



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

APLICACIÓN WEB PARA LA GESTIÓN DE UNA PYME DEL SECTOR DE LA CONSTRUCCIÓN

Alumno: Celia Macías Torres

Tutores: Prof. D. Miguel Ángel García Cumbreras
Prof^a. Dña. Salud María Jiménez Zafra

Dpto: Departamento de Informática

Septiembre, 2022



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don MIGUEL ÁNGEL GARCÍA CUMBRERAS y Dña. SALUD MARÍA JIMÉNEZ ZAFRA, tutores del Proyecto Fin de Carrera titulado: APLICACIÓN WEB PARA LA GESTIÓN DE UNA PYME DEL SECTOR DE LA CONSTRUCCIÓN, que presenta CELIA MACÍAS TORRES, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, SEPTIEMBRE de 2022

El alumno:

MACIAS
TORRES CELIA
- 77367311A
Firmado digitalmente por
MACIAS TORRES
CELIA - 77367311A
Fecha: 2022.09.09
16:24:39 +02'00'
CELIA MACÍAS TORRES

Los tutores:

GARCIA CUMBRERAS MIGUEL ANGEL - 52871272E 52871272E	Firmado digitalmente por GARCIA CUMBRERAS MIGUEL ANGEL - 52871272E Fecha: 2022.09.09 19:13:14 +02'00'	JIMENEZ ZAFRA SALUD MARIA - 77370897R 77370897R	Firmado digitalmente por JIMENEZ ZAFRA SALUD MARIA - 77370897R Fecha: 2022.09.09 18:47:29 +02'00'
---	--	---	--

MIGUEL ÁNGEL GARCÍA CUMBRERAS
SALUD MARÍA JIMÉNEZ ZAFRA

Resumen

En este trabajo de fin de grado se ha desarrollado una aplicación web para la gestión de una pyme del sector de la construcción. El objetivo principal del proyecto es la creación de una plataforma web para la gestión interna de la empresa PROYECTOS INTEGRALES JAENVIVIENDA SL. La aplicación consiste en un panel de administración para la gestión de usuarios y documentos.

Para el desarrollo de la aplicación, se ha utilizado la librería React js en el lado del cliente, y el framework Express de Node js en el lado del servidor. Se ha desarrollado localmente y para el diseño se ha utilizado la plantilla Lucid – Responsive ReactJS Admin Template, desarrollada en React js.

La gestión de archivos se ha implementado mediante la conexión con la API de Google Drive, para mantener su privacidad. En esta documentación se detalla en profundidad el proceso de diseño y desarrollo de la aplicación.

Abstract

In this final degree project, a web application has been developed for the management of an SME in the construction sector. The main objective of the project is the creation of a web platform for the internal management of the company PROYECTOS INTEGRALES JAENVIVIENDA SL. The application consists of an administration panel for the management of users and documents.

For the development of the application, the React js library has been used on the client side, and the Node js Express framework on the server side. It has been developed locally and the Lucid - Responsive ReactJS Admin Template, developed in React js, has been used for the design.

File management has been implemented by connecting to the Google Drive API, to maintain your privacy. This documentation details in depth the design and development process of the application.

Índice

Resumen	2
Abstract	2
1. INTRODUCCIÓN	7
1.1. Aspectos generales y motivación	7
1.2. Objetivos.....	8
1.3. Estudio de mercado	8
1.3.1. Factorial.....	9
1.3.2. GerSoft	11
1.3.3. Trello	13
1.3.4. Asana	14
1.3.5. Google Drive	16
1.3.6. Dropbox.....	18
1.4. Procedimiento a seguir en la elaboración de la plataforma	19
1.4.1. Estudio de mercado y selección de los usuarios	19
1.4.2. Estudio sobre los diferentes frameworks sobre programación.....	19
1.4.3. Diseño de la estructura de la plataforma	19
1.4.4. Implementación de la plataforma	21
1.4.5. Pruebas de usabilidad y validación	21
1.5. Estructura de la memoria	22
2. SOLUCIONES EXISTENTES EN EL MERCADO.....	23
2.1. Aplicaciones para la gestión de usuarios	23
2.2. Aplicaciones para la gestión de documentos	23
2.3. Aplicaciones para la gestión de usuarios y documentos.....	23
2.3.1. Dataprius	23
2.4. Conclusiones.....	27
3. POSIBLES TECNOLOGÍAS A UTILIZAR.....	28
3.1. Tecnologías frontend	28
3.1.1. HTML.....	28
3.1.2. CSS.....	28
3.1.3. Javascript.....	30
3.2. Frameworks y librerías frontend aplicables a nuestro proyecto	31
3.2.1. Angular.....	31

3.2.2.	React js	32
3.2.3.	Vue.js.....	34
3.3.	Tecnologías backend	35
3.3.1.	PHP.....	35
3.3.2.	Python.....	36
3.3.3.	Java EE	37
3.3.4.	Node.js.....	38
3.4.	Frameworks backend aplicables a nuestro proyecto	38
3.4.1.	Laravel.....	39
3.4.2.	Express	39
3.5.	Comparativa entre los frameworks estudiados	40
3.5.1.	Angular, React js y Vue.js.....	40
3.5.2.	Express y Laravel	41
3.6.	Conclusiones.....	43
4.	REQUISITOS DEL PROYECTO	44
4.1.	Planificación del trabajo	44
4.2.	Análisis de los costes.....	49
4.2.1.	Costes referentes al hardware.....	49
4.2.2.	Costes fijos	49
4.2.3.	Costes referentes al software	49
4.2.4.	Costes de personal	49
4.3.	Requisitos funcionales	50
4.4.	Requisitos no funcionales	55
5.	Diseño del proyecto	57
5.1.	Diseño arquitectónico	57
5.2.	Diagrama de casos de uso.....	59
5.3.	Diseño de Secuencia	62
5.4.	Modelo de datos.....	67
5.4.1.	Modelo Entidad-Relación	67
5.4.2.	Modelo Relacional	68
5.4.3.	Modelo Relacional MySQL	69
5.4.4.	Definición del modelo.....	70
5.5.	Storyboard	71
5.6.	Diseño web	71
6.	DESARROLLO DEL PROYECTO	81
6.1.	Estructura del código fuente	81

6.2.	Conexión de plantilla con backend.....	84
6.3.	Creación de inicio de sesión	85
6.4.	Creación de tablas de los usuarios en la aplicación.....	86
6.5.	Creación, edición y borrado de datos desde la web.....	89
6.6.	Asociación de tablas y vistas	92
6.7.	Conexión con API Google Drive.....	93
6.8.	Solicitud de archivos de Drive. Concesión de permisos	96
6.9.	Visualización de la interfaz según usuario	97
6.10.	Permisos y detalles de texto según usuario.....	99
6.11.	Testeo y corrección de errores	99
7.	EVALUACIÓN DEL PROYECTO	102
7.1.	Validación de los servicios	102
7.2.	Usabilidad.....	102
8.	CONCLUSIONES	105
9.	REFERENCIAS.....	108
	Bibliografía.....	108
	Anexo 1: Tablas de la base de datos.....	110
	Tabla de administradores.....	110
	Tabla de archivos.....	110
	Tabla de clientes	110
	Tabla de documentos	111
	Tabla de proyectos.....	111
	Tabla de trabajadores.....	112
	Anexo 2: Storyboard: Vistas en Marvell App	113
	Anexo 3: Manual de puesta en marcha.....	121
	Instalación de wampserver	121
	Creación la base de datos.....	121
	Creación de usuario de la base de datos	121
	Importar la base de datos	123
	Instalar Nodejs.....	123
	Descarga de aplicación	124
	Lanzamiento de aplicación	125
	Iniciar sesión.....	126
	Detener ejecución	126
	Anexo 4. Manual de usuario.....	127
	Inicio de sesión	127

Manual del Administrador.....	128
9.1.1. Manual del Cliente.....	135
9.1.2. Manual del trabajador	137

1. INTRODUCCIÓN

1.1. Aspectos generales y motivación

El desarrollo web cada vez es más requerido dentro de las empresas, no solo a nivel de ventas, sino a nivel de gestión interna. La automatización de procesos es uno de los factores más importantes de la digitalización. Muchas empresas están comenzando a automatizarse integrando sistemas de gestión interna para agilizar procesos y ahorrar tiempo y dinero.

En este proyecto se definirá el proceso de desarrollo de una “APLICACIÓN WEB PARA LA GESTIÓN DE UNA PYME DEL SECTOR DE LA CONSTRUCCIÓN”. Concretamente se trata de la empresa PROYECTOS INTEGRALES JAENVIVIENDA SL, localizada en Jaén capital. Esta empresa fue fundada en 2012 y surgió a raíz de la crisis del ladrillo de los años 2008-2011. En el año 2018, se constituye como sociedad, y se dedica al asesoramiento y realización de reformas e interiorismo. Ofrecen servicios de realización de planos, trámite de permisos, seguimiento de la obra y realización de presupuestos, entre otros y está formada por un grupo de profesionales experimentados en los oficios de fontanería, alicatado, albañilería y pintura.

Actualmente, la comunicación con sus clientes y empleados no está gestionada por ningún sistema web, ya que es una pequeña pyme que cuenta con un total de seis asalariados, y hasta el momento, la gestión de los empleados se llevaba a cabo de forma manual. Debido al volumen de proyectos en el último año, consideran que es el momento de automatizar la empresa, por lo que solicitan la creación de una plataforma de administración.

La plataforma que el administrador de la empresa solicita, consiste en una aplicación que gestione documentos y permita la comunicación de estos con clientes y trabajadores.

En el caso de los clientes, Jaenvivienda quiere agilizar el proceso de comunicación. Hasta el momento, la empresa informa de la evolución de sus obras por vía telefónica y por correo electrónico. La empresa necesita enviar archivos e

imágenes, para informar del estado actual de su trabajo. Por este motivo, en la aplicación se requiere un apartado de proyectos, que están asociados a un cliente.

En el caso de los trabajadores, la empresa quiere un sistema en el que pueda acceder fácilmente a los datos de sus trabajadores. Además solicita la subida de nóminas y contratos. Esto ayuda a que en cualquier momento tenga toda la información de su empresa en un clic, y a que el trabajador pueda acceder a sus documentos en cualquier momento.

1.2. Objetivos

Los objetivos de esta plataforma web son satisfacer los requisitos que exige el cliente. En términos generales, las funcionalidades que solicita el administrador de Jaenvivienda son las siguientes.

- Solicita un panel de administración con tres tipos de usuarios: administrador, cliente y trabajador. El administrador desea gestionar sus documentos mediante una plataforma propia para disponer de una organización clara y eficiente.
- Solicita la creación interna de usuarios. Nadie puede registrarse en la plataforma, solo el administrador puede registrar otros usuarios.
- Solicita la privacidad de los datos. Esta plataforma alojará archivos con datos privados, como lo son una nómina, una factura o cualquier imagen de una obra privada.
- Solicita compartir archivos entre usuarios. En el caso de los clientes, compartir proyectos que contienen dichos archivos. En el caso de trabajadores, compartir documentos propios de cada trabajador.

A lo largo de la documentación se profundizará en cada una de las funcionalidades y en el desarrollo del proyecto.

1.3. Estudio de mercado

Existen diversas aplicaciones para gestionar a los empleados de una empresa, como el software Factorial o GerSoft. En cuanto a la gestión de proyectos, las aplicaciones se centran más en la gestión interna de la empresa, y no en la relación

con el cliente. Algunos ejemplos son Trello y Asana. Para compartir archivos, existen lugares de almacenamiento online como son Google Drive y Dropbox. A continuación, se definen dichas aplicaciones y sus funcionalidades.

1.3.1. Factorial

Logotipo principal de Factorial¹:



Ilustración 1.1

Factorial es el software de RRHH todo en uno que te ayuda a gestionar documentos, nóminas, tiempo y talento para que evites errores, dejes de perder el tiempo en tareas administrativas y lleves a tu empresa al éxito. (Factorial, s.f.)

Factorial incluye múltiples soluciones para la gestión de trabajadores:

- Gestión de ausencias y vacaciones
- Control horario y planificación de turnos
- Gestión de nóminas
- Portal de acceso de los empleados
- Organigrama de empresa y gestión de equipos

¹ <https://factorialhr.es/>

- Gestión de documentos de empresa y empleados
- Firma digital

Se puede consultar la información de esta aplicación en el enlace <https://factorialhr.es/>.

Insertión de empleado en Factorial²:

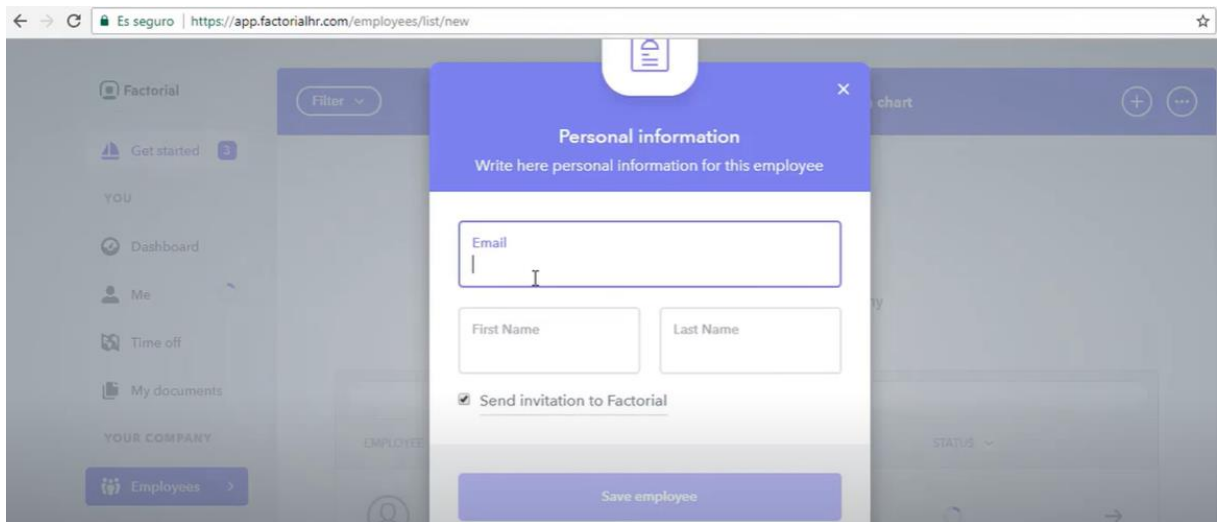
The screenshot shows a web browser window with the URL <https://app.factorialhr.com/employees/list/new>. The interface features a sidebar on the left with navigation options like 'Get started', 'Dashboard', 'Me', 'Time off', 'My documents', and 'Employees'. A central modal window titled 'Personal information' is open, prompting the user to 'Write here personal information for this employee'. It contains input fields for 'Email', 'First Name', and 'Last Name', along with a checked checkbox for 'Send invitation to Factorial' and a 'Save employee' button at the bottom.

Ilustración 1.2

Tabla de empleados en Factorial³:

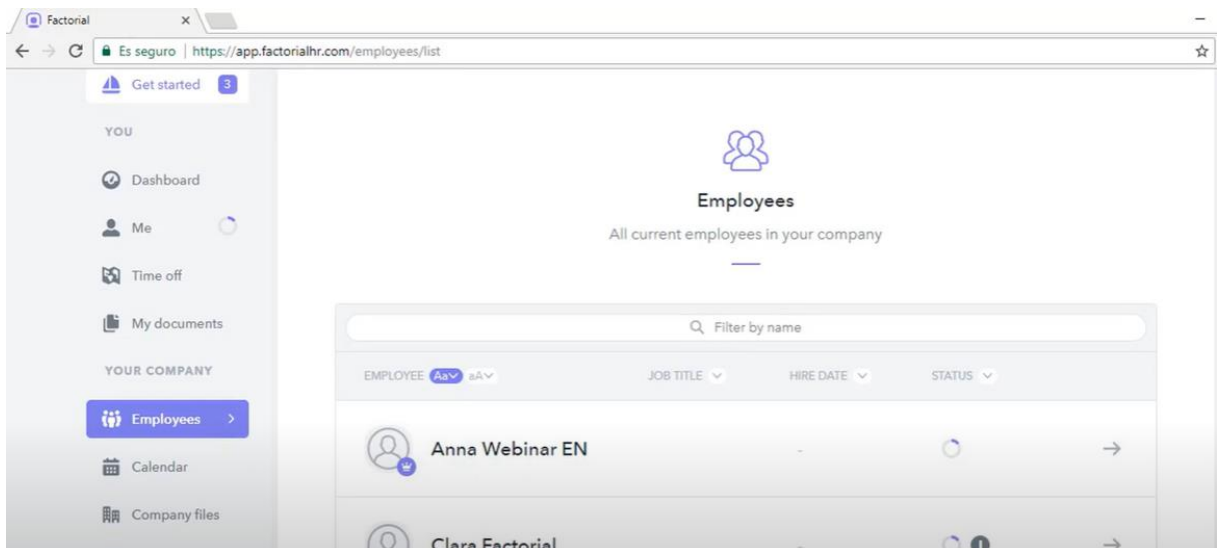
The screenshot displays the 'Employees' section of the Factorial HR application. The header shows 'Employees' with a subtitle 'All current employees in your company'. Below this is a table with columns for 'EMPLOYEE', 'JOB TITLE', 'HIRE DATE', and 'STATUS'. The table lists two employees: 'Anna Webinar EN' and 'Clara Factorial'. Each row includes a profile icon, the employee's name, and a status indicator (a circle with a checkmark or a circle with an exclamation mark). A search bar labeled 'Filter by name' is positioned above the table.

Ilustración 1.3

² <https://www.youtube.com/watch?v=MyWvLd5emns>

³ <https://www.youtube.com/watch?v=MyWvLd5emns>

Este software sería una buena solución para la gestión de trabajadores, pero no soluciona las gestiones entre administrador y clientes. Además, el coste supone un mínimo de 60 euros al mes, y la empresa no requiere todas las funcionalidades. Por lo que, en una empresa pequeña que no va a explotar la aplicación en su totalidad, el coste no es rentable.

1.3.2. GerSoft

Logotipo principal de GerSoft⁴:



Ilustración 1.4

La gestión interna es una aplicación orientada a gestionar los empleados de su empresa a través de internet. La solución aporta una serie de funcionalidades y nuevos servicios que mejora procesos internos y que simplifican la gestión y mejoran su rendimiento. Marcajes vía APP corporativo y/o web para el control de horario, gestión de permisos, gestión de tareas, gestión de incidencias... éxito. (GerSoft, s.f.)

GerSoft incluye las siguientes soluciones:

- APP corporativa
- Gestión de permisos
- Gestión de tareas
- Gestión de incidencias

⁴ <https://gersoft.eu/>

Se puede consultar la información de esta aplicación en el enlace <https://gersoft.eu/>.

Imagen de gestión de un albarán en GerSoft⁵:



Ilustración 1.5

Imagen de gestión de perfiles de usuario en GerSoft⁶:

⁵ <https://gersoft.eu/wp-content/uploads/2018/06/CARGADOCU.jpg>

⁶ <https://gersoft.eu/wp-content/uploads/2018/06/accesodocu.jpg>

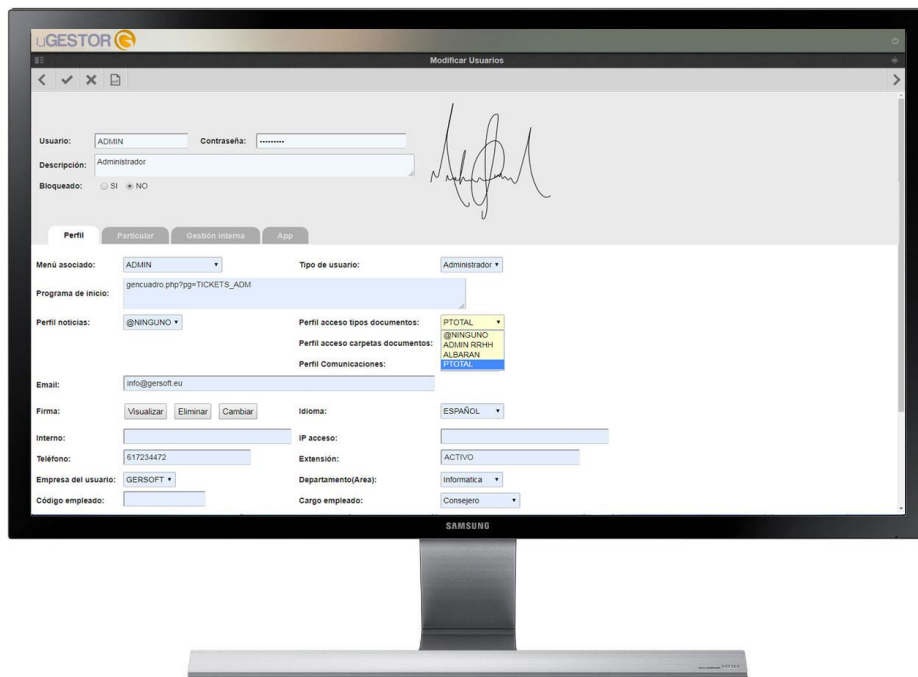


Ilustración 1.6

Esta plataforma soluciona la parte de visualización de los trabajadores. Pero no soluciona la gestión de documentos.

1.3.3. Trello

Logotipo principal de Trello⁷:



Ilustración 1.7

Colabora, gestiona proyectos y alcanza nuevos picos de productividad. Desde oficinas en edificios empresariales hasta trabajo remoto: la forma de trabajo de cada equipo es única, pero siempre se puede hacer con Trello. (Trello, s.f.)

⁷ <https://trello.com/es/login>

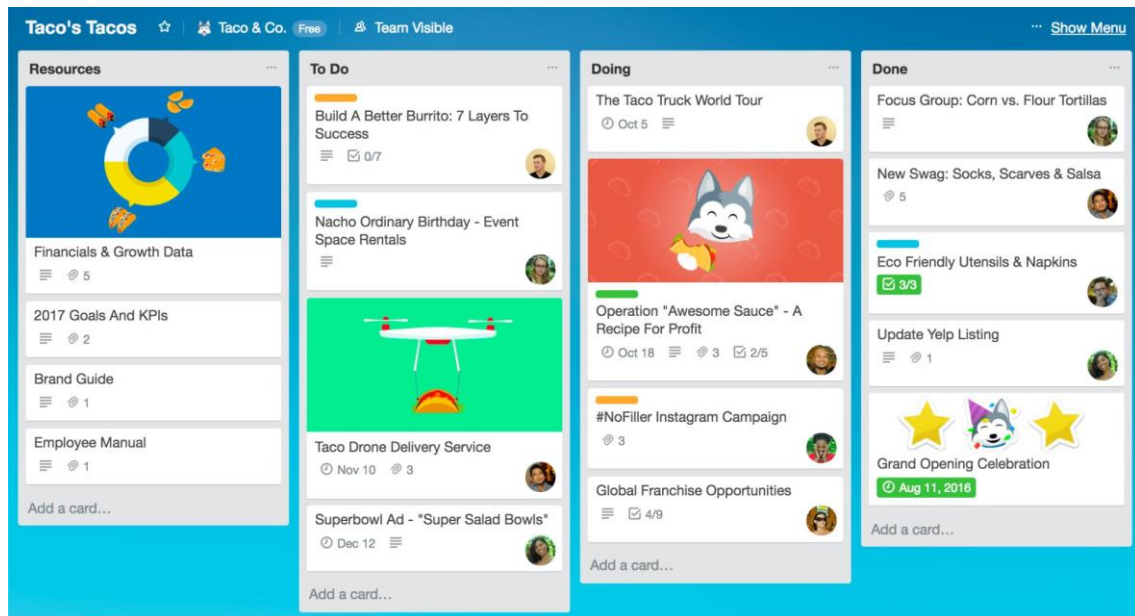
Gestión de proyectos en Trello⁸:

Ilustración 1.8

Se puede consultar la información de esta aplicación en el enlace <https://trello.com/es>.

Trello es una herramienta que permite la gestión de proyectos y flujos de trabajo. Sin embargo, la empresa no desea este tipo de gestión. Por lo que no solucionaría las peticiones anteriores.

1.3.4. Asana

Logotipo principal de Asana⁹:

⁸ <https://herramientasenlanube.com/wp-content/uploads/2019/08/gestion-proyectos-trello.jpg>

⁹ <https://assets.asana.biz/m/7342a96f034ee491/original/trademark-logo-lockup-horizontal.png>



Ilustración 1.9

Optimiza la planificación de tus proyectos, realiza un seguimiento de la información clave para tu equipo y mantén a todos informados, de modo que puedan trabajar mejor y más rápido. (Asana, s.f.)

Se puede consultar la información de esta aplicación en el enlace <https://asana.com/es/>.

Gestión de proyectos en Asana¹⁰:

¹⁰ <https://luna1.co/5eafef.png>

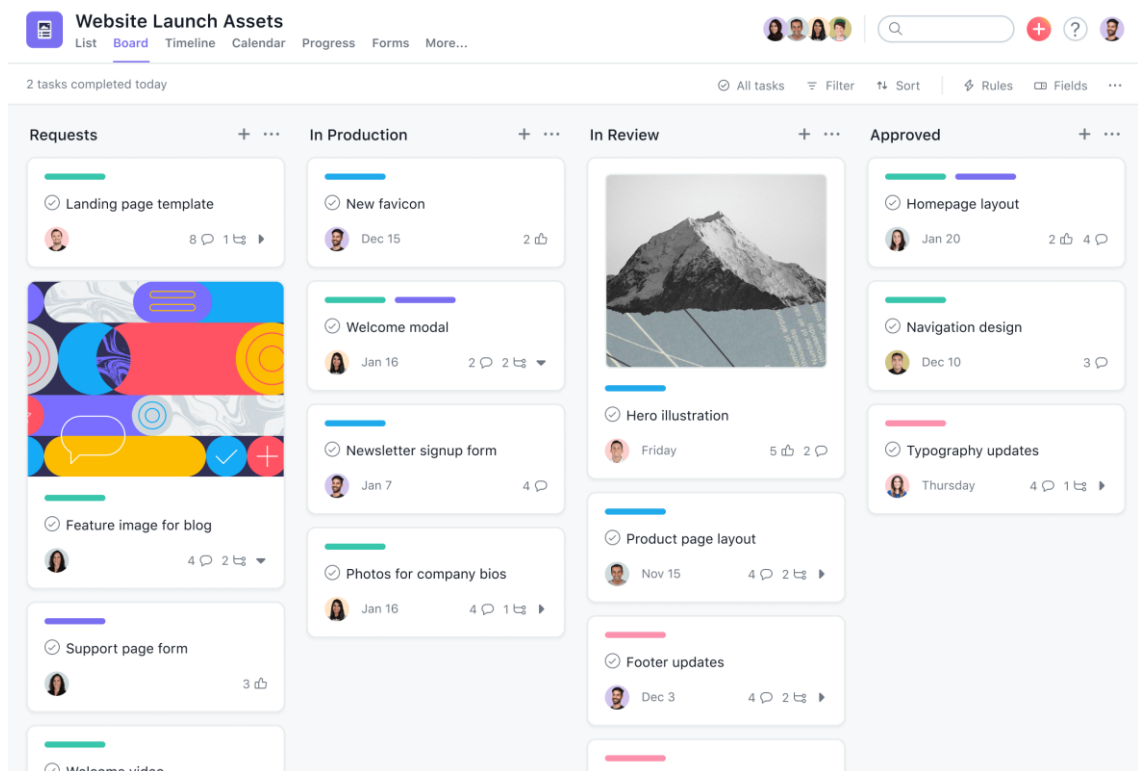


Ilustración 1.10

Asana es una herramienta similar a Trello, por lo que tampoco supera los requerimientos mencionados anteriormente.

1.3.5. Google Drive

Logotipo principal de Google Drive¹¹:

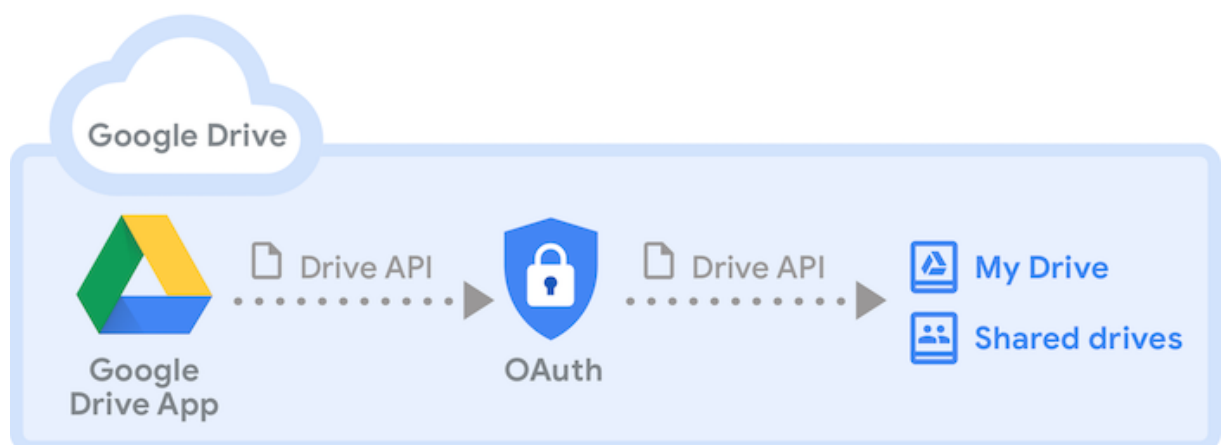
¹¹<https://servicesdownspana.com/img/google-drive-logo.png>



Ilustración 1.11

Drive puede proporcionar un acceso cifrado y seguro a tus archivos. Los archivos compartidos contigo se pueden analizar de forma proactiva y eliminar cuando se detecte software malicioso, spam, ransomware o suplantación de identidad (phishing). Además, como Drive es una aplicación nativa de la nube, no hace falta tener archivos locales y se minimizan los riesgos a los que se exponen tus dispositivos. (Google Drive, s.f.)

Funcionamiento de la API de Google Drive¹²:



¹² <https://developers.google.com/static/drive/images/drive-intro.png>

Ilustración 1.12

Se puede consultar la documentación sobre la API de Google Drive en el enlace <https://developers.google.com/drive/api/guides/about-sdk>.

Drive permite el uso compartido de archivos entre distintos usuarios, pero no es una herramienta específica para la gestión de empresas. Por este motivo, en este proyecto se utilizará esta plataforma como alojamiento de documentos, integrando la API en la aplicación.

1.3.6. Dropbox

Logotipo principal de Dropbox¹³:

**Ilustración 1.13**

Dropbox es el lugar en el que centralizas todo el contenido de tu equipo, el lugar en el que utilizas tus herramientas preferidas, el lugar en el que te ayudamos a superar el desorden y a destacar lo que más importa. (Dropbox, s.f.)

¹³ <https://developers.google.com/static/drive/images/drive-intro.png>

Se puede consultar información en la documentación oficial de Dropbox, en el siguiente enlace: <https://www.dropbox.com/developers/reference>.

Al igual que Drive, Dropbox sirve para el uso compartido de archivos en la nube y dispone de una API. Sin embargo, se ha elegido la opción de Google Drive porque ofrece 15GB de forma gratuita, frente a 2GB de Dropbox.

