



# **LABOREO: AppWeb para el control de empleados de una empresa**

Autor: Iván Rosillo Pérez  
Tutor: Jordi Ustrell Garrigos  
Profesor: Ferran Adell Español

Grado en Multimedia  
Ingeniería Web

Junio 2021

# Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento- NoComercial-SinObraDerivada [3.0 España de Creative Commons.](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Para el desarrollo del sistema se han utilizado recursos externos cuyas licencias se adjuntan a continuación:

## **Bootstrap**

Licencia MIT

<https://getbootstrap.com/docs/4.0/about/license/>

## **Fontawesome**

Licencia Creative Commons de Reconocimiento

<https://fontawesome.com/license/free>

## **Nodejs**

Copyright Node.js contributors. All rights reserved.

<https://raw.githubusercontent.com/nodejs/node/master/LICENSE>

## **Npm**

Licencia Artistic License 2.0

<https://www.npmjs.com/policies/npm-license>

## **Vue.js**

Licencia MIT

<https://vuejs.org>

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Diseño y desarrollo de una aplicación web para gestionar las jornadas laborales de los empleados de una empresa</i>
<b>Nombre del autor:</b>	<i>Iván Rosillo Pérez</i>
<b>Nombre del colaborador/a docente:</b>	<i>Jordi Ustrell Garrigos</i>
<b>Nombre del PRA:</b>	<i>Ferran Adell Español</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>06/2021</i>
<b>Titulación o programa:</b>	<i>Grado en Multimedia</i>
<b>Área del Trabajo Final:</b>	<i>Ingeniería Web</i>
<b>Idioma del trabajo:</b>	<i>Español</i>
<b>Palabras clave</b>	<i>Memoria, Trabajo de Final de Grado, Aplicación Web</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p><i>Laboreo</i> es una aplicación web con licencia libre pensada para cubrir las necesidades de una empresa en el control de las jornadas laborales de sus empleados, tanto desde el punto de vista del trabajador como del responsable de la organización.</p> <p>La aplicación web ha sido desarrollada utilizando las tecnologías HTML5, CSS y Javascript junto con una serie de <i>frameworks</i> con licencia libre que mejoran la experiencia del usuario. Todo el código fuente ha sido debidamente documentado para que pueda ser modificado en caso de necesidad por terceros.</p> <p>El resultado final es una <i>PWA (Progressive Web App)</i> de aspecto sobrio, con acciones muy intuitivas para los usuarios no experimentados, que ofrece tanto al personal como a la organización una gestión sencilla, práctica y veloz.</p>	
<b>Abstract (in English, 250 words or less):</b>	
<p><i>Laboreo</i> is an open-source project with a free license that it has been created to cover the enterprise requires about control employee workday and a control for the responsible for the organisation.</p> <p>This webapp has been developed with HTML5, CSS and Javascript technology with differents free licenses frameworks that improve the user experience. All of the source code has been documented to be modified if is necessary to improve or implement news features.</p> <p>The final result is an <i>PWA (Progressive Web App)</i> with a serious image, with intuitives actions for inexperienced users, that offer to workers and to the organization a simple, practice and quickly gestion.</p>	

## **Dedicatoria/Cita**

“La inseguridad sirve para saber lo que realmente deseas”

“Arrepiéntete de lo que has hecho, no de lo que no hayas hecho”

Las dudas siempre están presentes en la elaboración de un proyecto y a través de ellas, descartamos soluciones, descubrimos objetivos y avanzamos.

## Agradecimientos

A mi marido, por apoyarme durante tantos años a seguir mis estudios y no dejarlos. Gracias a su apoyo y su firmeza he conseguido llegar hasta el último peldaño de esta aventura.

A mi madre, que es la persona a la que más quiero en esta vida, y que nunca me ha fallado. Me siento muy orgulloso de ser su hijo, y le debo todo lo que tengo y lo que soy. Te quiero.

# Notaciones y Convenciones

## Titulo 1

Hace referencia al nombre del capítulo. Se utiliza la tipografía **Arial** en negrita de tamaño 20.

## Titulo 2

Hace referencia a los subapartados dentro del capítulo. Se utiliza la tipografía **Arial** en negrita de tamaño 13.

## Titulo 3

Hace referencia a los subapartados dentro del subapartado. Se utiliza la tipografía **Arial** en negrita de tamaño 13.

### *Xenismo*

Para indicar las palabras escritas en otros idiomas, se utiliza la tipografía *Arial* en cursiva de tamaño 10.

### **Palabra destacada**

Para destacar un nombre o palabra dentro de un apartado, se utiliza la tipografía **Arial** en negrita de tamaño 10.

### `Elemento Código`

Para mostrar fragmentos de código, se utiliza la tipografía **Consolas** regular en tamaño 10'5, con fondo negro, y color del texto según definidos en *Visual Studio Code*.

Para el contenido general de la memoria, se utiliza tipografía **Arial** regular de tamaño 10, con interlineado de 1'15 y texto justificado.

# Índice

<b>1. Introducción.....</b>	<b>11</b>
<b>1.1. Introducción/Prefacio .....</b>	<b>11</b>
<b>1.2. Descripción/Definición .....</b>	<b>13</b>
<b>1.3. Objetivos generales .....</b>	<b>15</b>
1.3.1. Objetivos principales .....	15
1.3.2. Objetivos secundarios .....	15
<b>1.4. Metodología y proceso de trabajo.....</b>	<b>16</b>
<b>1.5. Planificación .....</b>	<b>17</b>
1.5.1. Detalle de las tareas .....	17
1.5.2. Diagrama de Gantt .....	20
<b>1.6. Presupuesto.....</b>	<b>21</b>
<b>1.7. Estructura del resto del documento .....</b>	<b>22</b>
<b>2. Análisis de mercado.....</b>	<b>23</b>
<b>2.1. Público objetivo (i.e. <i>target audience</i>) y perfiles de usuario.....</b>	<b>23</b>
<b>2.2. Competencia .....</b>	<b>27</b>
<b>2.3. Análisis DAFO .....</b>	<b>32</b>
<b>3. Propuesta.....</b>	<b>33</b>
<b>3.1. Definición de objetivos/especificaciones del producto.....</b>	<b>33</b>
<b>4. Diseño.....</b>	<b>34</b>
<b>4.1. Arquitectura general de la aplicación/sistema/servicio.....</b>	<b>34</b>
<b>4.2. Arquitectura de la información y diagramas de navegación .....</b>	<b>34</b>
4.2.1. Almacenamiento de la información.....	34
4.2.2. Diagrama de clases .....	38
4.2.3. Diagrama de navegación (árbol de navegación).....	38
<b>4.3. Diseño gráfico e interfaces.....</b>	<b>41</b>
4.3.1. Estilos .....	41
4.3.2. Usabilidad /UX .....	45
<b>4.4. Lenguajes de programación y APIs utilizados.....</b>	<b>46</b>
4.4.1. Back-end.....	48

4.4.2. Front-end .....	49
<b>5. Implementación.....</b>	<b>51</b>
<b>5.1. Requisitos de instalación .....</b>	<b>51</b>
5.1.1. Requisitos para el servidor .....	51
5.1.2. Requisitos para cliente .....	51
<b>5.2. Instalación en Heroku.....</b>	<b>52</b>
<b>6. Demostración .....</b>	<b>56</b>
<b>6.1. Prototipos.....</b>	<b>56</b>
6.1.1. Prototipos Lo-Fi.....	56
<b>6.2. Test.....</b>	<b>60</b>
<b>6.3. Ejemplos de uso del producto (o guía de usuario) .....</b>	<b>60</b>
<b>7. Conclusiones y líneas de futuro .....</b>	<b>64</b>
7.1. Conclusiones.....	64
7.2. Líneas de futuro .....	65
<b>Bibliografía .....</b>	<b>67</b>
<b>Anexos .....</b>	<b>69</b>

# Figuras y tablas

## Índice de figuras

Figura 1: Icono de <i>Laboreo</i> .....	12
Figura 2: Diagrama de Gantt.....	20
Figura 3: Uso de TIC por las personas de 16 a 74 años (Instituto Nacional de Estadística, 2020).....	23
Figura 4: Tipo de conocimiento informático. Año 2020 (Instituto Nacional de Estadística, 2020).....	24
Figura 5: Activos por grupo de edad. Año 2020 (Instituto Nacional de Estadística, 2020).....	24
Figura 6: Página principal de Sesame ( <a href="https://www.sesametime.com/">https://www.sesametime.com/</a> ).....	28
Figura 7: Página principal de Factorial ( <a href="https://factorialhr.es/">https://factorialhr.es/</a> ).....	29
Figura 8: Página principal de OpenKM ( <a href="https://www.openkm.com/es/">https://www.openkm.com/es/</a> ).....	30
Figura 9: Página principal de iAccesos ( <a href="https://iaccesos.com/index.php/home/presentacion">https://iaccesos.com/index.php/home/presentacion</a> ).....	30
Figura 10: Arquitectura general del proyecto.....	34
Figura 11: Diagrama de clases.....	38
Figura 12: Árbol de navegación.....	39
Figura 13: Símbolo de Laboreo.....	41
Figura 14: Logotipo de Laboreo.....	42
Figura 15: Icono.....	43
Figura 16: Paleta de colores.....	44
Figura 17: Ejemplo de iconos de (Font Awesome).....	45
Figura 18: Logotipo de PostgreSQL. Fuente: (Wikipedia).....	47
Figura 19: Logotipos de Vue.js y Node.js.....	48
Figura 20: Ejemplo de formato JSON.....	49
Figura 21: Ejemplo de estructura de componentes en Vue.....	50
Figura 22: Ejemplo de un componente de Vue.....	50
Figura 23: Logotipo de Heroku. Fuente: (Wikipedia).....	52
Figura 24: Ejemplo de cómo desplegar una app en Heroku. Fuente: (Heroku).....	53
Figura 25: Instalación de la Laboreo desde Google Chrome.....	54
Figura 26: Ejecución de la aplicación de forma nativa.....	54
Figura 27: Estructura.....	56
Figura 28: Página principal.....	57
Figura 29: Menú.....	57
Figura 30: Página de fichajes.....	58
Figura 31: Página del calendario.....	58
Figura 32: Página de eventos.....	59
Figura 33: Página de centros de trabajo.....	59
Figura 34: Página de informes.....	60
Figura 35: Página del listado.....	60
Figura 36: Inicio de sesión.....	61
Figura 37: Pantalla de registro de entradas y salidas.....	61
Figura 38: Pantalla de edición de registros.....	62

Figura 39: Pantalla del calendario..... 62  
Figura 40: Pantalla de informes ..... 63

**Índice de tablas**

Mesa 1: Lista de tareas ..... 19  
Mesa 2: Presupuesto del proyecto..... 21  
Mesa 3: Presupuesto detallado del proyecto ..... 21  
Mesa 4: Perfil de usuario focal..... 25  
Mesa 5: Perfil de usuario focal ..... 26  
Mesa 6: Perfil de usuario focal..... 27  
Mesa 7: Tabla comparativa de servicios de control horario ..... 31  
Mesa 8: Análisis DAFO ..... 32  
Mesa 9: Entidades/tablas de la base datos..... 37  
Mesa 10: Tipografía y paleta de colores del logotipo ..... 43  
Mesa 11: Tipografía utilizada en la aplicación ..... 44  
Mesa 12: Tabla de usuarios de pruebas ..... 55

# 1.Introducción

## 1.1. Introducción/Prefacio

Tras la publicación del Real Decreto-Ley 8/2019, las empresas están obligadas a ofrecer a sus trabajadores alguna herramienta para controlar el registro diario del horario de inicio y de finalización de sus jornadas. Para ciertas empresas con unos recursos limitados, disponer de una herramienta digital puede suponer un esfuerzo considerable. A partir de esta necesidad nace la herramienta **Laboreo**<sup>1</sup>, que pretende ser un producto preparado para ser implantado rápidamente en cualquier organización, y que puede ofrecer tanto a la empresa como a sus empleados, las herramientas suficientes para tener un control total de las jornadas laborales.

La funcionalidad principal de Laboreo es la gestión de las jornadas de los empleados de una organización. El acceso se realiza mediante un usuario y contraseña, y según el rol de éste, el sistema ofrece más o menos funcionalidades. Por un lado, existe un rol de empleado que permite al usuario realizar el registro horario de su jornada laboral y consultar su calendario de trabajo (horario establecido, días libres, etc.). Por otro, el rol del responsable, que además de disponer del rol de empleado activo, puede realizar el mantenimiento de las fichas de los empleados, modificación de los calendarios, y gestión de las solicitudes del personal. La herramienta, además, ofrece una serie de informes -oficiales unos y otros más estadísticos-, de gran utilidad, tanto para los trabajadores como para la organización.

En la actualidad, existe una demanda, cada vez más fuerte, para hacer posible el teletrabajo. Por este motivo, a la hora de diseñar una nueva utilidad informática, uno de los requisitos más importantes es que sea accesible desde cualquier dispositivo que tenga acceso a internet. Por este motivo, esta herramienta es una aplicación web generada con HTML5 y JavaScript que permite ejecutarse en cualquier dispositivo con acceso a internet y con un navegador compatible con dichas tecnologías. Esta característica hace que el sistema tenga una accesibilidad prácticamente ilimitada, y no requiera de ningún tipo de instalación para su uso. Además, al no desarrollarse para una plataforma concreta (Windows, IOS, Android, etc.), la facilidad para utilizarlo aumenta, mientras que los costes de desarrollo se reducen.

En definitiva, esta aplicación es una herramienta de gran utilidad para administrar y almacenar información relevante de los puestos de trabajo de una organización. El sistema permite gestionar, de forma rápida, sencilla y accesible desde cualquier lugar, las principales tareas relacionadas con el departamento encargado de gestionar los puestos de trabajo, además de proporcionar a los

---

<sup>1</sup> La palabra “Laboreo” surge de la idea de coger las dos primera letras de las palabras *Laboring*, *Worker*, y *Register Hours*, dando como resultado la palabra compuesta “Laworeho”, y transformándola al castellano posteriormente.

empleados, toda la información de sus jornadas laborales. Por otro lado, al tratarse de una herramienta desarrollada como código abierto, cualquier organización puede utilizarla, si lo desea, como base para ampliar su funcionalidad.

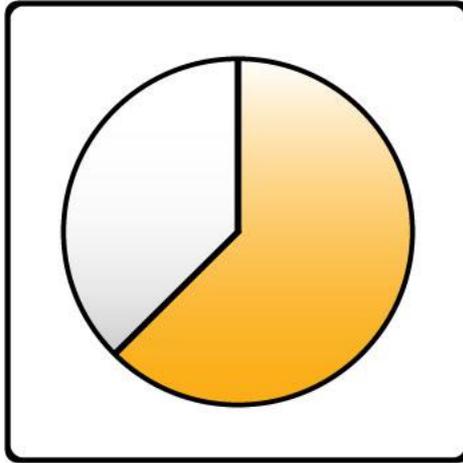


Figura 1: Icono de *Laboreo*

## 1.2. Descripción/Definición

El proyecto consiste en la creación de una aplicación web<sup>2</sup> que cubra las tareas principales del departamento de recursos humanos de una organización. En concreto, la aplicación permite gestionar el organigrama de la empresa, las fichas de los trabajadores -datos relevantes para los responsables del área como los datos de contacto, el puesto de trabajo, los contratos, etc.-, las jornadas laborales de los empleados -calendario de turnos, vacaciones, bajas-, y el control horario de los empleados.

El sistema se ha creado para ofrecer, a aquellas organizaciones que no tienen informatizado su negocio, una herramienta, de código libre, que les permita digitalizar la gestión de sus empleados con un acceso universal desde cualquier dispositivo, facilitando además el teletrabajo en los casos que lo requieran. Estas organizaciones verán reducido el esfuerzo por mantener actualizada la información de sus empleados, aumentando la productividad, reduciendo costes de integración, y, además, disponiendo, de forma instantánea, de datos sobre la disponibilidad de sus empleados junto con informes detallados sobre el rendimiento de estos.

Hoy en día la mayoría de las organizaciones están completamente informatizadas. Disponer de un correo electrónico es la primera de las fases en la informatización de una organización, y cada vez más, las empresas ven en la digitalización de sus datos un instrumento necesario e imprescindible. Laboreo es una pieza más en la digitalización de los pequeños negocios ya que permite informatizar una parte sumamente importante: la plantilla.

Hasta hace unos años, la información relacionada con los trabajadores de las organizaciones se gestionaba en papel y estaba localizada y centralizada en una ubicación concreta. La evolución de las nuevas tecnologías ha permitido que las condiciones que rodean los puestos de trabajo también cambien, de modo que los datos que se manejan son accesibles de forma remota desde cualquier lugar. Gracias a esta tecnología, como es el caso de “la nube<sup>3</sup>”, el *software* se convierte en un servicio para los usuarios que aporta mayor control. Laboreo cumple con esta característica por tratarse de una aplicación en línea, que es accesible desde cualquier lugar en cualquier momento, y está pensada para que pueda configurarse adaptándose a los diferentes tipos de organización y estructura que pueda tener una empresa.

