia

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

A continuación, en la ilustración 34, se muestran las clases que escuchan los eventos relacionados con las ausencias.

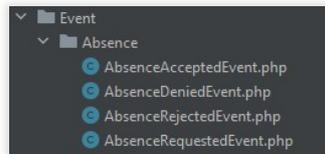


Ilustración 34. Eventos asociados a los permisos y ausencias

Como se ha visto previamente, el EventDispatcher lanza el evento, a continuación, en la ilustración 35, se muestra la clase suscrita a ese evento, la cual ejecuta el código necesario cuando se acepta la ausencia.

```
public static function getSubscribedEvents(): array
{
    return [
        AbsenceRequestedEvent::NAME => 'onAbsenceRequested',
        AbsenceAcceptedEvent::NAME => 'onAbsenceAccepted',
        AbsenceRejectedEvent::NAME => 'onAbsenceRejected',
    ];
}

public function onAbsenceRequested(AbsenceRequestedEvent $absenceRequestedEvent): void
{
    $absence = $absenceRequestedEvent->getAbsence();
    $this->vacationManagerService->absenceRequested($absence);
}
```

Ilustración 35. Clases suscrita a los eventos de los permisos y ausencias

Exception

En esta carpeta se guardan las excepciones personalizadas, considerado una buena práctica, permitiendo mejorar la detección y la captura de dichas excepciones.

Form

Una de las tareas más importantes y comunes dentro del desarrollo web se encuentra la creación de formularios, Symfony permite la creación de formularios mediante clases, las cuales se encuentran en este directorio

En este directorio también podemos encontrar DataTransformers, clases encargadas de convertir ciertos parámetros introducidos por el usuario, como se puede comprobar en la ilustración 36.

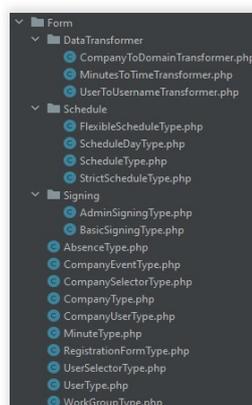


Ilustración 36. Estructura del directorio de formularios

Los formularios de Symfony integran la utilización de varios componentes, como son, la validación y la traducción, ya que, mediante determinadas funciones, se pueden validar los parámetros que se reciben en las peticiones.

Todos los formularios extienden de la clase `AbstractType`, la cual proporciona una serie de funciones, como puede ser, construir la vista del propio formulario, para posteriormente trasladarla a un archivo twig.

A continuación, en las ilustraciones 37, 38 y 39, podemos ver el proceso de creación del formulario de un usuario empleado y como se coloca en el archivo twig.

```
$form = $this->createForm( type: CompanyUserType::class, $companyUser);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid() {
```

Ilustración 37. Creación y validación de un formulario en el controlador

```
return $this->render( view: '/administrator/company_user/new', [
    'form' => $form->createView(),
]);
```

Ilustración 38. Creación de la vista y del formulario

```
class CompanyUserType extends AbstractType
{
    private LocalizationService $localizationService;

    /**
     * CompanyUserType constructor.
     * @param LocalizationService $localizationService
     */
    public function __construct(LocalizationService $localizationService)
    {
        $this->localizationService = $localizationService;
    }

    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        /** @var CompanyUser $companyUser */
        $companyUser = $options['data'];

        if (!$companyUser->getUser()) {
            $builder->add( child: 'user', type: UserSelectorType::class, [
                'help' => 'Username of the user that you want invite'
            ]);
        }

        $builder
            ->add( child: 'administrator', type: CheckboxType::class, [
                'label_attr' => ['class' => 'switch-custom'],
            ])
            ->add( child: 'jobPosition')
            ->add( child: 'workGroup', type: EntityType::class, [
                'class' => WorkGroup::class,
                'query_builder' => function (WorkGroupRepository $repository) use ($companyUser) {
                    return $repository->createQueryBuilder( alias: 'w')
                        ->where( predicates: 'w.company = :company')
                        ->orderBy( sort: 'w.name', order: 'ASC')
                        ->setParameter( key: 'company', $companyUser->getCompany());
                },
                'attr' => ['class' => 'chosen-select'],
            ])
            ->add( child: 'schedule', type: EntityType::class, [
                'class' => Schedule::class,
                'query_builder' => function (ScheduleRepository $repository) use ($companyUser) {
                    return $repository->createQueryBuilder( alias: 's')
                        ->where( predicates: 's.company = :company')
                        ->orderBy( sort: 's.name', order: 'ASC')
                        ->setParameter( key: 'company', $companyUser->getCompany());
                },
                'attr' => ['class' => 'chosen-select'],
            ])
            ->add( child: 'timeZone', type: ChoiceType::class, [
                'choices' => $this->localizationService->getTimezones(),
            ])
        ;
    }
}
```

Ilustración 39. Formulario de Usuarios de la empresa



Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Repositorios

Los repositorios son la capa que realiza la conexión con la base de datos, mediante el ORM o directamente realizando consultas a la base de datos.

Por cada entidad se ha creado un repositorio, cada uno de estos proporcionan una serie de métodos, que ofrecen la posibilidad de realizar consultas sobre la base de datos.

A continuación, en las ilustraciones 40 y 41, se muestra un ejemplo del uso de un repositorio.

```
public function getDataBetweenDates(Request $request, AbsenceRepository $absenceRepository): JsonResponse
{
    /** @var User $user */
    $user = $this->getUser();

    $params = $request->query->all();
    $result = $absenceRepository->findByCompanyUserBetweenDates($user->getCompanyUser(), $params);
    return $this->json($result, status: Response::HTTP_OK, [], ['groups' => 'absence:calendar']);
}
```

Ilustración 40. Controlador pidiendo los permisos de un empleado al repositorio

```
public function findByCompanyUserBetweenDates($companyUser, array $params = [])
{
    if (!array_key_exists( key: 'start', $params) || !array_key_exists( key: 'end', $params)) {
        return [];
    }

    return $this->createQueryBuilder( alias: 'a')
        ->innerJoin( join: 'a.companyUser', alias: 'cu')
        ->andWhere('cu = :companyUser')
        ->andWhere('a.start BETWEEN :start AND :end OR a.end BETWEEN :start AND :end')
        ->setParameter( key: 'companyUser', $companyUser)
        ->setParameter( key: 'start', $params['start'])
        ->setParameter( key: 'end', $params['end'])
        ->getQuery()
        ->getResult();
}
```

Ilustración 41. Método del repositorio encargado de obtener los permisos entre 2 fechas

Security

Dentro de esta carpeta encontramos la lógica involucrada en los accesos a las diferentes partes de la plataforma.

La plataforma cuenta con datos de las diferentes empresas, así como datos personales, por lo que este es un punto crítico del proyecto.

En esta carpeta se encuentra el sistema de autenticación, encargado de identificar al usuario una vez entra a la plataforma.

Los usuarios identificados son capaces de acceder a las empresas en las que hayan recibido acceso, después, dentro de la misma será capaz de desempeñar una serie de funciones, para controlar el acceso a dichas funciones se ha diseñado un sistema de votación encargado de ello.

El sistema de votación diseñado determina si el usuario que realiza la petición tiene acceso a dicho recurso, para ello se indica la acción y el objeto sobre el que se realiza.

A continuación, en las ilustraciones 42 y 43, se puede comprobar cómo se protege el acceso a la vista de un permiso o ausencia de un empleado:

```
$this->denyAccessUnlessGranted('attribute: AbsenceVoter::ABSENCE_VIEW', $absence);
```

Ilustración 42. Método de control en el controlador

```
class AbsenceVoter extends Voter
{
    public const ABSENCE_VIEW = 'ABSENCE_VIEW';
    public const ABSENCE_ADMIN = 'ABSENCE_ADMIN';

    protected function supports($attribute, $subject)
    {
        return in_array($attribute, [self::ABSENCE_VIEW, self::ABSENCE_ADMIN])
            && $subject instanceof Absence;
    }

    protected function voteOnAttribute($attribute, $subject, TokenInterface $token)
    {
        $user = $token->getUser();

        if (!$user instanceof UserInterface) {
            return false;
        }

        switch ($attribute) {
            case self::ABSENCE_VIEW:
                return $this->canView($subject, $user);
            case self::ABSENCE_ADMIN:
                return $this->canAdmin($subject, $user);
        }

        return false;
    }

    private function canView(Absence $absence, User $user): bool {...}

    private function canAdmin(Absence $absence, User $user): bool {...}
}
```

Ilustración 43. Voter encargado de los accesos sobre las ausencias

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Service

Esta carpeta alberga la lógica principal de la aplicación, los servicios abstraen y liberan a los controladores de la lógica de un proceso.

El uso de servicios permite mejorar la escalabilidad y reutilización del código.

Cada servicio debe desempeñar una funcionalidad concreta o englobar un conjunto de acciones relacionadas, a continuación, siguiendo el principio de responsabilidad única.