1.4. Procedimiento a seguir en la elaboración de la plataforma

1.4.1. Estudio de mercado y selección de los usuarios

Tras el análisis del mercado, se han encontrado funcionalidades que solucionan los requerimientos anteriormente descritos. Se tomará como referencia el sistema que utiliza factorial para la gestión de usuarios. Para la gestión de clientes y sus proyectos, se dispondrán los archivos compartidos utilizando la API que ofrece Google Drive.

El contenido de esta aplicación está dirigido exclusivamente a esta empresa, ya que las especificaciones de la plataforma se ajustan estrictamente a lo que dicha empresa solicita.

Sin embargo, la plataforma podría dirigirse a cualquier tipo de empresa similar, o a una empresa que ofrezca a sus clientes un servicio e información sobre este durante su desarrollo.

1.4.2. Estudio sobre los diferentes frameworks sobre programación

Una vez definido el estilo de la plataforma, se realizará un estudio sobre los diferentes frameworks del mercado. Algunos ejemplos son Angular, React y Vue.js.

1.4.3. Diseño de la estructura de la plataforma

Se diseña la estructura de la plataforma basándonos en lo que pide el cliente. Para ello, se suceden una serie de reuniones en las que se le realizan una serie de preguntas para comprender detalladamente de qué trata cada apartado y así especificar correctamente la base de datos. A continuación, se definen las preguntas y respuestas del administrador de la empresa.

- ¿Qué tipos de usuarios hay en la aplicación?
 - Hay 3, administradores, clientes y trabajadores
- ¿Existe un superadministrador que tenga permisos superiores a otros administradores?
 - No, todos los administradores tienen las mismas funciones.
- ¿Qué datos son necesarios en cada usuario?
 - Administrador: email, nombre.
 - Cliente: email, nombre y apellidos.
 - Trabajador: email, nombre y apellidos, DNI, teléfono, domicilio, código postal, población, número de la seguridad social, fecha de nacimiento, cargo en la empresa.
- ¿Qué documentos se asocian al usuario?
 - Contrato y nómina.
- Tendrá más de una nómina, ¿tendrá más de un contrato?
 - Sí, porque a veces se hacen contratos por obra y un mismo trabajador tiene varios contratos a lo largo del tiempo.
- ¿De qué tipo son estos archivos?
 - Por norma general, de tipo pdf.
- ¿Qué se asocia a los clientes?
 - Se le asigna el proyecto en el que se está trabajando.
- ¿Un cliente puede tener más de un proyecto al mismo tiempo?
 - Sí.
- ¿Un proyecto se puede asociar a varios clientes?
 - No, solo a uno.
- ¿Qué tiene un proyecto?
 - Un proyecto tiene imágenes, composiciones que realizo de las imágenes, facturas, un presupuesto...
- ¿Solo hay un presupuesto?
 - Por norma general, sí.
- ¿Qué tipo de archivo son esas composiciones?
 - Suelen ser de tipo pdf, pero también pueden ser dwg, el formato de AutoCAD.

- ¿Quieres guardar alguna descripción de esos archivos?
 - No, solo compartir el archivo.
- ¿Puede haber varias facturas?
 - Sí.
- ¿Qué datos necesita un proyecto?
 - Título y descripción.
- ¿Los trabajadores y clientes van a subir archivos?
 - No.
- ¿Los trabajadores y clientes van a editar o borrar archivos?
 - No, solo pueden verlos o descargarlos.
- ¿Los trabajadores y clientes pueden editar sus propios datos?
 - Sí, los clientes totalmente. Los trabajadores todo menos el cargo, que lo asignamos nosotros.
- ¿Los administradores pueden editar o eliminar al resto de usuarios?
 - Sí, a todos.

Estas preguntas son necesarias para la correcta implementación de la base de datos y de las funcionalidades de la plataforma. Una vez se aseguran los campos y necesidades, se genera un storyboard y se verifica por la empresa solicitante.

1.4.4. Implementación de la plataforma

La plataforma se implementará localmente con los frameworks elegidos, que finalmente son React js para la interfaz y Node js para la parte de backend. Además, se utiliza el servidor Wamp para alojar la base de datos MySQL en PhpMyAdmin.

En el capítulo III se describe de forma detallada cómo se ha llegado a esta decisión.

1.4.5. Pruebas de usabilidad y validación

Las pruebas de usabilidad se realizarán por parte la propia empresa, a la espera de su validación para la corrección de errores y futuras versiones de la plataforma.

1.5. Estructura de la memoria

La estructura del documento consta de las siguientes partes.

- Introducción
- Soluciones existentes en el mercado
- Tecnologías utilizadas
- Requisitos del proyecto
- Diseño del proyecto
- Implementación del proyecto
- Evaluación del proyecto
- Conclusiones

2. SOLUCIONES EXISTENTES EN EL MERCADO

2.1. Aplicaciones para la gestión de usuarios

Existen aplicaciones que pueden solucionar una parte de la aplicación, sobre todo para la parte de administración de trabajadores, pues existen plataformas que ofrecen solución a lo requerido, y amplían las funcionalidades para extenderse a cualquier tipo de empresa. Algunas de ellas son Factorial, GerSoft, Trello y Asana.

2.2. Aplicaciones para la gestión de documentos

En cuanto a la administración de proyectos, la mayoría de los resultados encontrados tratan del uso compartido de documentos en la nube, como son Google Drive, Dropbox o Wetransfer.

2.3. Aplicaciones para la gestión de usuarios y documentos

Se ha encontrado una aplicación dedicada a mantener un repositorio de archivos para los clientes, se trata de dataprius.

2.3.1. Dataprius

Logotipo principal de Dataprius¹⁴:



Ilustración 2.1

¹⁴ <https://observatorio.andaluciaconectada.es/wp-content/uploads/sites/7/2021/04/logo-dataprius-prof-1-1-e1623065188723.png>

Muchas empresas necesitan crear carpetas para compartirlas con los clientes. Dataprius dispone de los mecanismos necesarios para que nuestras carpetas y documentos sean accesibles por aquellos usuarios designados. Mantener un repositorio de archivos para los clientes es ya una forma habitual de trabajo para cientos de empresas que usan el sistema a diario. (Dataprius, s.f.)

Las principales características de esta plataforma son las siguientes:

- RGPD Firmado. Contratos con clientes.
- Soporte técnico cualificado
- Edición directa de los archivos en la nube.
- Compartir documentos con clientes de forma privada.
- Inmune a Ransomware.
- Compartir con Metaenlaces.
- Complementos para Office y Outlook.
- Mecanismos de protección de archivos.
- Acceso mediante varios medios. Adaptado a IOS, Android.
- Trabajo directo con cualquier tipo de archivo y editores online.
- Personalización con los logotipos de la empresa.
- Capacidad de almacenamiento y usuarios a demanda.
- Sencillo de usar. Se instala y se maneja en minutos.
- Almacenamiento de ficheros redundante.
- Con Backups en la propia Nube.
- Correo seguro integrado en aplicaciones.
- Firma digital de documentos PDF.
- Integración de cualquier aplicación web.
- Chat incluido.
- Videoconferencia incluida.

Como se puede apreciar, esta aplicación es bastante completa en cuanto a la gestión de archivos con los clientes. Se integra tanto en web como en Android e IOS

e incluye funcionalidades que pueden ser muy útiles para la comunicación con el cliente. Además de la gestión de archivos con clientes, se podría utilizar también con los trabajadores.

Este sistema tiene un precio variable, en función de la capacidad en TeraBytes (TB), el total de usuarios y algunos extras como Backups e indexación de archivos.

El coste mínimo es de un total anual de 192 euros si se solicita el tipo de pago anual. Este precio ascendería a 292,80 euros al año, si se decide pagar mensualmente, con un coste de 24,40 euros al mes.

Coste anual mínimo de la aplicación¹⁵:

The screenshot shows a pricing configuration interface. On the left, there are three sliders for 'Capacidad en TeraBytes (TB)' (set to 1), 'Total Usuarios' (set to 2), and 'Total Usuarios Web incluidos' (set to 400). Below these is a section for 'extras' with two dropdown menus: 'Backups' (set to '1 Gratuito') and 'Indexación de archivos' (set to 'Sin indexación'). On the right, there are two radio buttons for 'Pago Anual' (selected) and 'Pago Mensual'. Below them, a box displays 'Total Anual: 192 €'. Further down, it states 'Precio sin IVA. Las empresas europeas están sujetas a IVA.' and features a green button labeled 'SOLICITAR CUENTA'. Below the button, it says 'Contactamos con usted. No requiere pago inmediato.'

Ilustración 2.2

Coste mensual mínimo de la aplicación¹⁶:

¹⁵ <https://dataprius.com/precios-almacenamiento-trabajo-cloud.html>

¹⁶ <https://dataprius.com/precios-almacenamiento-trabajo-cloud.html>

Capacidad en TeraBytes (TB)

1

Total Usuarios

2

Total Usuarios Web incluidos

Usuarios externos para compartir archivos e información

400

extras

Backups

1 Gratis

Indexación de archivos

Sin indexación

Pago Anual

Pago Mensual

Total Mensual: 24.4 €

Precio sin IVA.
Las empresas europeas están sujetas a IVA.

SOLICITAR CUENTA

Contactamos con usted.
No requiere pago inmediato.

Ilustración 2.3

El coste máximo alcanzaría un total de 13872€ anuales si se solicita el tipo de pago anual, y ascendería a 15.952,80€ anual, seleccionando el pago mensual de 1329,40€.

Coste anual máximo de la aplicación, con máximo de usuarios, TB y todos los extras¹⁷:

Capacidad en TeraBytes (TB)

10

Total Usuarios

120

Total Usuarios Web incluidos

Usuarios externos para compartir archivos e información

24000

extras

Backups

12

Indexación de archivos

Incluir Indexación

Pago Anual

Pago Mensual

Total Anual: 13872 €

Precio sin IVA.
Las empresas europeas están sujetas a IVA.

SOLICITAR CUENTA

Contactamos con usted.
No requiere pago inmediato.

Ilustración 2.4

¹⁷ <https://dataprius.com/precios-almacenamiento-trabajo-cloud.html>

Coste mensual máximo de la aplicación, con máximo de usuarios, TB y todos los extras¹⁸:

Capacidad en TeraBytes (TB)

10

Total Usuarios

120

Total Usuarios Web incluidos

Usuarios externos para compartir archivos e información

24000

extras

Backups	12
Indexación de archivos	Incluir Indexación

☒ Pago Anual ☒ Pago Mensual

Total Mensual: **1329.4 €**

Precio sin IVA.
Las empresas europeas están sujetas a IVA.

SOLICITAR CUENTA

Contactamos con usted.
No requiere pago inmediato.

Ilustración 2.5

2.4. Conclusiones

Tras el estudio de mercado, se llega a la conclusión de que no existen soluciones en el mercado que solucionen todos los requerimientos, debido a que la aplicación se ajusta exquisitamente a las funcionalidades de necesita la empresa solicitante.

Sin embargo, existe una aplicación que se asimila a lo que pide el cliente, Dataprius, con la única objeción de que no existe una diferenciación de trabajadores y clientes, y no sabemos si se pueden guardar datos de los trabajadores como es el caso del número de la seguridad social.

¹⁸ <https://dataprius.com/precios-almacenamiento-trabajo-cloud.html>

3. POSIBLES TECNOLOGÍAS A UTILIZAR

Existen numerosas tecnologías que pueden utilizarse para desarrollo web. En primer lugar, es necesario distinguir los principales segmentos que componen el desarrollo web.

Por una parte, encontramos el frontend, la parte visible de la plataforma, en la que los usuarios interactúan y acceden al contenido. Por otra parte, el backend, encargado de la lógica, permitir que la interfaz funcione, procesar la información y conectarse con el servidor. Para gestionarlos, se utilizan una serie de lenguajes y frameworks, sobre los que vamos a profundizar.

3.1. Tecnologías frontend

Entre ellas encontramos HTML, CSS y Javascript. Las tres permiten el desarrollo de la parte gráfica del proyecto y la interfaz de usuario. (Blog, s.f.)

3.1.1. HTML

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) (Developer, s.f.) es un conjunto de códigos breves que describe el contenido de una plataforma web. Es un lenguaje de hipertexto de etiquetas que define el significado y la estructura del contenido web.

Las marcas HTML incluyen "elementos" especiales como <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output> (en-US), <progress>, <video>, , , y muchos otros.

3.1.2. CSS

CSS (Hoja de estilos en cascada) (Microsoft, s.f.) da forma a la representación visual de HTML, indicando cómo debe reflejarse. Proporciona el estilo y diseño de la web.

Declaración de CSS¹⁹:

¹⁹ <https://blog.hubspot.es/hubfs/queescssregla.jpeg>

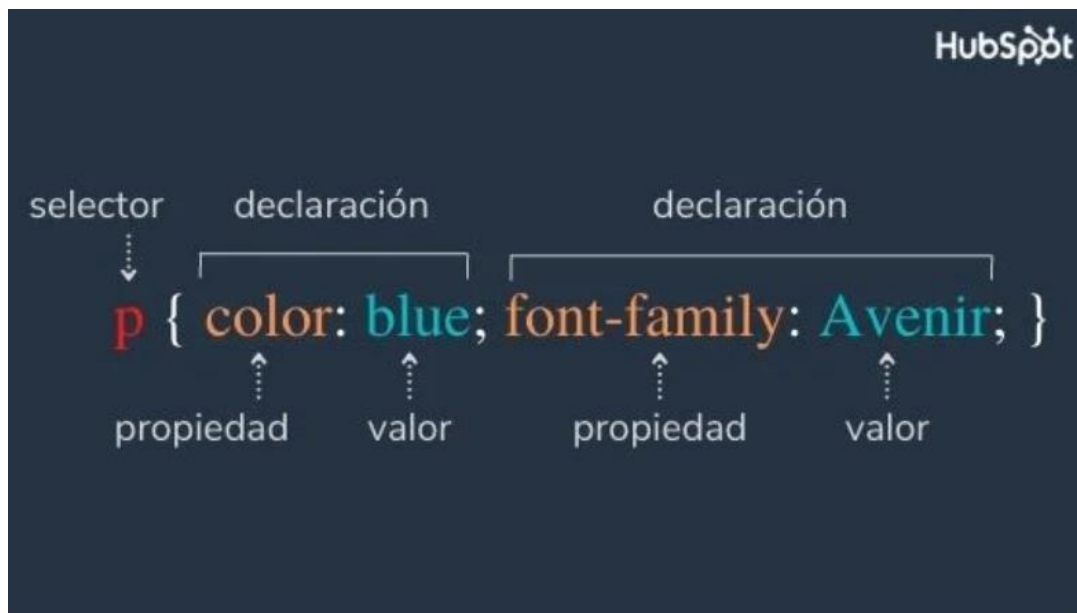
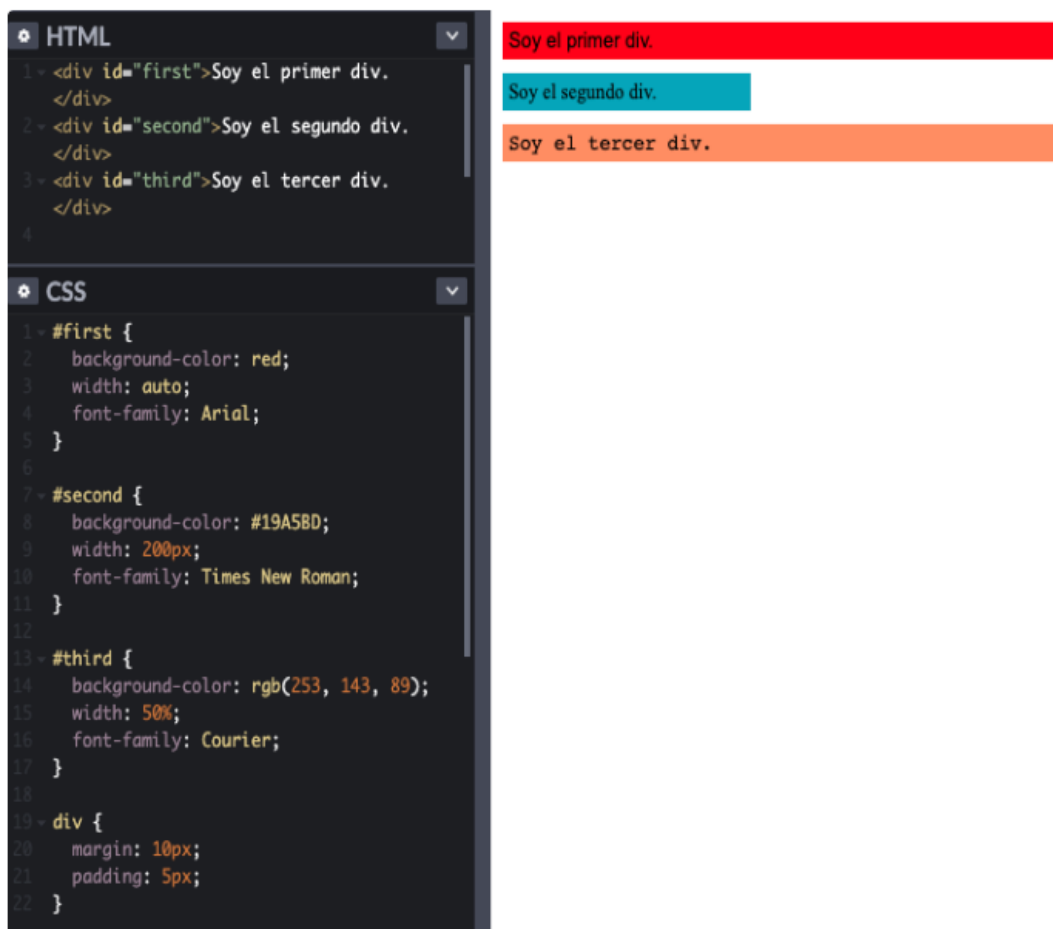


Ilustración 3.1

Ejemplo de reglas CSS aplicadas a HTML²⁰:



²⁰ <https://blog.hubspot.es/hubfs/queescssejemplodevalor.png>

Ilustración 3.2

3.1.3. Javascript

Es el encargado de aportar interactividad al desarrollo, programa el comportamiento de los elementos aportando dinamismo a la interfaz.

Ejemplo de función para lanzar alerta en una web:

Botón inicial²¹:

PLAYER 1: CHRIS

Ilustración 3.3

Función para cambiar el texto del nombre²²:

```
const para = document.querySelector('p');

para.addEventListener('click', updateName);

function updateName() {
  let name = prompt('Enter a new name');
  para.textContent = 'Player 1: ' + name;
}
```

Ilustración 3.4

Alerta que aparece al hacer clic²³:

Enter a new name

Celia Macías Torres

Aceptar

Cancelar

²¹ https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

²² https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

²³ https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Ilustración 3.2

Resultado final²⁴:



Ilustración 3.5

3.2. Frameworks y librerías frontend aplicables a nuestro proyecto

Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software. (Edix, s.f.)

Actualmente existe una gran diversidad de frameworks creados para el desarrollo de aplicaciones, como son Angular, React o Vue.JS.

3.2.1. Angular

Angular fue desarrollada en 2009 por Miško Hevery y Adams Abrons. En un principio consistía en un software detrás de un servicio de almacenamiento online de archivos JSON, pero tiempo después se abandonó dicho proyecto y se liberó AngularJS como una biblioteca de código abierto. (Gutiérrez, 2018)

Desde entonces se han lanzado numerosas versiones de AngularJS, hasta llegar a la actual: Angular 12, lanzada el 12/05/2021. (Hiberus, s.f.)

Aunque muchos usuarios lo confundan con una librería, AngularJS es, definitivamente, un framework 100% JavaScript, 100% del lado del cliente y compatible con navegadores de escritorio y móviles. (Angular JS, s.f.)

Ejemplo de código de Angular²⁵:

²⁴ https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

²⁵ <https://codigoonclick.com/conoce-angular-en-5-minutos/>



Ilustración 3.6

Las principales ventajas de Angular son: (Hostinger, s.f.)

- Enlace bidireccional de datos
- Directivas
- Estructura de código
- Pruebas
- Futuro brillante debido a su funcionalidad y popularidad
- Compatibilidad móvil y de escritorio

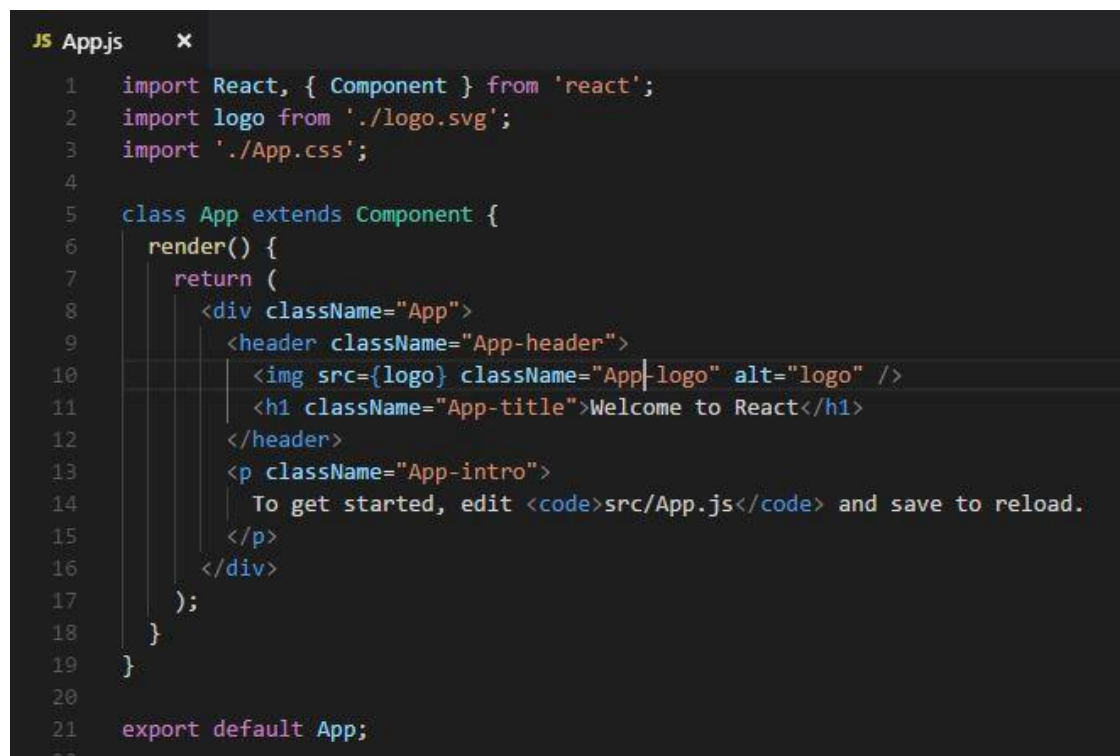
3.2.2. React js

React es una biblioteca de JavaScript declarativa, eficiente y flexible para construir interfaces de usuario. Permite componer IUs complejas de pequeñas y aisladas piezas de código llamadas “componentes”. (Reactjs, s.f.)

Fue creada por Jordan Walker, ingeniero de software en Facebook. En un principio, el prototipo se denominó “FaxJS” y se implementó por primera vez en las noticias de Facebook, en 2011. En 2013 se constituyó de código abierto. (htt)

Es un sistema basado en componentes encapsulados que manejan su propio estado. La lógica de los componentes está escrita en JavaScript, por lo que los datos se envían de forma sencilla.

Ejemplo de código de React²⁶:



```
JS App.js x
1  import React, { Component } from 'react';
2  import logo from './logo.svg';
3  import './App.css';
4
5  class App extends Component {
6    render() {
7      return (
8        <div className="App">
9          <header className="App-header">
10             <img src={logo} className="App-logo" alt="logo" />
11             <h1 className="App-title">Welcome to React</h1>
12          </header>
13          <p className="App-intro">
14            To get started, edit <code>src/App.js</code> and save to reload.
15          </p>
16        </div>
17      );
18    }
19  }
20
21  export default App;
22
```

Ilustración 3.7

React puede renderizar desde el servidor, usando Node y potenciar también aplicaciones móviles utilizando React Native.

Algunas ventajas de Reactjs son: (Doonamis, s.f.)

- Desarrollo rentable
- Entrega más rápida de proyectos de aplicaciones
- Aprovecha JavaScript
- Ventaja del código abierto
- Función de recarga activa
- Excelente rendimiento de la aplicación
- El diseño modular

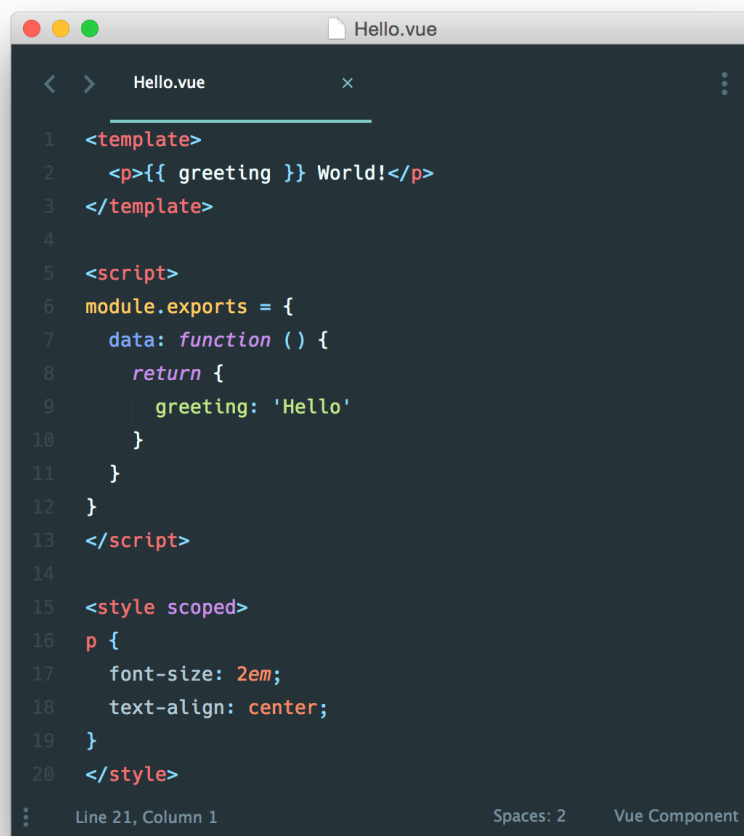
²⁶ <https://medium.com/@leonardellifernando/reactjs-primeros-pasos-e0176b1ce5b5>

3.2.3. Vue.js

Vue (pronunciado /vju:/, como view) es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo. (Vue JS, s.f.)

Vue js se basa en html, css y JavaScript estándar y fue creado por Evan You, extrabajador de Google, que decidió crear su propio framework en el año 2014.

Ejemplo de código de Vue js²⁷:



```
1 <template>
2   <p>{{ greeting }} World!</p>
3 </template>
4
5 <script>
6   module.exports = {
7     data: function () {
8       return {
9         greeting: 'Hello'
10      }
11    }
12  }
13 </script>
14
15 <style scoped>
16   p {
17     font-size: 2em;
18     text-align: center;
19   }
20 </style>
```

Line 21, Column 1 Spaces: 2 Vue Component

²⁷ <https://es.vuejs.org/v2/guide/single-file-components.html>

Ilustración 3.8

Algunas ventajas de Vue son: (Blog, s.f.)

- El proceso y la Curva de aprendizaje es suave, no es complicado.
- Ecosistema bien definido, si deseas solucionar un problema con Vue JS no tienes que dar tantas vueltas, todo lo que necesitas esta allí para usarlo.
- Cuenta con una línea de comandos llamada Vue CLI que nos facilita el trabajo mediante la ejecución de comandos específicos para trabajar rápido y eficientemente en nuestros proyectos.
- Es lo mejor de Angular JS y React JS, Evan You su creador trabajo en el desarrollo de Angular JS cuando estaba en Google y decidió construir una solución ligera extrayendo lo mejor de Angular JS.
- Vue JS pesa algo más de 20KB y tiene un rendimiento fantástico, es quizás uno de los Frameworks más rápidos en la actualidad.

3.3. Tecnologías backend

Este conjunto de lenguajes se encarga de implementar los comportamientos de la web en el servidor. En el backend se programa la interacción web, se almacena la información en el servidor, y es el encargado de conectarse con la base de datos para su actualización y recuperación de datos. (Blog, s.f.)

3.3.1. PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. (PHP, s.f.)

Ejemplo de código de php²⁸:

²⁸ <https://www.php.net/manual/es/intro-what-is.php>

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>

    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>

  </body>
</html>
```

Ilustración 3.9

En lugar de usar muchos comandos para mostrar HTML, las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!"). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del "modo PHP". (PHP, s.f.)

3.3.2. Python

Python es un lenguaje de programación de alto nivel caracterizado por definir una sintaxis muy limpia y legible. Soporta el paradigma orientado a objetos (como Java o Ruby), pero se puede seguir el paradigma de la programación estructurada (como C). (Nobbot, s.f.)

Ejemplo de código de Python²⁹:

²⁹ <https://www.nobbot.com/tecnologia/los-lenguajes-de-desarrollo-web-del-futuro-python/>

```

class Rectangle(Blob):

    def __init__(self, width, height,
                  color='black', emphasis=None, highlight=0):
        if width == 0 and height == 0 and \
            color == 'red' and emphasis == 'strong' or \
            highlight > 100:
            raise ValueError("sorry, you lose")
        if width == 0 and height == 0 and (color == 'red' or
                                           emphasis is None):
            raise ValueError("I don't think so")
        Blob.__init__(self, width, height,
                      color, emphasis, highlight)

```

Ilustración 3.10

3.3.3. Java EE

Java EE (Java Enterprise Edition) es un entorno independiente de la plataforma centrado en Java para desarrollar, crear e implementar en línea aplicaciones empresariales basadas en web. Java EE incluye muchos componentes de Java Standard Edition (Java SE). La plataforma Java EE consta de un conjunto de servicios, API y protocolos que proporcionan la funcionalidad necesaria para desarrollar aplicaciones basadas en web de varios niveles. (Java, s.f.)

Ejemplo de código de Python³⁰:

```

<div class="col-md-12 text-center">
  <div id="purchasePrice" data-price="0,00" class="total-price">#{indexBean.getPrice()} €</div>
  <c:choose>
    <c:when test="#{indexBean.buyer}">
      <button id="buy" type="button" class="btn btn-primary btn-lg">Buy</button>
    </c:when>
    <c:otherwise>
      <h:link value="Login to buy" outcome="login" class="btn btn-primary btn-lg"/>
    </c:otherwise>
  </c:choose>
</div>

```

Ilustración 3.11

³⁰ <https://picodotdev.github.io/blog-bitix/2016/04/aplicacion-de-ejemplo-usando-varias-especificaciones-de-java-ee-7/>

3.3.4. Node.js

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. En el siguiente ejemplo de "hola mundo", pueden atenderse muchas conexiones simultáneamente. Por cada conexión, se activa la devolución de llamada, pero si no hay trabajo que hacer, Node.js se dormirá. (Node.js, s.f.)

Ejemplo de código de Node.js³¹:

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hola Mundo');
});

server.listen(port, hostname, () => {
  console.log(`El servidor se está ejecutando en http://${hostname}:${port}/`);
});
```

Ilustración 3.12

Node.js entra en el bucle de eventos después de ejecutar el script de entrada y sale cuando no hay más devoluciones callbacks de llamada para realizar. Se comporta de una forma similar a JavaScript en el navegador.