Laboreo ofrece a las empresas los utensilios para crear una base de datos completa se sus trabajadores, definir los diferentes turnos que puedan existir en la organización, planificar las semanas

---

<sup>2</sup> También conocidas como *Progressive Web Application (PWA)*. Más información en: [https://developer.mozilla.org/es/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/es/docs/Web/Progressive_web_apps)

<sup>3</sup> La computación en la nube (del inglés *cloud computing*), conocida también como servicios en la nube, informática en la nube, nube de cómputo o simplemente «la nube», es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es internet. Fuente: Wikipedia. Disponible en: [https://es.wikipedia.org/wiki/Computación\\_en\\_la\\_nube](https://es.wikipedia.org/wiki/Computación_en_la_nube).

de trabajo de la plantilla, obtener informes para el control horario de los puestos de trabajo y conocer la disponibilidad de sus empleados. A los trabajadores de la empresa, la aplicación les permite disponer de los informes mensuales y anuales de sus jornadas laborales, gestionar las horas extras, consultar el calendario de turnos, vacaciones, bajas o ausencias, y facilitar el teletrabajo (para aquellos puestos que puedan realizarlo).

Con la creación de esta herramienta, se pretende ofrecer un producto asequible para cualquier organización que necesite disponer de un sistema informatizado para el control de las jornadas del personal. Al tratarse de una aplicación web creada con software libre, y siendo propiamente software libre, los costes para su instalación son mínimos por no requerir de licencias de pago. El sistema está pensado para que pueda ser instalado en un servidor web y con una base de datos PostgreSQL. Para el desarrollo de este proyecto se utilizó inicialmente un servidor con sistema operativo Linux (Debian), con el servicio web Apache y con base de datos PostgreSQL. Posteriormente, gracias a tecnología más avanzada, se ha prescindido del sistema operativo Linux y del servidor Apache al utilizar una plataforma externa para el hosting del programa.

## 1.3. Objetivos generales

### 1.3.1. Objetivos principales

Objetivos de la aplicación/producto/servicio:

- Crear una aplicación web de código libre que permita a organizaciones y empresas la gestión de las fichas de sus empleados, el control de los calendarios laborales y el registro horario de los trabajadores.
- Conseguir que la aplicación sea 100% adaptable a cualquier dispositivo desde el que se acceda.
- Crear una herramienta con un mantenimiento sencillo y fácil de mejorar y adaptar a las necesidades del cliente.

Objetivos para el cliente/usuario:

- Reducir costes en la implantación en comparación con otras herramientas similares.
- Aumentar la productividad a la hora de gestionar las jornadas de los empleados.
- Facilitar el control horario de los empleados de una organización para eliminar la documentación en papel.
- Digitalizar la información relativa a las fichas de empleados, cuadros horarios y registros laborales.
- Disponer de un sistema informatizado, centralizado y de fácil acceso para la organización.

Objetivos personales del autor del TF:

- Crear una *PWA (Progressive Web Application)* que permita una ejecución en línea como si fuera una aplicación local.
- Adquirir conocimientos tanto en la gestión de un proyecto completo, como en el desarrollo de este con las tecnologías utilizadas.
- Realizar un proyecto con licencia de código abierto.

### 1.3.2. Objetivos secundarios

Objetivos adicionales que enriquecen el TF.

- Ofrecer una herramienta básica de gestión de una empresa (recursos humanos) que pueda ser ampliada al disponer del código fuente y de la estructura de base de datos.

## 1.4. Metodología y proceso de trabajo

Dado que el proyecto consiste en una aplicación sin un cliente concreto, el desarrollo de ésta se realiza pensando en la creación de un producto completamente nuevo. Es decir, no existe ningún sistema previo que deba readaptarse o actualizarse, por lo que el análisis se hace partiendo de cero. Esta estrategia permite crear un sistema sin ningún tipo de interferencia o de influencia, y que pueda desarrollarse cumpliendo los requisitos marcados en el análisis inicial.

Para la investigación de las necesidades del proyecto se han realizado consultas por internet de otros productos similares, además de consultar con perfiles de usuarios potenciales de la herramienta. Estas entrevistas han proporcionado información de interés que ha sido reevaluada para su integración en el análisis inicial.

En cuanto a la metodología de desarrollo del proyecto, se han evaluado distintos métodos para observar cuál se adapta mejor a las necesidades. Dado que el desarrollo de todo el proyecto se ha llevado a cabo por una única persona, el método seleccionado es el de "cascada". Este procedimiento realiza el proyecto de forma secuencial, pasando por las diferentes fases que un proyecto suele tener: análisis de las necesidades, diseño, implantación, pruebas e integración. Aun así, para el desarrollo de algunas particularidades del producto, se ha utilizado otro tipo de metodología más cercana a la "ágil", ya que ésta se adapta mejor a cualquier posible cambio durante el desarrollo del proyecto, porque se generan ciclos de vida de desarrollo propio de cada tarea. Además, durante el transcurso del proyecto se han ido modificando y adaptando algunos requisitos para añadir mayor funcionalidad o se han eliminado otros que finalmente no se han considerado necesarios.

## 1.5. Planificación

La planificación de proyecto se ha dividido en cinco etapas con fechas de entrega clave. El proyecto se inicia el 17 de febrero de 2021, y se finaliza el 7 de junio del mismo año. A continuación, se detallan las diferentes fases por las que ha transcurrido el desarrollo:

### **Fase 1. Definición del proyecto [21 de febrero de 2021]**

La primera fase consiste en la presentación del proyecto a desarrollar indicando cuál es su finalidad, la motivación para llevarla a cabo, y una descripción breve de la posible tecnología utilizada.

### **Fase 2. Definición formal del proyecto y la pauta de trabajo [7 de marzo de 2021]**

En esta fase se realiza la definición definitiva del proyecto, indicando todas las características y requisitos que debe cumplir la aplicación. Se marca también la metodología a seguir para desarrollar el proyecto y la concreción de las diferentes entregas parciales durante el proceso de creación.

### **Fase 3. Inicio del desarrollo [4 de abril de 2021]**

La tercera fase se inicia con la revisión de los requisitos y los métodos de trabajo. Finalizado este paso, en esta misma fase se define la base de datos que será la estructura primordial de la aplicación, junto con los *wireframe* para concretar las interfaces.

### **Fase 4. Desarrollo de la versión beta [9 de mayo de 2021]**

La penúltima fase consiste en la creación de una primera versión de la aplicación o versión beta. Es la fase en la que se desarrollan la mayoría de las funcionalidades del sistema y en la que el proyecto comienza a cobrar vida.

### **Fase 5. Finalización del proyecto [7 de junio de 2021]**

En la última fase, se consolida el proyecto y se realizan las últimas revisiones, testeos y generación de documentación, junto con la publicación del trabajo realizado.

### 1.5.1. Detalle de las tareas

A continuación, se detallan las tareas que componen cada fase en modo de tabla, donde se detalla el inicio y fin de cada tarea, y el número de horas requerido para la posterior confección del diagrama de Gantt.

Tarea	Inicio	Fin	Horas
Fase 1. Definición del proyecto	17/02/21	21/02/21	24
1.1. Propuestas para el proyecto	17/02/21	19/02/21	8
1.2. Debate sobre las propuestas	20/02/21	20/02/21	8
1.3. Formalización de la propuesta	21/02/21	21/02/21	8
Fase 2. Definición formal del proyecto y la pauta de trabajo	22/02/21	07/03/21	69
2.1. Estudio y análisis del formato para la memoria	22/02/21	22/02/21	3
2.2. Descripción detallada del proyecto	23/02/21	23/02/21	6
2.3. Definición de los objetivos	24/02/21	26/02/21	12
2.4. Elección de la metodología de trabajo	27/02/21	27/02/21	5
2.5. Desglose de las tareas (planificación)	28/02/21	01/03/21	15
2.6. Cálculo del presupuesto	02/03/21	03/03/21	10
2.7. Preparar escenario de desarrollo	04/03/21	07/03/21	
2.7.1. Instalación de servidor Linux (Debian)	04/03/21	04/03/21	4
2.7.2. Instalación de Apache	05/03/21	05/03/21	4
2.7.3. Instalación de PostreSQL	06/03/21	07/03/21	4
Fase 3. Inicio del desarrollo	08/03/21	04/04/21	180
3.1. Documentación de la memoria del proyecto	08/03/21	04/04/21	12
3.2. Concepción del producto	08/03/21		
3.2.1. Recapitular requisitos	08/03/21	08/03/21	2
3.2.2. Selección de posibles <i>frameworks</i> y <i>pluggins</i>	08/03/21	08/03/21	6
3.3. Arquitectura de la información	09/03/21		
3.3.1. Definición de los contenidos	09/03/21	09/03/21	8
3.3.2. Elaboración del diagrama de flujo	10/03/21	10/03/21	8
3.3.3. Diseño de la interacción del usuario	11/03/21	11/03/21	8
3.3.4. Diseño funcional	12/03/21	12/03/21	8
3.4. Diseño de la interfaz	13/03/21	18/03/21	
3.4.1. Elaboración de <i>wireframes</i>	13/03/21	16/03/21	32
3.4.2. Elección del estilo gráfico	17/03/21	17/03/21	8
3.4.3. Definición de las retículas y guías de estilo	18/03/21	18/03/21	8
3.5. Diseño de base de datos	19/03/21	28/03/21	
3.5.1. Análisis funcional	19/03/21	22/03/21	32
3.5.2. Creación del modelo UML	23/03/21	28/03/21	48
Fase 4. Desarrollo de la versión beta	05/04/21	09/05/21	396
4.1. Redacción de la memoria del proyecto	05/04/21	09/05/21	16
4.2. Creación de scripts de generación de base de datos	05/04/21	06/04/21	16
4.3. Programación de la aplicación	07/04/21	03/05/21	

4.3.1. Maquetación de las páginas en HTML5 y PHP	07/04/21	21/04/21	112
4.3.2. Definición de las hojas de estilo CSS	19/04/21	03/05/21	112
4.3.3. Programación de interactividad con JavaScript	12/04/21	26/05/21	112
4.4. Pruebas de funcionamiento	03/05/21	03/05/21	8
4.5. Corrección de errores	04/05/21	09/05/21	20
Fase 5. Finalización del proyecto	10/05/21	07/06/21	212
5.1. Redacción de la memoria del proyecto	10/05/21	07/06/21	18
5.2. Repasar funcionalidades solicitadas e implementadas	10/05/21	10/05/21	4
5.3. Testeo de la aplicación	11/05/21	15/05/21	48
5.4. Modificaciones y correcciones	16/05/21	04/06/21	130
5.5. Publicación de la versión definitiva	05/06/21	07/06/21	12

Mesa 1: Lista de tareas

### 1.5.2. Diagrama de Gantt

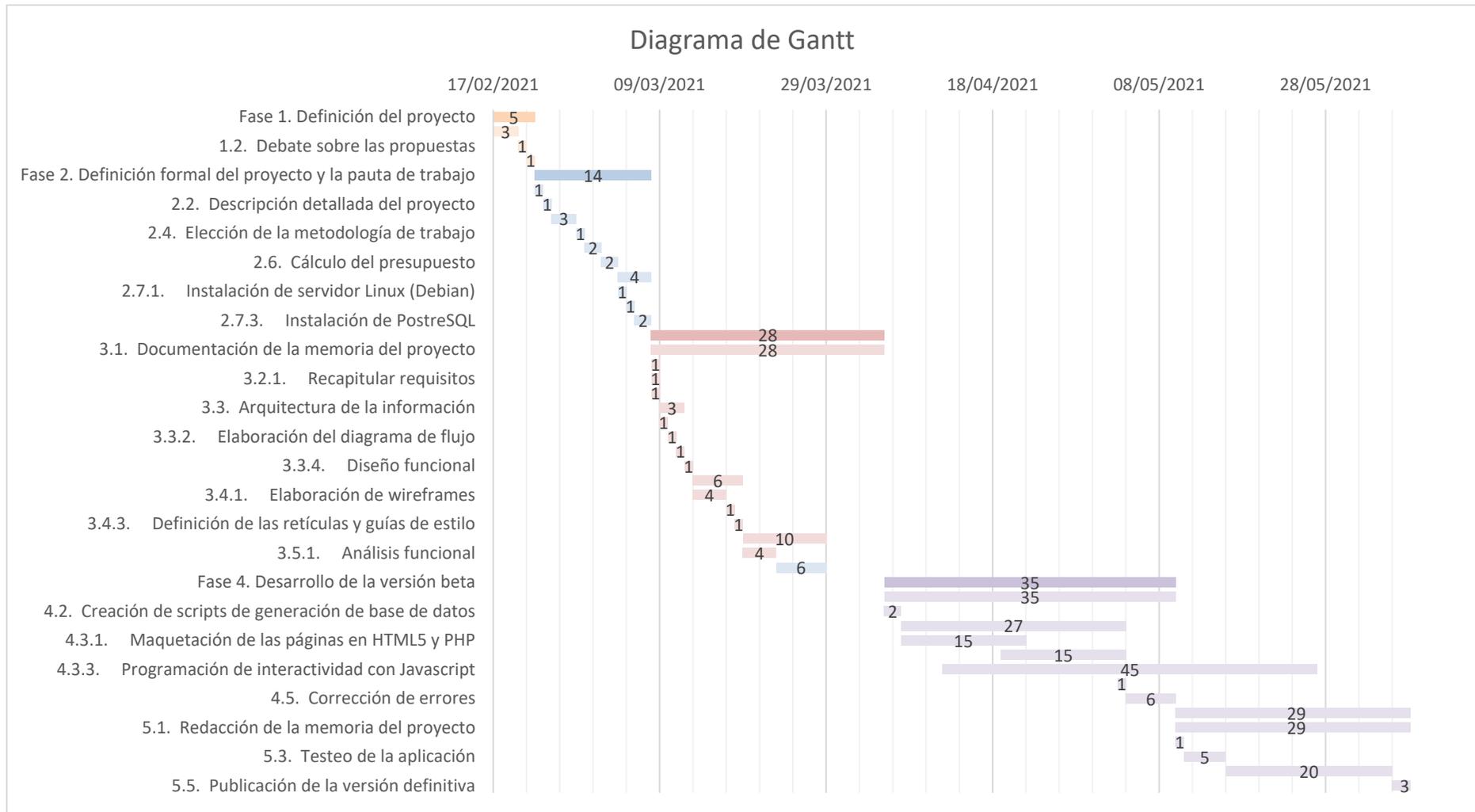


Figura 2: Diagrama de Gantt

## 1.6. Presupuesto

El presupuesto para el proyecto se basa en el coste que suponen las horas de desarrollo invertidas en la creación de la aplicación, por lo que la unidad para el cálculo es la hora de trabajo. Al utilizarse software libre, no existe ningún coste adicional de licencias. El importe del presupuesto es puramente informativo.

Servicio	Coste
Preparación de sistemas	94.00 €
Concepción del producto	144.00€
Diseño del producto	360.00 €
Desarrollo del producto	5,940.00 €
Documentación	480.00 €
Varios	500.00 €
<b>Total:</b>	<b>7,520.00 €</b>

Mesa 2: Presupuesto del proyecto

Tarea	Unidades	Precio/unidad	Coste
Instalación servidor Linux	4	8.00 €	32.00 €
Instalación Apache	4	8.00 €	32.00 €
Instalación PostgreSQL	4	8.00 €	32.00 €
<b>Preparación de sistemas</b>			<b>94.00€</b>
Análisis de requisitos	4	18.00 €	72.00€
Selección de <i>frameworks</i> y <i>pluggins</i>	4	18.00 €	72.00€
<b>Concepción del producto</b>			<b>144.00 €</b>
Arquitectura de la información	4	18.00 €	72.00 €
Diseño de la interfaz	6	18.00 €	108.00 €
Diseño de base de datos	10	18.00 €	180.00 €
<b>Diseño del producto</b>			<b>360.00 €</b>
Generación de base de datos	2	15.00 €	30.00 €
Programación	304	15.00 €	4,560.00 €
Testeo y correcciones	90	15.00 €	1,350.00
<b>Desarrollo del producto</b>			<b>5,940.00 €</b>
Documentación	40	12.00 €	480.00 €
<b>Documentación</b>			<b>480.00 €</b>
Varios (imprevistos, cambios en el diseño, etc.)	1	500.00 €	500.00 €
<b>Documentación</b>			<b>500.00 €</b>

Mesa 3: Presupuesto detallado del proyecto

## 1.7. Estructura del resto del documento

En el segundo punto de la memoria, se realiza un análisis del mercado para conocer el público al que va dirigida la aplicación web y el contexto en el que se utilizará. Dentro de este apartado se identifican los diferentes *stakeholders*<sup>4</sup> que afectan al desarrollo del proyecto.

En el apartado número tres, se detalla la propuesta de la aplicación a partir del análisis de mercado hecho en el capítulo anterior. En ella se destacan las particularidades que lo diferencian del resto de productos similares que hay en el mercado.

---

<sup>4</sup> El interesado, parte interesada o involucrado (del inglés *stakeholder*) es una persona, organización o empresa que tiene interés en una empresa u organización dada. Fuente: Wikipedia. Disponible en: [https://es.wikipedia.org/wiki/Interesado\\_\(empresa\)](https://es.wikipedia.org/wiki/Interesado_(empresa)).

## 2. Análisis de mercado

Para desarrollar la idea del producto, se ha realizado una búsqueda en internet en la que se han seleccionado aplicaciones que cumplen con los requisitos indicados anteriormente para poder comparar la oferta existente en el mercado con la del producto generado. A continuación, se detallan algunas de las observaciones que se han extraído de este análisis comparativo.