A continuación, en la ilustración 44 y 45, se muestran los servicios encargados de la lógica detrás de los fichajes y las ausencias.

```
class SigningCheckerService
{
    private CompanyUserRepository $companyUserRepository;
    private EntityManagerInterface $em;
    private EventDispatcherInterface $dispatcher;
    private UserPasswordEncoderInterface $encoder;

    public function __construct(
        CompanyUserRepository $companyUserRepository,
        EntityManagerInterface $entityManager,
        EventDispatcherInterface $dispatcher,
        UserPasswordEncoderInterface $encoder
    ){...}

    public function check(User $user): ?Signing{...}

    public function remoteCheck(Company $company, string $username, string $password): Signing{...}

    private function sign(CompanyUser $companyUser): Signing{...}

    private function computeCheckType($companyUser): string{...}
}
```

Ilustración 44. Servicio encargado de los fichajes de los usuarios

```
class AbsenceManagerService
{
    public const TRANSITION_ACCEPT = 'accept';
    public const TRANSITION_REJECT = 'reject';
    public const TRANSITION_DENY = 'deny';

    private EntityManagerInterface $entityManager;
    private WorkflowInterface $absenceStateMachine;
    private EventDispatcherInterface $eventDispatcher;

    public function __construct(
        EntityManagerInterface $entityManager,
        WorkflowInterface $absenceStateMachine,
        EventDispatcherInterface $eventDispatcher
    ){...}

    public function request(Absence $absence): Absence{...}

    public function accept(Absence $absence): Absence{...}

    public function reject(Absence $absence): Absence{...}

    public function deny(Absence $absence): Absence{...}

    public function delete(Absence $absence): void{...}

    private function checkAcceptable(Absence $absence): void{...}

    private function checkRejectable(Absence $absence): void{...}

    private function checkDeniable(Absence $absence): void{...}

    private function checkDeletable(Absence $absence): void{...}
}
```

Ilustración 45. Servicio administrador de ausencias

5.1.7 Translations

En esta carpeta se guardan todos los mensajes traducidos al castellano, que posteriormente serán utilizados por el componente Symfony translator.

Los archivos de traducción tienen el formato xml, en los cuales se establece una clave con su consiguiente mensaje asignado una traducción por cada traducción realizada, a continuación, en la ilustración 46, se pueden comprobar algunos ejemplos de cómo se lleva esta traducción.

```
<trans-unit id="Accept terms and conditions is necessary.">
  <source>Accept terms and conditions is necessary.</source>
  <target>Es necesario aceptar los términos y condiciones.</target>
</trans-unit>
<trans-unit id="There is already an account with this username.">
  <source>There is already an account with this username.</source>
  <target>El nombre de usuario ya esta en uso.</target>
</trans-unit>
<trans-unit id="There is already an account with this email.">
  <source>There is already an account with this email.</source>
  <target>El email ya esta en uso.</target>
</trans-unit>
<trans-unit id="The company with the selected domain does not exist.">
  <source>The company with the selected domain does not exist.</source>
  <target>La Empresa con el dominio seleccionado no existe.</target>
</trans-unit>
```

Ilustración 46. Traducciones de las validaciones

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

5.2 Aplicación Web

En este punto se va a mostrar el resultado de la solución propuesta, así como sus diferentes vistas, las cuales nos brindan las funcionalidades requeridas.

5.2.1 Página de presentación

Con esta vista se pretende aportar al usuario un resumen rápido de las soluciones ofrecidas, así como de sus principales características.

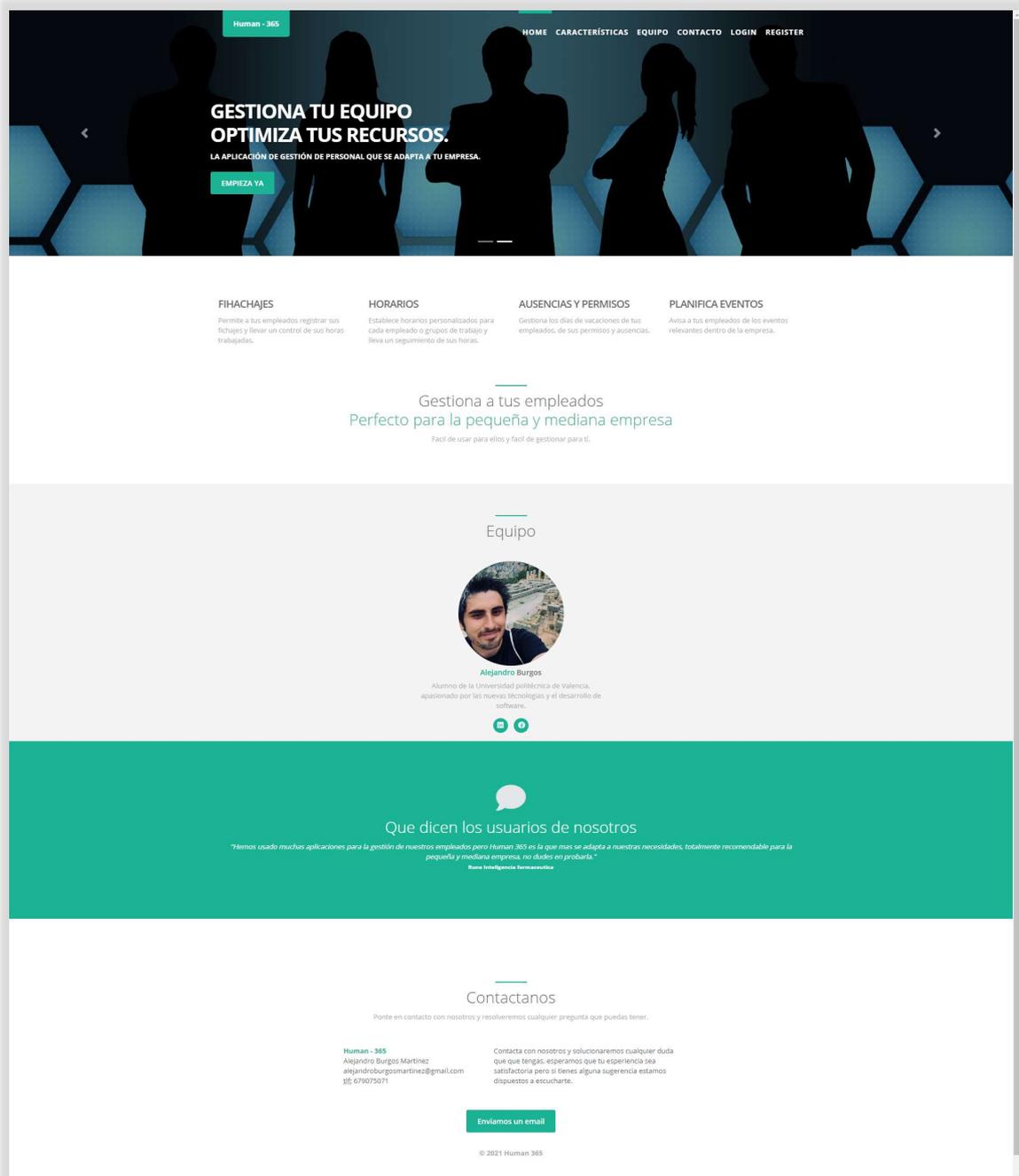


Ilustración 47. Página de la presentación

5.2.2 Registro

Vista encargada del registro de un usuario en la plataforma, para ello debe de rellenar los datos requeridos por el formulario.

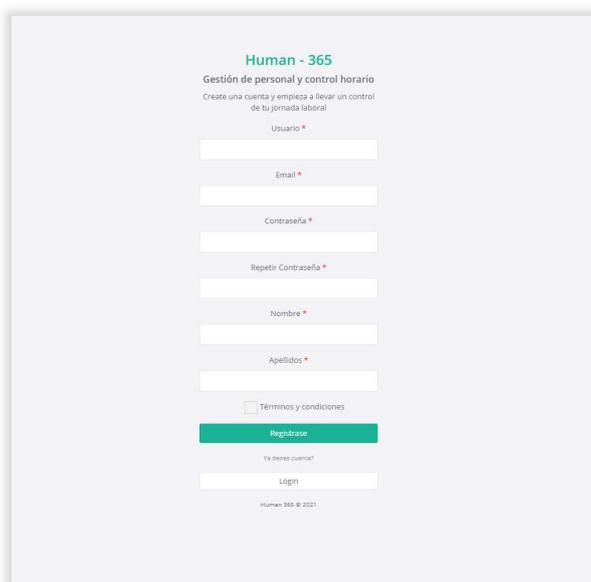


Ilustración 48. Página de registro

5.2.3 Login

En esta vista el usuario se podrá identificar en la plataforma.

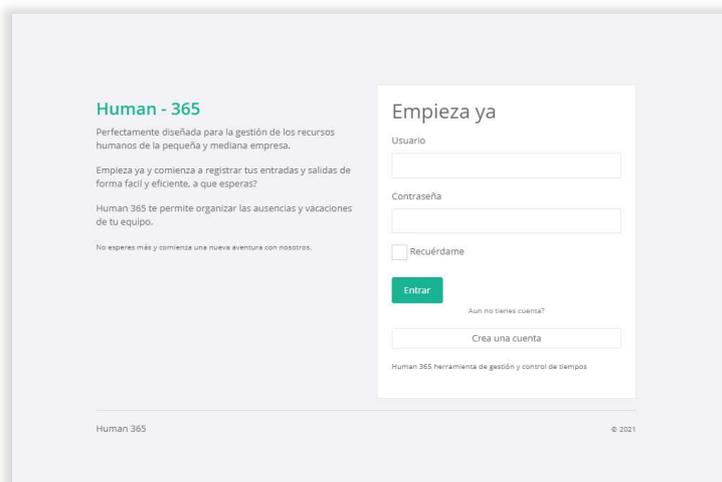


Ilustración 49. Página de Login

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

5.2.4 Pantalla de Inicio en la plataforma

Vista de inicio para los usuarios que se han identificado en la plataforma.

Para los usuarios que aún no se encuentran dentro de ninguna empresa dispondrán de 2 opciones:

- **Unirse a una empresa:** introduciendo el dominio de una empresa, en el caso de que haya sido dado de alta en la misma podrá acceder, siendo este redirigido al panel de inicio del empleado, de lo contrario se le informara que no tiene acceso.
- **Registrar una empresa:** Se da la posibilidad al usuario de registrar su propia empresa, en caso de elegir esta opción se le llevara a la vista de registro de la empresa.