3.4. Frameworks backend aplicables a nuestro proyecto

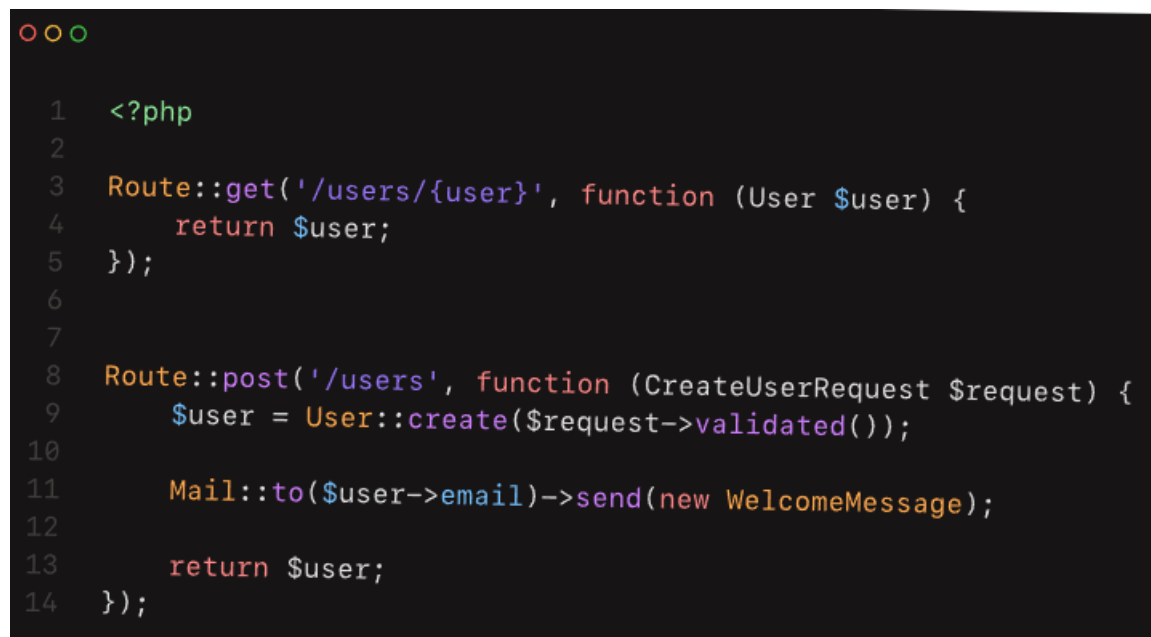
Los frameworks generan diversos beneficios en el desarrollo. El código web se encuentra organizado y estructurado desde el primer momento y se favorece el control y la seguridad de la plataforma filtrando la entrada y salida de datos. (Blog, s.f.)

³¹ <https://nodejs.org/es/about/>

3.4.1. Laravel

Laravel es un framework PHP. Es uno de los frameworks más utilizados y de mayor comunidad en el mundo de Internet. (Desarrollo Web, s.f.)

Ejemplo de código de Laravel³²:

A screenshot of a code editor with a dark background and syntax highlighting. The code is in PHP and defines two routes: a GET route for '/users/{user}' and a POST route for '/users'. The GET route returns the user object. The POST route creates a new user, sends a welcome message via email, and returns the user object. The code is numbered from 1 to 14.

```
1  <?php
2
3  Route::get('/users/{user}', function (User $user) {
4      return $user;
5  });
6
7
8  Route::post('/users', function (CreateUserRequest $request) {
9      $user = User::create($request->validated());
10
11      Mail::to($user->email)->send(new WelcomeMessage);
12
13      return $user;
14  });
```

Ilustración 3.13

Laravel trabaja con arquitectura de carpetas avanzada que promueve la separación de los archivos con un orden correcto, que guía a los integrantes del equipo de trabajo, siendo un estándar a lo largo de los distintos proyectos.

3.4.2. Express

Express es el framework web más popular de Node, y es la librería subyacente para un gran número de otros frameworks web de Node populares. Proporciona mecanismos para (Developer, s.f.):

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.

³² <https://laravel.com/>

- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

Ejemplo de código de Express³³:

```
var express = require('express');
var app = express();

app.get('/', function(req, res) {
  res.send('Hola Mundo!');
});

app.listen(3000, function() {
  console.log('Aplicación ejemplo, escuchando el puerto 3000!');
});
```

Ilustración 3.14

3.5. Comparativa entre los frameworks estudiados

3.5.1. Angular, React js y Vue.js

Los frameworks de JavaScript evolucionan rápidamente, por lo que existen muchas versiones, que se van actualizando continuamente. En las ofertas de empleo, este es el porcentaje de lenguaje más requerido, según un estudio en Indeed.com. (Developinginspanish, s.f.)

Gráfica comparativa³⁴:

³³ https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

³⁴ <http://developinginspanish.com/2019/03/08/reactjs-vs-angular5-vs-vue-js-cual-elegir/>

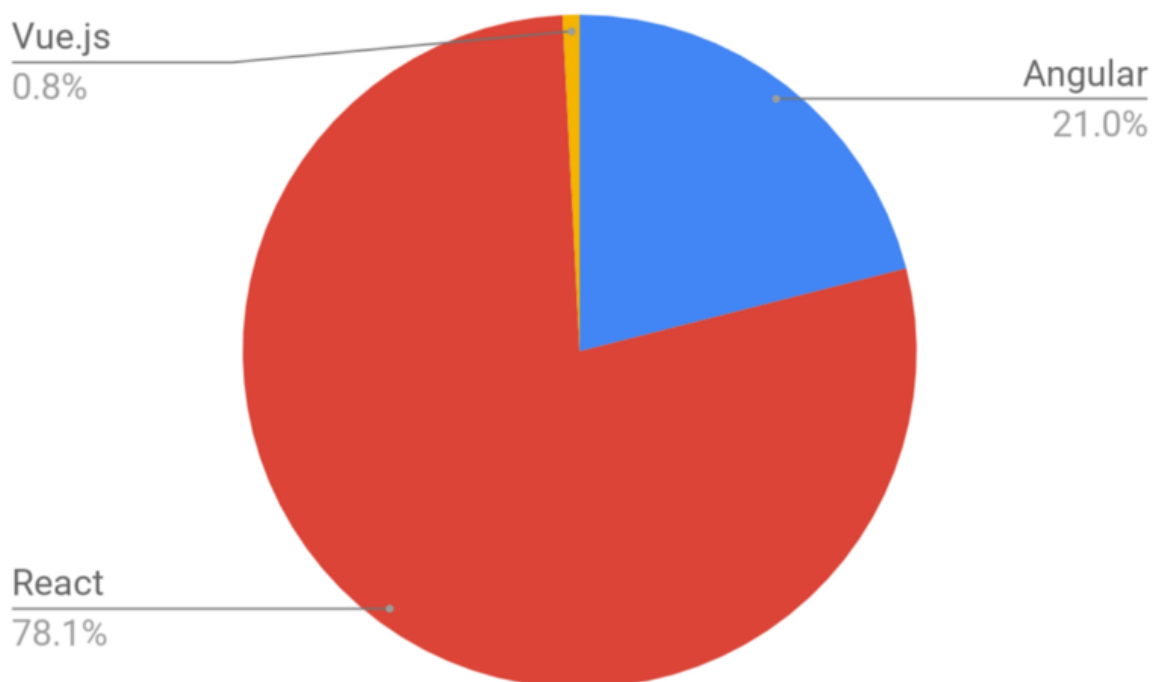


Ilustración 3.35

Tabla comparativa de Angular, React js y Vue.js:

	Angular	React js	Vue.js
DOM	DOM real	DOM virtual	DOM virtual
Tamaño	500KB aprox.	100KB aprox.	80KB aprox.
Flexibilidad	Ofrece todo lo que necesita, pero no es muy flexible.	Es el más flexible.	No demasiado flexible.

Tabla 3.1

En resumen, Vue y React ofrecen mejor rendimiento y flexibilidad que Angular y son más adecuados para aplicaciones sencillas. Angular, sin embargo, es más adecuado para aplicaciones de mayor amplitud. React y Angular tienen mayor demanda laboral y están respaldados por grandes empresas, Facebook y Google.

3.5.2. Express y Laravel

La principal diferencia entre Express y Laravel es que Express es un framework de Node.js y Laravel utiliza PHP. Por lo tanto, la ejecución en Express será asíncrona, frente a la ejecución síncrona de Laravel. La velocidad de Node.js es mayor que la de PHP, y la sincronización es no bloqueante, frente a la sincronización bloqueante de PHP. (Kinsta, s.f.).

Tabla comparativa de Express y Laravel³⁵:

	Express	Laravel
Tecnología	Node js	PHP
Gratuito	Sí	Sí
Cobertura de uso	Electrónica, tecnología, derecho, finanzas, artes, y otras categorías	Industria Pesada e Ingeniería
Sitios web rastreados	225.000	197.000

Tabla 3.2

En la tabla anterior podemos ver que Express es mucho más versátil y se está utilizando una variedad de mercados mayor que Laravel. A continuación, se muestran unas gráficas en las que se observa que Express es más utilizado y tiene un crecimiento constante.

Express domina claramente en el mercado frente a Laravel³⁶:

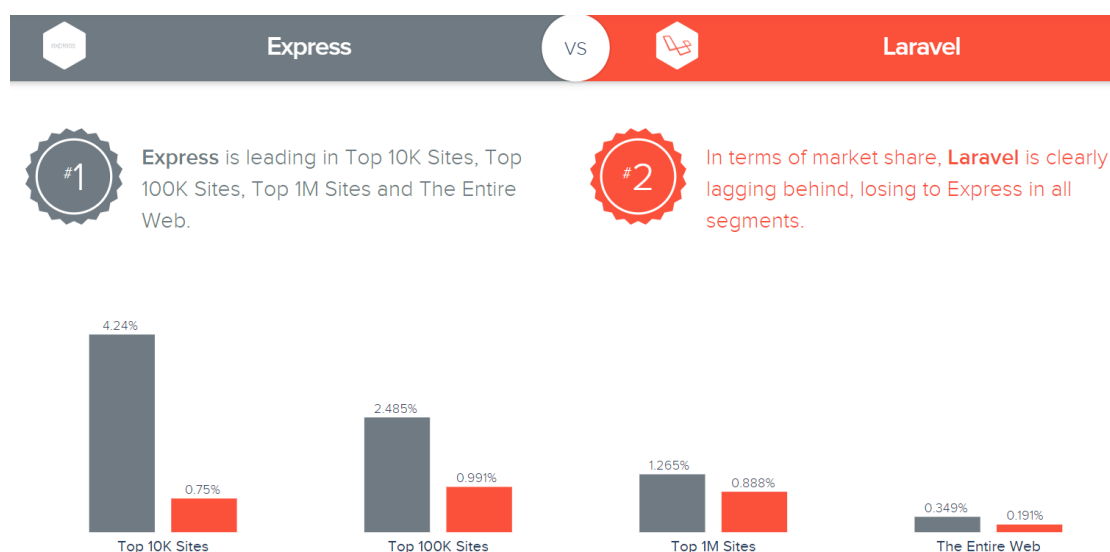


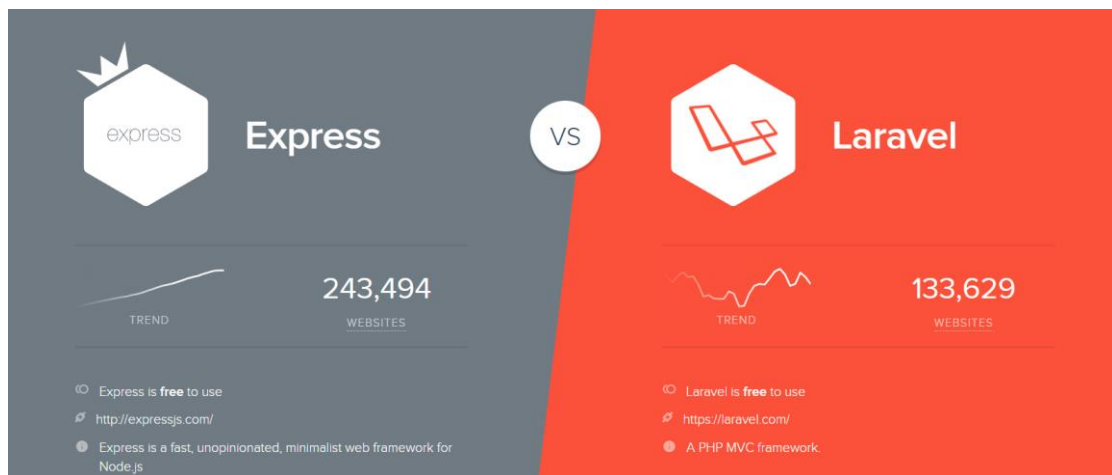
Ilustración 3.16

Imagen comparativa de la tendencia de uso de Express y Laravel³⁷:

³⁵ Datos obtenidos de: <https://code.tutsplus.com/es/tutorials/which-back-end-to-choose-nodejs-vs-php-cms-36890> y <https://www.wappalyzer.com/compare/laravel-vs-express>

³⁶ <https://www.similartech.com/compare/express-vs-laravel>

³⁷ <https://www.similartech.com/compare/express-vs-laravel>

**Ilustración 3.17**

En resumen, ambas son tecnologías que proporcionan numerosas ventajas en programación web, pero Express es el líder en comparativa de uso.

3.6. Conclusiones

Tras el análisis de las tecnologías y frameworks que pueden utilizarse para el desarrollo web, la conclusión final es que el proyecto se desarrollará en Reactjs, no solo por las alternativas que ofrece, sino por la gran cantidad de salidas profesionales que posee. Para el backend se utilizará, el framework de Express, el framework más popular de Node js.

4. REQUISITOS DEL PROYECTO

4.1. Planificación del trabajo

La planificación de este proyecto comienza en septiembre de 2021, y finaliza en septiembre de 2022. A continuación, se muestra el listado de tareas que se realizan durante el proyecto:

Listado de tareas parte 1³⁸:

▶ ✓ Investigar aplicaciones de gestión de usuarios	🕒 4	🟡
▶ ✓ Reunión con el tutor	🕒 2	🟡
▶ ✓ Definir los objetivos de la aplicación	🕒 1	🟡
▶ ◆ Reunión de inicio de TFG	🕒 1	🟡
▶ ✓ Recolectar la visión y objetivos del cliente	🕒 1	🔴 =
▶ ✓ Investigar aplicaciones de gestión de archivos	🕒 4	🟡
▶ ✓ Investigar aplicaciones de gestión de usuarios y archivos	🕒 4	🟡
▶ ✓ Investigar sobre tecnologías y frameworks	🕒 2	🟡
✓ Realizar tutorial con Angular		🟡
✓ Realizar tutorial de React.js		🟡
✓ Realizar tutorial de conexión de backend y frontend con Express, Node.js		🟡
✓ Descargar plantillas		🟡
✓ Hacer pruebas de ejecución y conexión de plantillas y backend		🟡
◆ Selección de plantilla final		🟡
✓ Crear base de datos		🟡
✓ Conectar la base de datos con el backend		🟡
✓ Hacer pruebas de inserción, edición y borrado desde el backend		🟡
▶ ✓ Conectar frontend y backend	🕒 1	🟡
▶ ✓ Crear de inicio de sesión	🕒 3	🟡

Ilustración 4.1

³⁸ Generado con: <https://app.clickup.com/>

Listado de tareas parte 2³⁹:

✓	Crear vista de Escritorio	⊖
✓	Crear vista de clientes	⊖
✓	Crear formulario de clientes	⊖
✓	Crear vista de trabajadores	⊖
✓	Crear formulario de trabajadores	⊖
✓	Crear vista de administradores	⊖
✓	Crear formulario de administradores	⊖
✓	Conectar aplicación con Google Drive	⊖
✓	Crear vista de proyecto	⊖
✓	Crear formulario de proyecto	⊖
✓	Crear formulario de facturas	⊖
✓	Creación de vista de facturas	⊖
✓	Crear formulario de documentos del trabajador	⊖
✓	Crear vista de documentos del trabajador	⊖
✓	Restringir acceso a vistas según usuario	⊖
✓	Hacer testeo y corregir errores	⊖
✓	Corregir errores	⊖
✓	Realizar documentación	

Ilustración 4.2

El diagrama de Gantt de un proyecto es una herramienta gráfica que muestra una lista de tareas y el tiempo asignado para terminirlas. A continuación, se muestra la evolución de tareas mediante un diagrama de Gantt.

Diagrama de Gantt parte 1⁴⁰:

³⁹ Generado con: <https://app.clickup.com/>

⁴⁰ Generado con: <https://app.clickup.com/>

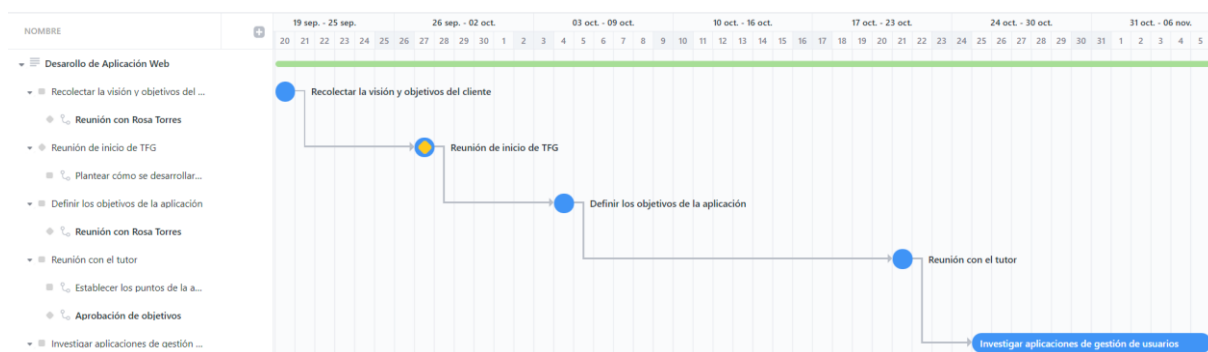


Diagrama de Gantt parte 2⁴¹:

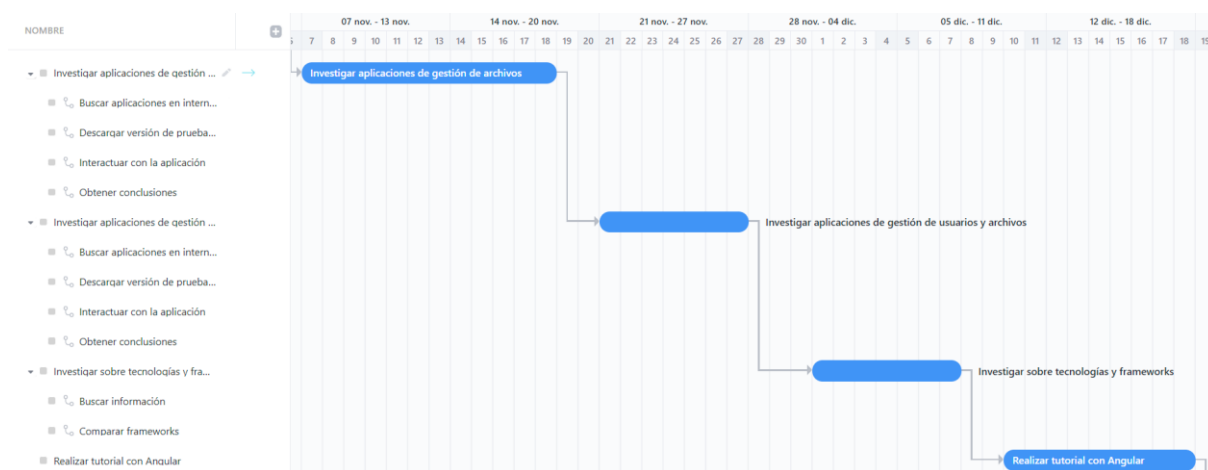


Diagrama de Gantt parte 3⁴²:

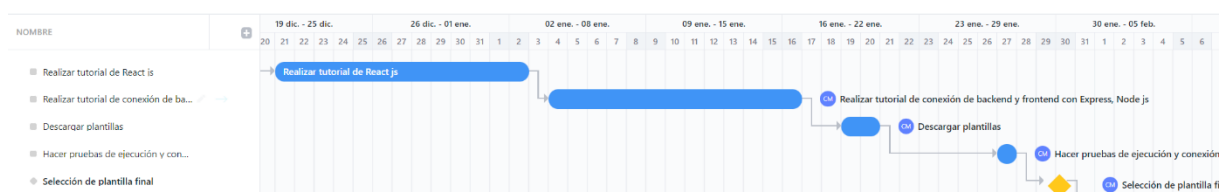


Diagrama de Gantt parte 4⁴³:

⁴¹ Generado con: <https://app.clickup.com/>

⁴² Generado con: <https://app.clickup.com/>

⁴³ Generado con: <https://app.clickup.com/>



Ilustración 4.6

Diagrama de Gantt parte 5⁴⁴:

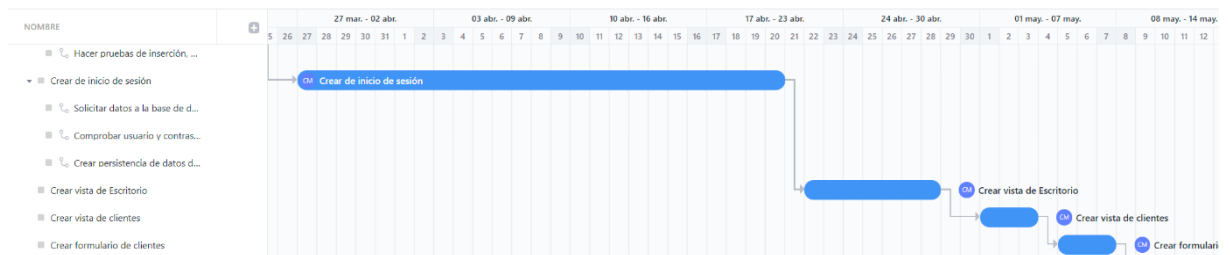


Ilustración 4.7

Diagrama de Gantt parte 6⁴⁵:

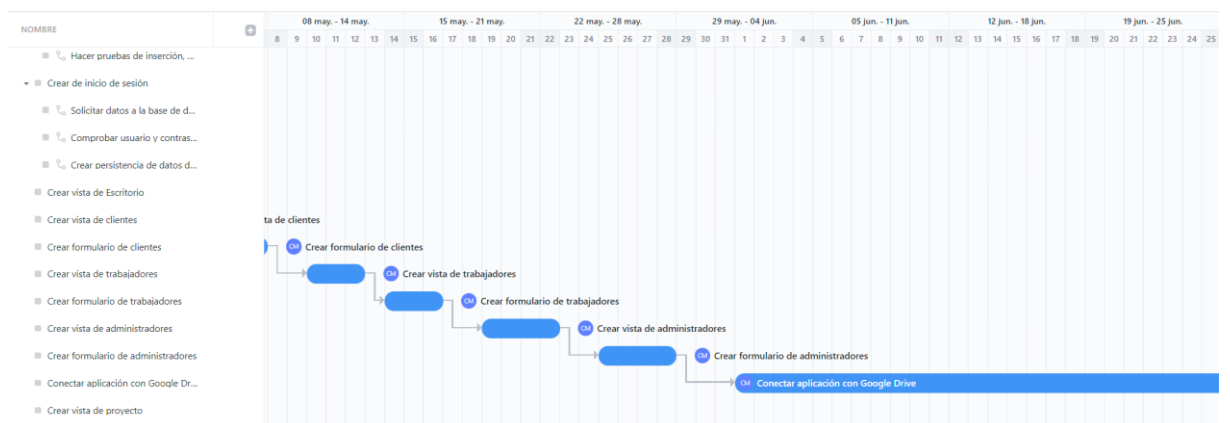


Ilustración 4.8

Diagrama de Gantt parte 7⁴⁶:

⁴⁴ Generado con: <https://app.clickup.com/>

⁴⁵ Generado con: <https://app.clickup.com/>

⁴⁶ Generado con: <https://app.clickup.com/>

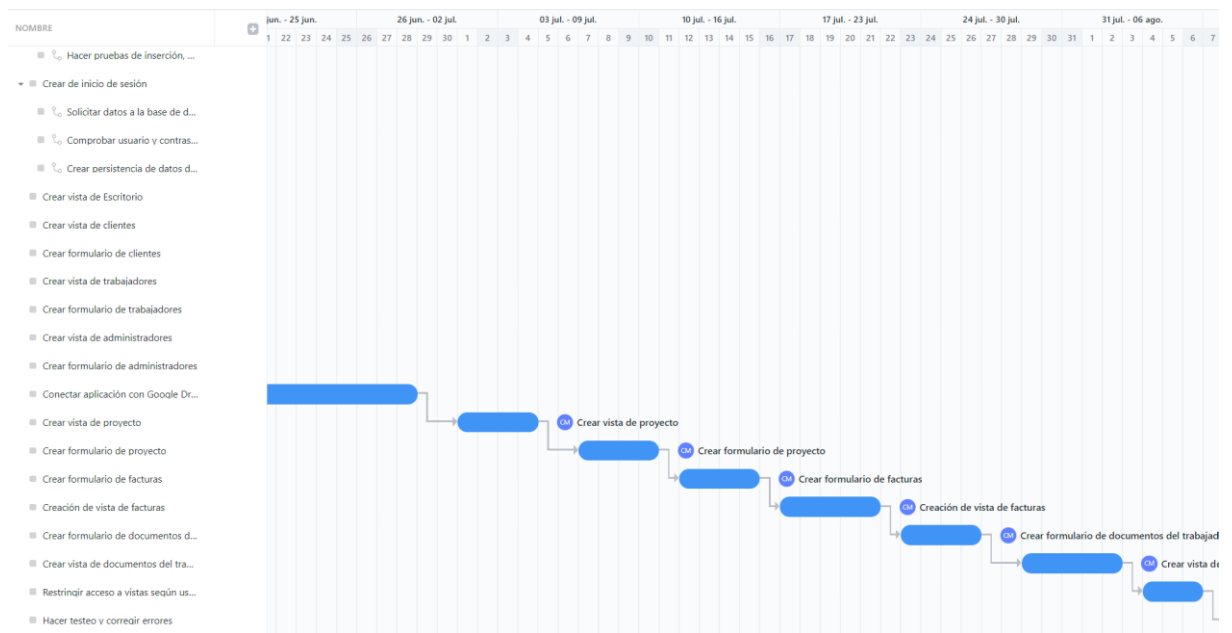
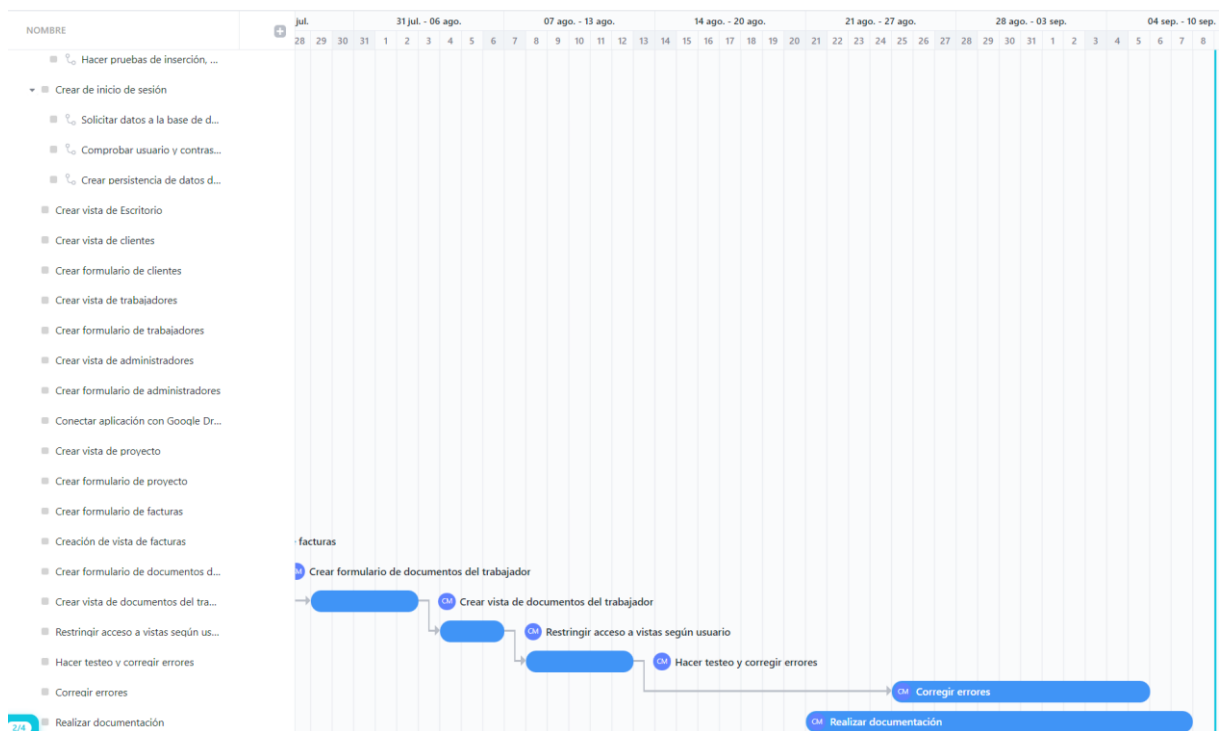


Diagrama de Gantt parte 7⁴⁷:



⁴⁷ Generado con: <https://app.clickup.com/>

4.2. Análisis de los costes

Se prevé que el coste total del desarrollo de la aplicación será de 4.692,94€, teniendo en cuenta la provisión completa de hardware y software al desarrollador de la aplicación. Este precio se desglosa en las siguientes tablas:

4.2.1. Costes referentes al hardware

Costes hardware:

Recurso	Coste (€)
Ordenador	1500
Pantalla Samsung	160
Ratón	10
Total	1670

Tabla 4.1

4.2.2. Costes fijos

Costes fijos:

Recurso	Meses	Coste/mes(€)	Coste (€)
Internet	6	60	360

Tabla 4.2

4.2.3. Costes referentes al software

Costes software:

Recurso	Horas	Meses	Coste/mes(€)	Coste (€)
Plantilla Reactjs	150	6	-	35
Visual Studio Code	150	6	0	0
Licencia Adobe Illustrator	150	6	62,99	377,94
GitMind	150	1	0	0
Total				412,94

Tabla 4.3

4.2.4. Costes de personal

Costes de personal:

Recurso	Horas	Coste (€)
Desarrollador web	150	2250

Tabla 4.4

4.3. Requisitos funcionales

Los requisitos funcionales se definen como las declaraciones de los servicios que proveerá el sistema. En el caso de esta aplicación web serán los siguientes:

REQ.F.001: Inicio de sesión

REQ.F.001	Inicio de sesión
Descripción	Inicio de sesión en la plataforma web mediante el uso del correo electrónico y contraseña.
Perfiles	Administrador, Cliente, Trabajador
Observaciones	El usuario debe haber sido registrado con antelación dentro de la plataforma. Para ello se genera un administrador desde la base de datos, que podrá comenzar a crear otros usuarios.