### 2.1. Público objetivo (i.e. *target audience*) y perfiles de usuario

Dado que la aplicación tiene una funcionalidad exclusivamente enfocada al mundo laboral, el público al que va dirigida es muy concreto. En este sentido, es necesario realizar un estudio de ciertos aspectos relacionados con el perfil de la población.

Al analizarlo, se ha considerado el tipo de usuario en relación a la capacidad de uso de las nuevas tecnologías. Así, según el Instituto Nacional de Estadística (INE), en 2020, el 93'2% de las personas entre 16 a 74 años hicieron uso de internet. La siguiente figura muestra el aumento del uso de internet en los últimos años.

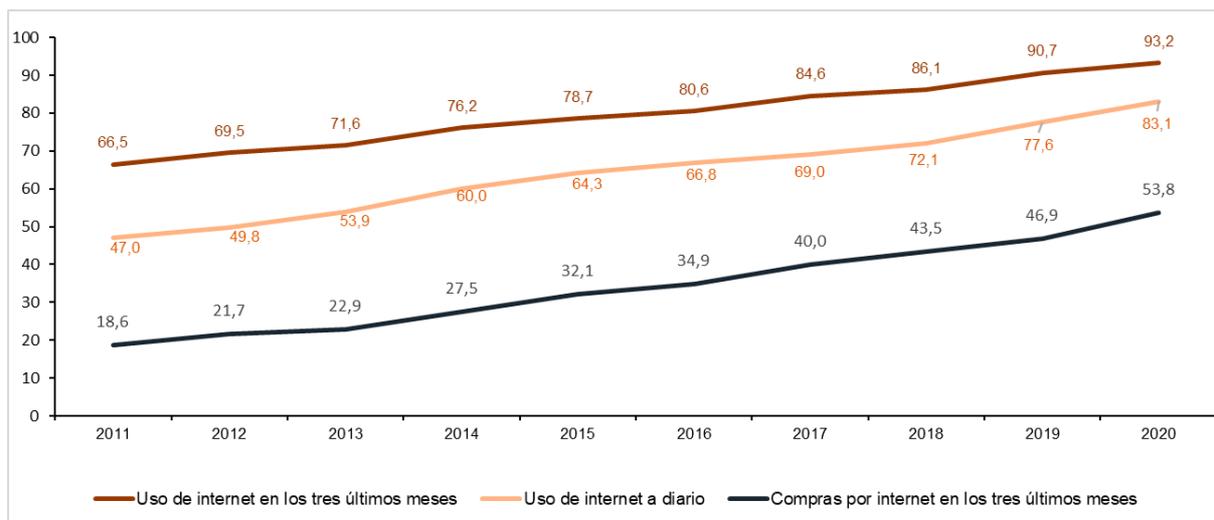


Figura 3: Uso de TIC por las personas de 16 a 74 años (Instituto Nacional de Estadística, 2020)

En cuanto a los conocimientos informáticos, el INE ha concluido que el 50'5% de la población activa posee habilidades avanzadas<sup>5</sup> en el uso de las TIC.

<sup>5</sup> El nivel de habilidades digitales se calcula según la metodología de organismo Eurostat (Oficina Europea de Estadística).

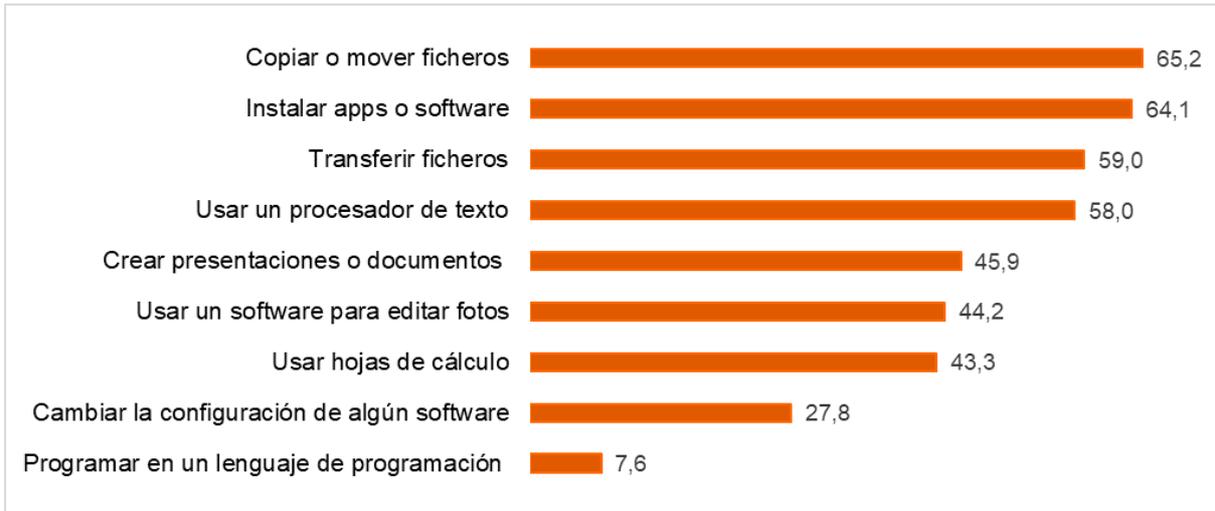


Figura 4: Tipo de conocimiento informático. Año 2020 (Instituto Nacional de Estadística, 2020)

En el año 2020, el porcentaje de ocupados en España por rango de edad muestra que la población mayoritariamente activa es la que se encuentra entre los 30 y 59 años.

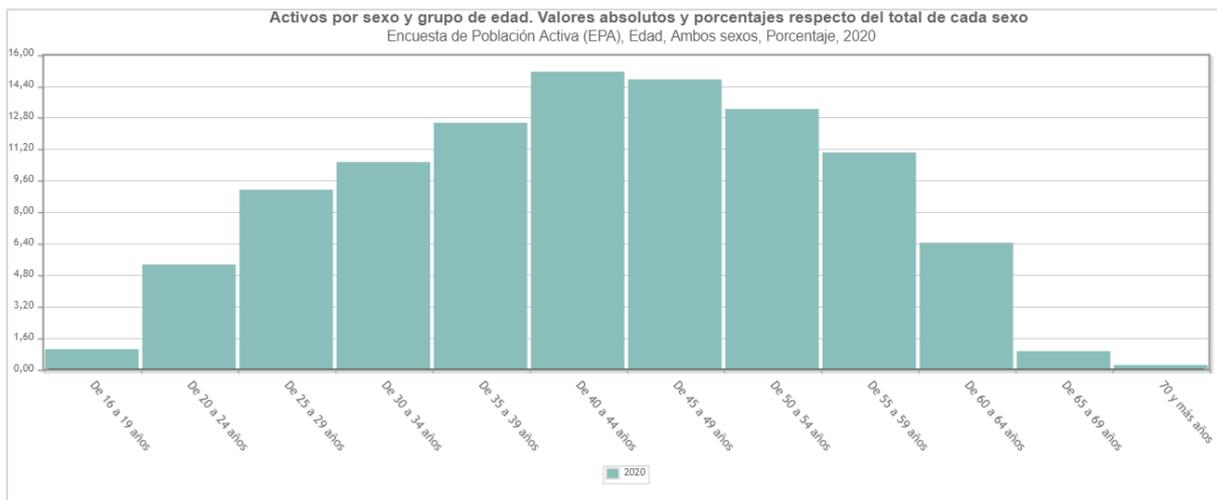


Figura 5: Activos por grupo de edad. Año 2020 (Instituto Nacional de Estadística, 2020)

Teniendo en cuenta estos datos, se puede concluir que el público objetivo para esta aplicación son personas que se encuentran activas laboralmente, en edades entre los 18 y los 65 años, y con conocimientos básicos de las nuevas tecnologías. Si bien es cierto que el rango de personas más jóvenes tiene un nivel más avanzado, hay que destacar que en el desarrollo de la aplicación se ha tenido en cuenta a la población mayoritaria activa sin hacer distinciones específicas.

Para obtener una descripción más detallada de los perfiles de los usuarios que utilizarán el producto, a continuación, se incluyen tres posibles tipos de usuarios:

Persona Focal	
<b>Nombre</b>	Sole
<b>Edad</b>	60
<b>Profesión</b>	Directora
<b>Género</b>	Femenino
<b>Lugar</b>	Barcelona



Foto obtenida en Pexels (Richelieu, 2020)

**Descripción**

Sole es directora en una organización que se dedica a la logística y el transporte de mercancía por todo el territorio nacional. La empresa está ubicada en Barcelona y Sole realiza tareas de dirección. Sole tiene conocimientos básicos de ofimática y también sabe utilizar la herramienta informática que les ayuda en las tareas diarias del negocio. Sin embargo, el sistema no les permite tener informatizada la información importante sobre el personal. Sole tiene en su despacho una estantería con varias carpetas donde almacena los datos de los empleados ordenados alfabéticamente en papel (contratos, bajas, y otros documentos). Tras la publicación de la normativa que obliga a las empresas a ofrecer un sistema para el control horario del personal, Sole ha creado unas hojas de cálculo que se imprimen para que puedan rellenarse manualmente con las horas de entrada y salida de cada mes, y que posteriormente recoge y archiva. Sole considera que este sistema es precario, y además le quita mucho tiempo, por lo que está valorando introducir en la empresa algún tipo de automatización para el registro horario y poder informatizar las fichas del personal. De este modo, ganaría tiempo y podría dedicarlo a otros asuntos más importantes para la empresa.

**Necesidades**

Introducir en la empresa un sistema que permita automatizar los fichajes de los empleados sin necesidad de instalar ningún tipo de registro físico, es decir, que pueda realizarse desde un ordenador personal o un terminal móvil.

Digitalizar los datos de las fichas del personal, además de poder registrar las bajas por enfermedad u otra causa.

Obtener un sistema económico y fácil de mantener.

Mesa 4: Perfil de usuario focal

Persona Focal	
<b>Nombre</b>	José
<b>Edad</b>	39
<b>Profesión</b>	Mozo de almacén
<b>Género</b>	Masculino
<b>Lugar</b>	Barcelona



Foto obtenida en Pexels (Lily, 2020)

**Descripción**

José trabaja como mozo de almacén en una empresa que se dedica a la logística y transporte de mercancía. Es una persona joven con estudios superiores, y con gran interés por las nuevas tecnologías. José maneja maquinaria especializada para las operativas de carga y descarga de mercancía en el almacén, y dispone de un ordenador para registrar los datos en el sistema de la empresa. Aunque no ha hecho formación específica de informática, tiene cierta habilidad para desenvolverse con el software en general. Cuando llega a su puesto de trabajo anota en su hoja de registro el día y la hora, y hace lo mismo cuando termina su jornada. José tiene diferentes turnos de trabajo, que varían por diversas circunstancias, y su gerente le informa del turno la semana anterior. Para poder disponer de días libre, José debe solicitarlos previamente y espera la confirmación/aceptación. Como en ocasiones hace horas extras, las anota en una libreta que tienen para posteriormente reclamarlas a su superior.

**Necesidades**

Tener un sistema automatizado para poder registrar desde un terminal móvil las entradas y salidas de sus jornadas laborales. Además, también se necesita que este sistema calcule automáticamente las horas extras que se han realizado en base a los registros horarios.

Disponer de un calendario para ver los diferentes turnos que se le han asignado desde central.

Poder consultar y solicitar los días libres que le pertenecen.

Persona Focal	
<b>Nombre</b>	Yolanda
<b>Edad</b>	44
<b>Profesión</b>	Administrativa
<b>Género</b>	Femenino
<b>Lugar</b>	Madrid



Foto obtenida en Pexels (cottonbro, 2020)

**Descripción**

Yolanda es empleada en una oficina de venta de puertas automáticas. Es madre soltera de un niño de seis años, por lo que tiene que organizarse muy bien su tiempo. En su trabajo se encarga de recoger los pedidos de los clientes, prepararlos y facturarlos. Normalmente trabaja la totalidad de su jornada con un ordenador de sobremesa en la oficina. Durante el último año, debido a la situación de pandemia, realiza la mayor parte de su trabajo desde casa. Esta opción le ha permitido poder gestionarse mejor el tiempo con el cuidado de su familia, por lo que ha solicitado a la empresa poder realizar el teletrabajo de forma permanente. Para el registro horario disponen de una hoja de cálculo donde se registran las entradas y salidas, que posteriormente se imprimen para enviarlas firmadas a su responsable. Hace tres semanas su hijo enfermó, por lo que tuvo que faltar durante dos días a su puesto de trabajo. Como algunos días trabaja desde casa, el resto de sus compañeros no fueron conscientes de que no estaba disponible en esas jornadas hasta pasadas ya varias horas, cuando Yolanda activó el “fuera de oficina” de su correo.

**Necesidades**

Disponer de una aplicación que permita realizar el registro horario para personal que hace teletrabajo.

Poder consultar en un calendario si el personal no está operativo, ya sea por vacaciones o por alguna baja.

Mesa 6: Perfil de usuario focal

## 2.2. Competencia

Existen múltiples opciones disponibles en internet para gestionar los registros horarios de una empresa. Solo hace falta lanzar una consulta en Google para obtener cientos de resultados. A continuación, se analizan una serie de opciones

## Sesame

Sesame es una de las primeras opciones que el buscador de *Google* devuelve. Esta herramienta ofrece una gestión del equipo de trabajo muy completa. Los responsables pueden tener acceso a la disponibilidad y ubicación de su plantilla en tiempo real, permite geolocalizar los fichajes y dispone de una gestión de vacaciones (solicitudes y aprobaciones), gracias a una App para Android e IOS. El servicio se ofrece en la nube y el cliente contrata los servicios durante un determinado tiempo mediante cuotas mensuales o anuales.

Esta herramienta es la principal competencia para Laboreo, ya que dispone de la mayoría de las funcionalidades que ofrece el proyecto.

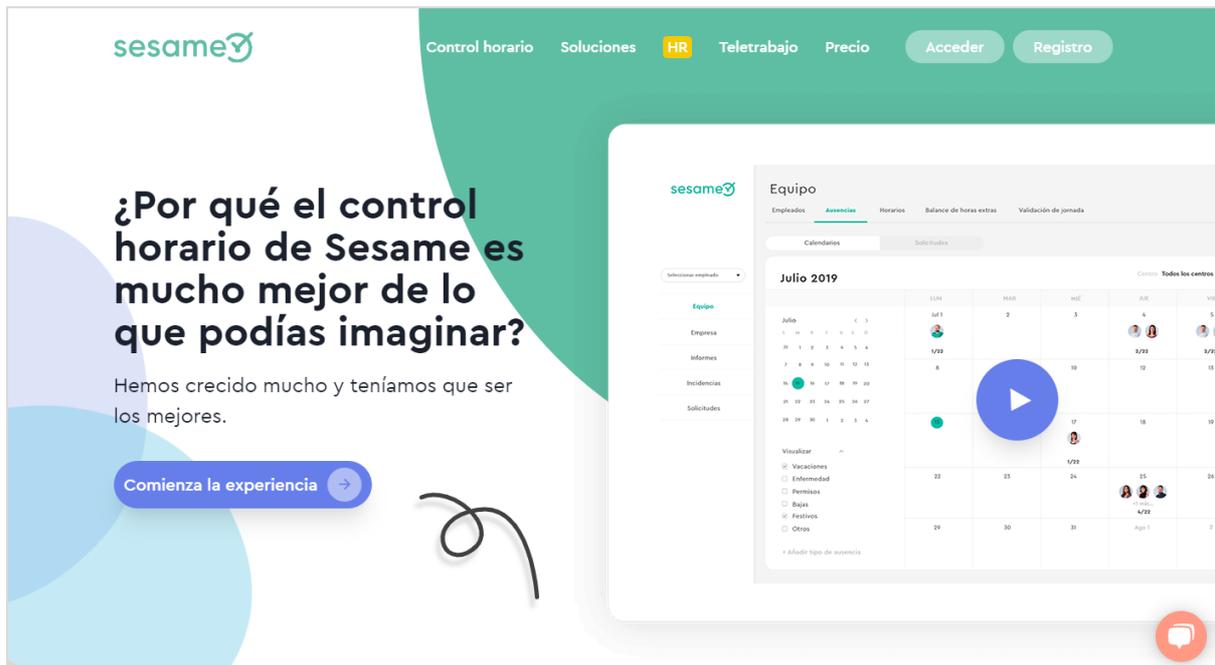


Figura 6: Página principal de Sesame (<https://www.sesametime.com/>)

## Factorial

Esta herramienta permite la gestión de las tareas administrativas del departamento de recursos humanos de una empresa. Dispone de la gestión de ausencias y vacaciones, control horario, gestión de nóminas, un gestor documental, una opción para realizar el seguimiento del proceso de selección de nuevo personal, e incluso la posibilidad de crear un portal de empleado. Al igual que en el caso anterior, dispone de una App destinada a Android e IOS para su utilización, y su uso está asociado a diferentes planes y precios.