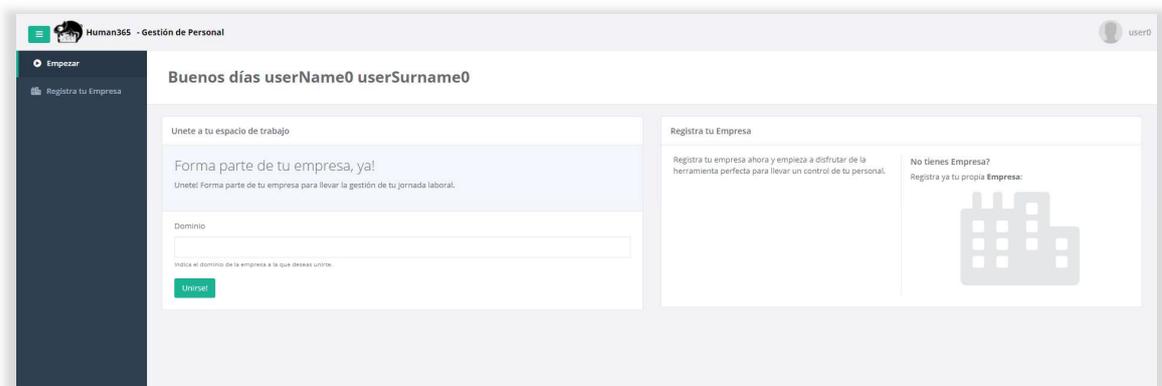


Ilustración 50. Página de inicio

5.2.5 Registro de empresa

En esta vista el usuario que lo desee podrá dar de alta su propia empresa y comenzar a administrarla, para ello debe de completar los datos del formulario.

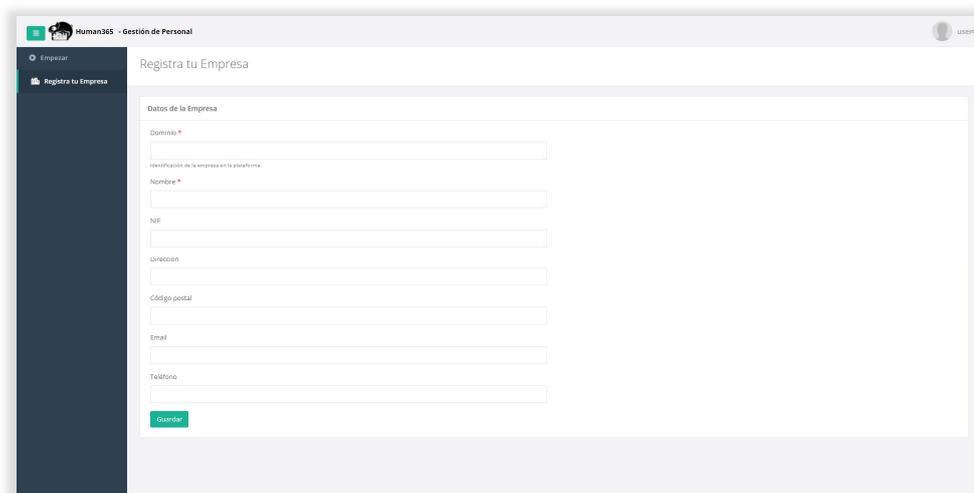


Ilustración 51. Página de Registro de Empresas

5.2.6 Perfil del usuario

En esta vista el usuario podrá ver los datos con los que se ha identificado, ofreciéndole la posibilidad de editar su perfil, así como administrar sus áreas de trabajo, otorgándole la opción de unirse o salirse de las mismas.

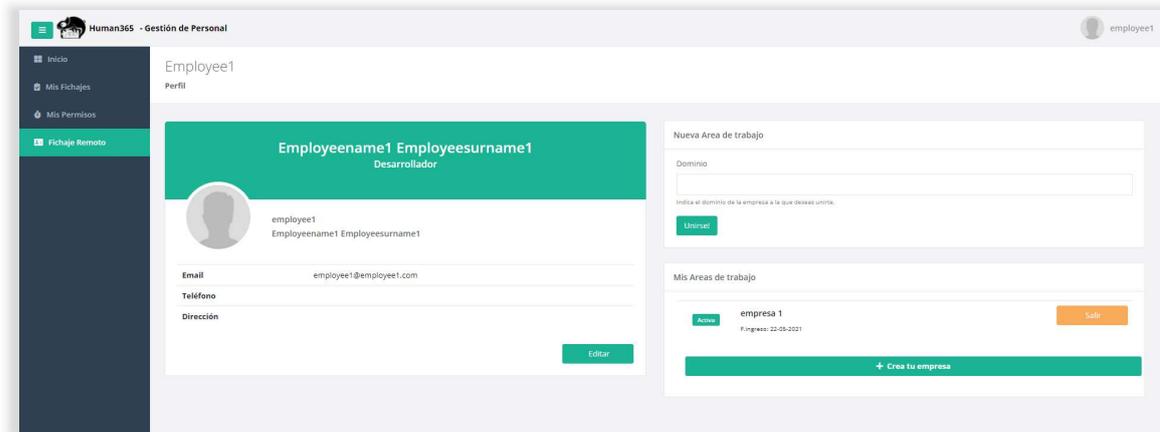


Ilustración 52. Página de perfil de usuario

5.2.7 Panel de Inicio del empleado

Ventana principal de un empleado, esta se mostrará cuando el usuario se identifique y se encuentre unido a una empresa.

En esta vista se le dará la posibilidad de realizar fichajes de forma fácil, así como varios resúmenes de su historial de registros, su horario y los próximos eventos en los que esté involucrado.

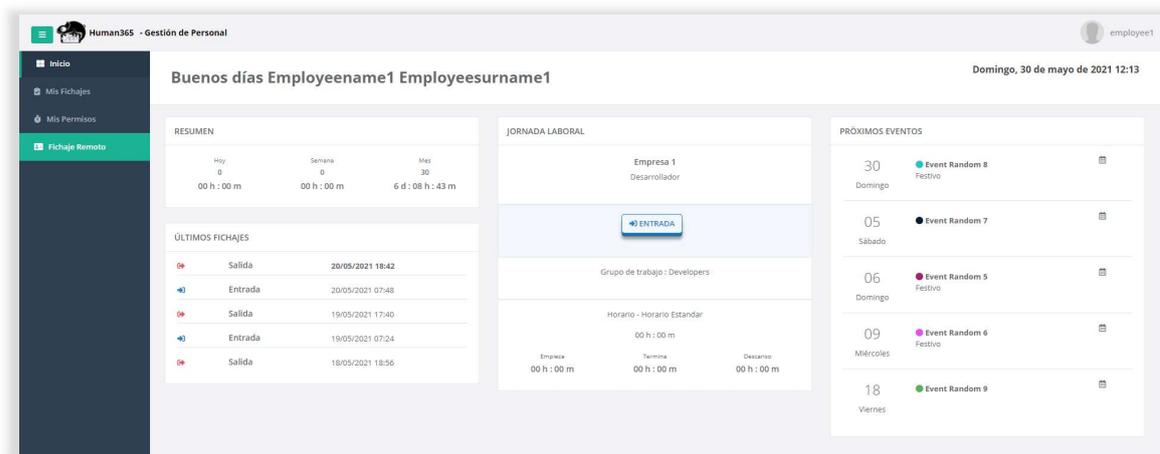


Ilustración 53. Página de Panel de inicio

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

5.2.8 Mis Fichajes

En esta vista el usuario de la empresa tiene la opción de consultar los fichajes que ha realizado, permitiendo acceder a todo su historial de registros de entradas y salidas.

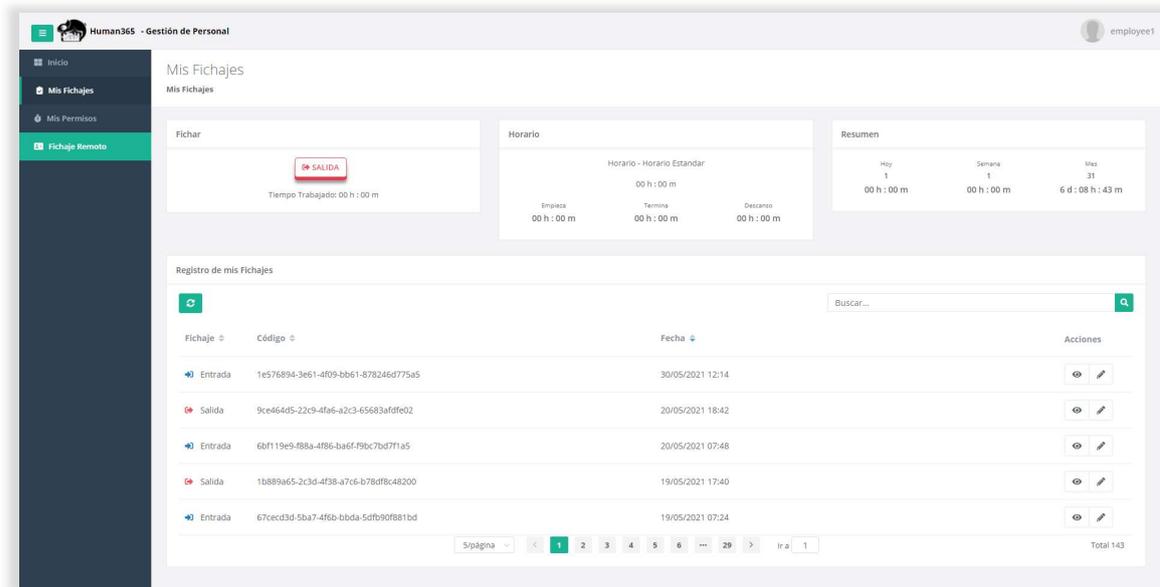


Ilustración 54. Página Mis fichajes

5.2.9 Mis Permisos

En esta ventana el empleado podrá llevar un seguimiento de sus permisos y ausencias, así como la posibilidad de solicitarlos,