Tabla 4.5

REQ.F.002: Cierre de sesión

REQ.F.002	Cierre de sesión
Descripción	Cierre de sesión en la plataforma
Perfiles	Administrador, Cliente, Trabajador
Observaciones	El usuario debe poder cerrar sesión desde el menú de la aplicación. Después, será redirigido al inicio de sesión.

Tabla 4.6

REQ.F.003: Creación de usuarios

REQ.F.003	Creación de usuarios
Descripción	Creación de usuarios de cualquier tipo: administrador, cliente y trabajador.
Perfiles	Administrador
Observaciones	El usuario debe poder crear cualquier tipo de usuario. Solo el administrador puede realizar esta acción. El trabajador y el cliente no deben poder crear usuarios.

Tabla 4.7

REQ.F.004: Visualización de usuarios

REQ.F.004	Visualización de usuarios
Descripción	Visualización de los datos de los usuarios
Perfiles	Administrador
Observaciones	Cualquier administrador puede ver los datos de los usuarios, ya sean administradores, trabajadores o usuarios. Los trabajadores y clientes no pueden ver estos datos.

Tabla 4.8

REQ.F.005: Eliminación de usuarios

REQ.F.005	Eliminación de usuarios
Descripción	Eliminación de un usuario de cualquier tipo: administrador, cliente y

	trabajador
Perfiles	Administrador
Observaciones	Cualquier administrador puede eliminar a un administrador, cliente o trabajador. Los clientes y trabajadores no pueden eliminar a ningún usuario.

Tabla 4.9

REQ.F.006: Edición de administradores

REQ.F.006	Edición de administradores
Descripción	Edición de los datos de los administradores
Perfiles	Administrador
Observaciones	Cualquier administrador puede editar los datos del resto. El motivo se debe a que el cambio de contraseña se realiza solo desde dentro de la aplicación, y, además, porque la empresa que solicita la web lo requiere de esta manera.

Tabla 4.10

REQ.F.007: Edición de clientes

REQ.F.007	Edición de clientes
Descripción	Edición de los datos de los clientes
Perfiles	Administrador, Cliente
Observaciones	Cualquier administrador puede editar los datos de los clientes. Los clientes solo pueden editarse a sí mismos. Un trabajador no puede editar clientes. No se puede editar el email del cliente una vez creado.

Tabla 4.11

REQ.F.008: Edición de trabajadores

REQ.F.008	Edición de trabajadores
Descripción	Edición de los datos de los trabajadores
Perfiles	Administrador, Trabajador
Observaciones	Cualquier administrador puede editar los datos de los trabajadores. Los trabajadores solo pueden editarse a sí mismos. Un cliente no puede editar trabajadores. No se puede editar el email del trabajador una vez creado.

Tabla 4.12

REQ.F.009: Creación de documentos de los trabajadores

REQ.F.009	Creación de documentos de los trabajadores
Descripción	Subida de documentos mediante la API de Google Drive.
Perfiles	Administrador
Observaciones	Un administrador debe poder subir documentos desde el panel, añadiendo nombre y el tipo de documento (nómina, contrato, protección de datos u otro). Se debe compartir el enlace del archivo en drive con el trabajador al que pertenece, para su posterior visualización.

Tabla 4.13

REQ.F.010: Edición de documentos de los trabajadores

REQ.F.010	Edición de documentos de los trabajadores
Descripción	Edición de los datos del documento (nombre y tipo) y sustitución del documento.
Perfiles	Administrador
Observaciones	Un administrador debe poder sustituir el documento actual desde el panel. Debe poder editar el nombre y el tipo de documento (nómina, contrato, protección de datos u otro). Se debe compartir el enlace del nuevo archivo en drive con el trabajador al que pertenece, para su posterior visualización. Los clientes no pueden editar documentos.

Tabla 4.14

REQ.F.011: Eliminación de documentos de los trabajadores

REQ.F.011	Eliminación de documentos de los trabajadores
Descripción	Eliminación de los documentos en la base de datos.
Perfiles	Administrador
Observaciones	Un administrador debe poder eliminar el documento asociado al trabajador. Clientes y trabajadores no pueden realizar esta acción.

Tabla 4.15

REQ.F.012: Visualización de documentos de los trabajadores

REQ.F.012	Visualización de documentos de los trabajadores
Descripción	Visualización de documentos mediante enlace de Google Drive y visualización de datos.
Perfiles	Administrador, trabajador
Observaciones	Un administrador debe poder visualizar el documento asociado a cualquier trabajador. Un trabajador debe poder visualizar los documentos que están asociados a sí mismo, pero no los del resto. Los clientes no pueden visualizar estos documentos.

Tabla 4.16

REQ.F.013: Creación de proyectos

REQ.F.013	Creación de proyectos
Descripción	Creación de proyectos asociados a un cliente
Perfiles	Administrador
Observaciones	Un administrador debe poder crear un proyecto y asociarlo a un cliente concreto. Trabajadores y clientes no deben poder realizar esta acción.

Tabla 4.17

REQ.F.014: Edición de proyectos

REQ.F.01.4	Edición de proyectos
Descripción	Edición del título y la descripción de un proyecto
Perfiles	Administrador
Observaciones	Un administrador debe poder editar los datos de un proyecto. Clientes y trabajadores no podrán.

Tabla 4.18

REQ.F.015: Eliminación de proyectos

REQ.F.015	Eliminación de proyectos
Descripción	Eliminación de proyecto en la base de datos
Perfiles	Administrador
Observaciones	Un administrador debe poder eliminar un proyecto de la base de datos. Clientes y trabajadores no podrán.

Tabla 4.19

REQ.F.016: Visualización de proyectos

REQ.F.016	Visualización de proyectos
Descripción	Visualización de datos de los proyectos.
Perfiles	Administrador, cliente
Observaciones	Un administrador debe poder visualizar cualquier proyecto. El cliente solo podrá visualizar el proyecto que le haya asociado. Trabajadores no podrán acceder a los proyectos.

Tabla 4.20

REQ.F.017: Creación de documentos de un proyecto

REQ.F.017	Creación de documentos de un proyecto
Descripción	Subida de documentos mediante la API de Google Drive.
Perfiles	Administrador
Observaciones	Un administrador debe poder subir documentos desde el panel, añadiendo nombre y el tipo de documento (factura, presupuesto u otro). Se debe compartir el enlace del archivo en drive con el cliente al que pertenece el proyecto, para su posterior visualización. Los clientes y trabajadores no pueden subir documentos.

Tabla 4.21

REQ.F.018: Edición de documentos de un proyecto

REQ.F.018	Edición de documentos de un proyecto
Descripción	Edición de los datos del documento (nombre y tipo) y sustitución del documento.
Perfiles	Administrador
Observaciones	Un administrador debe poder sustituir el documento actual desde el panel. Debe poder editar el nombre y el tipo de documento (factura, presupuesto u otro). Se debe compartir el enlace del nuevo archivo en drive con el cliente al que pertenece el proyecto, para su posterior visualización. Los clientes no pueden editar documentos.

Tabla 4.22

REQ.F.019: Eliminación de documentos de un proyecto

REQ.F.019	Eliminación de documentos de un proyecto
Descripción	Eliminación de los documentos en la base de datos.

Perfiles	Administrador
Observaciones	Un administrador debe poder eliminar el documento asociado al trabajador. Clientes y trabajadores no pueden realizar esta acción.

Tabla 4.23

REQ.F.020: Visualización de documentos de un proyecto

REQ.F.020	Visualización de documentos de un proyecto
Descripción	Visualización de documentos mediante enlace de Google Drive y visualización de datos.
Perfiles	Administrador, cliente
Observaciones	Un administrador debe poder visualizar el documento asociado a cualquier proyecto. Un cliente debe poder visualizar los documentos que pertenezcan a un proyecto que está asociado a sí mismo, pero no los del resto. Los trabajadores no pueden visualizar estos documentos.

Tabla 4.24

REQ.F.021: Creación de imágenes y archivos de un proyecto

REQ.F.021	Creación de imágenes y archivos de un proyecto
Descripción	Subida de imágenes y otros mediante la API de Google Drive.
Perfiles	Administrador
Observaciones	Un administrador debe poder subir documentos desde el panel. Los clientes y trabajadores no pueden subir documentos.

Tabla 4.25

REQ.F.022: Eliminación de imágenes y archivos de un proyecto

REQ.F.022	Eliminación de imágenes y archivos de un proyecto
Descripción	Eliminación de imágenes y otros en la base de datos.
Perfiles	Administrador
Observaciones	Un administrador debe poder eliminar documentos asociados a cualquier proyecto. Clientes y trabajadores no pueden realizar esta acción.

Tabla 4.26

REQ.F.023: Visualización de imágenes y archivos de un proyecto

REQ.F.023	Visualización de imágenes y archivos de un proyecto
Descripción	Visualización de imágenes en la web y archivos mediante enlace de Google Drive.
Perfiles	Administrador, cliente
Observaciones	Un administrador debe poder visualizar imágenes y documentos asociados a cualquier proyecto. Un cliente debe poder visualizar los archivos que pertenezcan a un proyecto que está asociado a sí mismo, pero no los del resto. Los trabajadores no pueden visualizar estos documentos.

Tabla 4.27

REQ.F.024: Prohibición de registro

REQ.F.024	Prohibición de registro
Descripción	Nadie puede registrarse en la web si no tiene acceso. Todos los usuarios serán creados desde dentro.
Perfiles	-
Observaciones	Para ello se genera el primer administrador desde la base de datos.

Tabla 4.28

4.4. Requisitos no funcionales

Los requisitos funcionales no se refieren directamente a funciones específicas del sistema, sino a sus propiedades.

REQ.NF.001: Implementación de base de datos

REQ.NF.001	Implementación de base de datos
Descripción	Creación de base de datos para guardar toda la información requerida por el sistema.
Observaciones	La base de datos debe tener coherencia y relaciones apropiadas entre sus tablas.

Tabla 4.29

REQ.NF.002: Conexión con la base de datos

REQ.NF.002	Conexión con la base de datos
Descripción	El sistema necesita una correcta conexión con la base de datos para funcionar correctamente.
Observaciones	Si el sistema se ejecuta localmente, es necesario disponer de un servidor local que pueda alojar la base de datos.

Tabla 4.30

REQ.NF.003: Conexión con la Google Drive

REQ.NF.003	Conexión con Google Drive
Descripción	El sistema necesita la conexión con Google Drive, para la correcta subida y visualización de archivos.
Observaciones	Los usuarios deben disponer de un correo electrónico de Gmail para la visualización y descarga de documentos.

Tabla 4.31

REQ.NF.004: Disposición de un administrador

REQ.NF.004	Disposición de un administrador
Descripción	La base de datos necesita disponer de un administrador inicial, con correo electrónico y contraseña, para el inicio de sesión en la plataforma y la posterior creación de otros usuarios desde la plataforma.
Observaciones	El programa no dispone de registro por petición del cliente, por lo que es imprescindible asignar un administrador desde la base de datos.

Tabla 4.32

REQ.NF.005: Ejecución de backend y frontend

REQ.NF.005	Ejecución de backend y frontend
Descripción	El frontend requiere del backend para su funcionamiento. En caso contrario, el inicio de sesión no será posible, y por lo tanto no se podrá acceder a ninguna vista.
Observaciones	Si se ejecuta el frontend, cargará la vista de inicio de sesión, pero no será posible realizar esta acción pues la verificación de usuarios no dispondrá de los datos necesarios.

Tabla 4.33

REQ.NF.006: Usuario y contraseña de usuario de base de datos

REQ.NF.006	Usuario y contraseña de usuario de base de datos
Descripción	El acceso a la base de datos se hará con autenticación de usuario.
Observaciones	Si la base de datos se introduce en un servidor local sin especificar el usuario que tendrá acceso, la conexión no se podrá realizar.

Tabla 4.34

REQ.NF.007: Usuario para la consola de Google Cloud

REQ.NF.007	Usuario para la consola de Google Cloud
Descripción	Se requiere un usuario con acceso a la consola de Google Cloud para utilizar la API de Google Drive.
Observaciones	Google Cloud creará un usuario para el proyecto, con unas claves necesarias para la conexión de la app y la carpeta de Google Drive.

Tabla 4.35

REQ.NF.008: Creación de carpeta en Google Drive compartida con el usuario de Google Drive

REQ.NF.008	Creación de carpeta en Google Drive compartida con el usuario de Google Drive
Descripción	Se requiere la creación de una carpeta que será el destino de los archivos subidos en la plataforma web. Se requiere compartir permisos de edición con el usuario de Google Cloud.
Observaciones	Los archivos que se suban a la web, serán subidos a la carpeta de Google Drive con el usuario de Google Drive, que se considerará el autor del archivo.

Tabla 4.36

REQ.NF.009: Necesidad de un Servidor

REQ.NF.009	Necesidad de un Servidor
Descripción	Se requiere disponer de un servidor para almacenar la base de datos y realizar la conexión.
Observaciones	Los datos se toman de la base de datos que necesita estar alojada en un servidor, ya sea local o virtual.

Tabla 4.37

5. Diseño del proyecto

Este proyecto consta de 14 vistas principales, en base a las cuáles, se han generado los distintos diagramas que se exponen a continuación:

1. Inicio de sesión
2. Escritorio
3. Proyecto
4. Administradores
5. Clientes
6. Trabajadores
7. Documentos del trabajador
8. Facturas y presupuestos del cliente
9. Formulario de Proyecto
10. Formulario de administrador
11. Formulario de cliente
12. Formulario de trabajador
13. Formulario de documento
14. Formulario de facturas y presupuestos

En este apartado se exponen los distintos esquemas que refieren al desarrollo de la aplicación web.

5.1. Diseño arquitectónico

El diseño arquitectónico está ligado con las metas establecidas para una aplicación web, con el contenido que se va a presentar, los usuarios que la visitarán y la filosofía de navegación adoptada. (Ingeniería Software)

La arquitectura de la aplicación web hace referencia a la forma en que la aplicación queda estructurada para administrar la interacción con el usuario.

El diseño arquitectónico de esta web parte del escritorio, donde se encontrará el usuario al iniciar sesión. Los primeros niveles de acceso se encontrarán en el menú principal.

Una vez el usuario se sitúe en uno de estos niveles podrá seguir navegando siguiendo las ramificaciones tal y como se presentan en las siguientes imágenes.

Diseño arquitectónico:

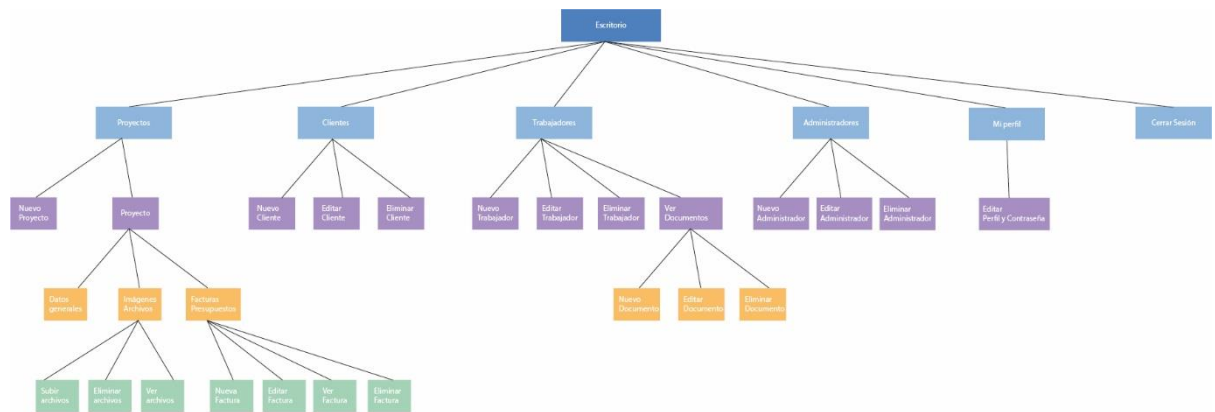


Ilustración 5.1

Diseño arquitectónico ampliado parte 1:

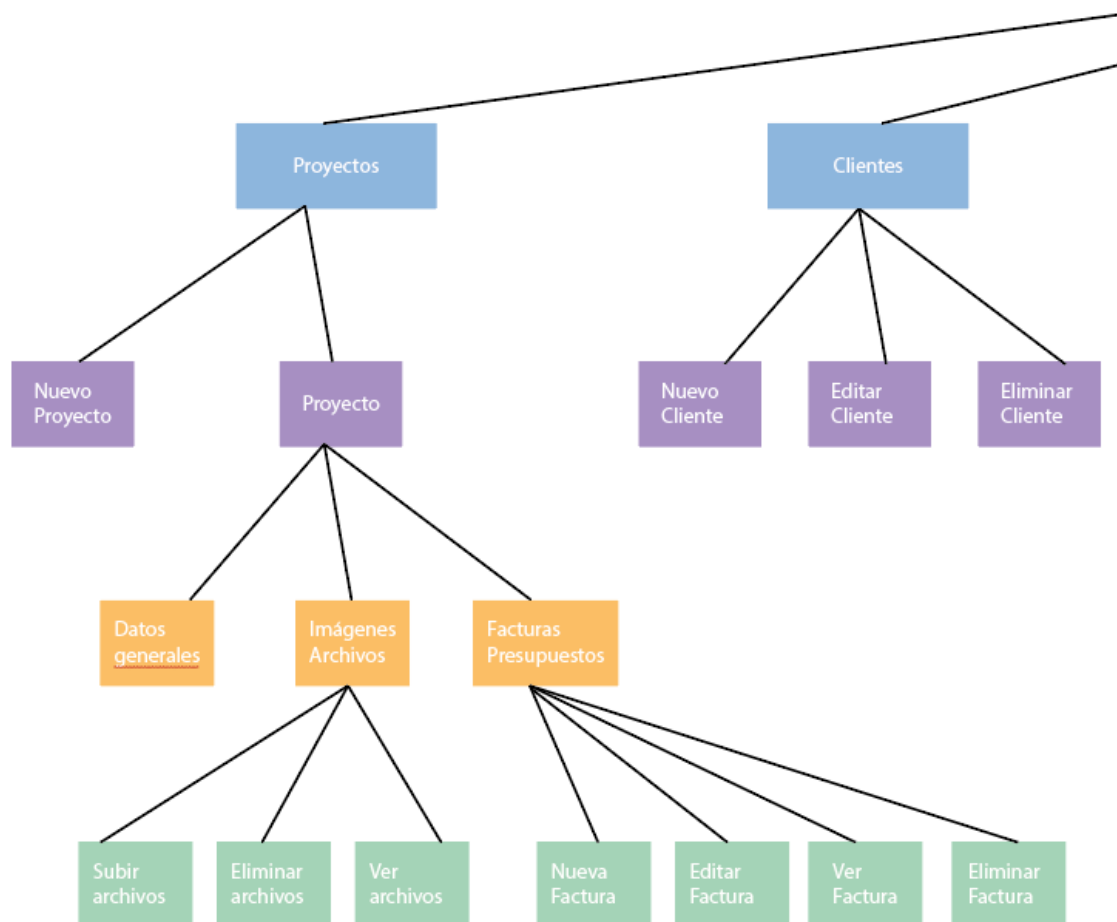


Ilustración 5.2

Diseño arquitectónico ampliado parte 2:

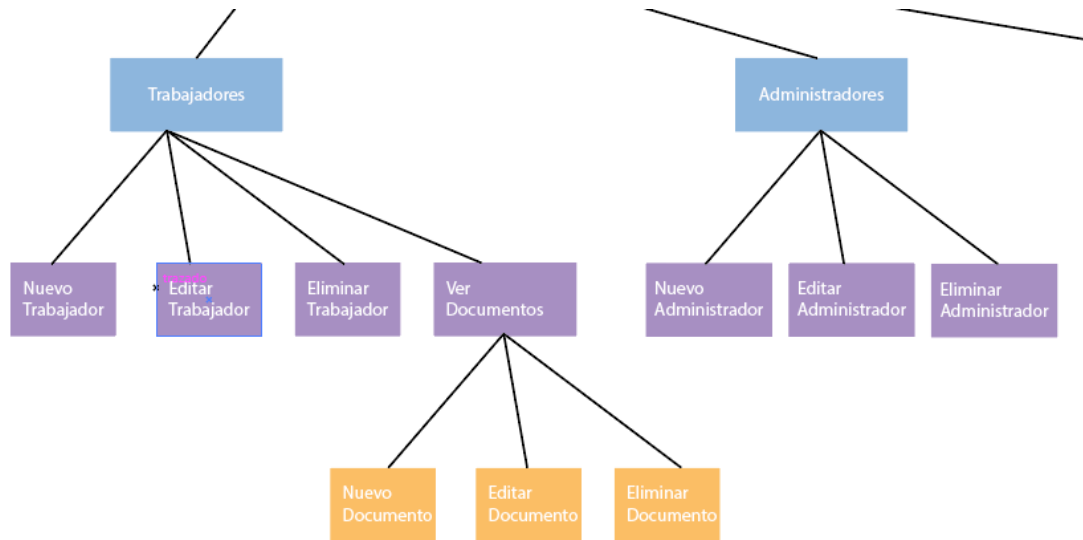


Ilustración 5.3

Diseño arquitectónico ampliado parte 3:

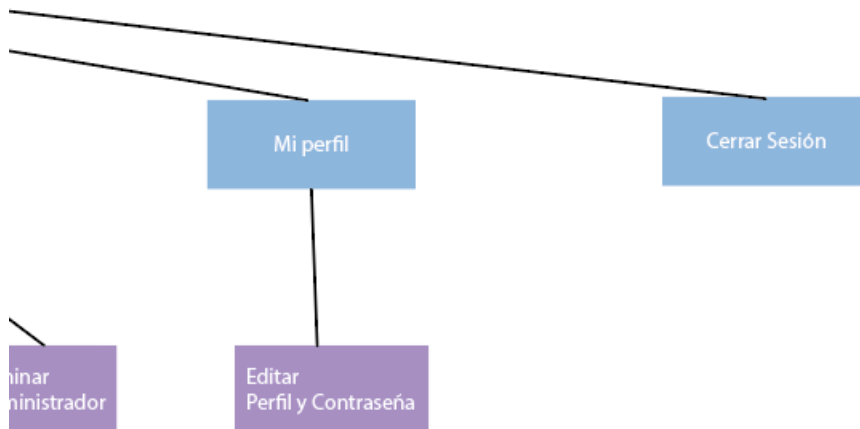


Ilustración 5.4

5.2. Diagrama de casos de uso

El diagrama de casos de uso es una forma de diagrama de comportamiento en lenguaje de modelado unificado (UML, del inglés Unified Modelling Language), con

la que se representan procesos empresariales, así como sistemas y procesos de programación orientada a objetos. (Ionos)

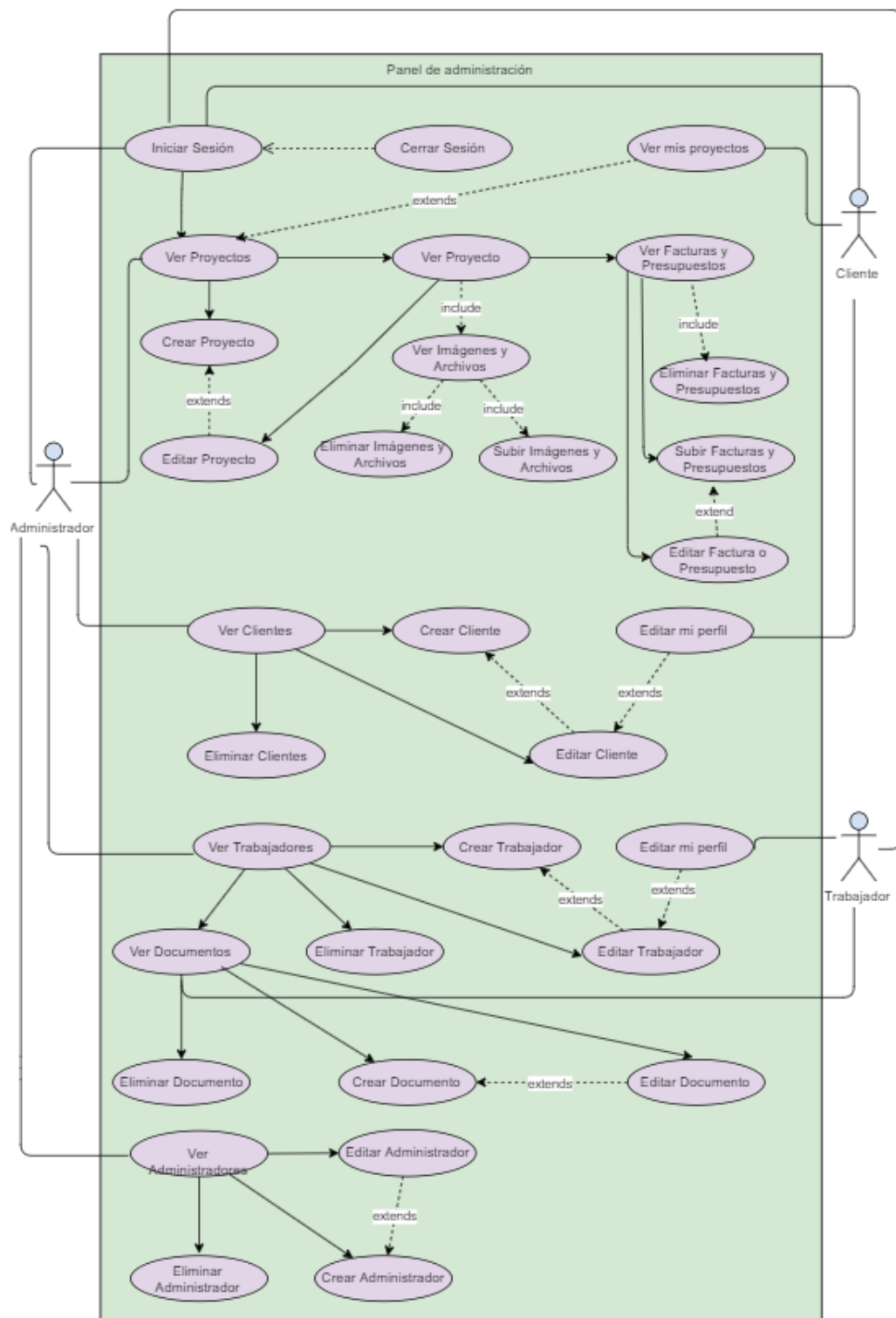
Los elementos del diagrama son⁴⁸:

- Actor: se representa con el dibujo de una figura humana esquemática.
- Sistema: el sistema al que se refiere el caso de uso tiene forma de rectángulo.
- Caso de uso: se muestra como una elipse que suele incluir un texto describiendo brevemente el proceso.

En esta aplicación, existen tres actores diferentes: administrador, cliente y trabajador. El administrador puede realizar todos los casos de uso. Trabajadores y clientes tienen limitaciones, como podemos ver a continuación:

Diagrama de casos de uso:

⁴⁸ <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>



5.3. Diseño de Secuencia

El diagrama de secuencia es un tipo de diagrama del lenguaje unificado de modelado (UML) que, a su vez, se trata de un lenguaje orientado a objetos y está compuesto por elementos gráficos. (lonos, s.f.)

A continuación, vemos un diagrama de secuencia por cada actor, ya que las acciones de cada usuario determinarán de una u otra forma dependiendo de sus permisos.

Por ejemplo, los tres comienzan en el inicio de sesión si aún no se han identificado en la plataforma. Tras iniciar sesión, tanto el administrador como el cliente continuarán en la ventana del escritorio, pero las llamadas al servidor para obtener los datos serán diferentes en uno y otro, ya que el administrador puede acceder a todos los proyectos y el cliente solamente a los que están asociados a él.

El trabajador, por su parte, continuará en la ventana de sus documentos, debido a que este usuario no tiene permisos para recibir información sobre ningún proyecto.