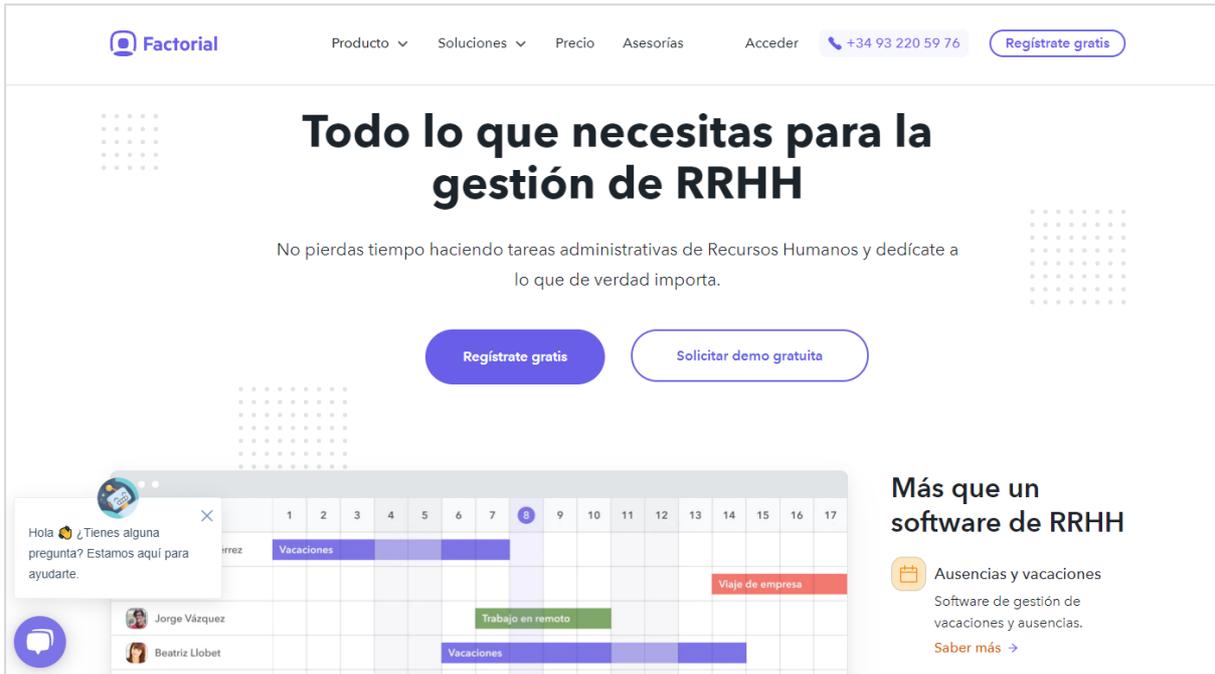


Figura 7: Página principal de Factorial (<https://factorialhr.es/>)

## OpenKM

OpenKM es una aplicación web de gestión documental para empresas desarrollado con tecnología *Open Source* que permite adaptarse a los diferentes tipos de sociedades. Dentro de este proyecto, existe un módulo de gestión de registros – *Record Management* – donde se puede llevar una gestión documental para el registro horario. Quizá no es la herramienta más compleja, pero se trata de una herramienta de código abierto.



Figura 8: Página principal de OpenKM (<https://www.openkm.com/es/>)

### iAccesos

Por último, iAccesos es una aplicación que permite el registro de entradas y salidas tanto en dispositivos de escritorio como en móviles y tabletas. La aplicación permite sacar informes, realizar justificantes por ausencias y obtener alertas por discrepancias en la geolocalización de los fichajes. La aplicación almacena los datos de registros de los empleados durante cuatro años. El servicio es gratuito porque pueden aparecer anuncios publicitarios para costear el proyecto.



Figura 9: Página principal de iAccesos (<https://iaccesos.com/index.php/home/presentacion>)

La siguiente tabla muestra una comparativa de ciertas características analizadas en las anteriores opciones que han sido seleccionadas como competencias del proyecto.

	Sesame	Factorial	OpenKM	iAccesos
Acceso desde Web	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
App para Android e IOS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Fichaje de entrada y de salida	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Disponibilidad del personal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Registro con geolocalización	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Control de horas extras	<input checked="" type="checkbox"/>			
Informes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

Integración con sistemas biométricos	<input checked="" type="checkbox"/>			
Alertas y notificaciones	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Gestión de ausencias y vacaciones	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Gestión de nóminas y documentos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Gestión de selección de personal		<input checked="" type="checkbox"/>		
Gestión de turnos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Portal del empleado		<input checked="" type="checkbox"/>		
Puntos de fichaje	<input checked="" type="checkbox"/>			
Organigrama de la empresa		<input checked="" type="checkbox"/>		
Precio	Empleado/mes	Empleado/mes	Gratuito	Empleado/año

Mesa 7: Tabla comparativa de servicios de control horario

Aunque existen herramientas en el mercado que cubren las mismas necesidades que Laboreo, estas son opciones de pago con un coste mensual por empleado. Además, al tratarse de herramientas que están en la nube, la información está gestionada por las organizaciones que ofrecen el servicio y, por lo tanto, dependen del correcto funcionamiento que ofrezcan. La ventaja de este tipo de sistemas es la despreocupación de mantenimiento de los equipos que ofrecen estas empresas, dejando toda la seguridad en sus manos. Por el contrario, si no se elige una empresa lo suficientemente estable y solvente, puede producirse que dejen de ofrecer el servicio en cualquier momento, con las consecuencias que ello tendría.

Las principales ventajas de Laboreo sobre el resto de las herramientas que hay en el mercado son:

1. El sistema es gratuito, y no tiene coste adicional por su uso ni por el número de usuarios que lo utilicen.
2. La información se almacena en una base de datos propia, pudiendo realizar consultas desde otros sistemas externos y realizar integraciones si fueran necesario.
3. Su capacidad de adaptabilidad y de mejora al tratarse de un proyecto de código libre.
4. La compatibilidad con cualquier dispositivo al no tener que realizar ningún tipo de instalación para su funcionamiento.

### 2.3. Análisis DAFO

	INTERNO	EXTERNO
NEGATIVO	<p>Disponer y mantener un servidor web.</p> <p>La integración con los <i>gadgets</i> del terminal es limitada.</p> <p>Tiempo limitado para el desarrollo.</p>	<p>Gran competencia en el mercado.</p> <p>Producto no es rentable.</p> <p>Baja demanda.</p>
POSITIVO	<p>Proyecto de código libre.</p> <p>Aplicación gratuita.</p> <p>Compatibilidad con todo tipo de terminales.</p>	<p>Aumento en la digitalización en organizaciones.</p> <p>Reducción en los costes.</p> <p>Programación colaborativa.</p>

Mesa 8: Análisis DAFO

## 3.Propuesta

### 3.1. Definición de objetivos/especificaciones del producto

A continuación, pasaremos a describir los objetivos y funcionalidades de la aplicación:

- Ofrecer la opción de definir el organigrama de la organización.
- Gestionar datos del personal de la organización.
- Generar los posibles turnos de trabajo.
- Realizar el registro horario con las entradas y salidas de las jornadas laborales.
- Ofrecer información extra sobre las jornadas realizadas (minutos transcurridos, minutos restantes, minutos extras, etc.)
- Registrar el lugar de trabajo (oficina, teletrabajo, casa cliente, exteriores, etc.)
- Gestionar el calendario de vacaciones, a partir de solicitudes por parte del personal y la posible confirmación de la empresa.
- Conocer la disponibilidad del personal en tiempo real.
- Aportar una serie de informes de utilidad para la organización, para obtener estadísticas sobre las ausencias de los empleados, las horas extras realizadas, etc.

## 4. Diseño

### 4.1. Arquitectura general de la aplicación/sistema/servicio

La arquitectura general de la aplicación se compone de tres niveles o capas:

- Base de datos. Consiste en el almacenamiento de la información.
- *Back-end*. Es la capa de acceso a la base de datos persistentes mediante peticiones a la base de datos. En esta capa se asegura que la funcionalidad con el sistema sea correcta. Es una capa que no es visible para el usuario que interactúa con la aplicación.
- *Front-end*. Es la capa con la que el usuario interactúa con el sistema, y por lo tanto, es la parte visual (diseño y elementos gráficos).

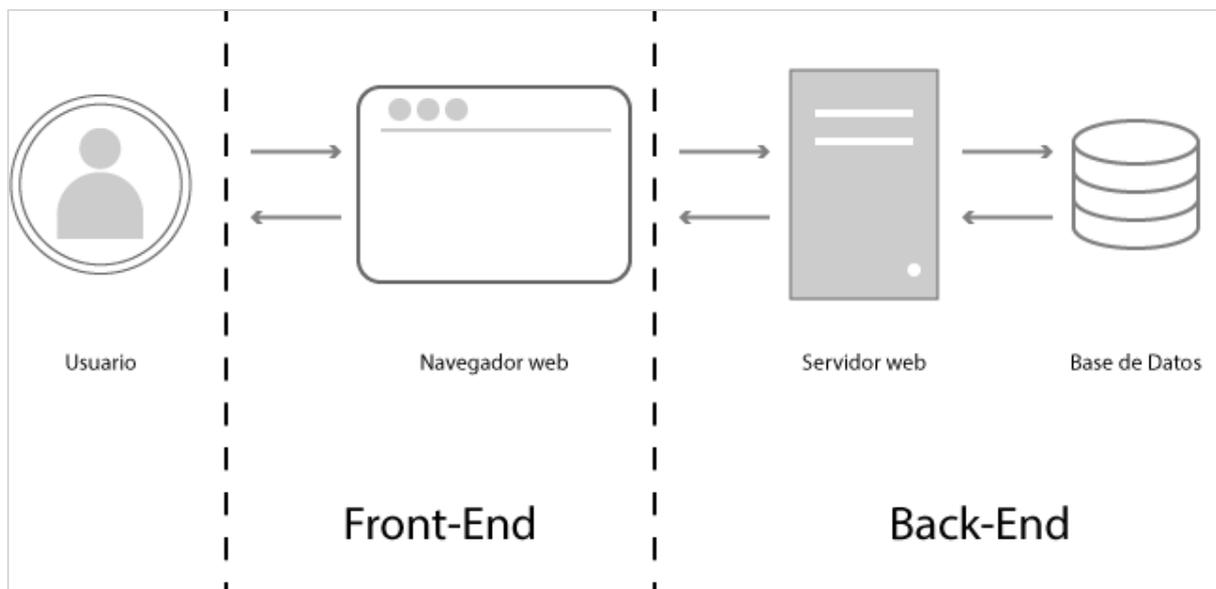


Figura 10: Arquitectura general del proyecto

### 4.2. Arquitectura de la información y diagramas de navegación

#### 4.2.1. Almacenamiento de la información

El sistema utiliza como sistema de almacenaje de la información una base de datos relacional PostgreSQL. El modelo de diseño utilizado se basa en la arquitectura cliente-servidor, donde la parte conectada con el cliente realiza peticiones al programa, y la parte del servidor responde devolviendo los datos al cliente.

El sistema de base de datos relacional seleccionado utiliza el lenguaje SQL<sup>6</sup> que permite el manejo completo de los datos almacenados. Este lenguaje permite realizar operaciones de creación de tablas para el almacenamiento de los datos de forma persistente, como la ejecución de consultas de información y la realización de operaciones de inserción, modificación o de borrado de datos.

Tras desarrollar el modelo ER<sup>7</sup>, se definen las siguientes tablas físicas para cubrir con las necesidades del proyecto.

Tabla	Descripción
<b>Organizacion</b>	Almacenamiento de datos de la organización. Es una tabla de un único registro para identificar a la organización que gestiona la aplicación. Dispone de atributos como el nombre, el cif, el registro mercantil y la posibilidad de subir una imagen del logo. <i>idOrganizacion, nombre, logotipo, cif, registro</i>
<b>CentroTrabajo</b>	Esta tabla contiene los diferentes centros de trabajo de los que se compone una organización. Pueden ser delegaciones, oficinas, almacenes, etc. Los atributos para identificar el centro de trabajo son el código identificativo, el nombre, y los datos de la dirección completa y datos de contacto. <i>idCentroTrabajo, idOrganizacion, código, nombre, dirección, codigoPostal, población, país, telefono</i> <i>{idOrganizacion} REFERENCES Organizacion (idOrganizacion)</i>
<b>Departamento</b>	Almacena los diferentes departamentos de los que se compone un centro de trabajo, como, por ejemplo, contabilidad, facturación, comercial, etc. Los campos habilitados son el código identificativo y su nombre. <i>idDepartamento, idCentroTrabajo, código, nombre, idEmpleado_responsable</i> <i>{idCentroTrabajo} REFERENCES CentroTrabajo (idCentroTrabajo),</i> <i>{idEmpleado_responsable} REFERENCES Empleado (idEmpleado)</i>
<b>Convenio</b>	En esta tabla se registran los diferentes tipos de convenios de trabajo con los que la organización va a trabajar. En ella se indican el nombre, una descripción breve, el número de días naturales y laborables de vacaciones totales por año y el total de horas de una jornada. <i>idConvenio, nombre, descripcion, diasNaturalesVacaciones,</i>

<sup>6</sup> SQL (Structure Query Language) es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Fuente Wikipedia. Disponible en: <https://es.wikipedia.org/wiki/SQL>.

<sup>7</sup> El modelo entidad-relación es una herramienta para el modelo de datos que facilita la representación de entidades de una base de datos. Fuente Wikipedia. Disponible en: [https://es.wikipedia.org/wiki/Modelo\\_entidad-relación](https://es.wikipedia.org/wiki/Modelo_entidad-relación).

	<i>diasLaborablesVacaciones, minutoJornada</i>
<b>TipoContrato</b>	<p>En este caso, se almacenan los diferentes tipos de contratos con lo que la organización tiene a sus empleados en nómina. Se identifican por el nombre.</p> <p><i>idTipoContrato, nombre</i></p>
<b>Turno</b>	<p>En esta entidad se almacenan los diferentes tipos de turnos. Permite realizar la configuración marcando los diferentes días de la semana y su horario de inicio y fin.</p> <p><i>idTurno, código, nombre, lunes, martes, miércoles, jueves, viernes, sábado, domingo, horaInicio, horaFin</i></p>
<b>RolUsuario</b>	<p>Esta tabla contiene los diferentes tipos de roles que puede tener un usuario que accede al sistema. Existe el rol de administrador, de supervisor y de usuario común.</p> <p><i>idRolUsuario, nombre, administrador, supervisor</i></p>
<b>Empleado</b>	<p>En la tabla de obserEmpleado se almacenan las fichas de los empleados. La organización puede añadir el nombre, el dni, datos de contacto (teléfono y email), número de la seguridad social, la dirección completa, la foto y el pin de acceso para el usuario. Además, se informa del convenio que tiene asociado el empleado, a qué departamento pertenece y cuál es el rol de trabajo con la aplicación.</p> <p><i>idEmpleado, idOrganizacion, nombre, apellidos, dni, teléfono, email, observaciones, pin, foto, numeroINSS, dirección, codigoPostal, población, país, idConvenio, idDepartamento, idContratoEmpleado_actual, idRolUsuario</i></p> <p><i>{idOrganizacion} REFERENCES Organizacion (idOrganizacion),</i>  <i>{idConvenio} REFERENCES Convenio (idConvenio),</i>  <i>{idDepartamento} REFERENCES Departamento (idDepartamento),</i>  <i>{idContratoEmpleado_actual} REFERENCES ContratoEmpleado (idContratoEmpleado),</i>  <i>{idRolUsuario} REFERENCES RolUsuario (idRolUsuario)</i></p>
<b>Jornada</b>	<p>En este caso, se almacenan las diferentes jornadas que tiene un empleado a lo largo de su vida laboral en la organización. No pueden existir dos registros que se solapen.</p> <p><i>idJornada, idEmpleado, idTurno, fechaInicio, fechaFin</i></p> <p><i>{idEmpleado} REFERENCES Empleado (idEmpleado),</i>  <i>{idTurno} REFERENCES Turno (idTurno)</i></p>
<b>Evento</b>	<p>Es una tabla abstracta para almacenar los eventos asociados a los empleados. Tiene una fecha de inicio y una fecha de fin del evento.</p>

	<p><i>idEvento, idEmpleado, fechaInicio, fechaFin</i>  <i>{idEmpleado} REFERENCES Empleado (idEmpleado)</i></p>
<b>Festivo</b>	<p>Es un evento de tipo festivo que sirve para almacenar los días festivos de los empleados. El empleado realiza una petición que posteriormente será confirmada por el empleado responsable del departamento.</p> <p><i>idEvento, idEmpleado, fechaInicio, fechaFin, autorizada, fechaAutorizacion, motivoRechazo, idEmpleado_autoriza</i>  <i>{idEmpleado} REFERENCES Empleado (idEmpleado),</i>  <i>{idEmpleado_autoriza} REFERENCES Empleado (idEmpleado)</i></p>
<b>Ausencia</b>	<p>Permite almacenar las ausencias de los empleados, indicando el motivo y el número de minutos que afecta.</p> <p><i>idEvento, idEmpleado, fechaInicio, fechaFin, motivo, minutos</i>  <i>{idEmpleado} REFERENCES Empleado (idEmpleado)</i></p>
<b>RegistroHorario</b>	<p>Esta especialización del evento sirve para almacenar las entradas y salidas para el control horario de los empleados. Adicionalmente se almacena la longitud y latitud según la geolocalización del terminal de acceso, y la fecha en la que se ha registrado el evento.</p> <p><i>idEvento, idEmpleado, fechaInicio, fechaFin, entrada, longitud, latitud, fechaRegistro</i>  <i>{idEmpleado} REFERENCES Empleado (idEmpleado)</i></p>
<b>Personal</b>	<p>Se trata de un evento personal del empleado, para que pueda anotar eventos que le afecten de forma personal. Puede registrar tareas de trabajo, recordatorios, etc. Consta de un título y una descripción.</p> <p><i>idEvento, idEmpleado, fechaInicio, fechaFin, titulo, descripcion</i>  <i>{idEmpleado} REFERENCES Empleado (idEmpleado)</i></p>
<b>Aviso</b>	<p>En esta tabla se almacenan los avisos que son de interés para el empleado y que se generan de forma automática según ciertas actividades. Por ejemplo, al confirmarse un evento de festivo.</p> <p><i>idAviso, idEmpleado, titulo, descripcion, leído</i>  <i>{idEmpleado} REFERENCES tEmpleado (idEmpleado)</i></p>
<b>CalendarioEmpleadoV</b>	<p>Esta vista permite tener una visión general de los eventos que afectan a un empleado. Sirve para poder ver la información de sus jornadas programadas, sus registros horarios, sus vacaciones y sus ausencias. No almacena nada de forma persistente.</p> <p><i>idEmpleado, fechainicio, fechafin, eventoNombre, eventoTipo</i>  <i>{idEmpleado} REFERENCES Empleado (idEmpleado)</i></p>