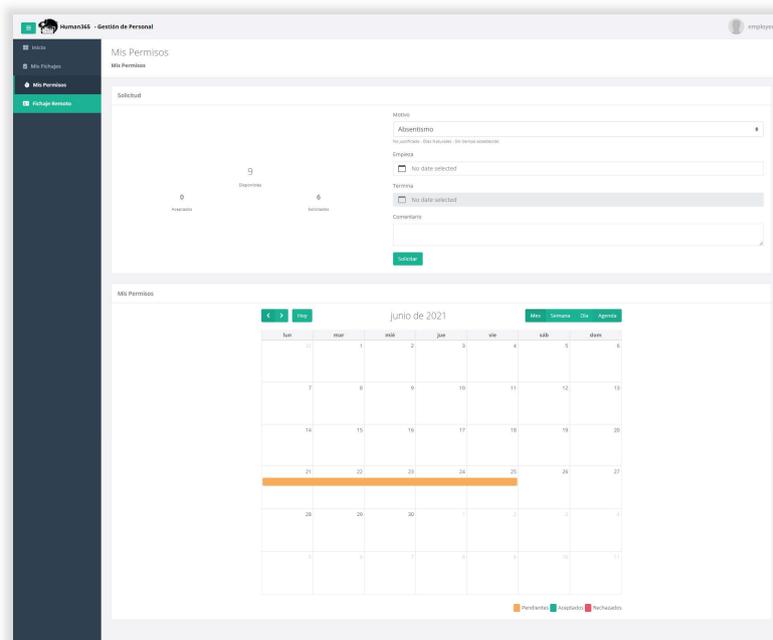


Ilustración 55. Página Mis permisos

5.2.10 Fichaje Remoto

La vista de fichaje remoto sirve para poder realizar fichajes sin la necesidad de iniciar sesión.

Vista que proporciona una forma de fichaje para cada una de las empresas, para acceder a esta vista no es necesario estar identificado a la hora de realizar el fichaje y será accesible de forma pública.

The screenshot shows a login form for 'empresa1'. At the top, it says 'Realiza de forma rápida tu fichaje gracias a Human365'. Below that, it asks for 'Ficha tu entrada y salida'. There are two input fields: 'Usuario *' and 'Contraseña *'. A green button labeled 'Realizar Fichaje' is at the bottom. At the very bottom, it says 'Human365 plataforma para la gestión de personal'.

Ilustración 56. Página de Fichaje remoto

5.2.11 Panel de control de Administración

Los usuarios administradores dentro de una empresa tendrán acceso a esta vista, la cual ofrece un resumen del estado actual y un cuadro de mandos con las principales funciones.

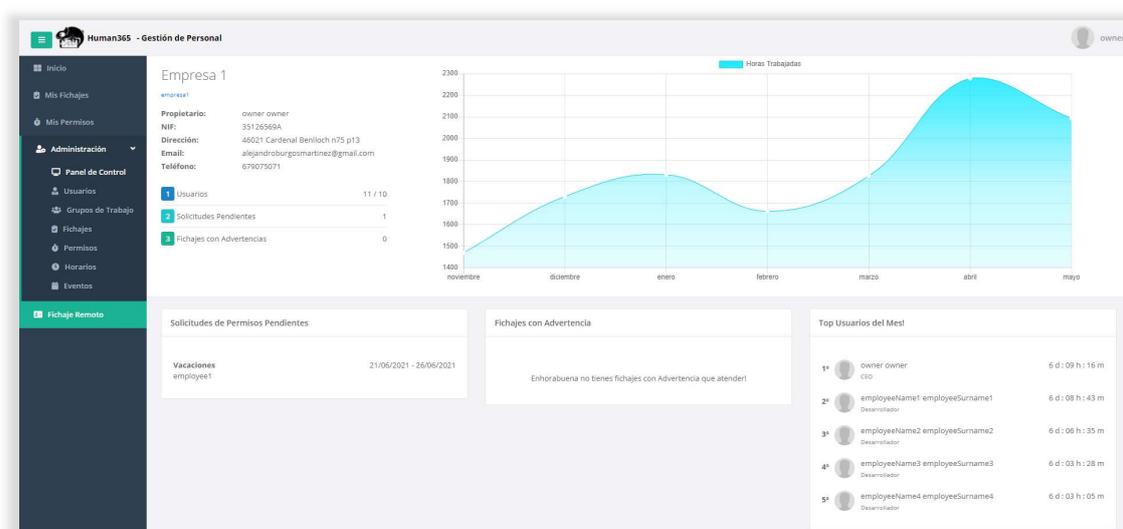


Ilustración 57. Página de Panel de control

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

5.2.12 Administración de empleados

Los usuarios administradores podrán consultar los datos de cualquier empleado, para ello se les ofrece una forma fácil de poder localizarlos y realizar acciones sobre los mismos.

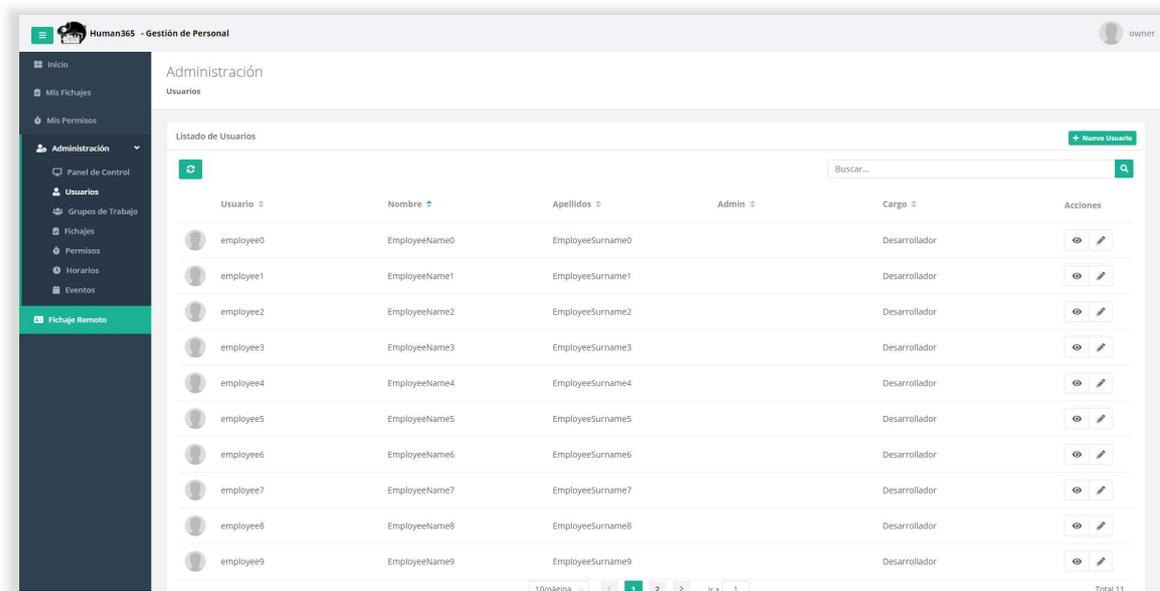


Ilustración 58. Página de Administración de empleados

5.2.13 Ficha del empleado

La ficha del empleado costa de tres partes principales:

Datos personales y los datos relacionados con el puesto de trabajo o la función que desempeña en la empresa.

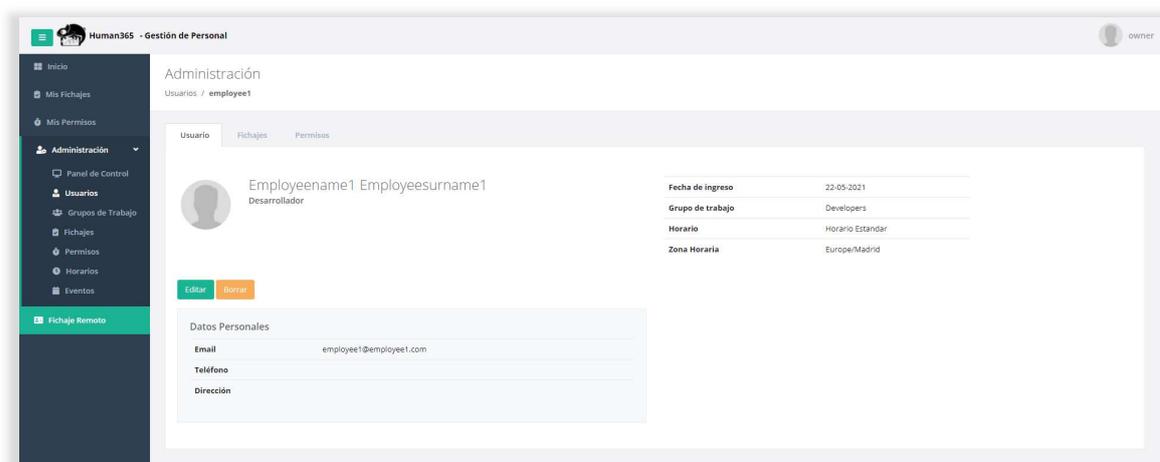


Ilustración 59. Página Ficha del empleado

Registro de fichajes realizados por el usuario y la opción de exportarlos a un formato excel.