Diagrama de secuencia parte 1:

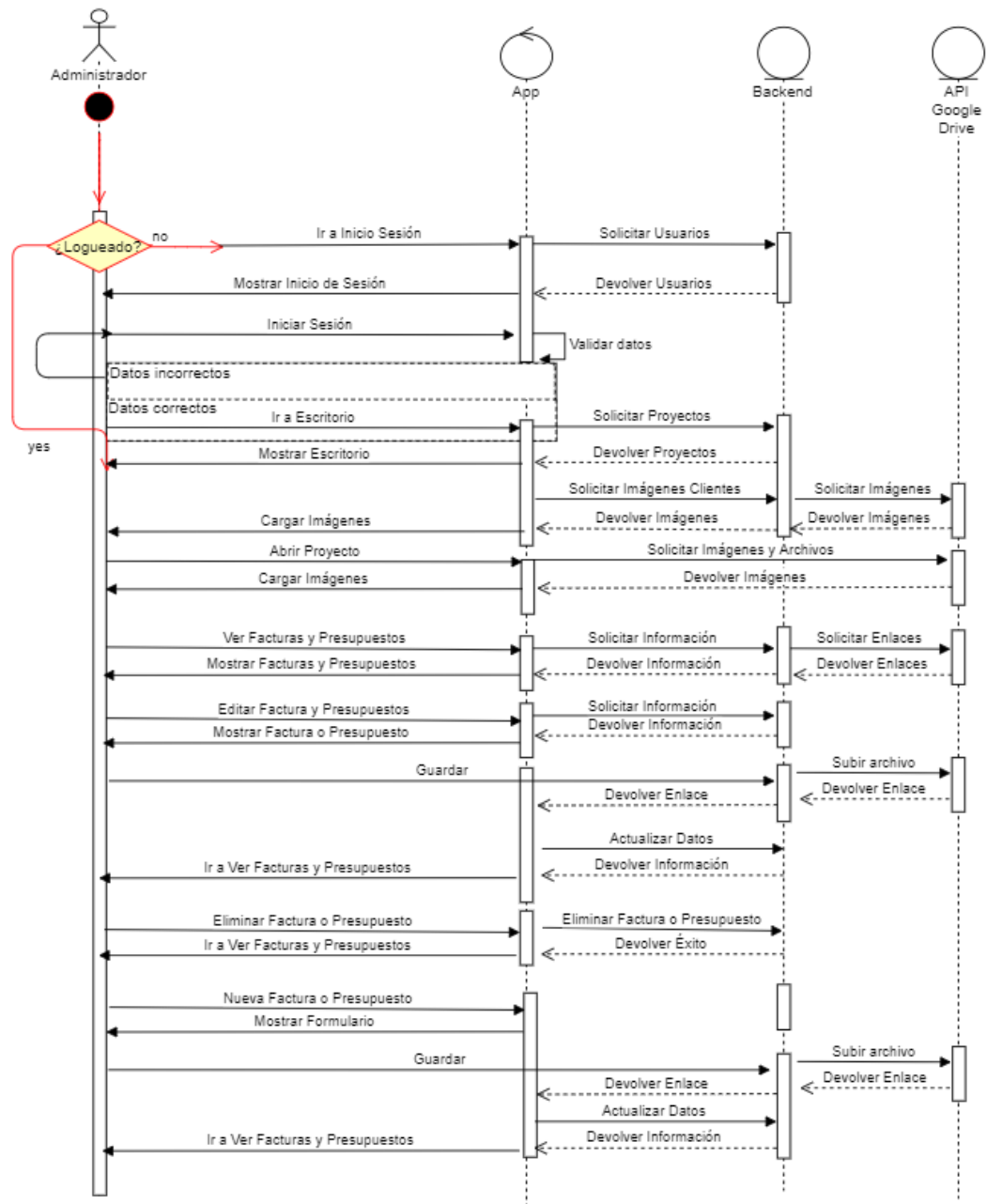


Ilustración 5.6

Diagrama de secuencia parte 2:

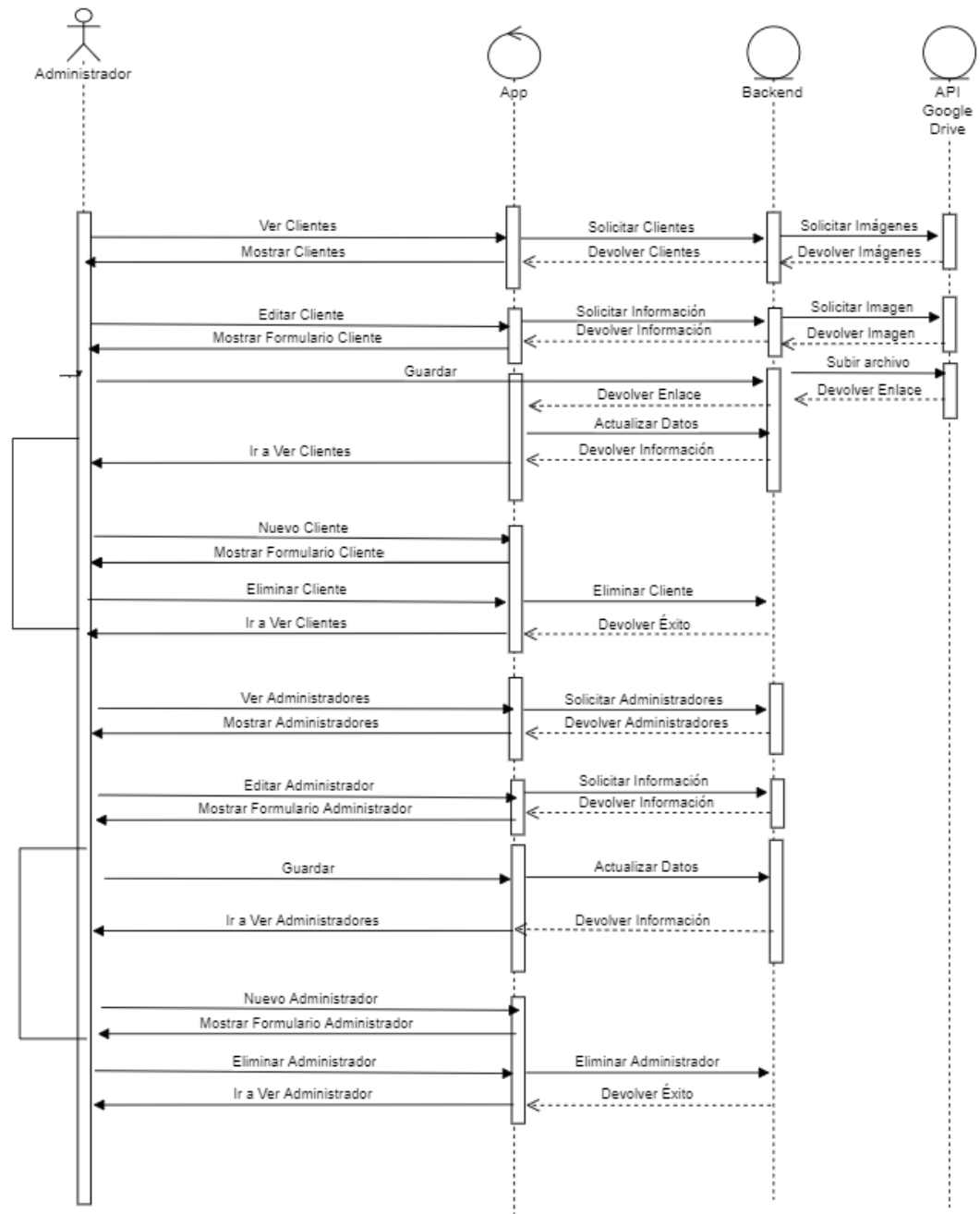


Ilustración 5.7

Diagrama de secuencia parte 3:

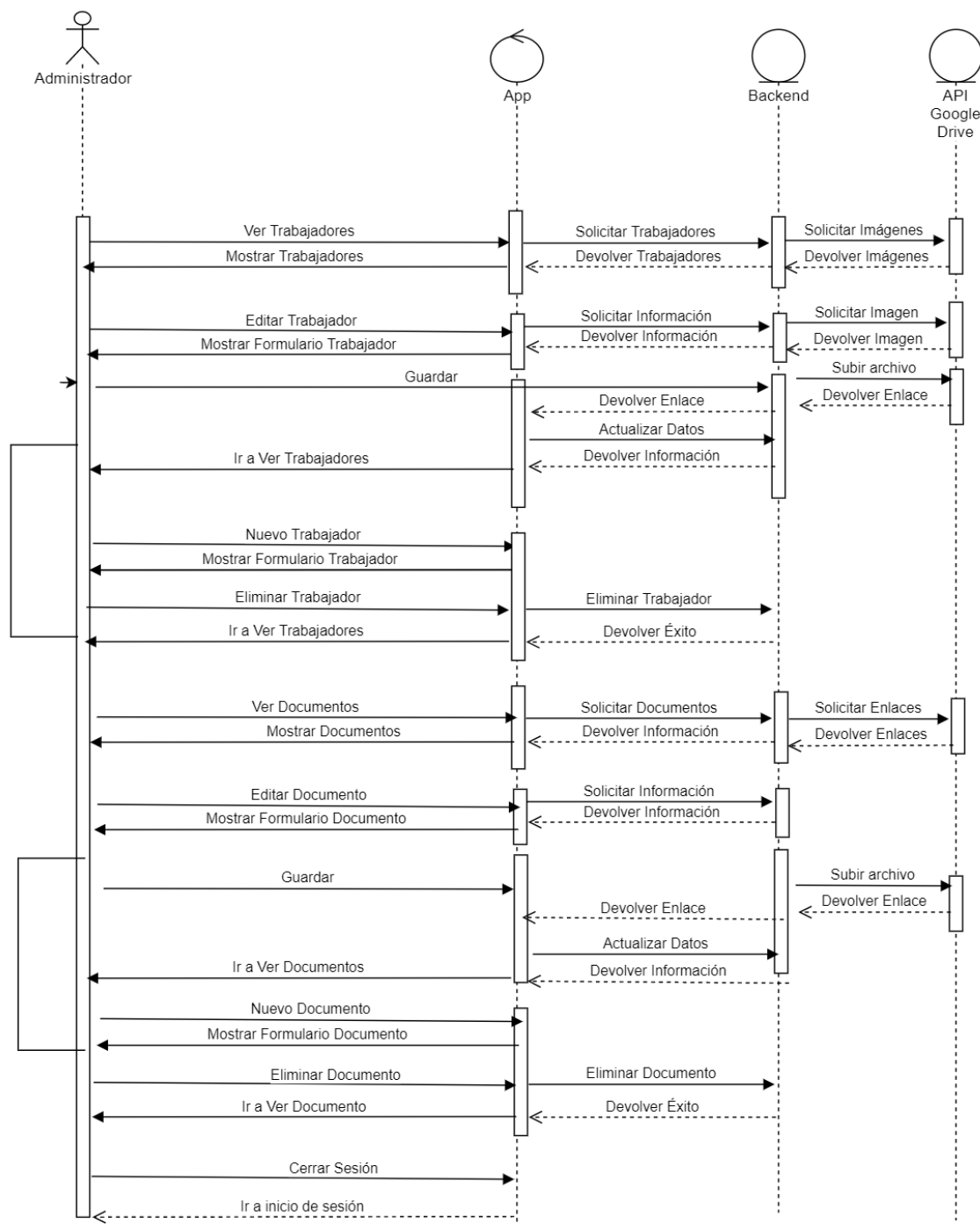


Ilustración 5.8

Diagrama de secuencia parte 4:

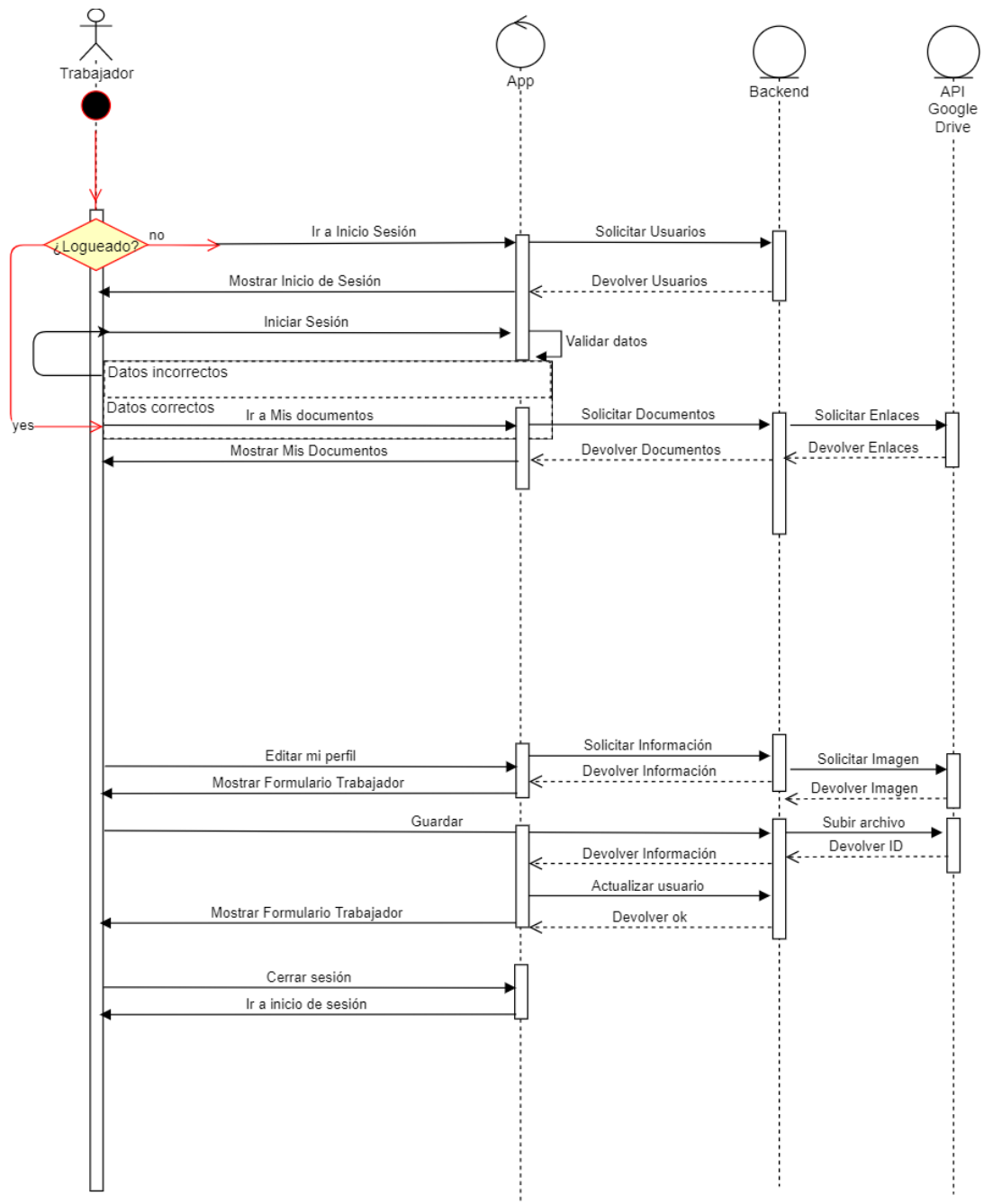


Ilustración 5.9

Diagrama de secuencia parte 5:

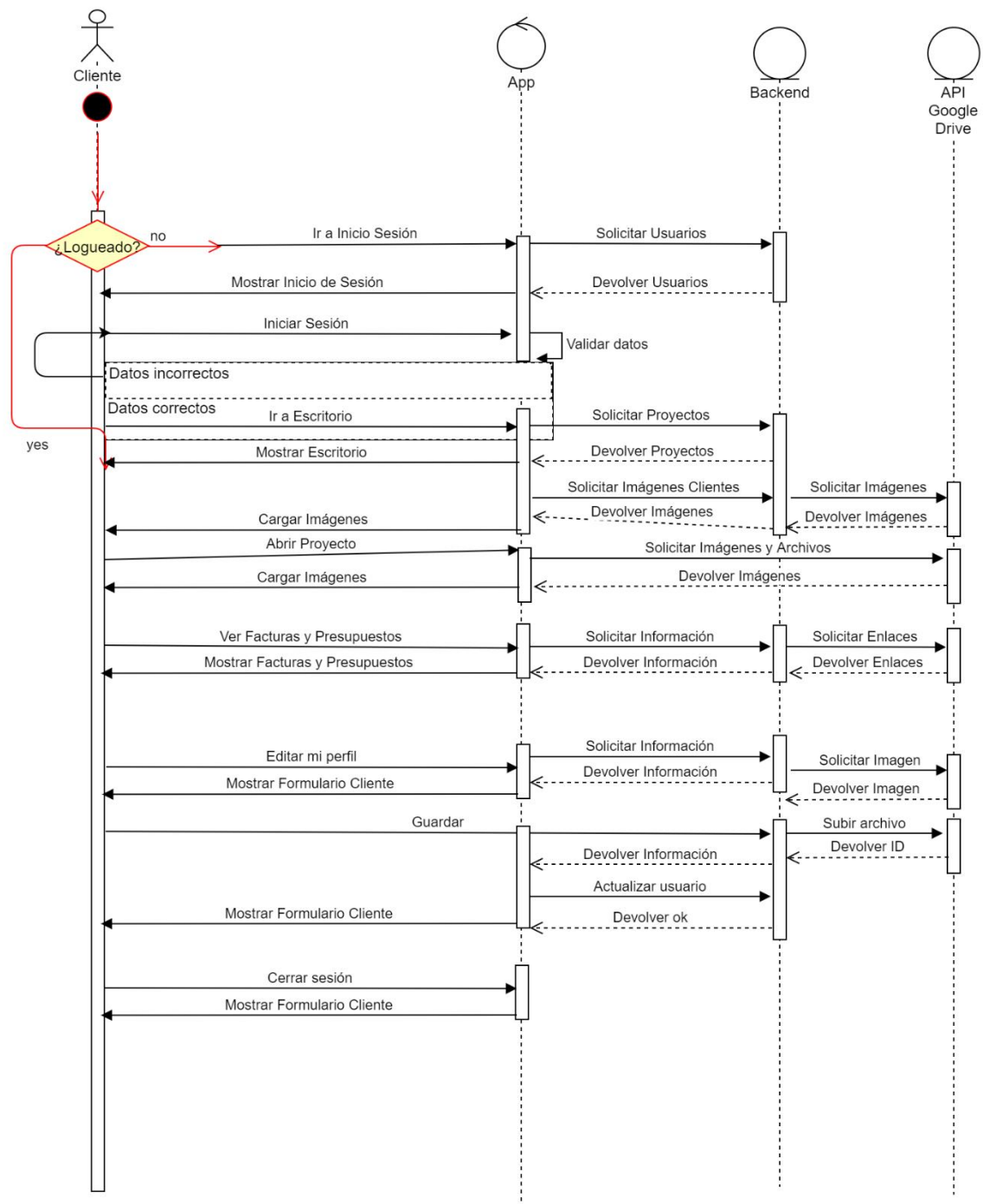


Ilustración 5.10

5.4. Modelo de datos

5.4.1. Modelo Entidad-Relación

El modelo entidad-relación es un tipo de diagrama de flujo que ilustra cómo las “entidades” se relacionan entre sí dentro de un sistema.

En este modelo se observa como partimos de un usuario, del que heredan los tres tipos: administrador, cliente y trabajador. Existen 4 relaciones, y todas son de uno a muchos:

- Relación 1: Un cliente puede tener muchos proyectos, pero un proyecto solo puede pertenecer a un cliente.
- Relación 2: A un trabajador le pueden pertenecer muchos documentos, pero un documento solo pertenece a un trabajador.
- Relación 3: Un proyecto puede tener muchos documentos, pero un documento solo pertenece a un proyecto
- Relación 4: Un proyecto puede tener muchos archivos, pero un archivo solo pertenece a un proyecto.

Modelo Entidad-Relación:

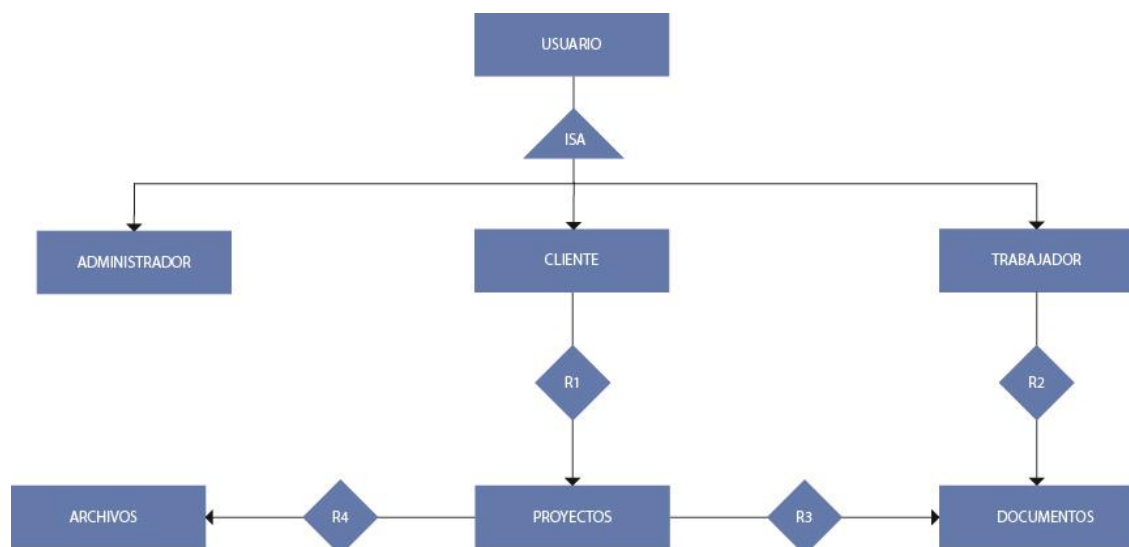


Ilustración 5.11

5.4.2. Modelo Relacional

El modelo relacional consiste en representar datos por medio de tablas relacionadas cuyas filas se llaman tuplas y las columnas, variables. De esta forma se conforma la base de datos.

A continuación, se muestra el Modelo Relacional, que se genera a partir del Modelo ER. Se transforman las herencias en tablas independientes y se muestran también los atributos.

Modelo Relacional:

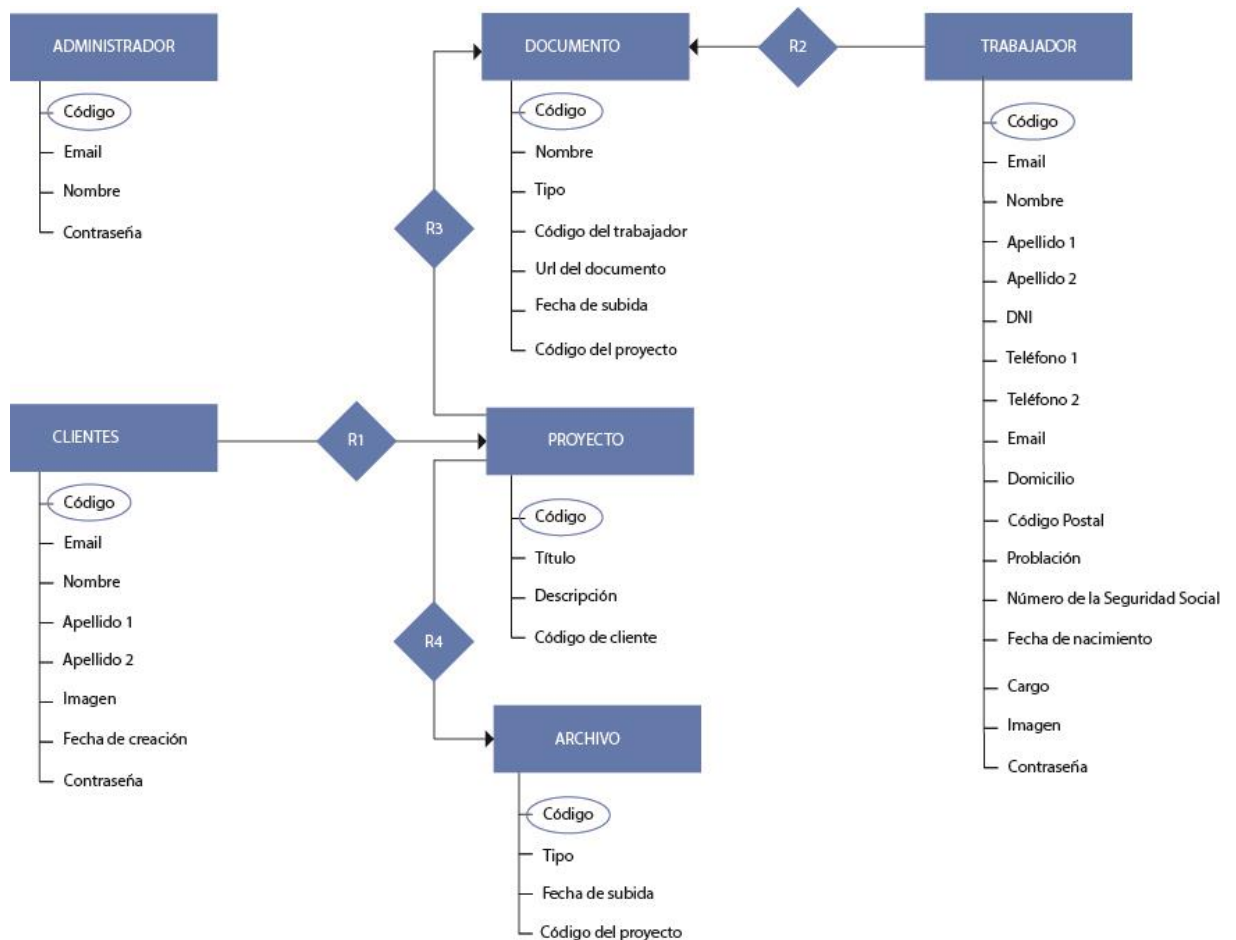


Ilustración 5.12

5.4.3. Modelo Relacional MySQL

Tras insertar las tablas en la base de datos, se obtiene el siguiente resultado:

Modelo Relacional phpMyAdmin:

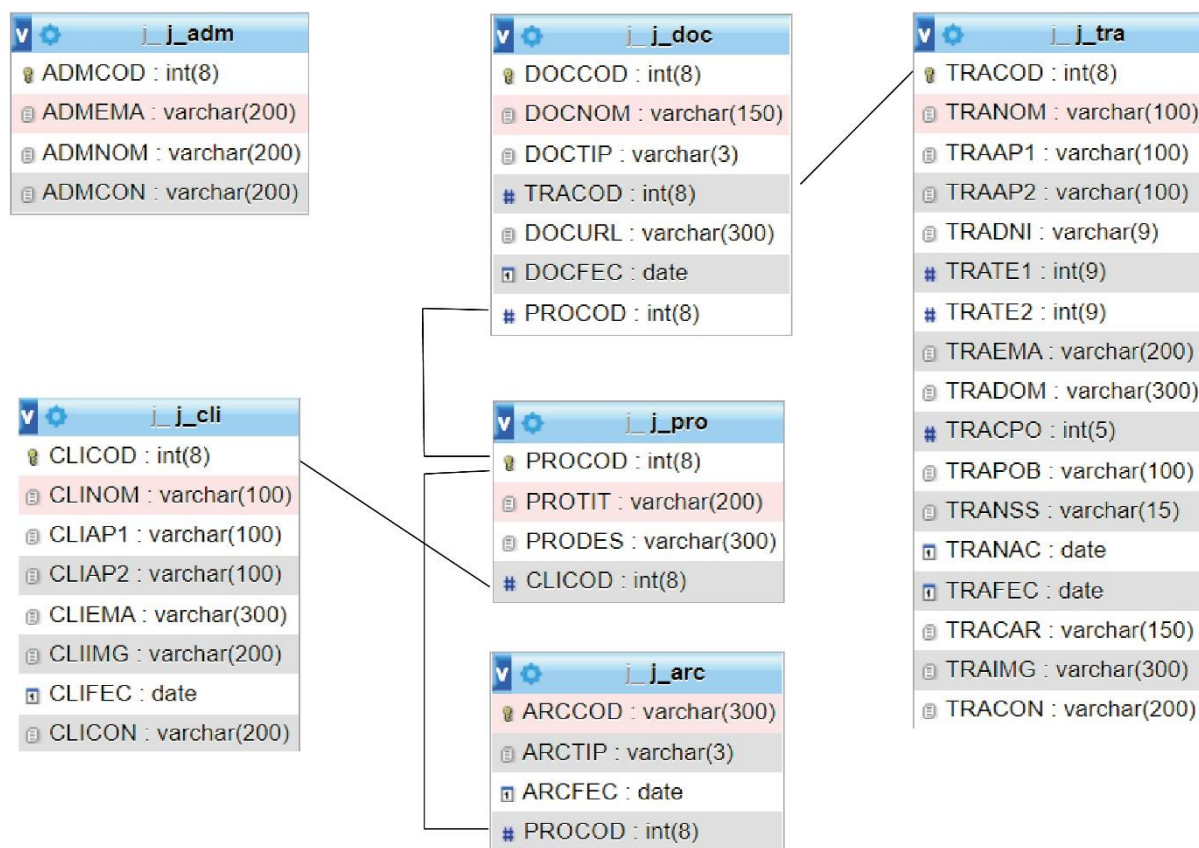


Ilustración 5.13

5.4.4. Definición del modelo

La base de datos consta de 6 tablas (ver anexo 1):

- j_adm: Tabla de administradores
- j_arc: Tabla de archivos de los proyectos
- j_cli: Tabla de clientes
- j_doc: Tabla de documentos de los trabajadores
- j_pro: Tabla de proyectos de los clientes
- j_tra: Tabla de trabajadores

Los nombres de las tablas constan de la inicial de la base de datos, y de 3 letras. Esta forma de denominar las tablas, evita futuros errores en la programación, debido a que, a mayor número de letras, mayor posibilidad de error, y, por lo tanto, mayor tiempo de ejecución.

Siguiendo este método de nombrado, cada uno de los atributos de las tablas, se nombra con las 3 iniciales de la tabla a la que pertenece, y con 3 letras que definan al atributo.

De esta forma, tenemos, por ejemplo, la tabla de administradores: “J_ADM”, donde alguno de sus atributos son el código: “ADMCOD”, o el email del usuario: “ADMEMA”.

En el caso de claves foráneas, tomamos el nombre de la clave principal, para evitar confusiones y distintos nombrados por tabla.

Por ejemplo, en la tabla “J_DOC”, el código del proyecto (clave foránea) mantiene el nombre de la tabla original PROCOD.

5.5. Storyboard

El Storyboard se define como un conjunto de viñetas en el que se puede representar de forma gráfica, aunque sencilla, distintos elementos que ayudan a visualizar ideas y conceptos sobre cómo será la web y cómo podrá el usuario interactuar con ella.

Se ha generado el storyboard a través de Marvel App. Se puede visualizar (ver anexo 2) e interactuar mediante el siguiente enlace:

<https://marvelapp.com/prototype/8156g46>

5.6. Diseño web

El diseño web es un área que engloba el conjunto de actividades destinadas a la planificación y maquetación de una web, el desarrollo de interfaces o plataformas digitales, su diseño, el layout y todo lo referente a su contenido. (Dobuss, s.f.)