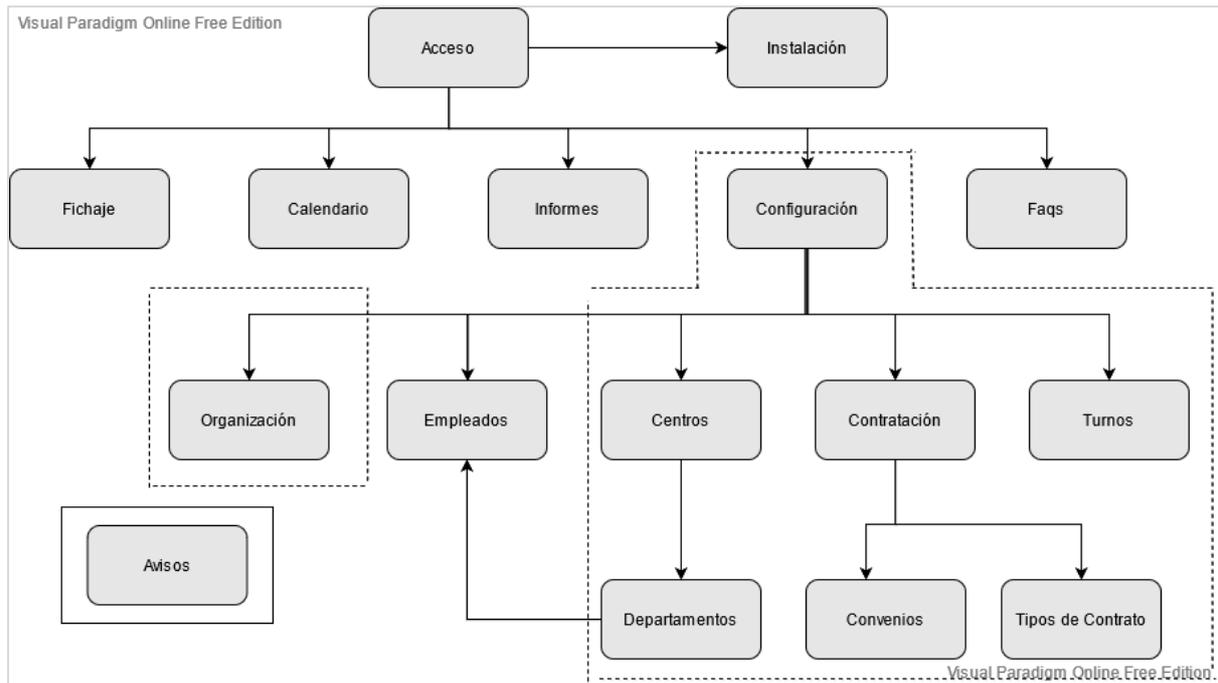


Figura 12: Árbol de navegación

### Acceso

Es la página principal de la aplicación y permite el acceso al sistema mediante un usuario y una contraseña. En el caso de acceder por primera vez (no existe la base de datos), se accede al apartado de instalación.

### Instalación

Este apartado se ejecuta automáticamente la primera vez que se accede y lanza un asistente de instalación de la aplicación.

### Fichaje

Desde la opción de fichajes el usuario puede registrar las entradas y salidas de su puesto de trabajo.

### Calendario

En esta opción se muestra un calendario por día, semana o mes, donde se reflejan datos relevantes para el usuario como los días de vacaciones o los turnos asignados. Según el rol del usuario la información mostrada puede ser mayor, mostrando datos de varios empleados a la vez.

### Informes

La opción de informes proporciona una serie de listados de utilidad al usuario. Los listados pueden variar según el rol asignado.

## **Configuración**

En este apartado se puede consultar información sobre la organización. Según el rol asignado al usuario algunas opciones estarán habilitadas o no, para poder realizar todas las tareas de mantenimiento. El usuario raso sólo tiene acceso a información relativa.

## **Personal**

Dentro del apartado anterior existe la sección que permite gestionar las fichas del personal. Desde aquí, el responsable puede realizar todo el mantenimiento de las fichas.

## **Centros**

Esta sección permite confeccionar la estructura de la organización, es decir los centros de trabajo y los departamentos que lo componen. Es una sección que solo será visible por ciertos roles.

## **Departamentos**

Dentro de los centros es posible gestionar los diferentes departamentos de los que se componen los centros. Sirve para gestionar los empleados

## **Contratación**

En este apartado se muestran las configuraciones relacionadas con las contrataciones de los empleados. En ella se pueden definir los turnos, los convenios de trabajo y los tipos de contratos.

## **Turnos**

Desde este apartado se pueden configurar los diferentes tipos de turnos de trabajo para asignarlos a los trabajadores.

## **Convenios**

Desde esta sección se definen los convenios de los trabajadores, donde se detalla el número de días de vacaciones y las horas máximas de las jornadas.

## **Tipos de contrato**

Esta opción permite dar de alta los diferentes tipos de contratos que la organización utiliza con los empleados.

## **Faqs**

Muestra una página con las preguntas y respuestas más habituales sobre el funcionamiento de la aplicación.

## Avisos

Por último, la sección de avisos permanecerá visible para el usuario en todo momento y permite visualizar los avisos que se generan automáticamente desde la organización (cambios de turnos, confirmación de solicitudes, etc.).

## 4.3. Diseño gráfico e interfaces

### 4.3.1. Estilos

#### Nombre comercial y logotipo

LABOREO surge de las siglas LAWOREHO, acrónimo de LAboring WOrker REgister HOurs, pero adaptándola a la ortografía, fonética y morfología del castellano, que hace que suene como el presente de indicativo del verbo laborear<sup>8</sup>.

Para diseñar el símbolo se ha realizado una interpretación muy simplificada de un reloj de fichaje que marca las ocho en punto. Se ha coloreado uno de los sectores en color amarillo que simboliza energía, luminosidad y también cierta similitud con el oro, y, por lo tanto, con las ganancias económicas. El diámetro de la circunferencia es de 3'3 unidades, y con un borde en color negro fino.

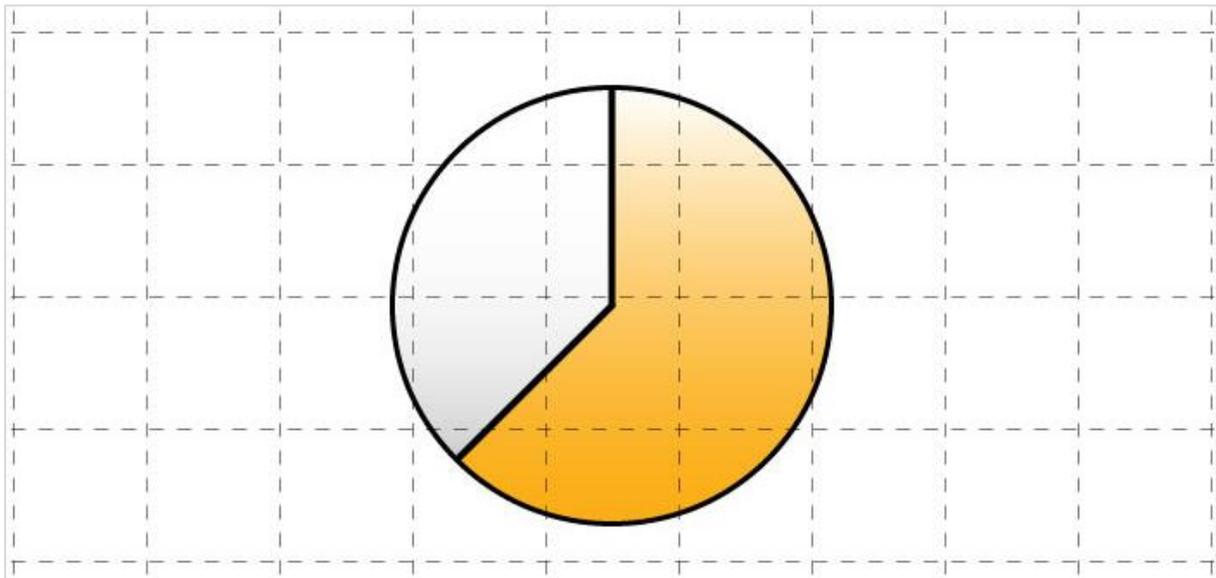


Figura 13: Símbolo de Laboreo

Para la creación del logotipo se ha utilizado la fuente *LuzSans-Book*<sup>9</sup> en mayúsculas y en color azul combinándolo con el símbolo anterior. Se sustituye la primera “O” por el símbolo, para crear un logotipo

<sup>8</sup> Labrar o trabajar algo. (<https://dle.rae.es/laborear>)

que permita relacionar el nombre con el símbolo. En el caso de tener que utilizar un fondo negro, se opta por cambiar el color de fondo de las letras y los bordes por blanco.

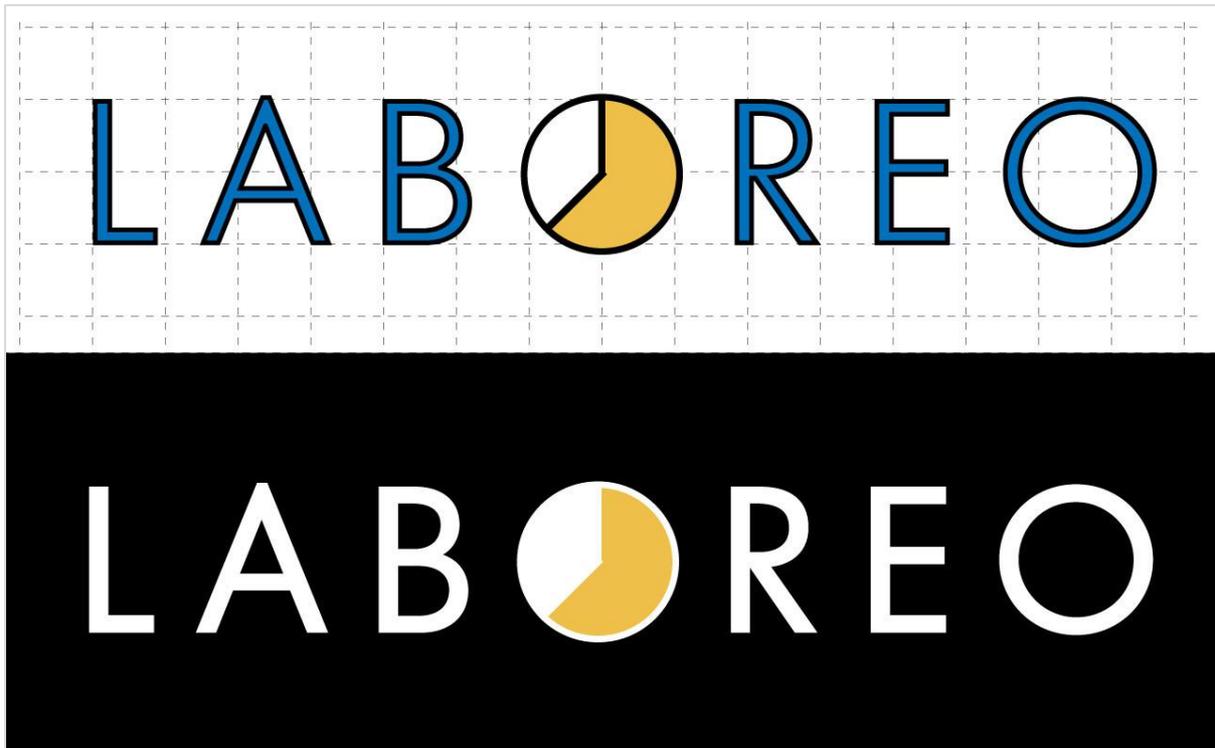


Figura 14: Logotipo de Laboreo

---

<sup>9</sup> Fuente de palo seco o sin gracias.

LuzSans-Book - Regular

abcdefghijklmnopqrstuvwxyz  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 1234567890,.-?¿!;{}[]+\*/



Hexadecimal:  
#0871B8

CMYK  
C=87 M=49 Y=0 K=0

RGB:  
R=8 G=113 B=184



Hexadecimal:  
#EFBF48

CMYK  
C=7 M=26 Y=79 K=0

RGB:  
R=239 G=191 B=72

Mesa 10: Tipografía y paleta de colores del logotipo

### Icono (favicon)

La siguiente figura muestra el icono utilizado para la aplicación web.

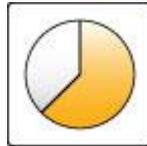


Figura 15: Icono

### Fuentes tipográficas

Para la tipografía del proyecto web se ha reducido el número de fuente a un total de dos. El motivo principal es el de simplificar el código y permitir la mayor compatibilidad posible con todos los navegadores web. La primera de ellas es una de las más utilizadas en páginas web y muy similar a la del logotipo, y se utiliza para los títulos de la aplicación principalmente. La segunda de ellas es la fuente más común y que se utiliza para el resto de los apartados de la aplicación web.

Calibri

abcdefghijklmnopqrstuvwxyz  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 1234567890,.-?¿!;{}[]+\*/

Arial

abcdefghijklmnopqrstuvwxyz  
 ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 1234567890,-.?¡;{}[]+\*/

Mesa 11: Tipografía utilizada en la aplicación

**Pautas y muestras de colores**

Los colores elegidos para la aplicación no son colores muy llamativos para centrarse más en la funcionalidad que en la estética. Así pues, se han elegido los colores: negro, blanco y dos tonalidades de grises. Aun así, se han conservado los colores del logotipo para ciertas partes del aplicativo.

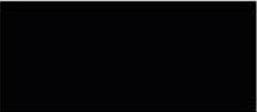
	Hexadecimal: #0871B8	CMYK C=87 M=49 Y=0 K=0	RGB: R=8 G=113 B=184
	Hexadecimal: #EFBF48	CMYK C=7 M=26 Y=79 K=0	RGB: R=239 G=191 B=72
	Hexadecimal: #BCBAB8	CMYK C=29 M=23 Y=24 K=4	RGB: R=188 G=186 B=184
	Hexadecimal: #828282	CMYK C=49 M=38 Y=38 K=19	RGB: R=130 G=130 B=130
	Hexadecimal: #FFFFFF	CMYK C=0 M=0 Y=0 K=0	RGB: R=255 G=255 B=255
	Hexadecimal: #000000	CMYK C=91 M=79 Y=62 K=97	RGB: R=0 G=0 B=0

Figura 16: Paleta de colores

## Fuentes, iconos, botones y otros elementos gráficos

En cuanto a la selección de los iconos y botones para la aplicación, se utilizan los iconos que ofrece *Fontawesome*<sup>10</sup> con licencia Creative Commons. Esta librería permite integrar en cualquier proyecto web, de forma rápida y sencilla, una serie de iconos que se ajustan a tamaños y colores deseados.