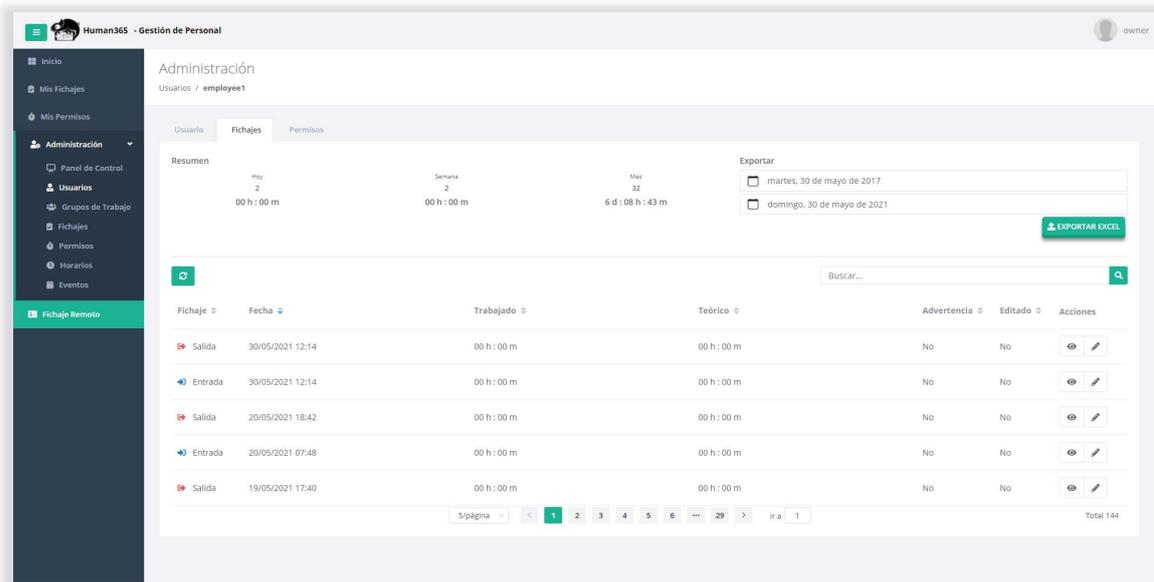


Ilustración 60. Página Fichajes del empleado

Gestión de permisos y ausencias solicitados por el empleado.

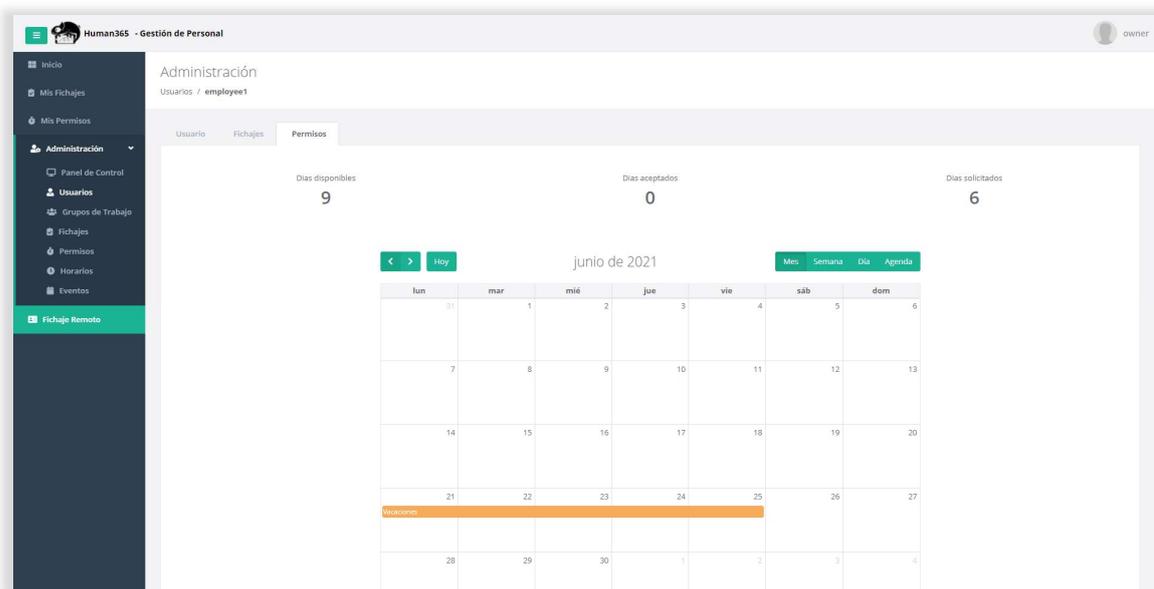


Ilustración 61. Página Permisos del empleado

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

5.2.14 Administración de Grupos de trabajo

Los grupos de trabajo tienen la función de establecer una asociación entre varios empleados facilitando su gestión.

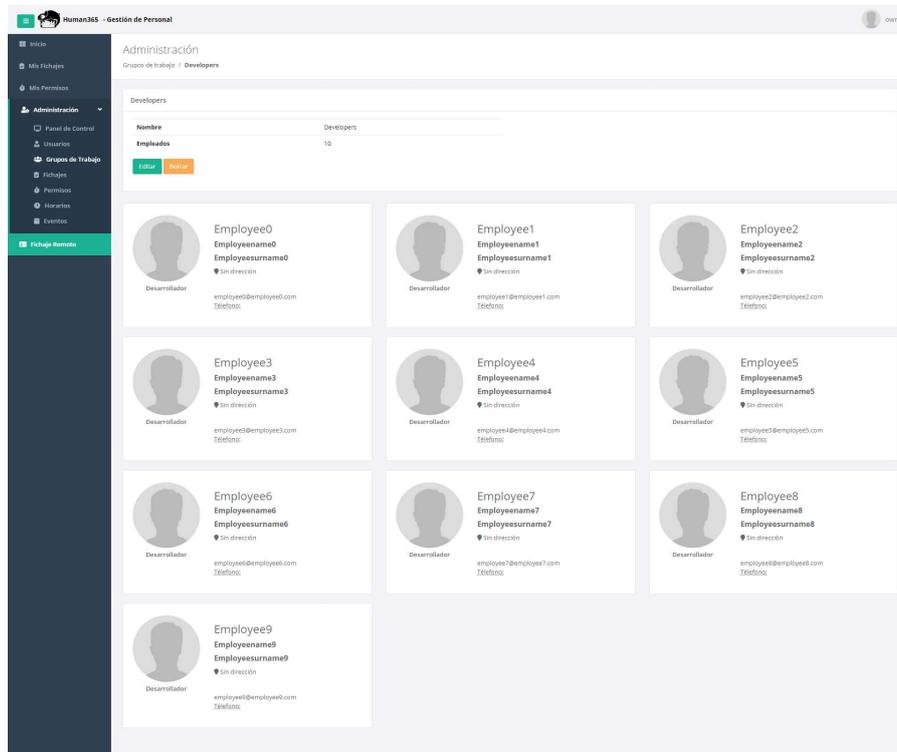


Ilustración 62. Página de Grupo de trabajo

5.2.15 Administración de Fichajes

Los usuarios encargados de la administración tienen acceso al registro completo de los fichajes realizados mediante esta vista.

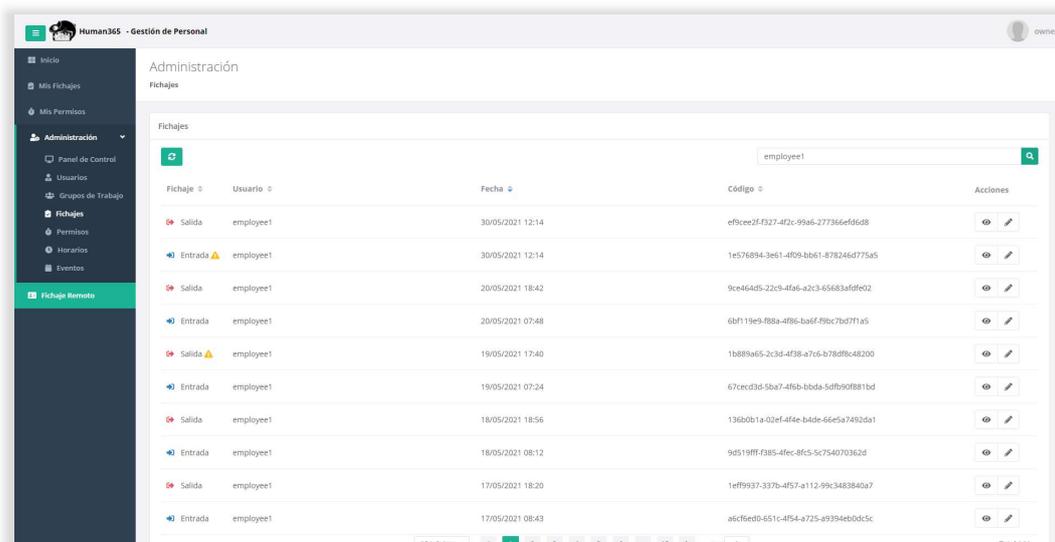


Ilustración 63. Página Administración de fichajes

5.2.16 Administración de Permisos

Los usuarios administradores de una empresa son los encargados de aceptar o rechazar los permisos realizados por los distintos empleados.

En esta vista se provee de un resumen detallado del permiso, así como, las acciones disponibles para el mismo.

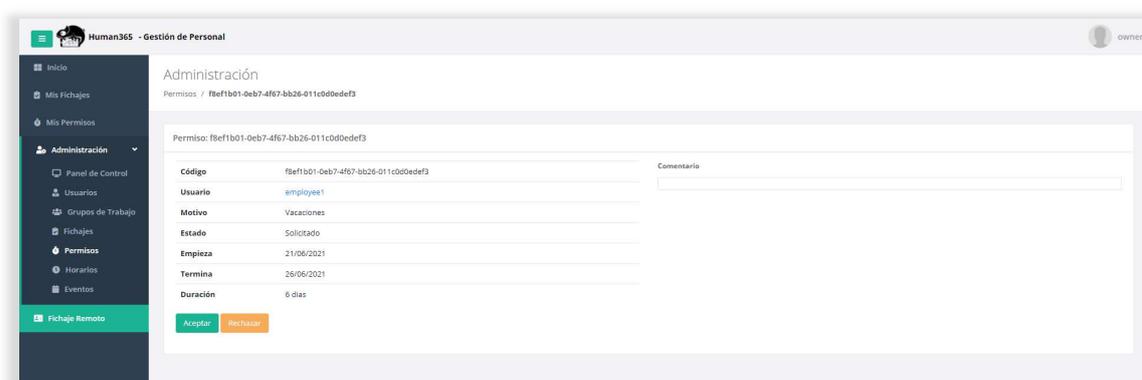


Ilustración 64. Página administración de Permisos y ausencias

5.2.17 Administración de horarios

La gestión de los horarios es otra de las funciones principales de la herramienta.