Para este proyecto, se realiza una selección de plantillas que facilitarán el desarrollo. En un principio, estas eran las 9 plantillas candidatas antes de elegir la final:

Plantillas descargadas:

Nombre	Fecha de modificación	Tipo
 architectui-react-free-1.0.0	30/01/2022 18:00	Carpeta de archivos
 argon-dashboard-react	13/02/2022 19:41	Carpeta de archivos
 argon-design-system-react-master	06/02/2022 19:53	Carpeta de archivos
 coreui-free-react-admin-template-main	12/04/2022 19:29	Carpeta de archivos
 crystal-dashboard	30/01/2022 18:13	Carpeta de archivos
 matx-react	06/02/2022 19:25	Carpeta de archivos
 react-reduction	06/02/2022 19:05	Carpeta de archivos
 shards-dashboard-react	06/02/2022 19:05	Carpeta de archivos
 sofia-react-template	20/02/2022 18:11	Carpeta de archivos

Ilustración 5.14

- ArchitectUI Bootstrap 5 ReactJS Theme FREE. Podemos consultar información sobre ella en el enlace:
 - <https://github.com/DashboardPack/architectui-react-theme-free.git>⁴⁹

Vista principal de la plantilla

⁴⁹ <https://github.com/DashboardPack/architectui-react-theme-free.git>

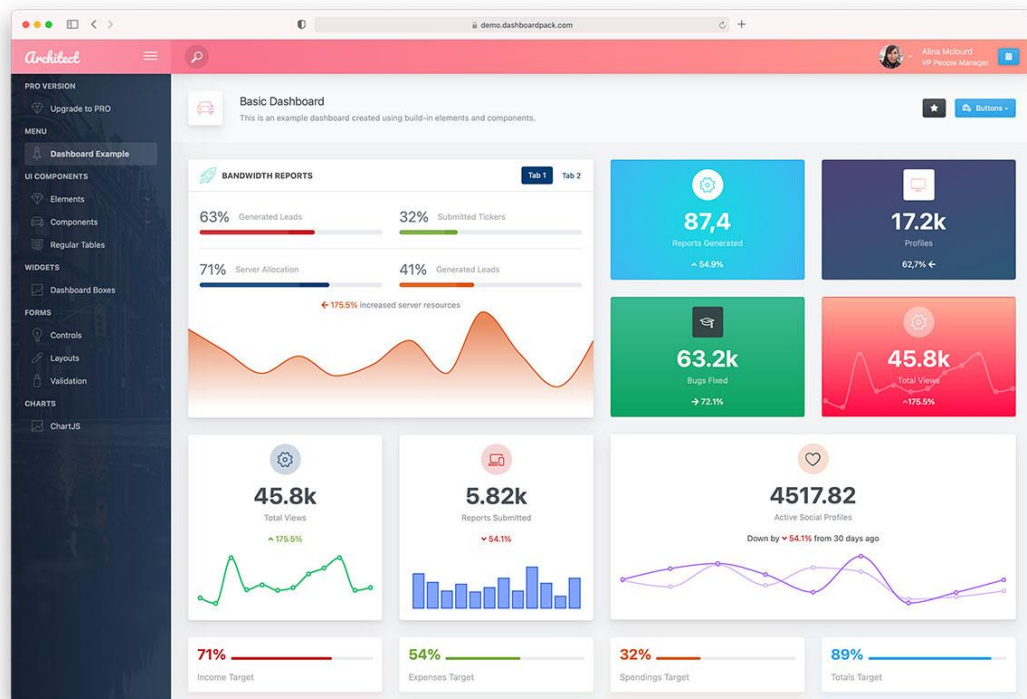


Ilustración 5.15

- Argon Dashboard React. Podemos consultar información sobre ella en el enlace:
 - <https://www.creative-tim.com/product/argon-dashboard-react>

Vista principal de la plantilla⁵⁰

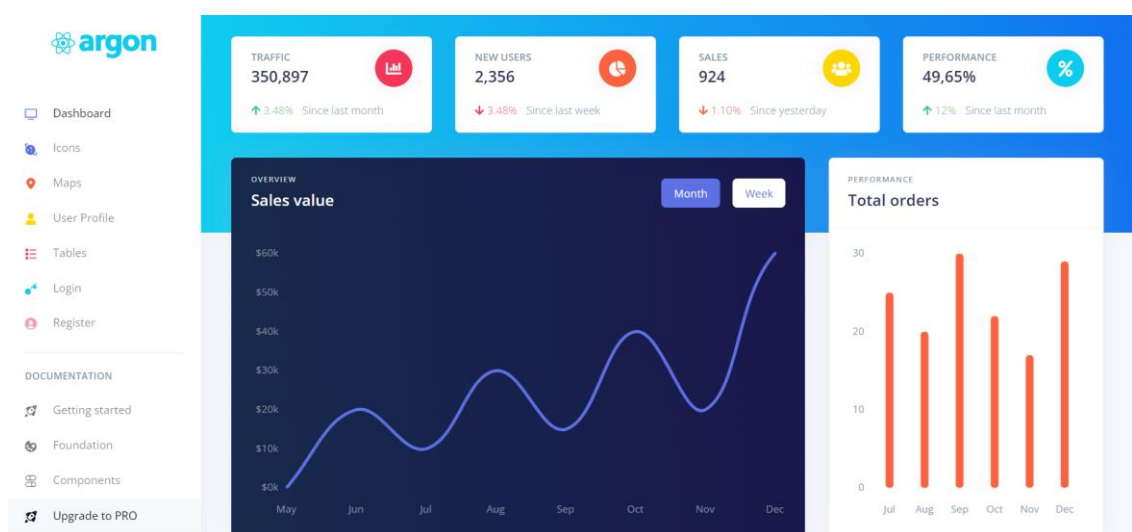


Ilustración 5.16

⁵⁰ https://demos.creative-tim.com/argon-dashboard-react/?_ga=2.100390174.1257038106.1662706702-2084690610.1662706702#/admin/index

- Argon Design System React. Podemos consultar información sobre ella en el enlace:
 - <https://www.creative-tim.com/product/argon-design-system-react>

Vista principal de la plantilla⁵¹

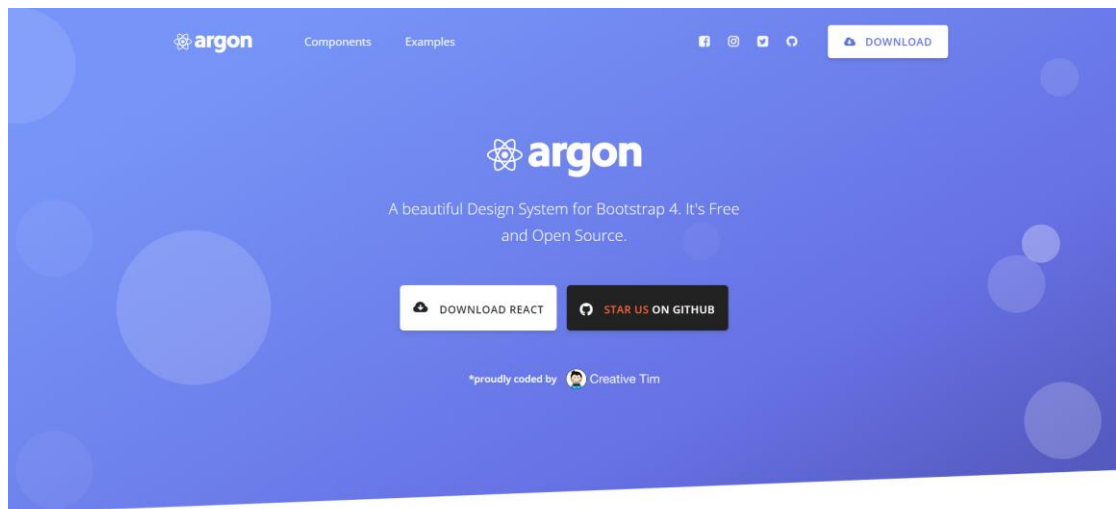


Ilustración 5.17

- CoreUI Free React Admin Template v4. Podemos consultar información sobre ella en el enlace:
 - <https://github.com/coreui/coreui-free-react-admin-template.git>

Vista general de la plantilla⁵²:

⁵¹ https://demos.creative-tim.com/argon-design-system-react/?_ga=2.88118939.1257038106.1662706702-2084690610.1662706702#/

⁵² <https://coreui.io/demos/angular/4.2/default-v3/#/dashboard>

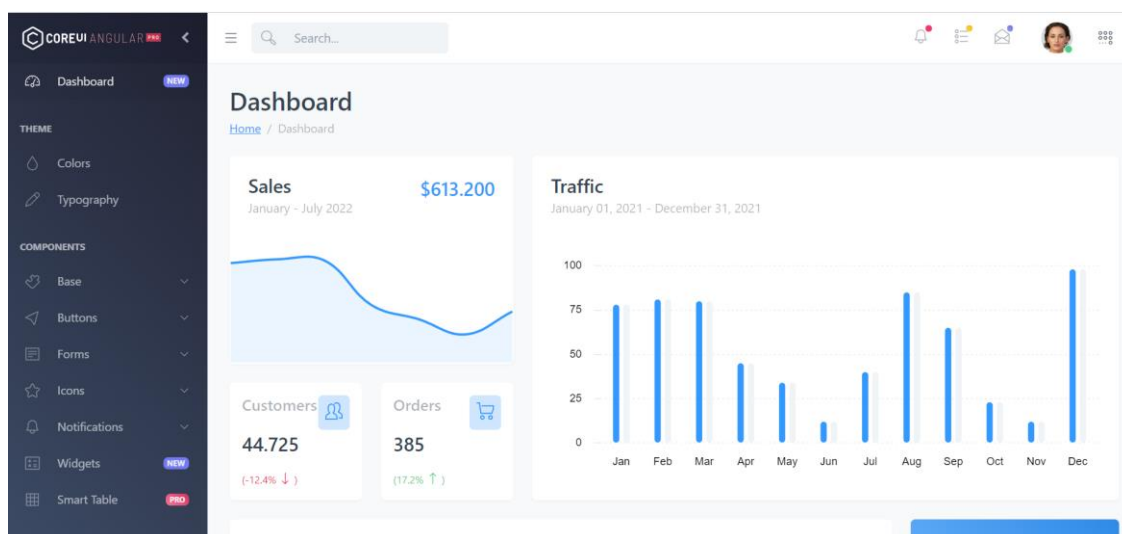


Ilustración 5.18

- Crystal React Bootstrap Dashboard. Podemos consultar información sobre ella en el enlace:
 - <https://github.com/JSLancerTeam/crystal-dashboard.git>

Vista general de la plantilla⁵³:

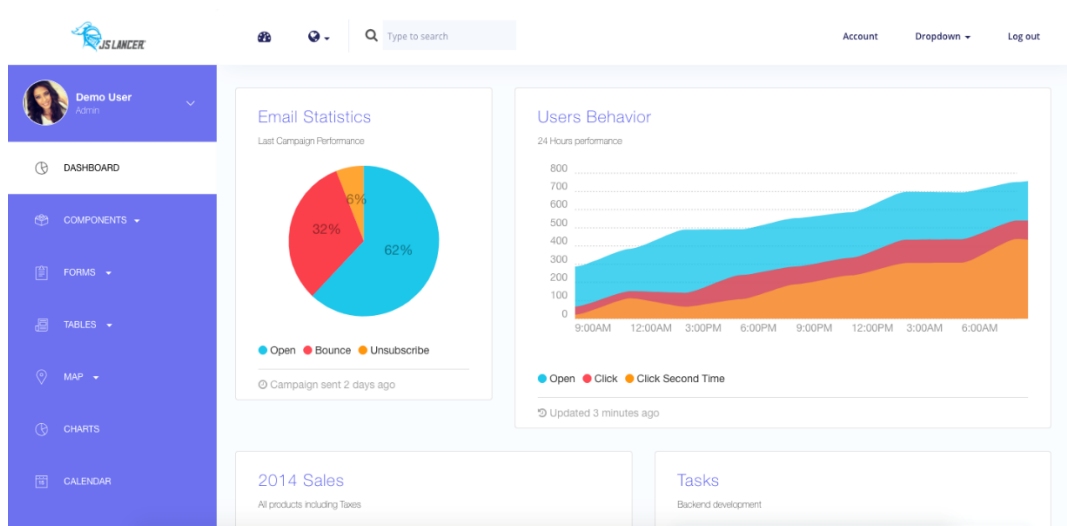


Ilustración 5.19

- Matx React Material Design Admin Dashboard Template. Podemos consultar información sobre ella en el enlace:
 - <https://github.com/uilibrary/matx-react.git>

⁵³ <https://github.com/JSLancerTeam/crystal-dashboard.git>

Vista general de la plantilla⁵⁴:

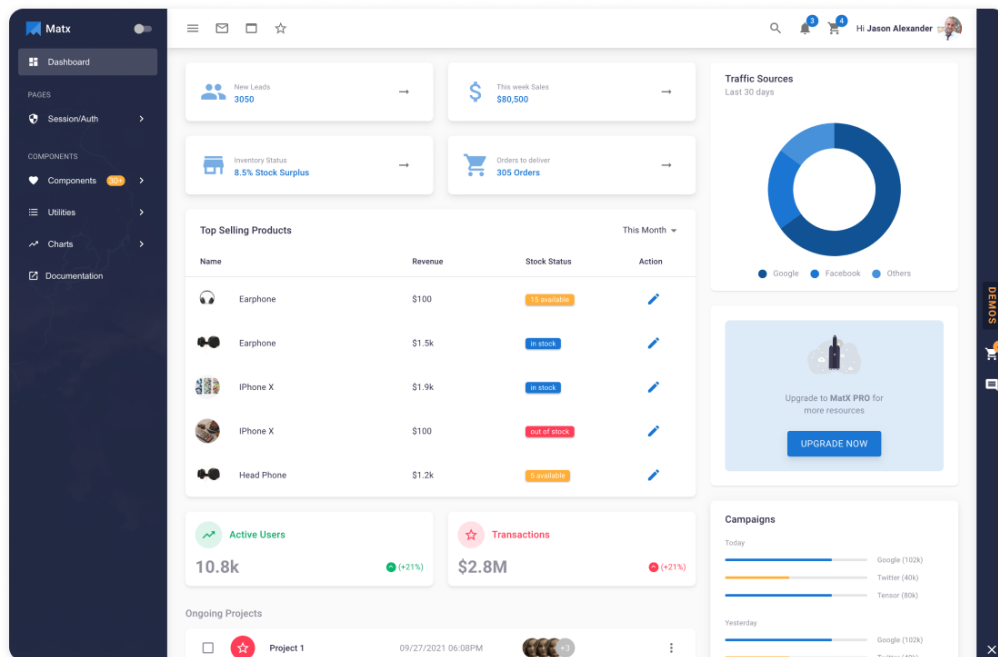


Ilustración 5.20

- React Reduction - Free Admin Template Built with React and Bootstrap4.
Podemos consultar información sobre ella en el enlace:
 - <https://github.com/reduction-admin/react-reduction>

Vista general de la plantilla⁵⁵:

⁵⁴ <https://github.com/uilibrary/matx-react>

⁵⁵ <https://github.com/reduction-admin/react-reduction>

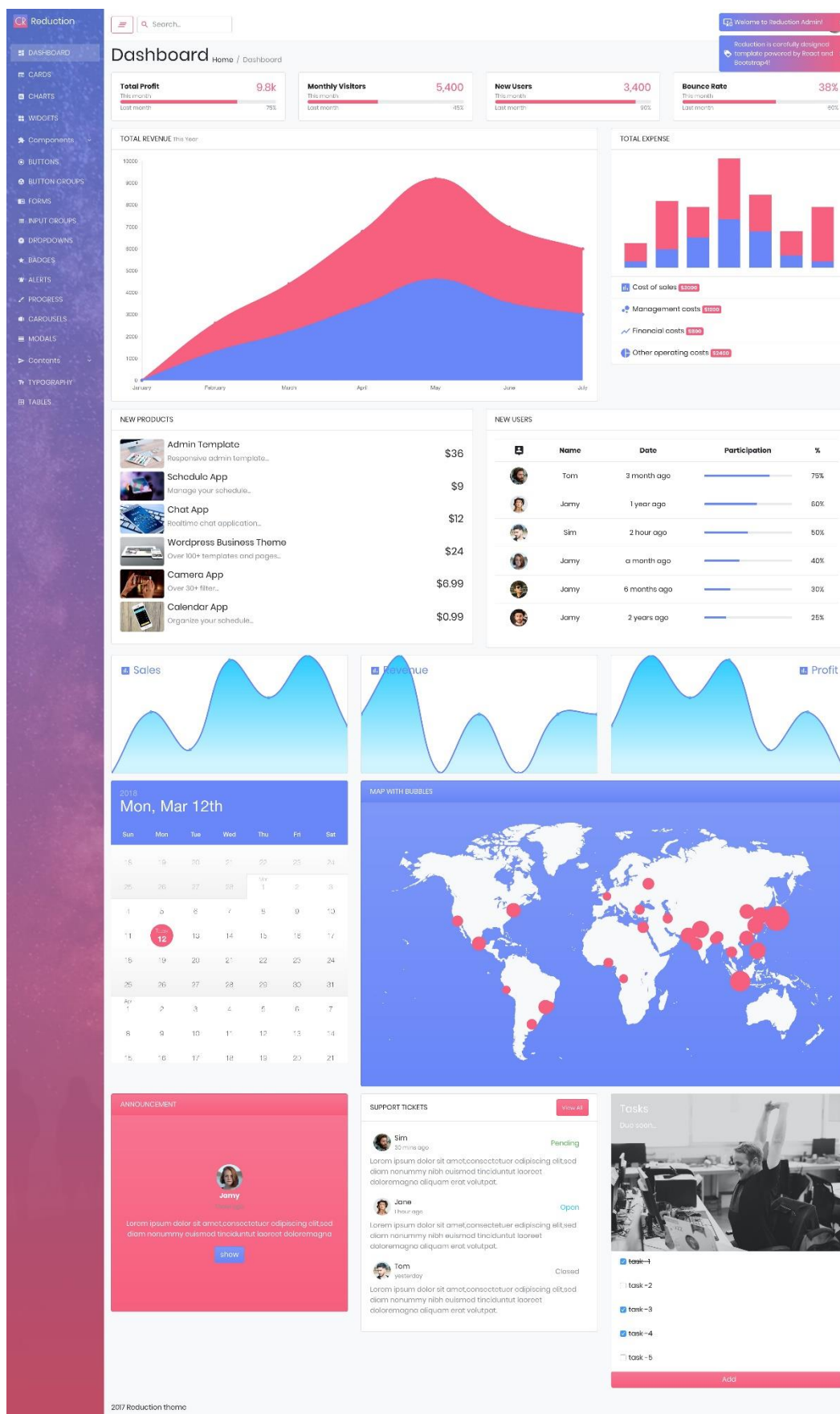


Ilustración 5.21

- Shards Dashboard Lite React. Podemos consultar información sobre ella en el enlace:
 - <https://designrevision.com/downloads/shards-dashboard-lite-react/>

Vista general de la plantilla⁵⁶:

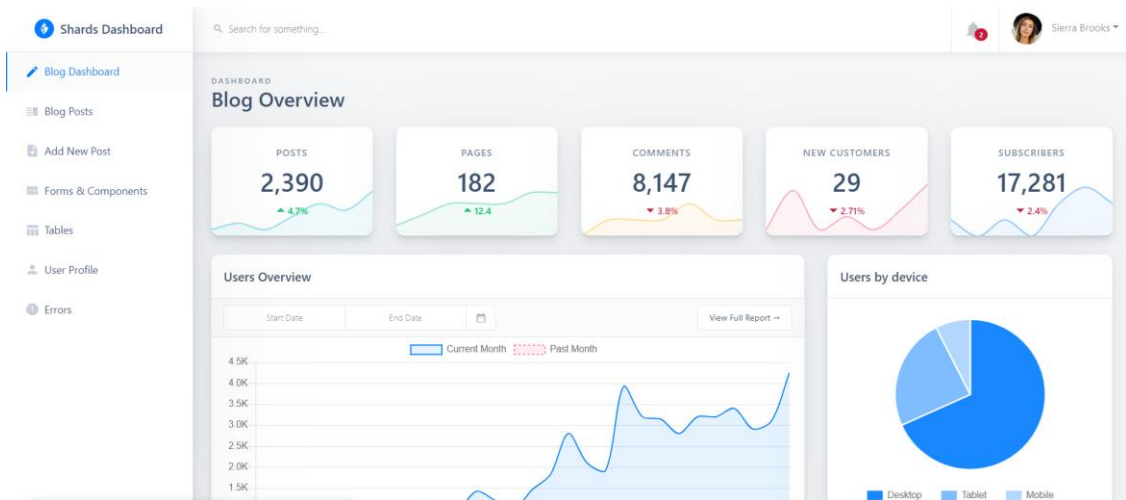


Ilustración 5.22

- Sofia React Template. Podemos consultar información sobre ella en el enlace:
 - <https://github.com/flatlogic/sofia-react-template.git>

Vista general de la plantilla⁵⁷:

⁵⁶ <https://designrevision.com/demo/shards-dashboard-lite-react/blog-overview>

⁵⁷ <https://github.com/flatlogic/sofia-react-template>



Ilustración 5.23

Finalmente, la plantilla seleccionada fue: Lucid – Responsive ReactJS Admin Template, cuya información encontramos en el enlace: <https://themeforest.net/item/lucid-react-admin-template/32023139#>.

Vista general de la plantilla⁵⁸:

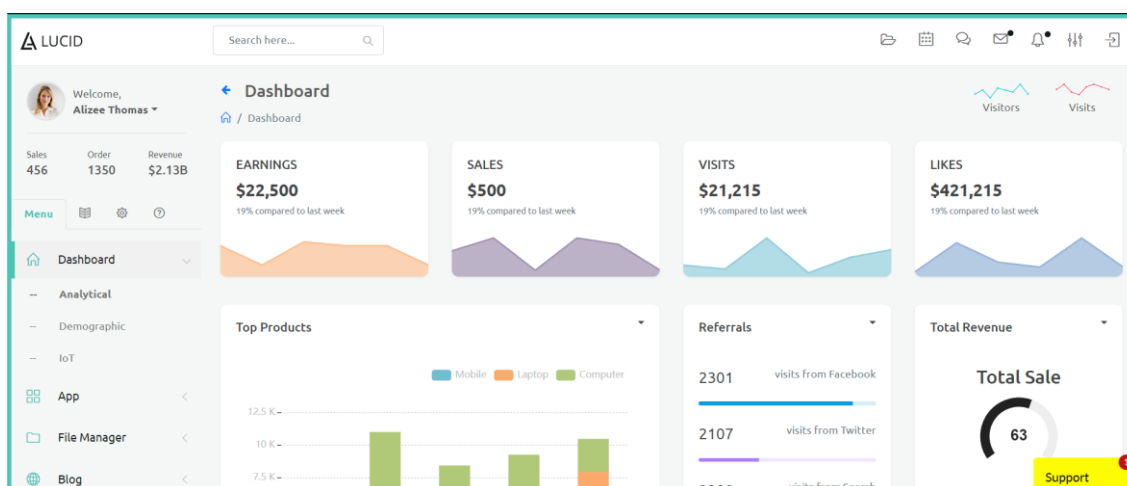


Ilustración 5.24

⁵⁸ https://preview.themeforest.net/item/lucid-react-admin-template/full_screen_preview/32023139?_ga=2.39699105.324811756.1662675051-807208824.1662675051

Esta decisión se tomó debido a que es la plantilla más completa, con vistas a un futuro desarrollo de la aplicación y que esta siga evolucionando. Para este proyecto se han usado algunos de los componentes que proporciona y también se han generados muchos nuevos, partiendo del mismo diseño.

6. DESARROLLO DEL PROYECTO

6.1. Estructura del código fuente

El código fuente presenta una estructura de 4 carpetas principales y 3 archivos:

Imagen de la estructura:

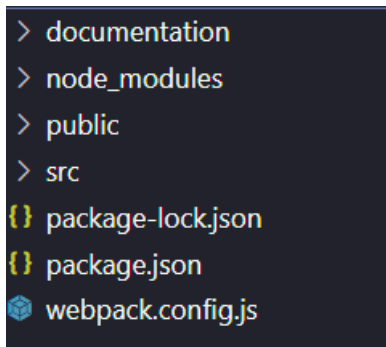


Ilustración 6.1

- En la documentación podemos encontrar ayudas e indicaciones sobre la plantilla.

Imagen de la carpeta documentation:

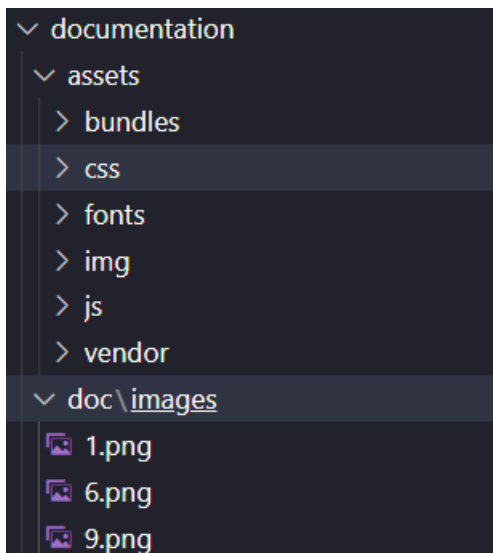


Ilustración 6.2

- La carpeta `node_modules` es un directorio que se crea en la carpeta raíz de nuestro proyecto cuando instalamos paquetes o dependencias mediante npm

Imagen de la carpeta `node_modules`:

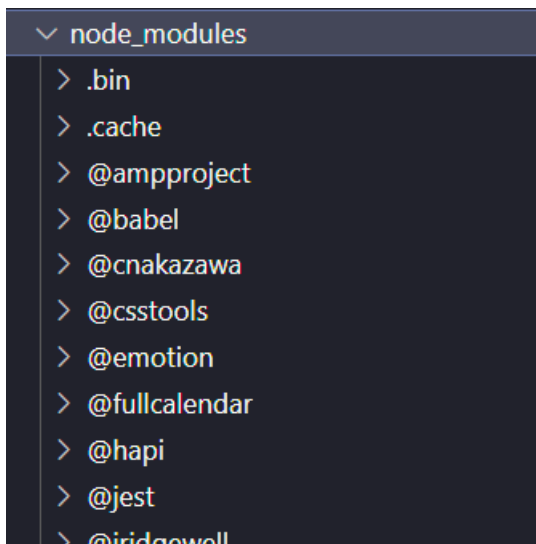


Ilustración 6.3

- La carpeta `public` contiene archivos públicos, como son los logos:

Imagen de la carpeta `public`:

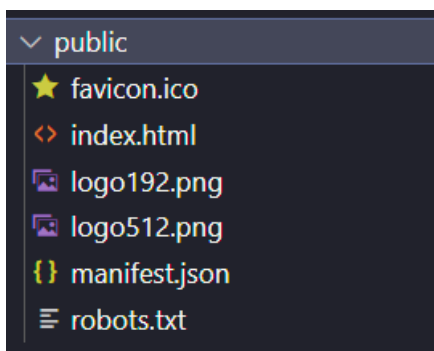


Ilustración 6.4

- La carpeta `src` es la carpeta principal de la aplicación, contiene todos los archivos que intervienen en la implementación de la aplicación:

Imagen de la carpeta `src`:

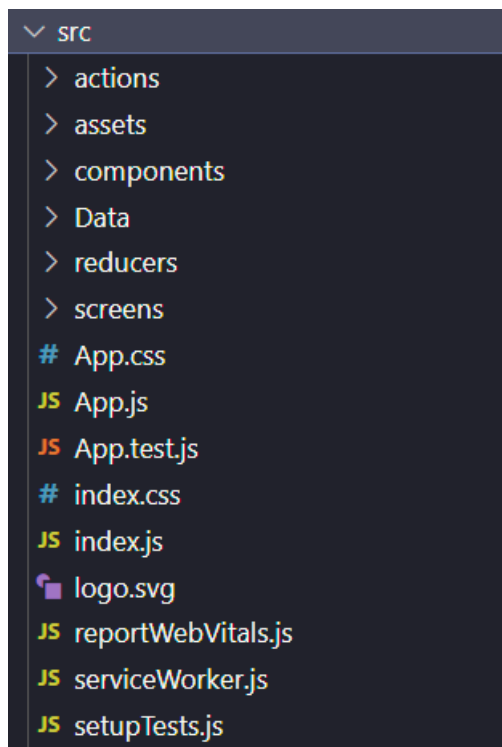


Ilustración 6.5

Como se puede apreciar, reactjs trabaja mediante componentes. La carpeta screens, posee las vistas principales, que están formadas por componentes.

Por ejemplo, la pantalla Dashboard está formada por distintos componentes, como puede ser el componente FileStorageStatusCard, al que le envía los datos necesarios para que se renderice.

Código del componente:

```

class FileStorageStatusCard extends React.Component {
  render() {
    const { TotalSize, UsedSize, Type, UsedPer, ProgressBarClass } = this.props;
    return (
      <div className="card modal-open m-b-5">
        <div className="body">
          <h6>{UsedSize}</h6>
          <p className="mb-0">
            {Type}{" "}
            <small className="text-muted float-right">of {TotalSize}</small>
          </p>
        </div>
        <ProgressBar
          className={ProgressBarClass}
          now={parseInt(UsedPer)}
          min={0}
          max={100}
        />
      </div>
    );
  }
}

```

Ilustración 6.6

Código de la pantalla que llama al componente:

```

> Blog 43
> Charts 44
> Dashbord 45
> FileManager 46
  JS Dashboard.js 47
  JS Documents.js 48
  JS Images.js 49
  JS Media.js 50
> Forms 51
> Maps 52

```

```

{fileStorageStatusCardData.map((data, index) => {
  return (
    <FileStorageStatusCard
      key={index + "sidjpidj"}
      TotalSize={data.TotalSize}
      UsedSize={data.UsedSize}
      Type={data.Type}
      UsedPer={data.UsedPer}
      ProgressBarClass={` ${data.ProgressBarClass}`}
    />
  );
})}

```

Ilustración 6.7

- El archivo package.json contiene los módulos que se necesitan instalar para que la aplicación funcione correctamente.
- El archivo package-lock.json se genera automáticamente al ejecutar la instalación: npm install.

6.2. Conexión de plantilla con backend

Para la conectar la plantilla y el backend, se lanzan los dos proyectos al mismo tiempo.

En el archivo package.json se añade el puerto del que partirá el frontend, 3001:

```
"start": "set PORT=3001 && react-scripts start",
```

En el archivo app.js del backend, se especifica el puerto 5500:

```
const port = 5500;
app.use(cors(corsOptions));
app.listen(port, () => {
  console.log(`Server is listening at http://localhost:${port}`);
});
```

De esta forma, la interfaz realizará las peticiones a la dirección <http://localhost:5500/> y además, se añade en el lado del servidor el origen de las peticiones:

```
const corsOptions = {
  origin: "http://localhost:3001",
  credentials: true, //access-control-allow-credentials:true
  optionSuccessStatus: 200,
};
```

En el archivo app.js se utilizan estos datos para la correcta conexión:

```
const db = mysql.createPool({
  host: "localhost",
  port: "3306",
  user: "jaenvivienda",
  password: "*****",
  database: "J_",
});
```

Y de este momento, ya es posible realizar consultas desde el backend.

6.3. Creación de inicio de sesión

Para realizar este proceso, en primer lugar, la vista del inicio de sesión, realiza una petición de los usuarios de la plataforma, que quedarán guardados en un estado.

```
const base = "http://localhost:5500/api/getAdministradores";
const baseCliente = "http://localhost:5500/api/getClientes";
const baseTrabajador =
"http://localhost:5500/api/getTrabajadores";
```

Utilizamos axios para realizar las peticiones al servidor:

```
const getAdministradores = async () => {  
  await axios  
    .get(base)  
    // fetch(base, { mode: 'no-cors', })  
    .then((response) => {  
      this.setState({  
        admins: response.data  
      })  
    });  
};
```

El estado de los administradores, en este momento, contiene los datos de los administradores que existen en la base de datos.

Cuando el estado de las entradas del formulario cambia, se realizan comparaciones y se determina si el usuario y la contraseña son correctos. Si lo son, guardaremos en las cookies el estado del usuario.

```
localStorage.setItem("authenticated", true);
```

6.4. Creación de tablas de los usuarios en la aplicación

Se crean las estructuras de las tablas como componentes. Para cada tabla de usuarios se crea su propio componente, ya que hay variaciones entre ellas en el apartado de acciones.

La pantalla principal es la que solicita los datos de los usuarios, para enviárselos al componente.

En el caso de los trabajadores, por ejemplo, vemos la siguiente pantalla.