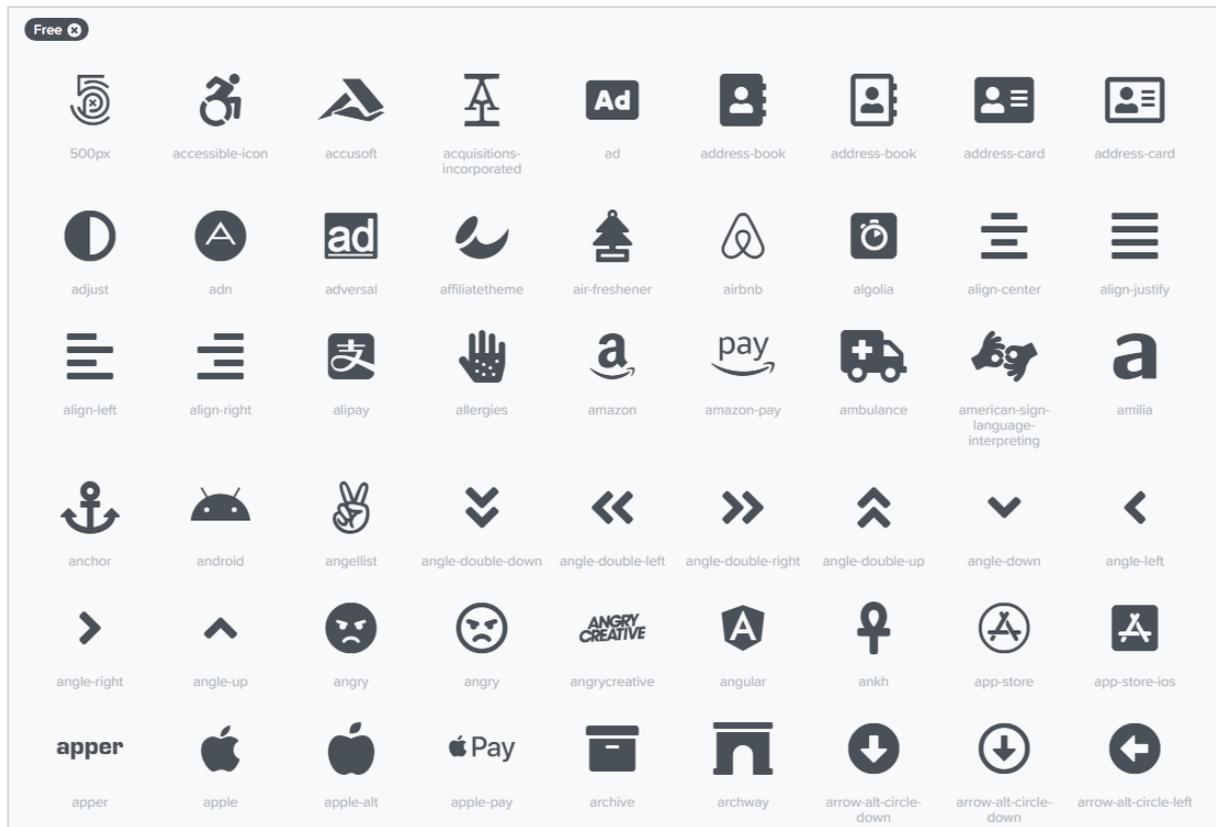


Figura 17: Ejemplo de iconos de (Font Awesome)

### 4.3.2. Usabilidad /UX

Para hacer que la aplicación web sea de fácil uso e intuitiva, se han seguido los 10 principios básicos de usabilidad web que redactó Jakob Nielsen<sup>11</sup>.

1. **Visibilidad del estado del sistema.** En la aplicación, las acciones intentan devolver siempre una página para indicar el estado en el que se encuentra la petición lanzada al sistema, como barras de progreso y las barras de navegación o *breadcrumbs*<sup>12</sup>.
2. **Relación entre el sistema y el mundo real.** El sistema utiliza un lenguaje conciso, claro y familiar para que sea comprensible para cualquier usuario, mayor de edad. Los iconos para identificar las acciones son los habituales en la mayoría de los sistemas actuales, pero además se ha incluido texto para facilitar su comprensión.

<sup>10</sup> Librería disponible en: <https://fontawesome.com>

<sup>11</sup> Los 10 principios básicos de Jakob Nielsen:

<https://www.nngroup.com/articles/ten-usability-heuristics/>

<sup>12</sup> Wikipedia: [https://es.wikipedia.org/wiki/Miga\\_de\\_pan\\_\(informática\)](https://es.wikipedia.org/wiki/Miga_de_pan_(informática))

3. **Control y libertad del usuario.** Todas las acciones están pensadas para que puedan deshacerse o modificarse.
4. **Consistencia y estándares.** Se han recogido los diferentes estándares utilizados en otras aplicaciones. Además, se intenta mantener una consistencia en todos los apartados y opciones para seguir una misma línea de edición que sea comprensible para el usuario.
5. **Prevención de errores.** Se introducen ciertas acciones de validaciones para evitar posibles errores humanos. Además, se ofrecen al usuario todas las opciones de forma que esto permita guiarlo en su trabajo diario.
6. **Reconocer antes que recordar.** Las acciones están pensadas para que sean reconocidas y evitar que los usuarios tengan que memorizar acciones para poder hacer su trabajo.
7. **Flexibilidad y eficiencia de uso.** Los procesos se han simplificado al máximo para adaptarse a cualquier tipo de usuario.
8. **Diseño estético y minimalista.** Para evitar distracciones al usuario se ha evitado sobrecargar el diseño y simplificarlo al máximo. Se ha creado un diseño limpio con colores sobrios.
9. **Ayudar a los usuarios a reconocer, diagnosticar y corregir los errores.** Se ha intentado encapsular todos los posibles errores para mostrar mensajes que sean de utilidad al usuario y ofrezcan algo más que un mensaje de error.
10. **Ayuda y documentación.** Se ha generado un pequeño manual de funcionamiento para indicar las acciones y posibilidades que tiene la aplicación.

Teniendo en cuenta todos estos principios, se ha creado una aplicación web que se asemeja a la mayoría de las aplicaciones que existen en la actualidad, para que de este modo los usuarios puedan hacer un uso de ella de forma intuitiva y sin necesidad de leer ningún manual de usuario.

#### 4.4. Lenguajes de programación y APIs utilizados

Laboreo es una aplicación pensada para que pueda ejecutarse desde un navegador web sin importar el tipo de dispositivo que se utilice. Es por este motivo que para poder desarrollarlo se necesita recurrir a un lenguaje de programación que pueda adaptarse perfectamente con HTML y que le proporcione funcionalidad e interactividad al usuario. Javascript ha evolucionado mucho en los últimos años y permite realizar tareas en el lado del servidor que antes sólo podían hacer ciertos lenguajes de programación como PHP. Por este motivo, se ha elegido este lenguaje de programación ya que es capaz de ejecutarse en la parte del cliente pero lanzando acciones que afectan al lado del servidor.

Gracias a esta nueva tecnología de Javascript la aplicación puede ser ejecutada en cualquier servidor web, sin importar cual sea su sistema operativo o el lenguaje de servidor instalado. Además, al ejecutarse en el lado del cliente, el uso de recursos en el servidor se reduce considerablemente, porque la mayor parte del renderizado y procesamientos se ejecutan en el dispositivo que lo ejecuta.

Inicialmente, para el desarrollo del proyecto se optó por elegir una plataforma Linux, en concreto el sistema Debian, por no disponer de licencias de pago para su uso. Posteriormente, para la publicación se ha elegido una plataforma como servicio de computación en la nube, que permite hospedar la aplicación web, además de ofrecer el servicio de base de datos con múltiples opciones. En el caso de Laboreo, PostgreSQL.

El almacenaje de la información se realiza en una base de datos relacional. Existen diferentes sistemas, pero continuando con el enfoque del código abierto, y que tenga una integración completa con todos los requisitos anteriores, PostgreSQL se ajusta perfectamente a las necesidades.

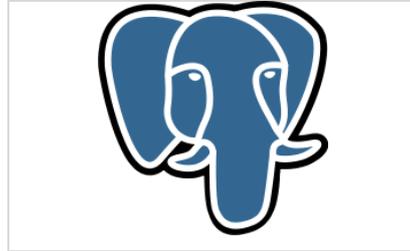


Figura 18: Logotipo de PostgreSQL. Fuente: (Wikipedia)

Para el desarrollo de las páginas web y para darle un aspecto más atractivo a la interfaz del usuario se han utilizado las hojas de estilo en cascada o CSS<sup>13</sup> que junto con el lenguaje Javascript hacen que la usabilidad y la interacción sean completas.

Para facilitar el desarrollo de la interfaz en los diferentes dispositivos (tamaños de pantallas) y hacer que la aplicación sea completamente *responsive*, se utiliza el conjunto de herramientas Bootstrap, que permite gestionar la estructura de la página web con una rejilla de 12 columnas para combinarse de diferentes formas y adaptarse a las diferentes resoluciones.

En cuanto a la creación de la estructura de la aplicación se ha utilizado NPM como sistema de gestión de paquetes por defecto para Node.js, en los entornos de ejecución para JavaScript. Esta aplicación permite tener el control de las dependencias de la aplicación con otros paquetes, y de esta manera, poder conocer los requisitos que la aplicación necesita para su desarrollo y su ejecución.

Algunas funcionalidades adicionales, como la geolocalización, necesitan de APIs externas para su correcto funcionamiento. En este caso, aunque existen APIs de Google, HTML5 incorpora en la mayoría de los nuevos navegadores una API de geolocalización que permite obtener la latitud y longitud del dispositivo que lo está ejecutando. Para localizar visualmente en un mapa las ubicaciones almacenadas en el sistema, se recurre a *OpenStreetMap*, que permite incrustar sus mapas con posiciones de forma fácil y sin licencias de uso.

En la construcción de la interfaz gráfica y para que se comporte como una aplicación nativa, además de facilitar el mantenimiento del código fuente, se utiliza el *framework* Vue.js que funciona con

---

<sup>13</sup> CSS (siglas en inglés de *Cascading Style Sheets*)

Javascript, que es de código abierto, y que permite, además, crear aplicaciones de una sola página<sup>14</sup> de forma rápida y sencilla.

Para poder hacer uso de este *framework* de forma completa, es necesario utilizar la herramienta de compilación Node.js junto con el sistema de gestión de paquetes (npm) por defecto de Node.js.



Figura 19: Logotipos de Vue.js y Node.js

En cuanto a la conectividad entre la base de datos y la interfaz gráfica, se ha realizado una API REST con Javascript, que son las encargadas de gestionar la interactividad entre el *front-end* y el *back-end*.

Para la programación del proyecto se ha utilizado Visual Studio Code<sup>15</sup> que permite una integración completa con los lenguajes y *frameworks* utilizados para el desarrollo. Para realizar las pruebas del funcionamiento de las API se ha utilizado Postman Agent<sup>16</sup>, que es una plataforma para desarrolladores de API que permite lanzar peticiones de todo tipo y obtener las respuestas.

#### 4.4.1. Back-end

El *back-end* se ha desarrollado en lenguaje Javascript a través de creación de un API REST que permite la interactividad entre la base de datos y el *front-end*. Este desarrollo es la base de la aplicación para poder almacenar y consultar los datos de la base de datos.

Para el intercambio de datos en el *back-end* se utiliza el formato JSON (*JavaScript Object Notation*) que es un formato independiente de los lenguajes de programación, muy simple, con un bajo peso para el intercambio de datos, y con una legibilidad para humanos muy grande, que permite el intercambio de datos entre el servidor y la página web (Shin, 2010).

---

<sup>14</sup> *Single-Page Applications* o SPA es una aplicación web que se ejecuta en una sola página para simular una aplicación de escritorio.

<sup>15</sup> Visual Studio Code. Editor de código. Disponible en: <https://code.visualstudio.com/>

<sup>16</sup> Postman Agent. Plataforma de desarrollo para API. Disponible en: <https://www.postman.com/>

```
[
  {
    "idcentrotrabajo": "1",
    "idorganizacion": "1",
    "codigo": "CEN",
    "nombre": "Central",
    "direccion": "Dirección de Central",
    "codigopostal": "00001",
    "poblacion": "Población de Central",
    "pais": "País de Central",
    "telefono": "(+99) 999 999 999"
  }
]
```

Figura 20: Ejemplo de formato JSON

#### 4.4.2. Front-end

Para el desarrollo del *front-end* se ha realizado el *framework* Vue.js, que permite la creación de interfaces de usuario, de una manera fácil de utilizar y de integrar con otras librerías, además de permitir la creación de proyectos *Single-Page Application*. Vue permite integrarse fácilmente en cualquier página web, pero con Vue-Cli es posible crear una estructura ordenada de aplicación para su desarrollo y posterior compilación con Node.js.

Esta herramienta permite separar la aplicación en diferentes “componentes” (archivos con extensión Vue) que pueden ser reutilizados y ejecutados en diferentes partes del programa.

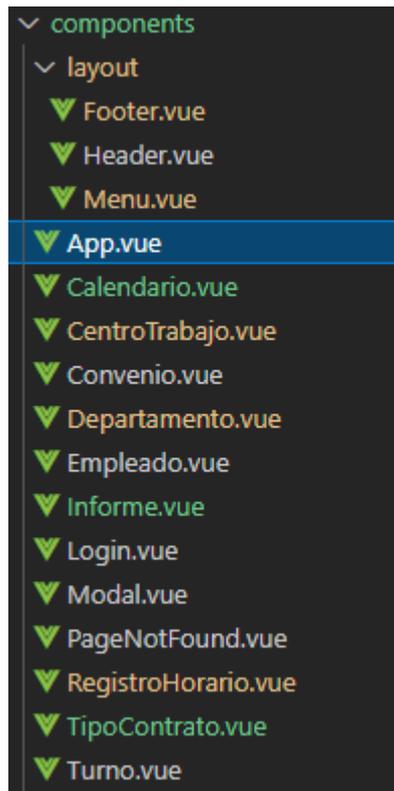


Figura 21: Ejemplo de estructura de componentes en Vue

De esta forma, la estructura del código de la aplicación queda separada y ordenada, permitiendo un mejor mantenimiento del código.

Los componentes de Vue tienen una estructura que permite separar, por un lado la plantilla HTML (*template*), de la interacción (*script*) y de los estilos.

```

<template>
  <div>
    <!-- Plantilla en HTML -->
  </div>
</template>
<script>
  // Scripts en javascript
</script>
<style scoped>
  /* Estilos */
</style>

```

Figura 22: Ejemplo de un componente de Vue

## 5. Implementación

### 5.1. Requisitos de instalación

#### 5.1.1. Requisitos para el servidor

Se necesita disponer de una estructura de servidores para publicar la aplicación en internet y para que sea accesible desde cualquier lugar. En el caso de realizar la instalación en un servidor propio de forma local, la aplicación requiere de los siguientes recursos necesarios para su correcto funcionamiento.

##### **Sistema operativo**

El proyecto está preparado para ser instalado en una plataforma Linux, ya que no requiere de licencias de pago para su uso. Entre los sistemas disponibles de la plataforma Linux se recomienda la instalación de *Debian* en su versión 10.8.0 AMD64x. Este sistema operativo dispone, en su instalación, de la posibilidad de incluir un servidor web Apache.

##### **Sistema de base de datos**

La aplicación necesita disponer de una base de datos para almacenar la información de forma persistente. Se apuesta por una base de datos relacional para poder cumplir con requisitos como la gestión de transacciones, la integridad referencial y la creación de procedimientos almacenados, entre otros. Entre las opciones disponibles, dado que se trata de un proyecto sin financiación, lo mejor es utilizar un gestor de base de datos que sea gratuito y libre. De entre todas las que cumplen con estos criterios, se apuesta por PostgreSQL por ser una herramienta completamente libre, de código abierto, y por tener un gran recorrido y experiencia.

##### **Servidor Web**

El servidor web Apache cumple con las necesidades del proyecto, que son ofrecer un servidor web.

#### 5.1.2. Requisitos para cliente

La aplicación web está pensada para que pueda ser ejecutada en cualquier plataforma sin necesidad de realizar ningún tipo de instalación.

## 5.2. Instalación en Heroku

Dada que la opción de disponer de todos los requisitos para la publicación de la app queda fuera del desarrollo de este proyecto, la implementación del proyecto se realiza en Heroku, que es una plataforma que ofrece un servicio de computación en la Nube y que soporta distintos lenguajes de programación.



Figura 23: Logotipo de Heroku. Fuente: (Wikipedia)

Este servicio permite, mediante la creación de una cuenta de usuario, implementar diferentes aplicaciones web, publicarlas y ejecutarlas desde sus servidores.

Para realizar la instalación de la aplicación web en los servicios de Heroku, es necesario disponer del código fuente del proyecto. En concreto se necesita la carpeta *src* del proyecto, junto con los ficheros *package.json* y *package-lock.json*. Estos dos últimos ficheros, permiten gracias a Node.js, instalar todos los módulos requeridos para el proyecto.

El siguiente comando lanza la instalación de todos los módulos requeridos por el proyecto según la información de los ficheros *package.json* y *package-lock.json*.

```
$ npm i
```

Una vez instalados los módulos, es necesario configurar los datos de acceso a la base de datos. Dentro del directorio *src/routes/database* existe el fichero *pool.js* con los datos de conexión.

```
// Conexión con PostgreSQL
const Pool = require('pg').Pool;
const pool = new Pool({
  user: 'prbnnoeime1clv',
  host: 'ec2-52-21-153-207.compute-1.amazonaws.com',
  database: 'd12l13vokbc54e',
  password: 'cd6fbd126316451006956166d79a2dc9da5cef13d023f3cc9631dab14d829fc
0',
  port: 5432,
  ssl: {
    require: true, // This will help you. But you will see nwe error
    rejectUnauthorized: false // This line will fix new error
  }
});
```

```
module.exports = pool;
```

Los datos de conexión a la base de datos pueden ser proporcionados por Heroku si la base de datos se localiza en este servicio. En caso de que exista una base de datos en otro servidor, se deberán introducir, igualmente, los datos de los mismos.

Una vez modificados los datos, se debe realizar el despliegue en el servidor. En el caso de Heroku, es necesario realizar la instalación de Heroku CLI que permite realizar el despliegue mediante una serie de comandos que facilita en su página web.

The screenshot displays the Heroku deployment guide for Git. It is organized into several sections:

- Deploy using Heroku Git:** A note stating to use git in the command line or a GUI tool to deploy the app.
- Install the Heroku CLI:** Instructions to download and install the Heroku CLI, followed by a terminal command: `$ heroku login`.
- Create a new Git repository:** Instructions to initialize a git repository in a new or existing directory, followed by terminal commands: `$ cd my-project/`, `$ git init`, and `$ heroku git:remote -a laboreoappweb`.
- Deploy your application:** Instructions to commit code to the repository and deploy it to Heroku using Git, followed by terminal commands: `$ git add .`, `$ git commit -am "make it better"`, and `$ git push heroku master`.
- Existing Git repository:** Instructions for existing repositories to add the heroku remote, followed by the terminal command: `$ heroku git:remote -a laboreoappweb`.