En esta vista se establecerán las opciones necesarias para la creación y asignación de los horarios.

Actualmente se pueden establecer dos tipos de horarios:

- **Flexible:** se establece un total de horas semanal, sin importar en qué momento se deben de realizar los fichajes.
- **Estricto:** En este horario se establece un momento de entrada y de salida, así como un tiempo de descanso máximo.

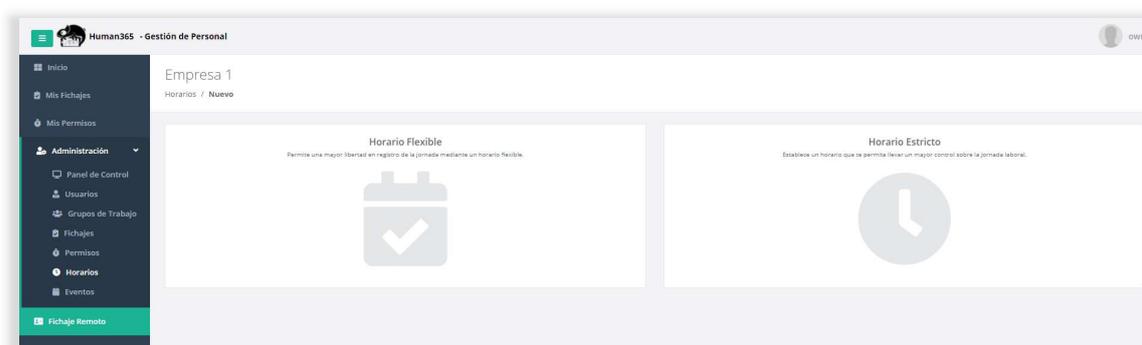


Ilustración 65. Página de Creación de Horarios

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

Human365 - Gestión de Personal

Empresa 1
Horarios / Nuevo

Nuevo Horario

Nombre *

Zona horaria #
Europe/Madrid

Empleados ...

Días de trabajo

#	Empieza	Termina	Descanso
Lunes	Hora : Minutos	Hora : Minutos	Hora : Minutos
Martes	Hora : Minutos	Hora : Minutos	Hora : Minutos
Miércoles	Hora : Minutos	Hora : Minutos	Hora : Minutos
Jueves	Hora : Minutos	Hora : Minutos	Hora : Minutos
Viernes	Hora : Minutos	Hora : Minutos	Hora : Minutos
Sábado	Hora : Minutos	Hora : Minutos	Hora : Minutos
Domingo	Hora : Minutos	Hora : Minutos	Hora : Minutos

Guardar

Ilustración 66. Página de creación de un Horario Estricto

5.2.18 Administración de eventos de la empresa

En esta vista se pueden visualizar los eventos de la empresa, así como ofrecer la posibilidad de creación de nuevos eventos.

Human365 - Gestión de Personal

Administración
Eventos

Eventos de la Empresa

mayo de 2021

MAY - Semana Día Agenda

lun	mar	mié	jue	vie	sáb	dom
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

11:42 Event Random

15:52 Event Random

10:47 Event Random

12:04 Event Random

12 Event Random 2

16:17 Event Random

12:27 Event Random

13:13 Event Random

Nuevo Evento

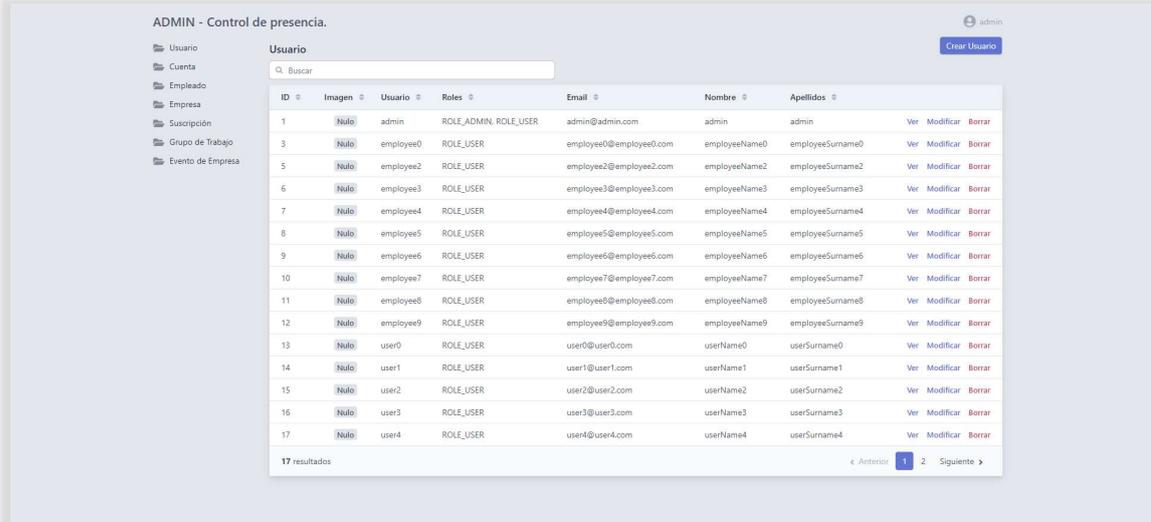
Ilustración 67. Página de Administración de eventos de la empresa

5.2.19 Administración de la plataforma

Con el fin de poder controlar y dar soporte a la plataforma se ha creado un sistema de administración de la misma.

Los usuarios con el rol administrador son capaces de llevar un control sobre las diferentes entidades.

En esta vista se muestran los listados de los registros de la base de datos, además de la posibilidad de realizar las acciones de creación, borrado y actualización.



ADMIN - Control de presencia.

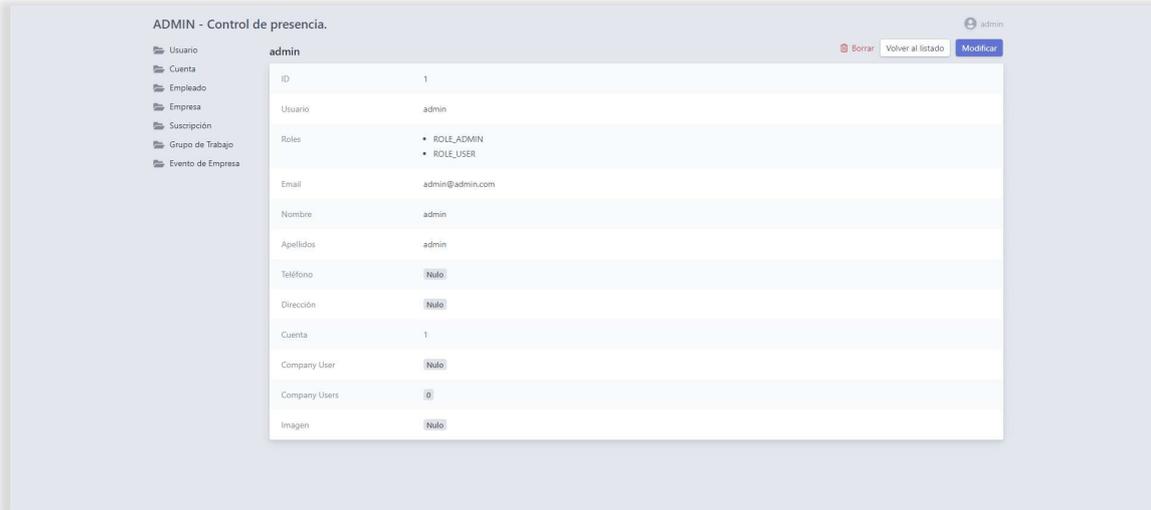
Usuario

Crear Usuario

ID	Imagen	Usuario	Roles	Email	Nombre	Apellidos	
1	Nulo	admin	ROLE_ADMIN, ROLE_USER	admin@admin.com	admin	admin	Ver Modificar Borrar
3	Nulo	employee0	ROLE_USER	employee0@employee0.com	employeeName0	employeeSurname0	Ver Modificar Borrar
5	Nulo	employee2	ROLE_USER	employee2@employee2.com	employeeName2	employeeSurname2	Ver Modificar Borrar
6	Nulo	employee3	ROLE_USER	employee3@employee3.com	employeeName3	employeeSurname3	Ver Modificar Borrar
7	Nulo	employee4	ROLE_USER	employee4@employee4.com	employeeName4	employeeSurname4	Ver Modificar Borrar
8	Nulo	employee5	ROLE_USER	employee5@employee5.com	employeeName5	employeeSurname5	Ver Modificar Borrar
9	Nulo	employee6	ROLE_USER	employee6@employee6.com	employeeName6	employeeSurname6	Ver Modificar Borrar
10	Nulo	employee7	ROLE_USER	employee7@employee7.com	employeeName7	employeeSurname7	Ver Modificar Borrar
11	Nulo	employee8	ROLE_USER	employee8@employee8.com	employeeName8	employeeSurname8	Ver Modificar Borrar
12	Nulo	employee9	ROLE_USER	employee9@employee9.com	employeeName9	employeeSurname9	Ver Modificar Borrar
13	Nulo	user0	ROLE_USER	user0@user0.com	userName0	userSurname0	Ver Modificar Borrar
14	Nulo	user1	ROLE_USER	user1@user1.com	userName1	userSurname1	Ver Modificar Borrar
15	Nulo	user2	ROLE_USER	user2@user2.com	userName2	userSurname2	Ver Modificar Borrar
16	Nulo	user3	ROLE_USER	user3@user3.com	userName3	userSurname3	Ver Modificar Borrar
17	Nulo	user4	ROLE_USER	user4@user4.com	userName4	userSurname4	Ver Modificar Borrar

17 resultados

Ilustración 68. Página de Administración de la plataforma



ADMIN - Control de presencia.

admin

Borrar Volver al listado Modificar

ID	1
Usuario	admin
Roles	<ul style="list-style-type: none">ROLE_ADMINROLE_USER
Email	admin@admin.com
Nombre	admin
Apellidos	admin
Teléfono	Nulo
Dirección	Nulo
Cuenta	1
Company User	Nulo
Company Users	0
Imagen	Nulo

Ilustración 69. Página de Administración de un Usuario de la plataforma

6. Implantación

En este apartado se va a explicar la puesta en marcha de la plataforma y los pasos necesarios para conseguirlo.