Tabla de trabajadores:

Trabajadores

Trabajadores

Nuevo trabajador

#	Acciones	Nombre	Email	DNI	Teléfono	Domicilio	Seg. Social	Cargo	
44		▲	Celia Macías Torres	limaalimon1@gmail.com	77367311A	662194280 953121212	Mi calle Jaén 23001	58641636SAD	Estupenda
45		▲	Paqui Torres Díaz	paquitorres65@gmail.com	12345678N	123456789 953123456	Jaén 23002		
46		▲	David Mérida Mesa	david@gmail.com	12344567A				
55		▲	Prueba Prueba PRueba	prueba@prueba.com					

Ilustración 6.80

Esta vista consta de 3 componentes. El primero es PageHeader, al que pasamos los datos del título y la url a la que redirige el link.

```
<PageHeader
  HeaderText="Trabajadores"
  Breadcrumb=[
    { name: "Trabajadores", navigate: "trabajadores" },
  ]
/>
```

El segundo es un pop up de aviso de inserción de datos. Cuando creas o modificas un usuario, aparece un aviso en pantalla de que el resultado ha sido satisfactorio.

```
<Toast
  id="toast-container"
  show={true}
  className={`toast-bottom-right toast-container`}
  style={{ background: "transparent" }}
  onClose={() => { }}
>
  {notifyData.map((data, i) => {
    if (data.position === "bottom-right") {
      setTimeout(() => {
        this.props.notifyOnClose(i);
      }, 10000);
    }
  })}
```

```

        return (
          <Toast.Body
            key={"keys" + i}
            className={`toast-${data.type} mb-10 ml-0`}
            style={{ opacity: 1 }}
          >
            <strong className="mr-auto"> {data.dialogText}

</strong>

            <button
              className="toast-close-button"
              onClick={() => {
                this.props.notifyOnClose(i);
              }}
            >
              x
            </button>
          </Toast.Body>
        );
      }
    }
  </Toast>

```

Ventana de éxito:

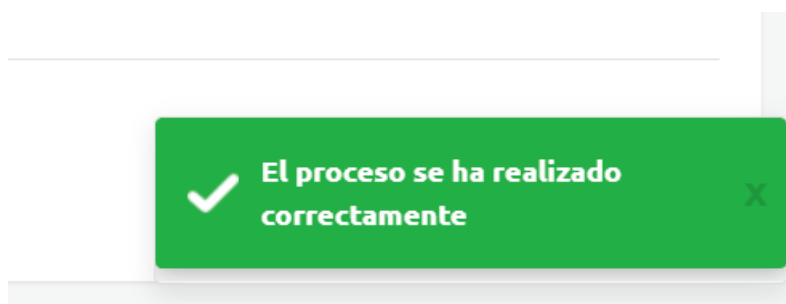


Ilustración 6.91

El tercero, es la el componente de la tabla, al que se le pasan los datos de los trabajadores.

```

<div className="row clearfix">
  <Trabajadores
    users={this.state.users}

```

```

      imagen={this.state.imagen}
    />
  </div>

```

Este componente, llama a su vez a otro componente, que es un desplegable con las acciones posibles:

Menú de acciones:

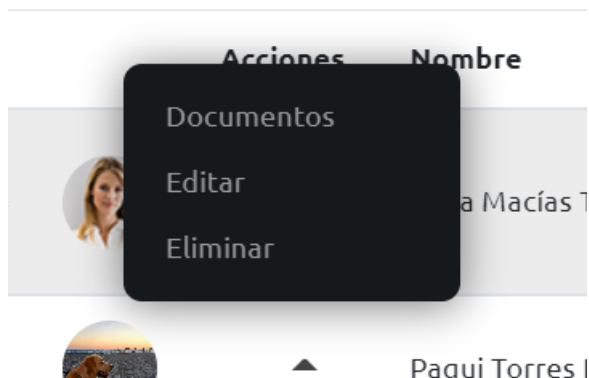


Ilustración 6.102

```

<Dropdown.Toggle
  variant="success"
  as="li"
  id="dropdown-basic"
></Dropdown.Toggle>

```

6.5. Creación, edición y borrado de datos desde la web

Para la creación y edición de usuarios, se utiliza un componente con el formulario. En caso de que nos encontremos en edición del usuario, la vista principal recibirá los datos y los enviará al componente para que los muestre:

Edición de Trabajador:

Editar Trabajador

Trabajadores / Editar Trabajador

Nombre* Celia

Primer Apellido* Macías

Segundo Apellido* Torres

Email* limaalimon1@gmail.com

DNI 77367311A

Teléfono 1 662194280

Teléfono 2 953121212

Domicilio Mi calle

Población Jaén

Código Postal 23001

Número de la Seguridad Social 586416365AD

Cargo del trabajador Estupenda

Nueva Contraseña Contraseña

Editar imagen Seleccionar

Guardar

Ilustración 6.113

Nuevo Trabajador:

Nuevo Trabajador

Trabajadores / Nuevo Trabajador

Nombre* Aitor

Primer Apellido* Suarez

Segundo Apellido* Martínez

Email* email@email.com

DNI 77123456A

Teléfono 1 662123456

Teléfono 2 953232323

Domicilio Carretera de Otiñar

Población Jaén

Código Postal 23002

Número de la Seguridad Social 555501234

Cargo del trabajador Albañil

Contraseña* Contraseña

Elige una imagen Seleccionar

Guardar

Ilustración 6.124

Se trabaja con el estado de los datos, y finalmente se envían los datos mediante el método POST, para su creación o edición en el servidor.

```
handleSubmit = async () => {
  if (this.handleEditing()) {
    if (this.state.file !== null) {
      console.log('archivo SI hay ');
      await this.subirImagen();
    } else {
```

```
        console.log('archivo no hay ');
    }
    const editar = this.state.editar + "";
    let url = "http://localhost:5500/api/nuevoTrabajador";
    if (editar === "true") {
        url =
`http://localhost:5500/api/editarTrabajador/${this.state.tracod}`;
    }
    let form = [];
    form = {
        tranom: this.state.tranom,
        traap1: this.state.traap1,
        traap2: this.state.traap2,
        traema: this.state.traema,
        tradni: this.state.tradni,
        trate1: this.state.trate1,
        trate2: this.state.trate2,
        tradom: this.state.tradom,
        tracpo: this.state.tracpo,
        trapob: this.state.trapob,
        transs: this.state.transs,
        tracar: this.state.tracar,
        traimg: this.state.traimg,
        tracon: this.state.tracon,
    };
    try {
        fetch(url, {
            method: "POST",
            headers: {
                'Content-type': 'application/json'
            },
            body: JSON.stringify(form),
        })
        .then((result) => result.json())
        .then((info) => {
            if (info) {
                localStorage.setItem("showSuccessAlert", true);
                if (this.state.tipoUsuario === 'ADMIN') {
                    window.location.href = "trabajadores";
                } else {
                    window.location.reload();
                }
            }
        })
    }
```

```

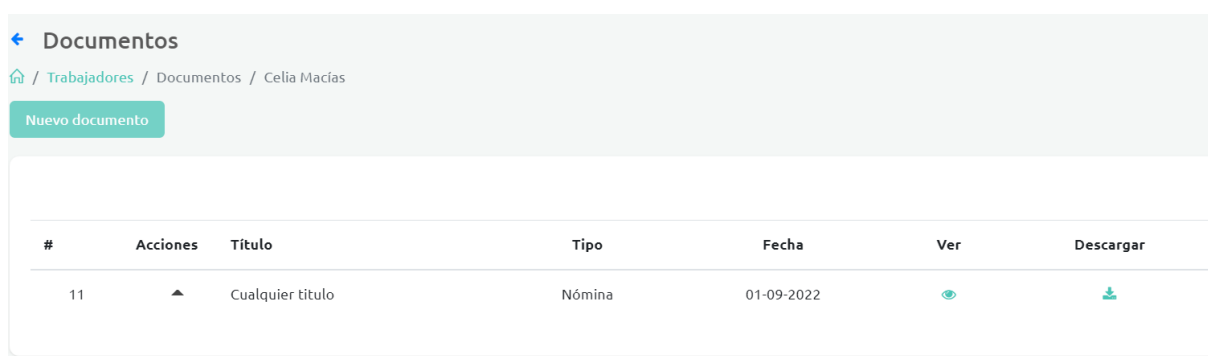
    }
  } else {
    localStorage.setItem("showFailAlert", true);
  }
});
} catch (error) {
  console.log("error");
  console.log(this.state.error_envio);
  this.setState({
    error_envio: true,
  });
}
}
};

```

6.6. Asociación de tablas y vistas

Dentro de las acciones de los trabajadores, se puede acceder a sus documentos, para ello se envía el parámetro del id del trabajador en la url. Por ejemplo: <http://localhost:3001/documentos/44> muestra solamente los documentos del usuario 44.

Documentos:



#	Acciones	Título	Tipo	Fecha	Ver	Descargar
11	▲	Cualquier titulo	Nómina	01-09-2022	👁	📄

Ilustración 6.134

En la parte del servidor, llamamos a la base de datos de la siguiente forma:

```

app.get("/api/getDocumentos/:id", (req, res) => {
  db.query(
    "SELECT * FROM J_DOC WHERE TRACOD=" + `${req.params.id}`,

```

```
function (error, results, fields) {  
  if (error) {  
    res.send(null);  
    // throw error;  
  }  
  res.send(results);  
}  
);  
});
```

6.7. Conexión con API Google Drive

Para subir imágenes se realiza una conexión con Google Drive. Para ello, es necesario acceder a la consola de Google con un usuario de Gmail.

Buscamos Google Drive para añadirlo:

Búsqueda Google Drive:

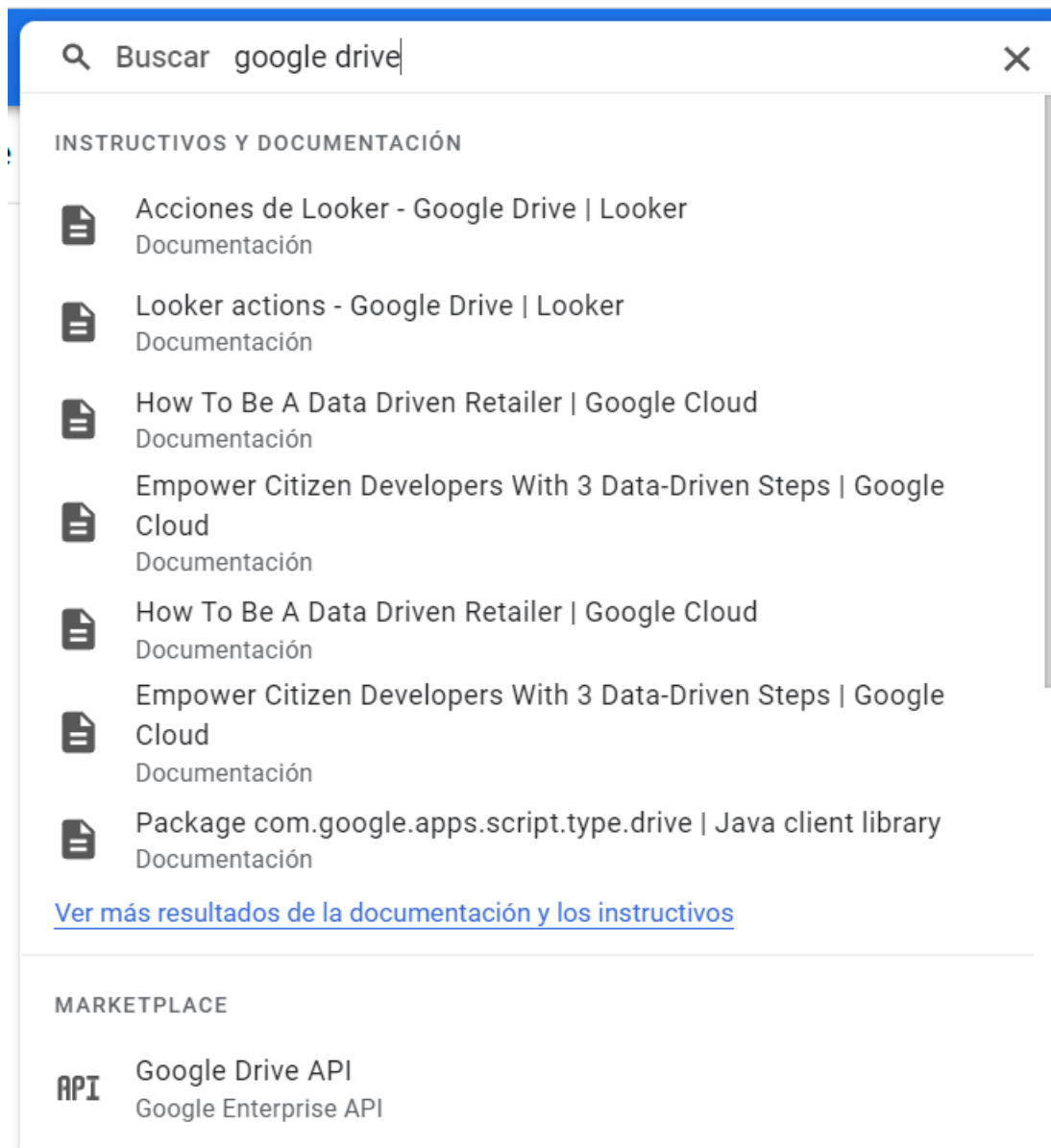


Ilustración 6.145

Una vez añadida, creamos una cuenta de servicio que permitirá la conexión.

Cuentas de servicio:

Cuentas de servicio [Administrar cuentas de servicio](#)

☐	Correo electrónico	Nombre ↑	Acciones
☐	radiant-pen-336710@appspot.gserviceaccount.com	App Engine default service account	
☐	test-subir-archivos@radiant-pen-336710.iam.gserviceaccount.com	test-subir-archivos	

Ilustración 6.16

Es necesario crear una carpeta en Drive y compartirla con el usuario que se ha generado automáticamente, utilizando una clave. Después hay que localizar el ID de la carpeta creada, compartirla con el usuario, y así, vemos como las imágenes de la web se alojan en dicha carpeta.

Carpeta:

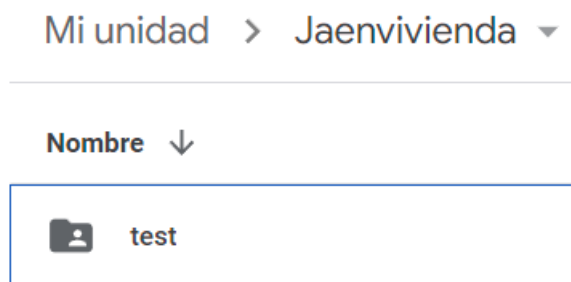


Ilustración 6.157

Imágenes de la carpeta:



Ilustración 6.168

Mismas imágenes en la aplicación:

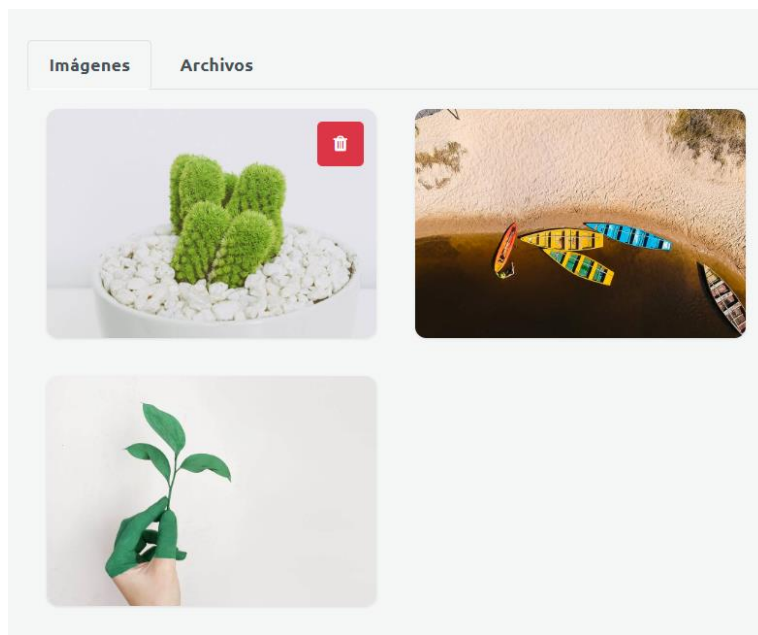


Ilustración 6.179

6.8. Solicitud de archivos de Drive. Concesión de permisos

Cuando se suben los archivos mediante la API, esta nos devuelve el id del archivo en la carpeta a la que se ha subido.

Para mostrar las imágenes, se transforman en base 64 y se muestran directamente en la web. Cualquier usuario con acceso a una vista concreta puede ver esas imágenes.

Para los archivos, hacemos una petición del enlace del archivo. Pero, para que un usuario que no tiene acceso a la carpeta de drive pueda verlo, es necesario concederle permisos.

```
const darPermisos = async (email, file) => {  
  const auth = authenticateGoogle();  
  const driveService = google.drive({ version: "v3", auth });  
  const response = await driveService.permissions.create({  
    fileId: file,  
    resource: {  
      role: 'reader',  
      type: "user",  
      emailAddress: email,  
    },  
  });  
};
```

```
return response;  
};
```

6.9. Visualización de la interfaz según usuario

Una vez creadas todas las vistas desde el administrador, se procede a administrar permisos según usuario. Por ejemplo, un trabajador no puede ver los proyectos, ni otros usuarios. Un cliente no puede ver a otros usuarios, ni los proyectos de otros usuarios.

Vista de inicio de un administrador:

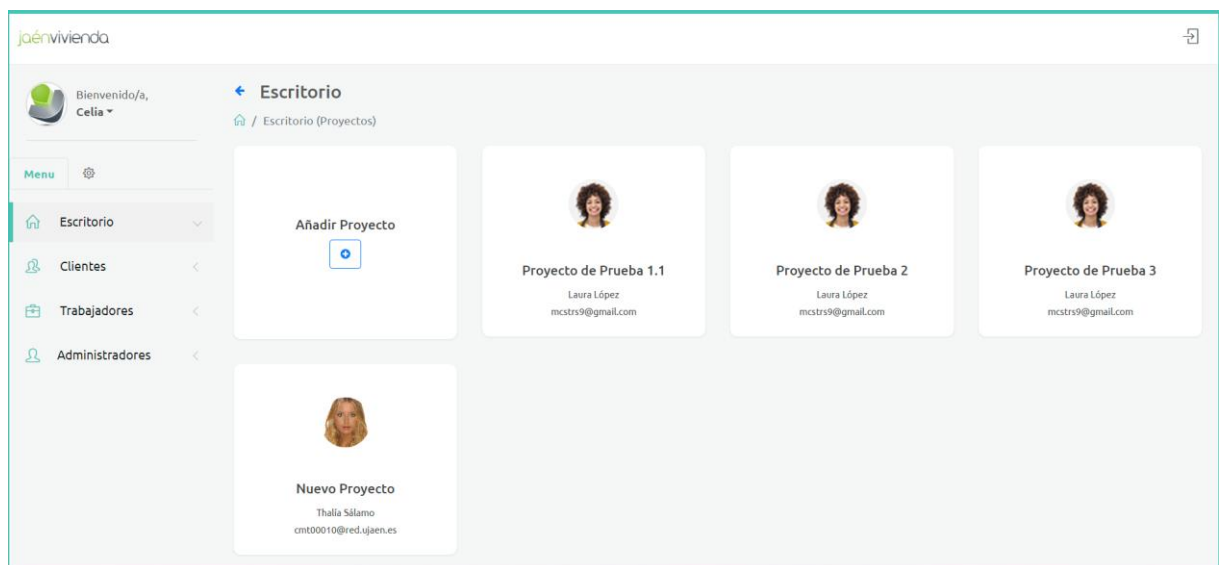


Ilustración 6.189

Vista de inicio de un cliente:

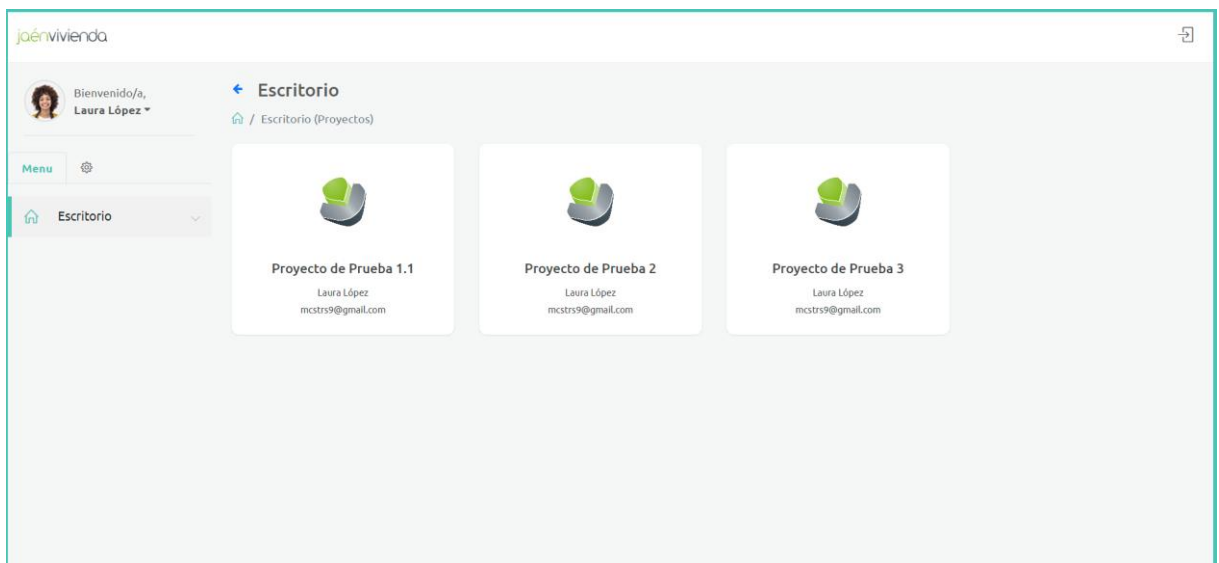


Ilustración 6.20

Vista de inicio de un trabajador:

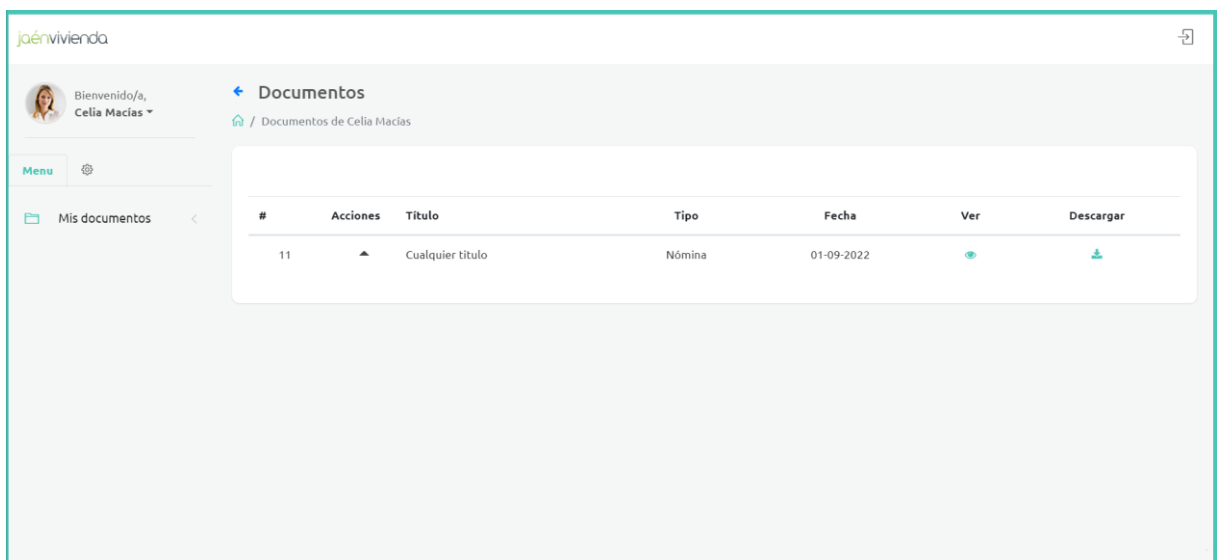


Ilustración 6.21

Para que esto suceda, se introducen diferentes urls de redirección tras el inicio de session:

```

window.location.href = '/escritorio';
window.location.href = '/documentos/' + element.TRACOD;

```

Para los administradores y clientes será el escritorio, y para los trabajadores, sus documentos. Una vez el cliente se encuentra en la misma vista que el administrador, la solicitud de documentos es distinta.

```
if (this.state.tipoUsuario !== 'ADMIN') {
    base = "http://localhost:5500/api/getMisProyectos/" +
this.state.codigoUsuario;
}
```

El cliente obtiene solo sus proyectos, y el administrador los obtiene todos. También se han realizado redirecciones en caso de que algún usuario intente acceder a una vista a la que no debe tener acceso.

6.10. Permisos y detalles de texto según usuario

Además de limitar el acceso a vistas según usuario, y hacer distintas peticiones a la base de datos, se han tenido en cuenta títulos, textos y campos que un usuario puede ver de forma diferente a otro. Por ejemplo, un trabajador puede editar sus datos, pero no el cargo que ocupa en la empresa. Un administrador sí puede editar los campos.

6.11. Testeo y corrección de errores

A continuación, se muestra una tabla en la que se define el objetivo, la forma en la que se ha comprobado y la solución.

Tabla de corrección de errores:

Objetivo	Testeo	Errores y solución	Objetivo cumplido
Prohibición de acceso en caso de no haber iniciado sesión.	Inicio de sesión con perfil de administrador en una ventana del navegador. Apertura de ventana de incógnito. Copiar y pegar el enlace de cada vista del administrador en la ventana de incógnito para comprobar si puede acceder o no.	No.	Sí.
Prohibición de acceso para clientes y trabajadores a las vistas del administrador.	Inicio de sesión en dos ventanas del navegador: primero con administrador y trabajador, y luego con administrador y cliente.	El trabajador podía acceder a los proyectos del cliente. Solución: restringir la vista al usuario trabajador.	Sí.
Prohibición de visualización de	Tanto en el trabajador como en el cliente,	No.	Sí.

información de otros usuarios en la misma vista.	probar el acceso a enlaces de otros usuarios cambiando los parámetros del enlace.		
Acceso a archivos propios por parte del cliente y del trabajador.	Intentar acceder a un archivo de Google Drive sin haber iniciado sesión. Iniciar sesión con el usuario correspondiente.	Al principio siempre accedía a los archivos desde un navegador en el que la sesión de google drive está iniciada, por lo que creía que el enlace del archivo contenía los permisos para su visualización. Solución: integrar un sistema de permisos de lectura en el momento de subida del archivo.	Sí.
Prohibición de acceso a los archivos no propios.	Crear un archivo y compartirlo con un cliente. Intentar acceder a Google Drive con el correo asociado al trabajador.	No.	Sí.
Todos los usuarios deben tener un email asociado.	Intento de creación de usuarios dejando el campo email vacío, insertando caracteres aleatorios.	No.	Sí.
Los clientes no pueden eliminar archivos del proyecto.	Error que no se había tenido en cuenta.	Los clientes podían ver el icono de la papelera en los archivos. Solución: Crear el icono solo si el usuario que accede a la vista es administrador.	Sí.
Correcta navegación desde cualquier punto de la web	Utilización de cualquier botón o enlace desde cualquier punto de la web.	El logo de la web redirige al escritorio con los proyectos y los trabajadores no tienen acceso a esta vista, por lo que, hacer clic en el logo los llevaba a una pantalla en blanco. Solución: Cambiar el enlace del logo dependiendo del tipo de usuario.	Sí.
Correcto cierre de sesión	Iniciar sesión y abrir dos pestañas distintas con cualquier vista. Cerrar sesión en una de ellas. Tratar de acceder a cualquier sitio desde la otra.	No. La sesión se cierra totalmente y la pestaña en la que no se cerró sesión redirige al inicio de sesión.	Sí.

Tabla 6.1

Todos los errores han sido corregidos por parte del desarrollador. Este proceso debe repetirse por parte de distintos usuarios, ya que una persona que ha diseñado, programado y testeado una aplicación puede no apreciar ciertos errores debido a que conoce perfectamente la aplicación.

7. EVALUACIÓN DEL PROYECTO

7.1. Validación de los servicios

Actualmente, este proyecto no está funcionando en un servidor. Todos los servicios se han evaluado localmente: la base de datos existe en un servidor local, y tanto el lado del cliente como el del servidor se han lanzado localmente.

El proyecto cumple las funciones solicitadas por la empresa y en un futuro se prevé que este proyecto siga evolucionando y adquiriendo nuevas funcionalidades. Cuando se decida publicar en un servidor, habrá que modificar redirecciones locales y validar de nuevo.

7.2. Usabilidad

La aplicación es sencilla e intuitiva. Se ha procurado que los formularios y vistas sigan la misma estética y procedimiento. Por ejemplo, en las tablas, siempre hay un botón con acciones, que tiene unas posibilidades u otras dependiendo de la tabla en la que se aloje.

Es una plataforma responsiva, puede utilizarse en pantallas de ordenador y de móvil o Tablet.

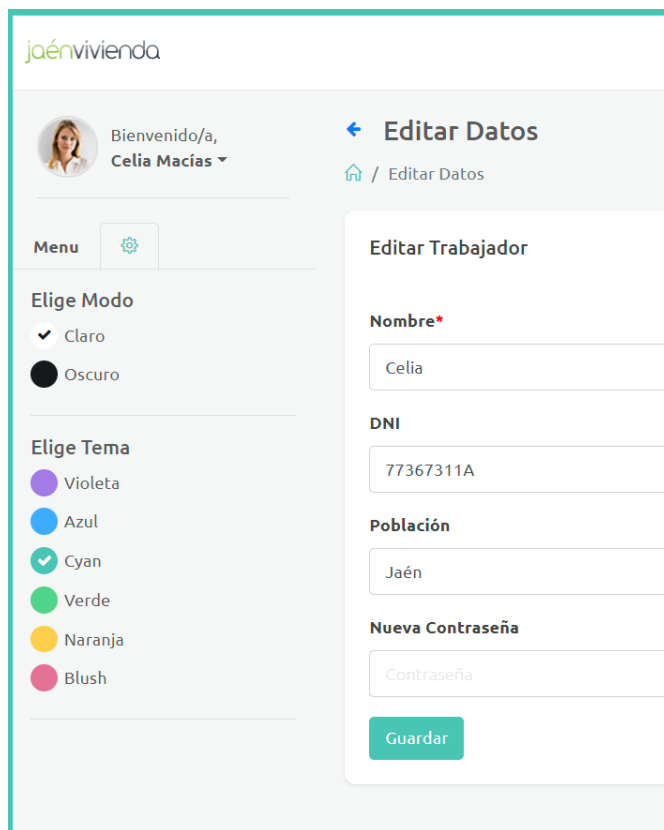
Tiene opciones en el menú para cambiar el tema y los colores. Para que cada usuario utilice la versión que desee.

Tema oscuro:

The screenshot displays the 'jaénvivienda' web application interface. At the top left, the logo 'jaénvivienda' is visible. Below it, a user profile section shows a welcome message 'Bienvenido/a, Celia Macías' next to a circular profile picture. To the right of the profile, there is a blue arrow icon and the text 'Editar Datos'. Below this, a home icon and the text '/ Editar Datos' are shown. The sidebar on the left contains a 'Menu' button with a gear icon, a section titled 'Elige Modo' with two radio buttons ('Claro' and 'Oscuro', where 'Oscuro' is selected), and a section titled 'Elige Tema' with six color-coded buttons ('Violeta', 'Azul', 'Cyan', 'Verde', 'Naranja', and 'Blush', where 'Blush' is selected). The main content area is titled 'Editar Trabajador' and contains several input fields: 'Nombre*' (with the value 'Celia'), 'DNI' (with the value '77367311A'), 'Población' (with the value 'Jaén'), and 'Nueva Contraseña' (with the value 'Contraseña'). A green 'Guardar' button is located at the bottom of the form.

Ilustración 7.1

Tema claro:



The screenshot displays the 'jaénvivienda' web application interface. On the left, a sidebar contains a 'Menu' button, a 'Bienvenido/a, Celia Macías' greeting with a profile picture, and options to 'Elige Modo' (Claro, Oscuro) and 'Elige Tema' (Violeta, Azul, Cyan, Verde, Naranja, Blush). The main content area is titled 'Editar Datos' and features a 'Editar Trabajador' form. The form includes fields for 'Nombre*' (Celia), 'DNI' (77367311A), 'Población' (Jaén), and 'Nueva Contraseña' (Contraseña), followed by a 'Guardar' button.

Ilustración 7.2

También encontramos distintas formas de navegar, ya sea por el menú principal situado a la izquierda, o por las cabeceras que encontramos en cada vista. De esta forma el usuario puede habituarse con el entorno de la forma que prefiera.