A blue callout box contains a note: "You can now change your main deploy branch from 'master' to 'main' for both manual and automatic deploys, please follow the instructions [here](#)."

Figura 24: Ejemplo de cómo desplegar una app en Heroku. Fuente: (Heroku)

Gracias a este servicio, la aplicación de Laboreo ha podido ser publicada en el siguiente enlace:

<https://laboreoappweb.herokuapp.com/>

Al tratarse de una PWA, la aplicación puede “instalarse” en aquellos dispositivos desde donde se ejecute. En el caso de ejecutar la herramienta desde un ordenador de sobremesa, algunos navegadores como Google Chrome ofrecen la opción de instalación para que pueda ejecutarse como si fuera una aplicación de escritorio.

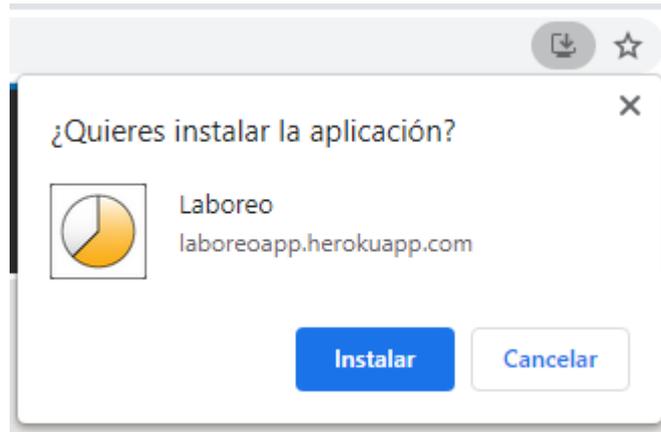


Figura 25: Instalación de la Laboreo desde Google Chrome.

Al instalar la aplicación, su ejecución se realiza en una ventana propia como si fuera una aplicación nativa. Este mismo proceso se puede realizar también en teléfonos inteligentes y tabletas.



Figura 26: Ejecución de la aplicación de forma nativa.

En la siguiente tabla se muestran los usuarios de prueba que pueden utilizarse para ver la funcionalidad según el rol asignado.

	<b>Administrador</b>	<b>Supervisor</b>	<b>Empleado</b>
<b>DNI/NIE</b>	ADMIN001	SUPER001	EMPL001
<b>Contraseña</b>	1234	1234	1234

Mesa 12: Tabla de usuarios de pruebas

## 6. Demostración

### 6.1. Prototipos

#### 6.1.1. Prototipos Lo-Fi

La estructura de la aplicación para *smartphone* se divide en tres secciones, una cabecera, un cuerpo y un pie. La versión para escritorio también se divide en tres secciones, pero con la particularidad de que el cuerpo se subdivide a su vez en tres más: un cuerpo central con el lateral izquierdo para el menú y el derecho para mostrar los registros.

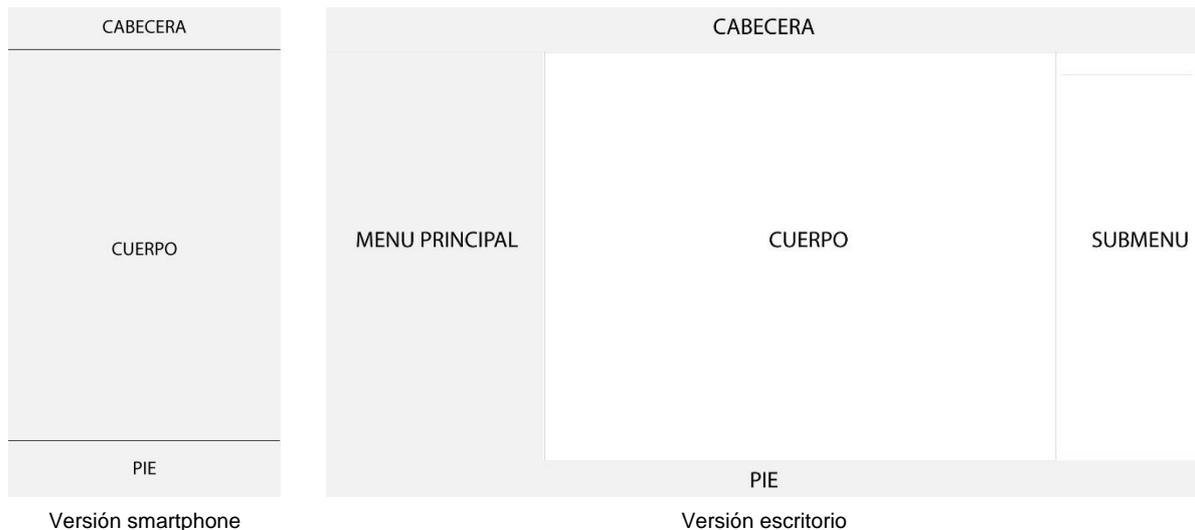


Figura 27: Estructura

A continuación, se adjuntan los *wireframes* Lo-Fi de las principales pantallas de trabajo en dos anchos de pantallas, para *smartphone* y para escritorio.

#### **Pantalla de acceso o de bienvenida para smartphone**

La pantalla principal o de acceso consiste en una pantalla de solicitud de usuario y contraseña. Esta pantalla tiene un diseño propio que no se corresponde con la estructura del resto de la aplicación.

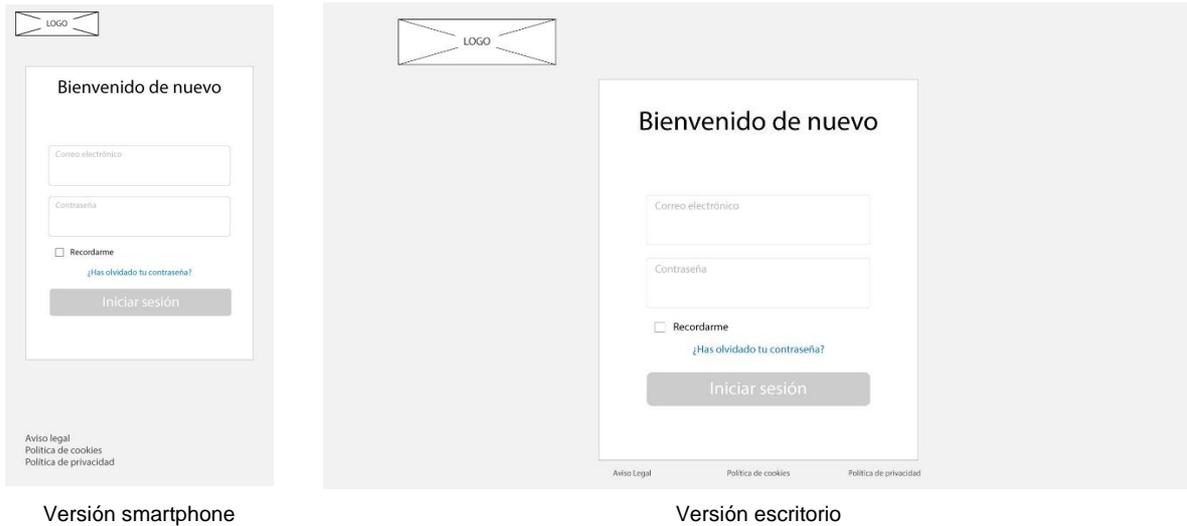


Figura 28: Página principal

### Menú principal

El menú principal aparece siempre visible para el usuario. En el caso de las pantallas reducidas, se ha simplificado en cuatro opciones, y el resto de las opciones se muestran en forma de listado al clicar sobre el botón de más opciones. Para el caso de escritorio, las opciones siempre están visibles en el panel del menú. No todas las opciones son accesibles y visibles para todos los perfiles de usuario. Solo los responsables y administradores tienen acceso a opciones de configuración.

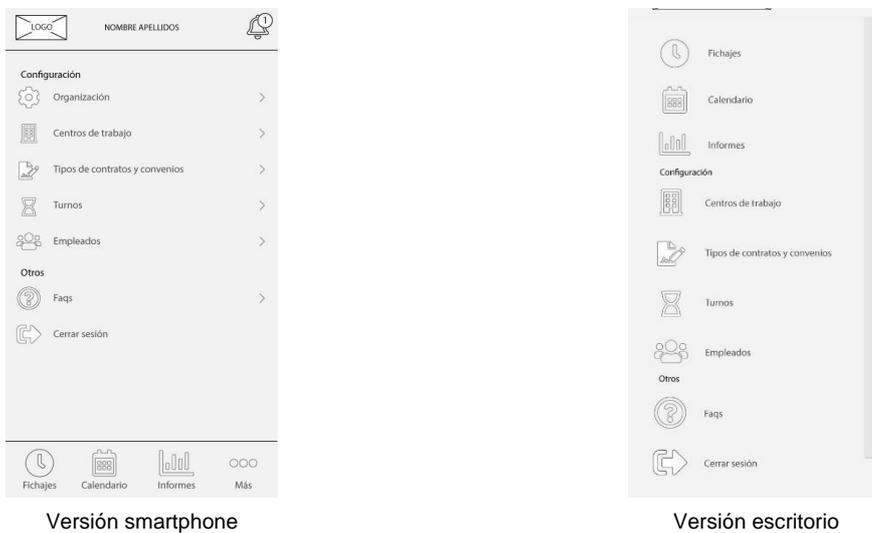


Figura 29: Menú

### Pantalla de fichajes

Al acceder a la aplicación se accede por defecto a la página de fichajes. Esta página sirve para gestionar las entradas y salidas del puesto de trabajo del usuario. En la pantalla se muestra una barra de progreso donde se indica la duración total de la jornada y el tiempo transcurrido. En la pantalla

central se muestra un botón de grandes dimensiones para realizar la acción correspondiente (entrada o salida), y también se muestra un pequeño listado con los últimos registros del día.



Figura 30: Página de fichajes

### Pantalla de calendario

Como se indica en el nombre se muestra un calendario mensual en el que se aparecen cada día los diferentes eventos que se han registrado. Para los usuarios con responsabilidad de departamento, la información que se muestra es la de todos los empleados del usuario responsable. Clicando sobre cada día se muestra el detalle de los eventos para esa jornada. Desde esta pantalla también se pueden registrar nuevos eventos (vacaciones, bajas, ausencias puntuales, etc.)



Figura 31: Página del calendario

### Pantalla de eventos

La pantalla de eventos corresponde con la edición de un evento del calendario. Se muestran en ella una serie de campos para rellenar o modificar.

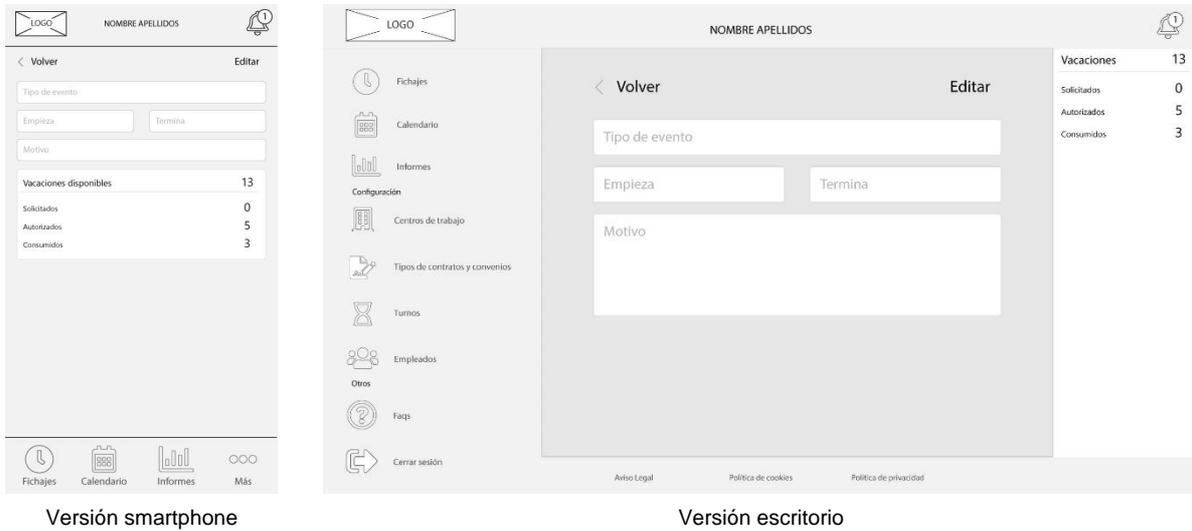


Figura 32: Página de eventos

### Pantalla de centros

La pantalla de gestión de centros se corresponde con un formulario de edición junto con el acceso a datos relacionados con los departamentos y empleados. El resto de las pantallas de configuración siguen esta misma estructura.



Figura 33: Página de centros de trabajo

### Pantalla de informes

La opción de informes muestra diferentes tipos de listados que se lanzan mediante una pantalla previa que permite filtrar ciertos datos.



Figura 34: Página de informes

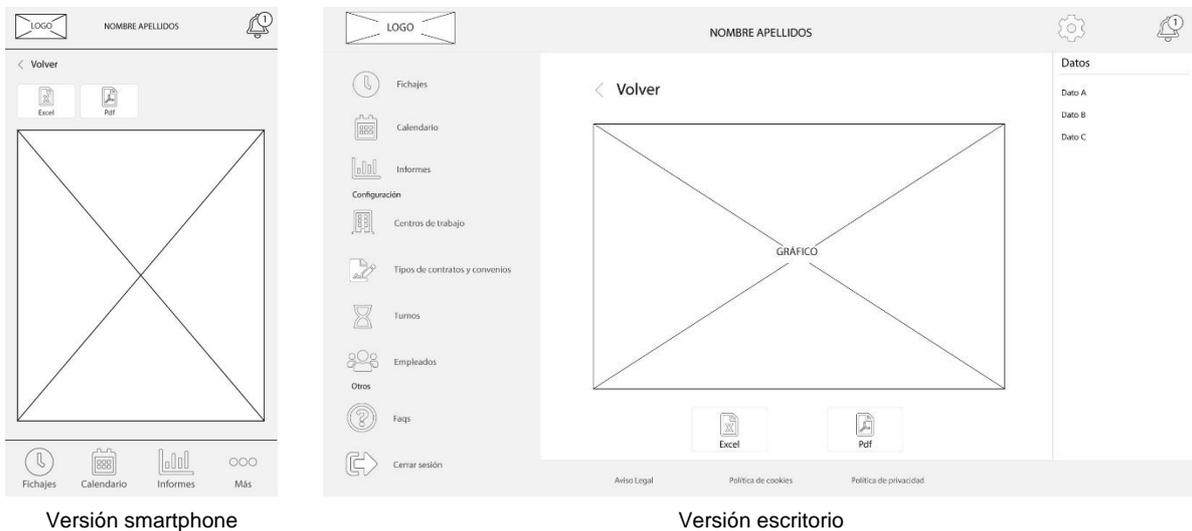


Figura 35: Página del listado

## 6.2. Test

### 6.3. Ejemplos de uso del producto (o guía de usuario)

El acceso a la aplicación se realiza mediante un formulario de inicio de sesión mediante un DNI/NIE del usuario y un pin.

Para el primer acceso, se puede utilizar el usuario ADMIN con el pin 2021, que permite acceder al sistema para poder gestionar los datos de la organización y dar de alta a los usuarios.



Versión smartphone



Versión escritorio

Figura 36: Inicio de sesión

Al acceder al sistema, por defecto se muestra la pantalla de fichajes, que permite registrar las entradas y salidas del puesto de trabajo.



Versión smartphone



Versión escritorio

Figura 37: Pantalla de registro de entradas y salidas

Mediante el menú, se pueden acceder a las diferentes opciones de la aplicación, entre las que se encuentran los mantenimientos de ciertos elementos como turnos de trabajo, contratos, convenios, centros de trabajo departamentos, etc.

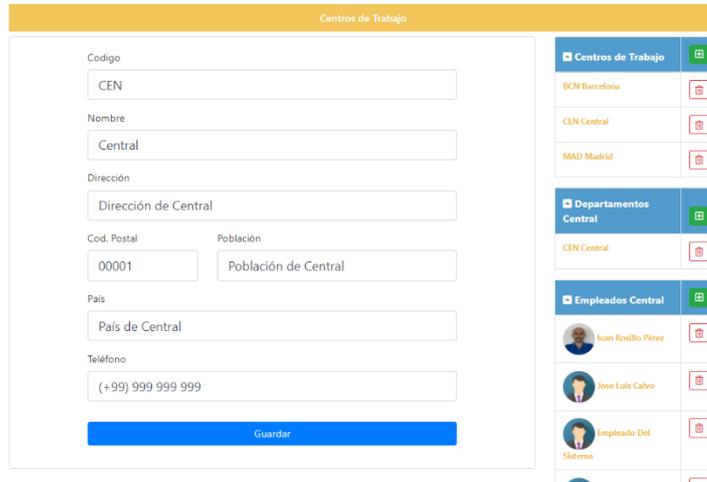
En la siguiente figura se puede ver un ejemplo de cómo es la gestión de este tipo de elementos. Hay que destacar también que todos los mantenimientos tienen la misma estructura.

En la versión para pantallas pequeñas, primeramente, se muestran los diferentes registros que existen en el sistema para poder ser consultados, y seguidamente se muestra el formulario. En la versión para

escritorio, en la parte izquierda se visualiza el formulario y a la derecha, los registros disponibles. Por defecto, el sistema entra en modo “alta” para insertar un nuevo registro.