Para albergar un proyecto Symfony es necesario tener un servidor, en el cual se necesita tener instalado un servidor web como puede ser Apache¹⁰ o Nginx¹¹ que permita una conexión HTTP.

Mediante los servicios de AWS¹², se ha creado un servidor privado virtual con Ubuntu como sistema operativo, en el cual se ha instalado Nginx como servidor http y se ha configurado una instancia de MySQL.

Las aplicaciones que se han de instalado en el sistema operativo para hacer funcionar la aplicación han sido las siguientes: Php7.4, nodejs, Composer2, Yarn

El despliegue se ha realizado con la herramienta de deployer¹³, la cual permite mediante el archivo de configuración deploy.php (ilustración 70) establecer los pasos y la configuración necesaria para poder desplegar la aplicación web.

```
require 'recipes/symfony.php';

// Project name
set(:name, :application, :value, 'presence_control_app');

// Project repository
set(:name, :repository, :value, 'git@gitlab.com:presence_control/presence_control_app.git');

// [Optional] Allocate tty for git clone. Default value is false.
set(:name, :git_tty, :value, true);

// Shared files/dirs between deploys
add(:name, :shared_files, [:env, '.env.local']);
add(:name, :shared_dirs, [:var/sessions, 'var/log', 'public/upload']);

// Writable dirs by web server
add(:name, :writable_dirs, ['var/sessions', 'var/log', 'public/upload']);
set(:name, :allow_anonymous_stats, :value, false);

// Hosts

host(['prod'])

  >hostname(hostname, 'human665.com')
  >set(:deploy_path, '/var/www/{{application}}')
  >set(:branch, function () {
    return input().getOption(:name, 'branch') ?: 'master';
  });

// Tasks

task(:name, 'deploy:js-routing:dump', function () {
  cd(:dir, '{{release_path}}');
  run(:command, 'bin/console fos:js-routing:dump --env=prod --format=json --target=public/js/fos_js_routes.json');
});

task(:name, 'deploy:front:install', function () {
  cd(:dir, '{{release_path}}');
  run(:command, 'yarn install');
});

task(:name, 'deploy:front:generation', function () {
  cd(:dir, '{{release_path}}');
  run(:command, 'yarn run encore production');
});

task(:name, 'build', function () {
  run(:command, 'cd {{release_path}} && build');
});

// [Optional] if deploy fails automatically unlock.
after(:@, 'deploy:failed', @, 'deploy:unlock');

// Migrate database before symlink new release.
before(@, 'deploy:symlink', @, 'database:migrate');
```

Ilustración 70. Archivo deploy.php

¹⁰ <https://httpd.apache.org/>

¹¹ <https://www.nginx.com/>

¹² <https://aws.amazon.com/>

¹³ <https://deployer.org/>

Implantación en Empresas

El primer paso con la finalidad de implantar la herramienta en el mercado, se ha proporcionado acceso a un reducido número de empresas:

- Rune Inteligencia farmacéutica.
- Patrin Tech Solutions
- Cherubino Valsangiacomo

Actualmente se está realizando un proceso de validación, toma de requisitos y la implantación de las mejoras recopiladas por parte de un pequeño de usuarios piloto.

Mediante este proceso se espera adaptar la herramienta a las demandas de las empresas mencionadas anteriormente, con el fin de poder realizar una implantación del sistema con el resto de sus empleados.

Una aceptación por parte de las empresas, permitirá destinar más recursos a la plataforma y demostraría la viabilidad de la aplicación.

7. Pruebas

Las pruebas en la ingeniería del software consisten en realizar comprobaciones preliminares con el objetivo de encontrar fallas lo suficientemente graves para rechazar una versión del software.

PHP provee de una serie de herramientas capaces de facilitarnos esta tarea, como son, PHPUnit¹⁴ y PHPStan¹⁵.

Durante el desarrollo del proyecto, se ha establecido un proceso de detección de fallos mediante el uso de Gitlab y su herramienta de integración continua y con la herramienta mencionadas anteriormente.

7.1 PHPUnit

PHPUnit es el entorno que se ha elegido para realizar pruebas unitarias, con la idea de poder detectar errores y garantizar el comportamiento de los diferentes módulos dentro de la aplicación.

Symfony provee de una integración con este framework diseñado para poder ejecutar tests de manera sencilla, aportando una serie de reglas y validaciones.

Para completar a Phpunit se ha utilizado la librería de panther¹⁶, también utilizada para realizar scraping de páginas web, permitiendo simular el comportamiento humano y poder realizar operaciones sobre un navegador, con estas dos librerías se ha podido desarrollar una serie de tests funcionales.

A continuación, en la ilustración 71 podemos ver el test realizado para el registro de un usuario dentro de la plataforma:

```
public function testRegister()
{
    $client = self::createClient();

    $crawler = $client->request('method' => 'GET', 'uri' => '/register');

    self::assertResponseIsSuccessful();

    $username = StringUtil::randomString( length: 5);
    $email = $username . '@' . StringUtil::randomString( length: 3) . '.com';
    $password = StringUtil::randomString( length: 6);

    $formButton = $crawler->selectButton( value: 'Register');
    $form = $formButton->form();
    $formName = $form->getName();

    $client->submit($form, [
        $formName.[username] => $username,
        $formName.[email] => $email,
        $formName.[plainPassword][first] => $password,
        $formName.[plainPassword][second] => $password,
        $formName.[name] => StringUtil::randomString( length: 6),
        $formName.[surname] => StringUtil::randomString( length: 6),
        $formName.[agreeTerms] => '1',
    ]);

    $client->followRedirect();
    self::assertRouteSame( expectedRoute: 'app_login');

    $container = $client->getContainer();
    /** @var EntityManagerInterface $em */
    $em = $container->get( id: 'doctrine' )->getManager();
    $user = $em->getRepository( className: User::class )->findOneBy( ['username' => $username]);
    self::assertInstanceOf( expected: User::class, $user);
}
```

Ilustración 71. Test de Registro

¹⁴ <https://phpunit.de/>

¹⁵ <https://phpstan.org/>

¹⁶ <https://github.com/symfony/panther>

7.2 GitLab CI/CD

El control y la gestión del código es una parte fundamental en el desarrollo de software, ya que constantemente se producen modificaciones y actualizaciones sobre este, por lo que la integración continua es una práctica con la cual los desarrolladores son capaces de gestionar todos estos cambios.

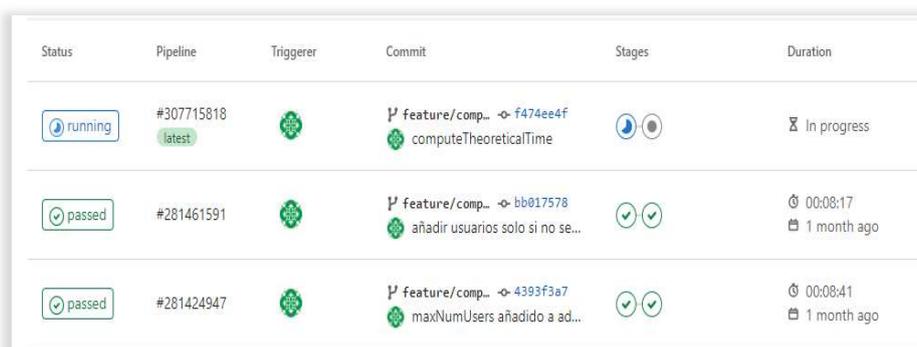
La integración continua se centra sobre todo en la fase de desarrollo y mantenimiento del proyecto en el cual mediante pruebas automatizadas mejoramos el proceso de publicación e integración de cambios.

La necesidad principal del uso de esta práctica surge cuando se empieza a trabajar en equipo, anteriormente era habitual que los desarrolladores trabajasen de manera aislada, pero con el tiempo, con el aumento de los proyectos y la necesidad de mantenerlos a largo plazo, el uso de esta práctica se vuelve fundamental para mejorar el rendimiento, la detección y prevención de errores.

La integración continua consiste en establecer un conjunto de reglas a la hora de realizar cambios sobre el código, mediante un repositorio compartido central y un sistema que se encargó de ejecutar una serie de verificaciones y acciones cuando se realiza alguna modificación sobre el código, las cuales permiten controlar y mantener el código libre de errores antes de la puesta en producción.

Como parte fundamental de la integración continúa llevada a cabo en este proyecto podemos constatar que se encuentra la plataforma de Gitlab y su herramienta de CI/CD.

GitLab nos permite guardar nuestro proyecto en su repositorio de forma totalmente gratuita, así como ejecutar de forma automática una serie de pasos automatizados cada vez que se realiza una subida o actualización de código sobre el repositorio como podemos ver en la ilustración 72.



Status	Pipeline	Triggerer	Commit	Stages	Duration
running	#307715818 latest		feature/comp... -> f474ee4f computeTheoreticalTime		In progress
passed	#281461591		feature/comp... -> bb017578 añadir usuarios solo si no se...		00:08:17 1 month ago
passed	#281424947		feature/comp... -> 4393f3a7 maxNumUsers añadido a ad...		00:08:41 1 month ago

Ilustración 72. Listado de pasos automatizados al introducir cambios en el código

Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas

La automatización y configuración de estos pasos se realiza mediante un archivo de configuración dentro del proyecto, en la ilustración 73 podemos comprobar el resultado de las ejecuciones de las validaciones necesarias para permitir la integración de nuevas funcionalidades en nuestro proyecto.

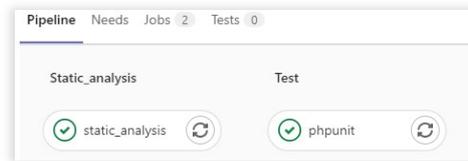


Ilustración 73. Validaciones realizadas en la integración continua

Los pasos necesarios para que la ejecución del pipeline sea satisfactoria debe de ser la siguiente:

- **Static_analysis:** este paso consiste en la ejecución de la librería phpstan, la cual realiza un análisis estableciendo un nivel de requisitos mínimos que aseguren la calidad de nuestro código, detectando posibles fallos.
- **Phpunit:** en este paso se ejecutan los tests realizados en el proyecto, los cuales se han diseñado para garantizar que las funcionalidades principales de la plataforma funcionan de forma correcta.

El archivo de configuración “.gitlab-ci.yml” del proyecto contiene los pasos necesarios para realizar el proceso de integración continua, mediante una máquina virtual se realizara el despliegue y validación de la plataforma.

A continuación, en la ilustración 74 se incluye el archivo de configuración gitlab-ci.yml.

```
default:
  image: php:7.4-fpm
  services:
    - name: mysql:8.0
      alias: mysql

cache:
  key: ${CI_COMMIT_REF_SLUG}
  paths:
    - vendor/
    - node_modules/
    - .yarn

variables:
  MYSQL_ROOT_PASSWORD: pass_test
  MYSQL_DATABASE: myapptest
  MYSQL_USER: myapptest
  MYSQL_PASSWORD: myapptest
  DATABASE_URL: 'mysql://myapptest:myapptest@mysql:3306/myapptest'

before_script:
  - sh scripts/ci-php-install.sh
  - chmod 755 bin/composer.phar
  - mv bin/composer.phar /usr/local/bin/composer
  - composer install -o
  - apt-get install nodejs -y
  - curl -o- -L https://yarnpkg.com/install.sh | bash
  - export PATH="$HOME/.yarn/bin:$HOME/.config/yarn/global/node_modules/.bin:$PATH"
  - yarn config set cache-folder .yarn
  - yarn install --frozen-lockfile
  - yarn build
  - php bin/console fos:js-routing:dump --format=json --target=public/js/fos_js_routes.json

static_analysis:
  stage: static_analysis
  script:
    - php -d memory_limit=512M bin/phpstan.phar analyse src --level=1

phpunit:
  stage: test
  script:
    - sh scripts/php_test_run.sh

stages:
  - static_analysis
  - test
```

Ilustración 74. Archivo de configuración gitlab-ci.yml

8. Conclusiones

En este último apartado se presenta una evaluación a nivel global del proyecto realizado.

El proyecto a nivel personal considero que ha sido de gran ayuda a la hora de documentarme y adquirir conocimientos en el desarrollo de aplicaciones web, así como el mundo de la empresa y en la búsqueda de proporcionar soluciones para la misma.

En cuanto a objetivos planteados podemos concluir que se han realizado la mayoría e incluso se ha avanzado un poco más en la parte del desarrollo y en proporcionar una herramienta funcional para múltiples usuarios.

He podido realizar el desarrollo de una aplicación web completa, con una tecnología presente en el mundo laboral, se han abordado todos los aspectos del desarrollo principales de los cuales consta una plataforma profesional, desde el diseño de interfaces, la lógica de la aplicación y el manejo de bases datos.

Por otro lado, el realizar una aplicación de forma individual me ha llevado a explorar todas las áreas del desarrollo y he intentado enfocarlo en todo momento desde un punto de vista que permitiese incrementar el número de desarrolladores utilizando una metodología de trabajo e integración continua desde el principio, así como una serie de estándares relacionados con los frameworks utilizados.

También me he encontrado con problemas que he sido capaz de abordar, el hecho de tratar grandes cantidades de datos y proporcionar un rendimiento me ha obligado a buscar un método por el cual he necesitado optimizar las consultas a la base de datos, lo que me lleva a estar contento con el resultado realizado.

Otra problemática encontrada ha sido la implantación, la configuración del servidor y la puesta en marcha de la plataforma; esta parte ha supuesto todo un reto debido a mis conocimientos básicos sobre los servidores, sin embargo, como lección aprendida podemos destacar la documentación del proceso, lo que me proporcionara una ayuda de cara a futuros desarrollos.

Sin embargo, la mayor dificultad con la que me he encontrado ha sido el compaginar el trabajo laboral con el TFG, en primera instancia no pensaba que esto pudiese suponer un problema, pero ha sido un esfuerzo extra en el día a día, retrasando en gran medida la finalización del presente TFG, incluso en cierto momento me llevo a la suspensión temporal del mismo.

Por último y a nivel personal estoy gratamente satisfecho con el trabajo realizado, se ha podido desarrollar una primera versión funcional de la plataforma, ha sido todo un reto poder crearla desde cero, intentar cumplir con todas las partes de una solución web profesional, me hacen estar seguro que los conocimientos adquiridos me serán útiles en mi futura vida laboral.



8.1 Relación del trabajo realizado con los estudios cursados

En mi caso he realizado la especialización de Ingeniería del software en la cual se provee de la base para abordar y permitir planificar la estructura de cualquier proyecto software.

En la asignatura de Diseño del Software se indican los principios fundamentales para dotar a tu software de las características principales para construir una estructura robusta y escalable.

Análisis y especificación de requisitos es otra de las asignaturas que me ha ayudado, buscando las soluciones tecnológicas que más se podían adaptar a los requisitos para la creación de la plataforma.

Deontología y Profesionalismo ha resultado ser una de las asignaturas que han proporcionado al alumno una mejora a la hora de realizar búsquedas y contrastar información, lo que ha sido especialmente útil en el apartado de estado del arte y análisis del problema.

En conclusión, cursar el grado en ingeniería informática ha sido fundamental para proporcionarme los conocimientos para poder abordar proyectos de software, estableciendo las bases para crecer y desenvolverse en el mundo laboral.

9. Trabajos futuros

El proyecto realizado ha sido muy ambicioso y aunque considero que se ha abordado en gran medida todos los objetivos planteados, existen dos puntos principales que me gustaría continuar con su desarrollo en el futuro, estos son:

Añadir funcionalidades a los grupos de trabajo

Considero fundamental mejorar la parte de los grupos de trabajo dentro de la plataforma.

Se plantea un futuro desarrollo que busca corregir esto, mediante el uso usuarios responsables del grupo, los usuarios administradores podrán delegar parte de la administración únicamente sobre los miembros del grupo que deseen.

Monetización de la aplicación

Actualmente existe una configuración del límite de empresas y empleados por cada cuenta, pero es necesario implementar un sistema de pagos que permita a los usuarios aumentar estos límites.

Se plantea como desarrollo futuro un sistema de suscripciones, el cual busca aportar flexibilidad a los usuarios dependiendo de sus necesidades.

Esta mejora en la aplicación permitirá aumentar la infraestructura de los servidores, así como aumentar los recursos de desarrollo.

10. Referencias

1. **Social, Ministerio De Empleo Y Seguridad.** iberley. *iberley*. [En línea] <https://www.iberley.es/legislacion/rd-ley-8-2019-8-mar-medidas-urgentes-proteccion-social-lucha-contra-precariedad-laboral-jornada-trabajo-26206060>.

2. **CCOO.** Horas extra no pagadas y absentismo. [En línea] <https://www.ccoo.es/2b364fac2e0fbc15cfe49721e0acc2dc000001.pdf>.

3. **trabajadores, Dudas frecuentes del registro de la jornada de los.** *cuestioneslaborales*. *cuestioneslaborales*. [En línea] 4 de 2021. <https://www.cuestioneslaborales.es/obligatoriedad-registro-jornada-trabajo/#legislacion>.

4. **trabajadores, Estatuto de los.** ccoo-servicios. *ccoo-servicios*. [En línea] <https://www.ccoo-servicios.es/archivos/itdeusto/EstatutoTrabajadores.pdf>.

5. **Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.** boe.es. *boe.es*. [En línea] <https://www.boe.es/eli/es/lo/2018/12/05/3/con>.

6. **App, LOPD: La ley de protección de datos que debe cumplir tu.** fp.uoc.fje.edu. *fp.uoc.fje.edu*. [En línea] <https://fp.uoc.fje.edu/blog/lopd-la-ley-de-proteccion-de-datos-que-debe-cumplir-tu-app/>.

7. **Real Decreto Legislativo 5/2000, de 4 de agosto, por el que se aprueba el texto refundido de la Ley sobre Infracciones y Sanciones en el Orden Social.** boe.es. *boe.es*. [En línea] <https://www.boe.es/buscar/act.php?id=BOE-A-2000-15060>.

8. **jornada, Sentencia Audiencia Nacional descuento.** cdn.cuestioneslaborales.es. *cdn.cuestioneslaborales.es*. [En línea] <https://cdn.cuestioneslaborales.es/wp-content/uploads/2019/07/Sentencia-Audiencia-Nacional-descuento-jornada.pdf>.

9. **informatico, Cuanto gana un ingeniero.** universidadeuropea. *universidadeuropea*. [En línea] <https://universidadeuropea.com/blog/cuanto-gana-un-ingeniero-informatico>.

10. **introducción, PHP.** php.net. *php.net*. [En línea] <https://www.php.net/manual/es/intro-what-is.php>.