Versión smartphone



Versión escritorio

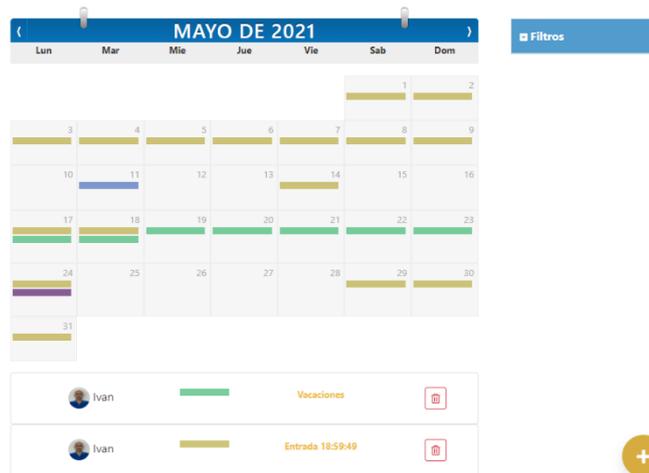
Figura 38: Pantalla de edición de registros

Entre las funcionalidades principales, se encuentra el calendario, que permite visualizar mes a mes, los diferentes eventos que el usuario tiene asignados. En el caso de los supervisores y administradores, el calendario ofrece información de todos los empleados.

Desde el calendario es posible visualizar eventos pasados y registrar los nuevos. La sección de filtros permite sesgar la información según el tipo de evento.



Versión smartphone



Versión escritorio

Figura 39: Pantalla del calendario

Por último, el apartado de Informes permite obtener diferentes listados que pueden ser de utilidad para la organización. Estos listados se muestran por pantalla, pero pueden ser lanzados a la impresora.

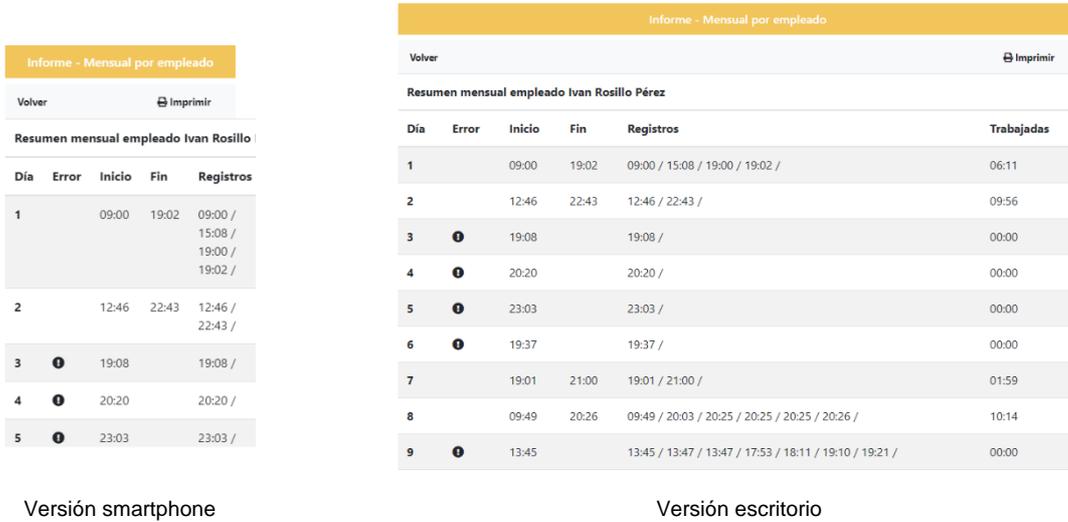


Figura 40: Pantalla de informes

## 7. Conclusiones y líneas de futuro

### 7.1. Conclusiones

El desarrollo de un proyecto de ingeniería web tiene una estructura con diferentes fases por las que hay que pasar para poder finalizarlo con éxito. Según las características del proyecto los pasos pueden variar, pero existen algunas fases que son comunes en la mayoría de los proyectos.

En este sentido, y tras finalizar este proyecto, una de las conclusiones que he podido extraer es la importancia que tiene realizar un detallado análisis inicial para poder organizar todos los contenidos y también para tener claro el punto de partida. La confección de un análisis de calidad, tanto en el ámbito de los requisitos que se deben cumplir, como en de las herramientas que se van a utilizar para desarrollarlo, son de suma importancia. Si durante el proceso de recogida de los requisitos, éstos no quedan claramente delimitados, es muy probable que, al finalizar el proyecto, la funcionalidad deseada no cumpla con las expectativas. Por otro lado, si las herramientas propuestas tampoco son las adecuadas, es posible que el coste del proyecto aumente considerablemente, provocando el fracaso del mismo.

En el caso del proyecto que nos ocupa, la idea inicial de crear una aplicación web que cubriera las necesidades básicas para una organización se han logrado. En este sentido, la primera versión es completamente funcional y cumple con la totalidad de los objetivos principales que se habían propuesto, como, por ejemplo, la creación de una aplicación adaptable a cualquier dispositivo o la creación de una PWA.

Quisiera destacar también que durante el desarrollo de esta aplicación he adquirido, en relación con la generación de aplicaciones web, nuevos conocimientos que han contribuido a que el programa mejore. Entre ellos, la idea de crear una aplicación SPA (*Single Page Application*), que consiste en la creación de una interfaz de usuario donde todas las pantallas del sistema se muestran en la misma página sin necesidad de recarga el navegador. Gracias a esta mejora, la experiencia del usuario con el programa es mucho más agradable y sus interacciones mucho más fluidas de forma que la sensación es la misma que usar una aplicación de escritorio.

El hecho de introducir estos cambios durante el proceso de desarrollo ha sido motivado por cumplir con los estándares de programación actuales. Si bien han supuesto un mayor esfuerzo en la creación de la aplicación, estas metodologías han permitido la integración fácilmente de otras librerías y *frameworks*, que mejoran y agilizan el desarrollo del código, además de ayudar a separar el *front-end* del *back-end*.

Por otra parte, la aplicación de este nuevo método de trabajo (prescindir de PHP y utilizar Javascript para la gestión del servidor), ha supuesto que no se haya seguido fielmente la planificación inicial del

proyecto y que se hayan incorporado otras tecnologías no consideradas al principio en el desarrollo del mismo. Si bien es cierto que la planificación sirve para tener una guía de los tiempos necesarios para cumplir con las entregas, en ocasiones, algunos factores pueden modificar los tiempos asignados a las diferentes fases. En el caso que nos concierne, aún con imprevistos, se ha cumplido los tiempos marcados, y el resultado obtenido es óptimo y satisfactorio.

Por último, el poder realizar este trabajo me ha servido, por un lado, para terminar de confirmar la complejidad que tiene el desarrollo de una aplicación web y, por otro, y más interesante, por los conocimientos adquiridos, para aprender nuevos métodos de desarrollo de aplicaciones, investigar y actualizarme con la creación de proyectos web. Definitivamente, finalizar este proyecto me ha hecho crecer como profesional y como estudiante.

## 7.2. Líneas de futuro

Esta primera versión de la aplicación se intentan cubrir las necesidades principales que se recogen en el punto 1.3 de esta memoria. La aplicación web se ha realizado teniendo en cuenta el análisis realizado y existen ciertas opciones que deben ser mejoradas en las siguientes versiones.

La primera versión de Laboreo cubre unas necesidades básicas para una organización y el control de sus empleados, pero el diseño de la aplicación está pensado para que pueda ser ampliado con nuevas opciones, e incluso mejorar las ya existentes. Como se ha indicado anteriormente, al tratarse de una aplicación con código abierto, cualquier organización podría coger el proyecto como base para poder crear una aplicación mayor y más completa. Al igual ocurre con el *front-end* de la app, que puede generarse de forma completamente nueva, ya que la interacción con la base de datos se realiza mediante consultas por API.

Algunas mejoras que podrían llevarse en las siguientes versiones son:

- Introducir mejoras en la seguridad para evitar ataques informáticos.
- Mejorar el diseño del *front-end* para hacerlo más intuitivo y fácil de usar tras recibir sugerencias y quejas por parte de los usuarios.
- Ampliar el modelo de las clases (tablas) para incluir nuevos atributos que proporcionen mejor información a la organización.
- Añadir más funcionalidades en el apartado de informes.
- Ampliar la sección de FAQs.
- Crear una gestión documental que permita adjuntar documentos en algunos de los procesos (añadir los contratos firmados por los empleados, incluir los registros mensuales verificados, etc.)
- Aumentar los comentarios en el código de la aplicación para ofrecer una mayor comprensión.

- Optimizar el código fuente, para crear más componentes que puedan ser reutilizados.
- Modificar la estructura de las carpetas para mejorar la organización de los archivos.

## Bibliografía

- Abuse, Stack. 2020.** Digital Ocean. Community. *Cómo usar módulos Node.js con npm y package.json*. [En línea] 15 de abril de 2020. [Citado el: 1 de junio de 2021.] <https://www.digitalocean.com/community/tutorials/how-to-use-node-js-modules-with-npm-and-package-json-es>.
- Arenzana, David. 2016.** Semrush Blog. *Principios de usabilidad web de Jakob Nielsen: diseño UX*. [En línea] 30 de Marzo de 2016. [Citado el: 18 de Marzo de 2021.] <https://es.semrush.com/blog/usabilidad-web-principios-jakob-nielsen/>.
- Arimetrics. Arimetrics. Glosario Digital.** [En línea] [Citado el: 5 de junio de 2021.] <https://www.arimetrics.com/glosario-digital/>.
- Azaustre, Carlos. 2017.** Youtube. *API REST con Node.js + Mongo*. [En línea] abril de 2017. [Citado el: 13 de mayo de 2021.] [https://www.youtube.com/watch?v=l03\\_z5DBgp4](https://www.youtube.com/watch?v=l03_z5DBgp4).
- cottonbro. 2020.** Pexels. *Moda hombre gente mujer*. [En línea] 11 de Julio de 2020. <https://www.pexels.com/es-es/foto/moda-hombre-gente-mujer-5137950/>.
- CSS Script. 2020.** CSS Script. *HTML & CSS Only FAQ Accordion*. [En línea] 28 de octubre de 2020. [Citado el: 24 de mayo de 2021.] <https://www.cssscript.com/html-faq-accordion/>.
- Fazt. Faztweb. Aprende a crear la web por ti mismo.** [En línea] <https://www.faztweb.com>.
- Font Awesome. Font Awesome. Font Awesome.** [En línea] <https://fontawesome.com>.
- Gardey, Julián Pérez Porto y Ana. 2015.** Definición de. [En línea] 2015. [Citado el: 5 de junio de 2021.] <https://definicion.de/>.
- Instituto Nacional de Estadística. 2020.** Encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares. *Año 2020*. [En línea] 16 de Noviembre de 2020. [Citado el: 5 de Marzo de 2021.] [https://www.ine.es/prensa/tich\\_2020.pdf](https://www.ine.es/prensa/tich_2020.pdf).
- Jiménez, Juan Diego Pérez. 2019.** OpenWebinars. *Qué es HTML5*. [En línea] 20 de enero de 2019. [Citado el: 5 de junio de 2021.] <https://openwebinars.net/blog/que-es-html5/>.
- Lily, Tiger. 2020.** Pexels. *Hombre tecnología efeco desenfocado en pie*. [En línea] 25 de Mayo de 2020. <https://www.pexels.com/es-es/foto/hombre-tecnologia-efecto-desenfocado-en-pie-4484078/>.
- Love, Chris. 2019.** Why Progressive Web Applications Are Not Single Page Applications. [En línea] 3 de octubre de 2019. [Citado el: 8 de abril de 2021.] <https://love2dev.com/blog/pwa-spa/>.
- MDN Web Docs. 2021.** Estructura de una aplicación web progresiva. [En línea] 8 de mayo de 2021.
- Ministerio de Industria, Comercio y Turismo.** Ministerio de Industria, Comercio y Turismo. *DAFO*. [En línea] [Citado el: 5 de junio de 2021.] <https://dafo.ipyme.org/Home>.
- Nori, Luigi. 2019.** ma-no. *Como Construir Una Api Restful - Guía Paso a Paso*. [En línea] 6 de septiembre de 2019. [Citado el: 17 de abril de 2021.] <https://www.ma-no.org/es/programacion/como-construir-una-api-restful-guia-paso-a-paso>.
- Oqotech Colombia. 2017.** Beneficios de la informatización y validación de procesos. [En línea] 2 de agosto de 2017. [Citado el: 26 de febrero de 2021.]

<https://www.oqotech.com/blog/informatizacion-de-procesos/beneficios-de-la-informatizacion-y-validacion-de-procesos/>.

**Rascia, Tania. 2020.** Node.js, Express.js, and PostgreSQL: CRUD REST API example. [En línea] 23 de diciembre de 2020. [Citado el: 17 de abril de 2021.]

<https://blog.logrocket.com/nodejs-expressjs-postgresql-crud-rest-api-example/>.

**Richelieu, August de. 2020.** Pexels. *Traje mujer escritorio oficina*. [En línea] 18 de Mayo de 2020.

**Sánchez, Kelvin. 2021.** Cómo construir una PWA desde cero con HTML, CSS y JavaScript. [En línea] 19 de enero de 2021. [Citado el: 8 de abril de 2021.]

<https://www.freecodecamp.org/espanol/news/como-construir-una-pwa-desde-cero-con-html-css-y-javascript/>.

**Shin, Sang. 2010.** Presentation [www.javapassion.com](http://www.javapassion.com). [En línea] 2010. [Citado el: 9 de mayo de 2021.] <http://www.cse.iitd.ac.in/~cs5090250/JSON.pdf>.

**Terron, Mario. 2017.** Como subir nuestra aplicación Node a Heroku. [En línea] 28 de mayo de 2017. [Citado el: 2 de mayo de 2021.]

<https://medium.com/@MarioTerron/como-subir-nuestra-aplicacion-node-js-a-heroku-7ca91595297f>.

**Wikipedia.** PHP. [En línea] [Citado el: 19 de Marzo de 2021.] <https://es.wikipedia.org/wiki/PHP>.

—. Servidor HTTP Apache. [En línea] [https://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache).

# Anexos

## Anexo A: Glosario

Glosario de terms y acrónimos utilizados en el trabajo (sólo aquellos mencionados en el presente documento) con breves definiciones de cada uno de ellos.

- **API:** Es una sigla de la expresión en inglés *Application Programming Interface*, que hace referencia a los procesos, funciones y métodos que ofrece una determinada biblioteca de programación a modo de capa de abstracción para que se empleada por otro programa informático.
- **AppWeb:** Es la abreviación para definir el término “aplicación web”.
- **CSS:** Las hojas de estilo (en inglés *Cascading Stylesheets*) se utilizan para dar estilos y posicionamiento visual al contenido de una página web.
- **DAFO:** El DAFO (iniciales de Debilidades, Amenazas, Fortalezas y Oportunidades) es una herramienta que permite al empresario analizar la realidad de su empresa, marca o producto para poder tomar decisiones de futuro.
- **Framework:** Un framework es un marco o esquema de trabajo utilizado por programadores para realizar el desarrollo de software.
- **HTML5:** Corresponde con la quinta revisión del lenguaje utilizado para el *World Wide Web*, conocido como *Hyper Text Markup Lenguaje*.
- **Javascript:** Es un lenguaje de programación que permite implementar funciones complejas en páginas web.
- **Plugins:** Se tratan de pequeños programas que complementan y amplían funcionalidades para las aplicaciones web.
- **Wireframe:** Se trata de un esquema a modo de guía visual para representar el esqueleto de un sitio web.

## Anexo B: Entregables del proyecto

Lista de archivos entregados y su descripción.

- Informe de autoevaluación. [*PEC\_FINAL\_infeval\_RosilloPerez\_Ivan.pdf*]
- Presentación en diapositivas del proyecto. [*PEC\_FINAL\_prs\_RosilloPerez\_Ivan.pptx*]
- Proyecto completo, con código fuentes y scripts de bases de datos. [*PEC\_FINAL\_prj\_RosilloPerez\_Ivan.zip*]

## Anexo C: Currículum Vitae

Iván Rosillo Pérez

[irosillo@uoc.edu](mailto:irosillo@uoc.edu)

### Formación académica

Grado en Multimedia

Universitat Oberta de Catalunya

Año: 2021

C.F.G.S. de Desarrollo de Aplicaciones Informáticas

I.E.S. Esteve Terrades i Illa

Año: 2002

### Experiencia laboral

*Abril 2012 – Actualidad.* Analista-programador en SaaSTi 2011, S.L.

- Análisis de requerimientos.
- Programación de aplicaciones en lenguaje Delphi con base de datos SQL Server.
- Mantenimiento y desarrollo de aplicaciones web en ASP y Javascript.
- Mantenimiento y desarrollo de webs en lenguaje PHP y CSS.

*Agosto 2002 – Enero 2012.* Programador junior en RAL Asistencia Informática, S.L.

- Helpdesk. Asistencia telefónica de las aplicaciones propias de la empresa.
- Análisis y desarrollo de nuevos módulos de aplicaciones.
- Programación de aplicaciones en lenguajes (Summer87/Clipper, Delphi).
- Manejador de Base de Datos (SQL Server, MySQL)
- Páginas Web. Diseño de aplicaciones informáticas en formato Web (HTML, JavaScript, ASP, PHP y JQuery entre otros)