8. CONCLUSIONES

Los objetivos fijados para este TFG, indicados anteriormente, son los siguientes:

1. Objetivo 1: Creación de un panel de administración con tres tipos de usuarios: administrador, cliente y trabajador.
2. Objetivo 2: El administrador desea gestionar sus documentos mediante una plataforma propia para disponer de una organización clara y eficiente.
3. Objetivo 3: Creación interna de usuarios.
4. Objetivo 4: Prohibición de registro en la plataforma de forma externa, solo el administrador puede registrar otros usuarios.
5. Objetivo 5: Privacidad de los datos. Esta plataforma alojará archivos con datos privados, como lo son una nómina, una factura o cualquier imagen de una obra privada.
6. Objetivo 6: Sistema para compartir archivos entre usuarios. En el caso de los clientes, compartir proyectos que contienen dichos archivos. En el caso de trabajadores, compartir documentos propios de cada trabajador.

A continuación, se describe el grado de cumplimiento de los mismos:

1. Objetivo 1: se ha alcanzado y superado este objetivo utilizando distintas tablas en la base de datos para cada tipo de usuario. Además, en el inicio de sesión se guardan detalles sobre el usuario en cookies con la finalidad de restringir el acceso a vistas y datos que no le pertenecen. De esta forma se diferencian claramente tres usuarios con diferentes permisos.
2. Objetivo 2: se ha superado correctamente este objetivo, ya que el administrador puede gestionar sus documentos permitiendo la subida y correcta disposición de ellos.
3. Objetivo 3: se ha logrado este objetivo mediante el uso de formularios y la correcta conexión con el backend. De esta forma, la información de los usuarios se inserta correctamente en la base de datos, permitiendo el acceso al panel a los distintos usuarios.

4. Objetivo 4: se ha alcanzado y superado este objetivo, ya que no existen enlaces de registro desde la vista de inicio de sesión. Además, se ha restringido el acceso a cualquier vista excepto la de inicio de sesión a cualquier usuario que no esté registrado.
5. Objetivo 5: se ha logrado este objetivo utilizando la API de Google Drive, que utiliza un cifrado para los archivos. Todos los archivos se suben a una carpeta de Google Drive a la que solo tiene acceso el administrador. Para que trabajadores y usuarios tengan acceso a sus propios archivos, se les asignan permisos de lectura de cada archivo en concreto. En la aplicación se comparte el enlace del archivo con el usuario al que pertenece. Esto evita que si hubiera un posible error y alguien accediera al enlace del archivo, tuviera acceso a él, pues para visualizarlo es necesario iniciar sesión en Google Drive.
6. Objetivo 6: este objetivo se ha alcanzado y superado, ya que la base de datos se ha enlazado correctamente y la plataforma solicita a Google Drive el enlace del archivo mediante el Id generado en la subida del archivo.

React JS es una biblioteca que no conocía antes de realizar este proyecto. En mi opinión, tiene un comienzo complicado cuando estás habituado a otros tipos de lenguaje. Existen muchas versiones, y muchas de las plantillas que utilicé en un principio, utilizaban módulos obsoletos, que no funcionaban en otras.

Tras estos meses de investigación y de programación, he descubierto un “framework” muy útil y eficaz para el desarrollo web. El sistema que maneja reactjs del estado de los componentes, evita costes de tiempo cuando se trata de ciertas funcionalidades, que en otros frameworks habría que solucionar mediante, por ejemplo, la combinación de Ajax y javascript.

El sistema de componentes es una solución muy cómoda para ahorrar código y reutilizar escenas en varios sitios de la web. Además, la forma del lenguaje tiene un cierto parecido con html, que es muy intuitivo para el desarrollador.

En definitiva, este proyecto me ha llevado a interesarme en profundidad por esta biblioteca, por lo que, en un futuro, seguiré investigando en ella y solucionando

errores que a día de hoy pueda cometer, para así perfeccionar mi técnica, crear nuevos proyectos y seguir avanzando en el perfeccionamiento de este.

9. REFERENCIAS

Bibliografía

- (s.f.). Obtenido de https://ciberninjas.com/react/#La_Historia_de_React
- Angular JS. (s.f.). Obtenido de <https://docs.angularjs.org/misc/faq#is-angularjs-a-library-framework-plugin-or-a-browser-extension>
- Asana. (s.f.). Obtenido de Asana: <https://asana.com/es/product>
- Benkler, Y. (2012). *El Pingüino y el Leviatán: Por qué la cooperación es nuestra arma más valiosa para mejorar el bienestar de la sociedad*.
- Blog. (s.f.). Obtenido de <https://blog.nubecollectiva.com/que-es-vue-js-historia-y-tu-primer-hola-mundo/>
- Blog. (s.f.). Obtenido de <https://www.armadilloamarillo.com/blog/cuales-son-las-mejores-tecnologias-para-desarrollo-web/>
- Dataprius. (s.f.). Obtenido de <https://dataprius.com/compartir-archivos-con-clientes.html>
- Desarrollo Web. (s.f.). Obtenido de <https://desarrolloweb.com/home/laravel>
- Developer. (s.f.). Obtenido de <https://developer.mozilla.org/es/docs/Web/HTML>
- Developer. (s.f.). Obtenido de https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction
- Developinginspanish. (s.f.). Obtenido de <http://developinginspanish.com/2019/03/08/reactjs-vs-angular5-vs-vue-js-cual-elegir/>
- Dobuss. (s.f.). Obtenido de <https://www.dobuss.es/que-es-el-diseno-web-y-para-que-sirve/>
- Doonamis. (s.f.). Obtenido de <https://www.doonamis.es/para-que-sirve-react-js-beneficios-para-tus-apps/>
- Dropbox. (s.f.). Obtenido de https://www.dropbox.com/es_ES/features
- Edix. (s.f.). Obtenido de <https://www.edix.com/es/instituto/framework/>
- Factorial. (s.f.). Obtenido de Factorial: <https://factorialhr.es/>
- GerSoft. (s.f.). Obtenido de GerSoft: <https://gersoft.eu/gestion-interna>
- GitHub. (s.f.). Obtenido de <https://github.com/DashboardPack/architectui-react-theme-free.git>
- Google Drive. (s.f.). Obtenido de https://www.google.com/intl/es_es/drive/
- Gutiérrez, M. B. (2018). *El gran libro de Angular*. Barcelona, España: Alfaomega Grupo Editor, S.A. de C.V.
- Hiberus. (s.f.). Obtenido de <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/#:~:text=Versiones%20de%20Angular&text=Angular%20se%20desarroll%C3%B3%20en%202009,el%2012%2F05%2F2021>
- Hostinger. (s.f.). Obtenido de <https://www.hostinger.es/tutoriales/que-es-angular>
- Ingeniería Software. (s.f.). Obtenido de <https://sites.google.com/site/talleringenieriasoftwaregaby/unidad-5-diseno-para-aplicaciones-web/5-6-diseno-arquitectonico>
- Ionos. (s.f.). Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>
- Ionos. (s.f.). Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de-secuencia/>
- Java. (s.f.). Obtenido de <https://www.java.com/es/download/help/techinfo.html>
- Kinsta. (s.f.). Obtenido de <https://kinsta.com/es/blog/node-js-vs-php/#nodejs-vs-php-tabla-comparativa-lado-a-lado>
- Microsoft. (s.f.). Obtenido de <https://docs.microsoft.com/es-es/dotnet/architecture/modern-web-apps-azure/common-client-side-web-technologies>

Nobbot. (s.f.). Obtenido de <https://www.nobbot.com/tecnologia/los-lenguajes-de-desarrollo-web-del-futuro-python/>

Node.js. (s.f.). Obtenido de <https://nodejs.org/es/about/>

PHP. (s.f.). Obtenido de <https://www.php.net/manual/es/intro-what-is.php>

Reactjs. (s.f.). Obtenido de <https://es.reactjs.org/tutorial/tutorial.html#what-is-react>

Trello. (s.f.). Obtenido de Trello: <https://trello.com/es>

Velasco, J. G. (2009). *Energías renovables*. Editorial Reverte.

Vue JS. (s.f.). Obtenido de <https://es.vuejs.org/v2/guide/>

Anexo 1: Tablas de la base de datos

La base de datos consta de 6 tablas:

- j_adm: Tabla de administradores
- j_arc: Tabla de archivos de los proyectos
- j_cli: Tabla de clientes
- j_doc: Tabla de documentos de los trabajadores
- j_pro: Tabla de proyectos de los clientes
- j_tra: Tabla de trabajadores

Tabla de administradores

Base de datos: j_, Tabla: j_adm, Propósito: Estructura de tabla

Columna	Tipo	Nulo	Predeterminado	Comentarios
ADMCOD	int(8)	No		
ADMEMA	varchar(200)	Yes	NULL	Email administrador
ADMNOM	varchar(200)	Yes	NULL	Nombre Administrador
ADMCON	varchar(200)	Yes	NULL	contraseña

Tabla de archivos

Base de datos: j_, Tabla: j_arc, Propósito: Estructura de tabla

Columna	Tipo	Nulo	Predeterminado	Comentarios
ARCCOD	varchar(300)	No		Codigo de imagen, el mismo que se asigna en drive
ARCTIP	varchar(3)	Yes	NULL	Tipo: IMG imagen, OTR otro
ARCFEC	date	Yes	NULL	Fecha subida
PROCOD	int(8)	Yes	NULL	codigo del proyecto

Tabla de clientes

Base de datos: j_, Tabla: j_cli, Propósito: Estructura de tabla

Columna	Tipo	Nulo	Predeterminado	Comentarios
CLICOD	int(8)	No		
CLINOM	varchar(100)	No		
CLIAP1	varchar(100)	No		
CLIAP2	varchar(100)	Yes	NULL	
CLIEMA	varchar(300)	No		
CLIIMG	varchar(200)	No		Imagen del cliente
CLIFEC	date	Yes	NULL	Fecha de creacion
CLICON	varchar(200)	Yes	NULL	Contraseña

Tabla de documentos

Base de datos: j_, Tabla: j_doc, Propósito: Estructura de tabla

Columna	Tipo	Nulo	Predeterminado	Comentarios
DOCCOD	int(8)	No		
DOCNOM	varchar(150)	Yes	NULL	Nombre
DOCTIP	varchar(3)	Yes	NULL	Tipo: Para trabajadores: 'CON'-Contratos, 'NOM'- nominas, 'PRD' proteccion de datos,'OTR' otros Para proyectos: 'PRE' -> presupuesto 'FAC' -> Facturas
TRACOD	int(8)	Yes	NULL	Codigo del trabajador
DOCURL	varchar(300)	Yes	NULL	Url en la que se encuentra el documento
DOCFEC	date	Yes	NULL	Fecha subida
PROCOD	int(8)	Yes	NULL	Si es un documento de un proyecto, codigo del proyecto

Tabla de proyectos

Base de datos: j_, Tabla: j_pro, Propósito: Estructura de tabla

Columna	Tipo	Nulo	Predeterminado	Comentarios
PROCOD	int(8)	No		Código
PROTIT	varchar(200)	No		Título
PRODES	varchar(300)	Yes	NULL	Descripción
CLICOD	int(8)	No		Código cliente

Tabla de trabajadores

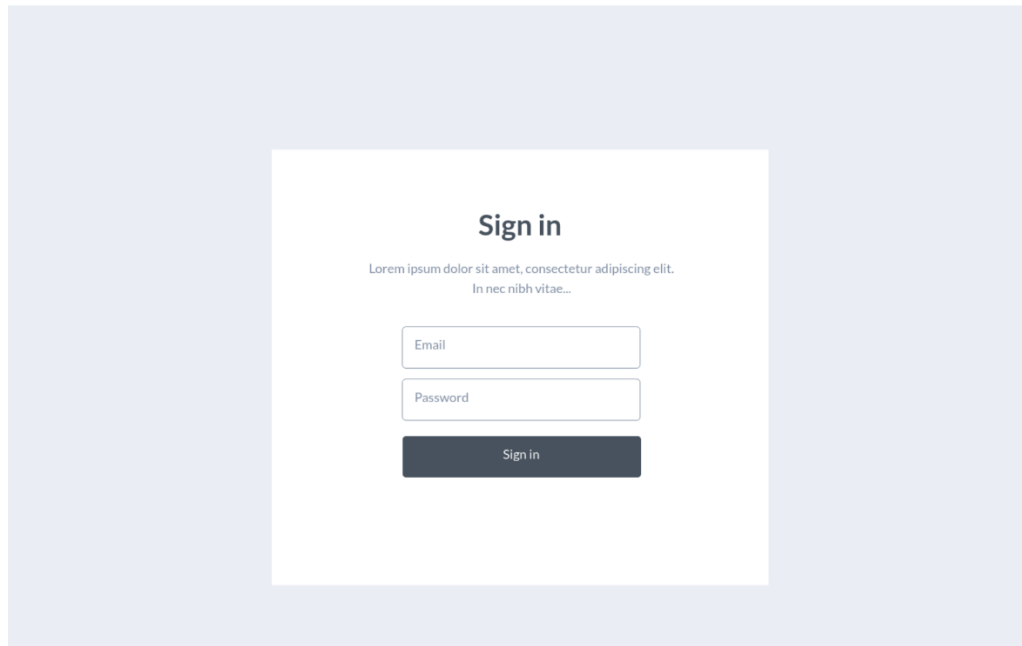
Base de datos: j_, Tabla: j_tra, Propósito: Estructura de tabla

Columna	Tipo	Nulo	Predeterminado	Comentarios
TRACOD	int(8)	No		Código del trabajador
TRANOM	varchar(100)	Yes	NULL	Nombre
TRAAP1	varchar(100)	Yes	NULL	Apellido 1
TRAAP2	varchar(100)	Yes	NULL	Apellido 2
TRADNI	varchar(9)	Yes	NULL	DNI 8 digitos y una letra
TRATE1	int(9)	Yes	NULL	Telefono 1
TRATE2	int(9)	Yes	NULL	Telefono 2
TRAEMA	varchar(200)	Yes	NULL	Email trabajador
TRADOM	varchar(300)	Yes	NULL	Domicilio
TRACPO	int(5)	Yes	NULL	Código Postal
TRAPOB	varchar(100)	Yes	NULL	Poblacion
TRANSS	varchar(15)	Yes	NULL	Numero Seguridad Social
TRANAC	date	Yes	NULL	Fecha de nacimiento
TRAFEC	date	Yes	NULL	Fecha de alta en la aplicación
TRACAR	varchar(150)	Yes	NULL	Cargo en la empresa
TRAIMG	varchar(300)	Yes	NULL	Url de la imagen
TRACON	varchar(200)	Yes	NULL	Contraseña

Anexo 2: Storyboard: Vistas en Marvell App

El Storyboard consta de 16 vistas principales:

1. Inicio de sesión:



2. Escritorio:



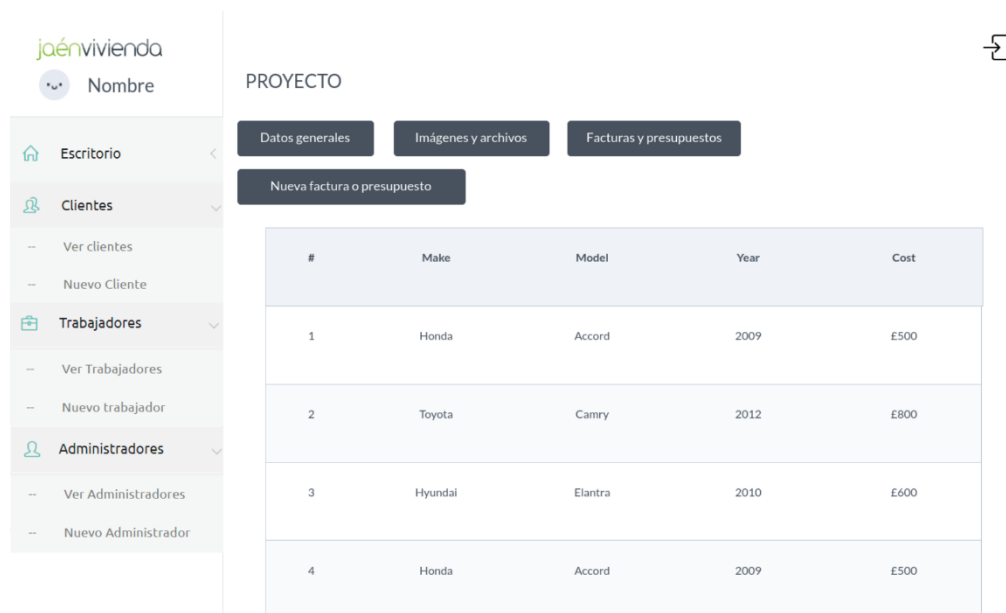
3. Nuevo proyecto:

The screenshot shows the 'jaénvivienda' web application interface. On the left is a sidebar with a user profile 'Nombre' and a menu with options: Escritorio, Clientes (with sub-options 'Ver clientes' and 'Nuevo Cliente'), Trabajadores (with sub-options 'Ver Trabajadores' and 'Nuevo trabajador'), and Administradores (with sub-options 'Ver Administradores' and 'Nuevo Administrador'). The main content area is titled 'PROYECTO' and contains three tabs: 'Datos generales' (active), 'Imágenes y archivos', and 'Facturas y presupuestos'. Under the 'Datos generales' tab, there are input fields for 'Titulo', 'Cliente', and 'Descripción', followed by a 'Guardar' button.

4. Imágenes del proyecto:

This screenshot shows the same 'jaénvivienda' application but with the 'Imágenes y archivos' tab selected. A 'Subir archivos' button is visible at the top of the main content area. Below it, there is a grid of six placeholder images, each represented by a light blue square with a diagonal 'X'.

5. Facturas y presupuestos:



jaénvivienda

Nombre

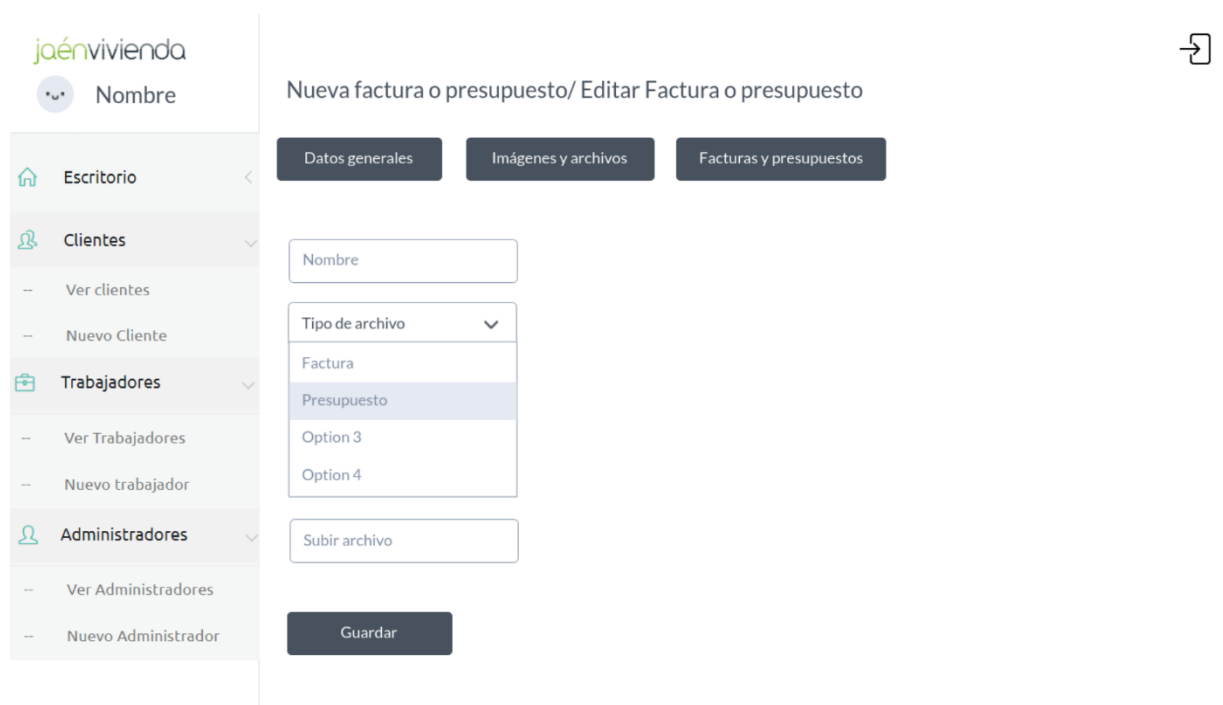
PROYECTO

Datos generales Imágenes y archivos Facturas y presupuestos

Nueva factura o presupuesto

#	Make	Model	Year	Cost
1	Honda	Accord	2009	£500
2	Toyota	Camry	2012	£800
3	Hyundai	Elantra	2010	£600
4	Honda	Accord	2009	£500

6. Formulario Factura:



jaénvivienda

Nombre

Nueva factura o presupuesto/ Editar Factura o presupuesto

Datos generales Imágenes y archivos Facturas y presupuestos

Nombre

Tipo de archivo

Factura

Presupuesto

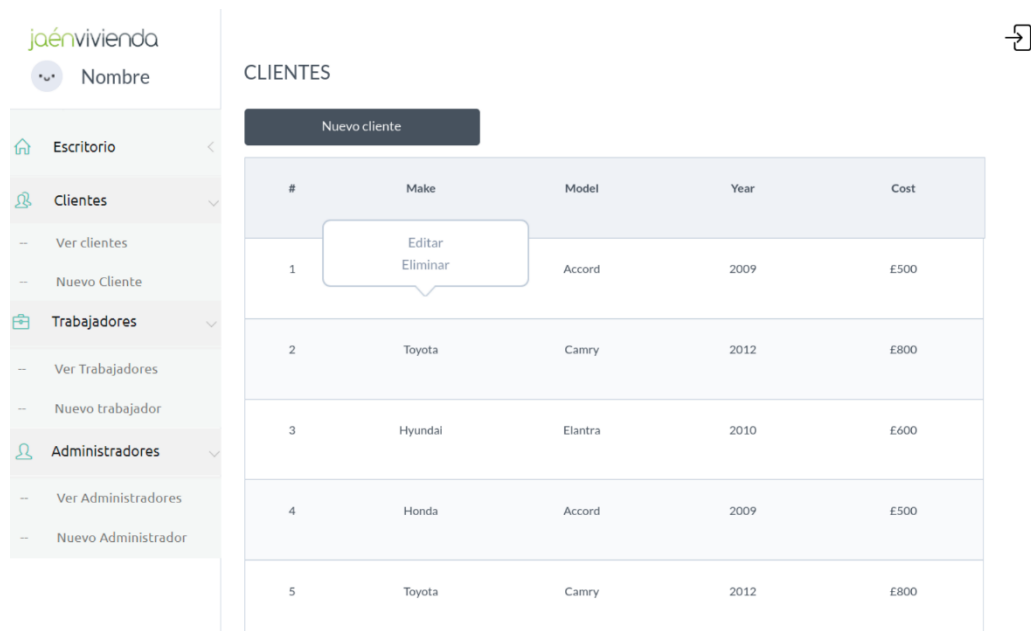
Option 3

Option 4

Subir archivo

Guardar

7. Clientes:



jaénvivienda

Nombre

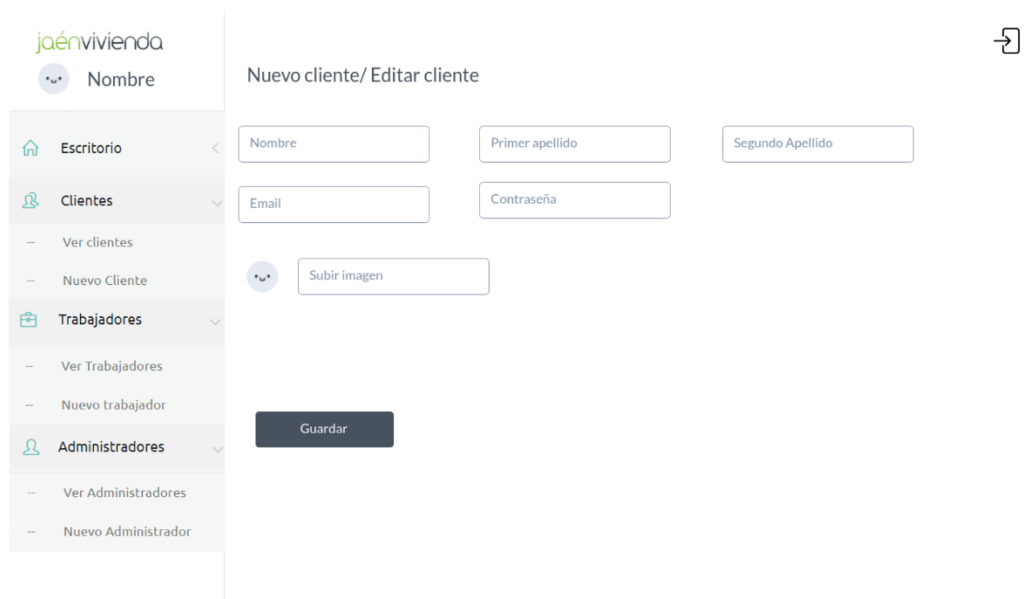
CLIENTES

Nuevo cliente

#	Make	Model	Year	Cost
1		Accord	2009	£500
2	Toyota	Camry	2012	£800
3	Hyundai	Elantra	2010	£600
4	Honda	Accord	2009	£500
5	Toyota	Camry	2012	£800

Editar
Eliminar

8. Formulario Cliente:



jaénvivienda

Nombre

Nuevo cliente/Editar cliente

Nombre

Primer apellido

Segundo Apellido

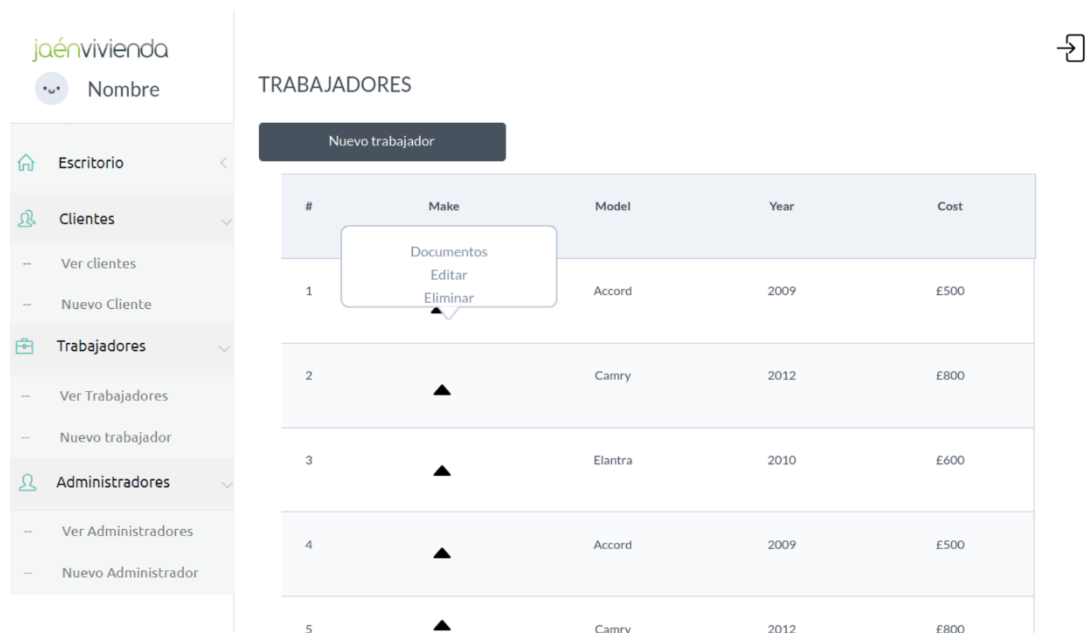
Email

Contraseña

Subir imagen

Guardar

9. Trabajadores:



jaénvivienda Nombre

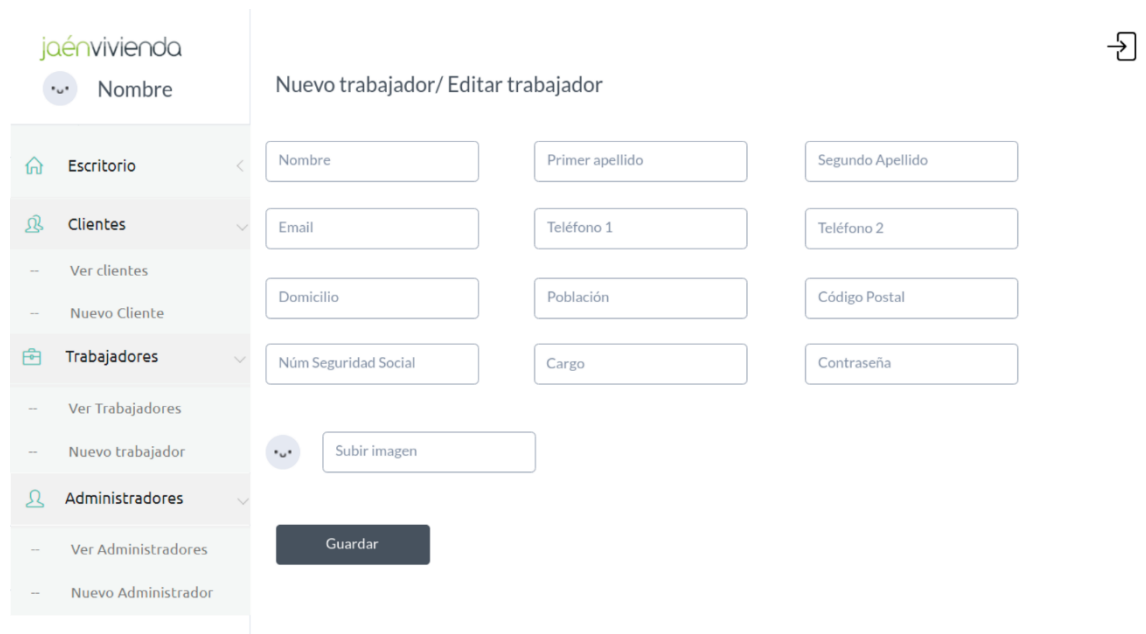
TRABAJADORES

Nuevo trabajador

#	Make	Model	Year	Cost
1		Accord	2009	£500
2	▲	Camry	2012	£800
3	▲	Elantra	2010	£600
4	▲	Accord	2009	£500
5	▲	Camry	2012	£800

Documentos
Editar
Eliminar

10. Formulario Trabajador:



jaénvivienda Nombre

Nuevo trabajador/ Editar trabajador

Nombre

Primer apellido

Segundo Apellido

Email

Teléfono 1

Teléfono 2

Domicilio

Población

Código Postal

Núm Seguridad Social

Cargo

Contraseña

Subir imagen

Guardar

11. Documentos:

jaénvivienda

Nombre

DOCUMENTOS DEL TRABAJADOR

Nuevo documento

#	Make	Model	Year	Cost
1		Accord	2009	
2	Toyota	Camry	2012	
3	Hyundai	Elantra	2010	
4	Honda	Accord	2009	
5	Toyota	Camry	2012	

Editar
Eliminar

12. Formulario Documento:

jaénvivienda

Nombre

Nueva documento/ Editar documento del trabajador

Nombre

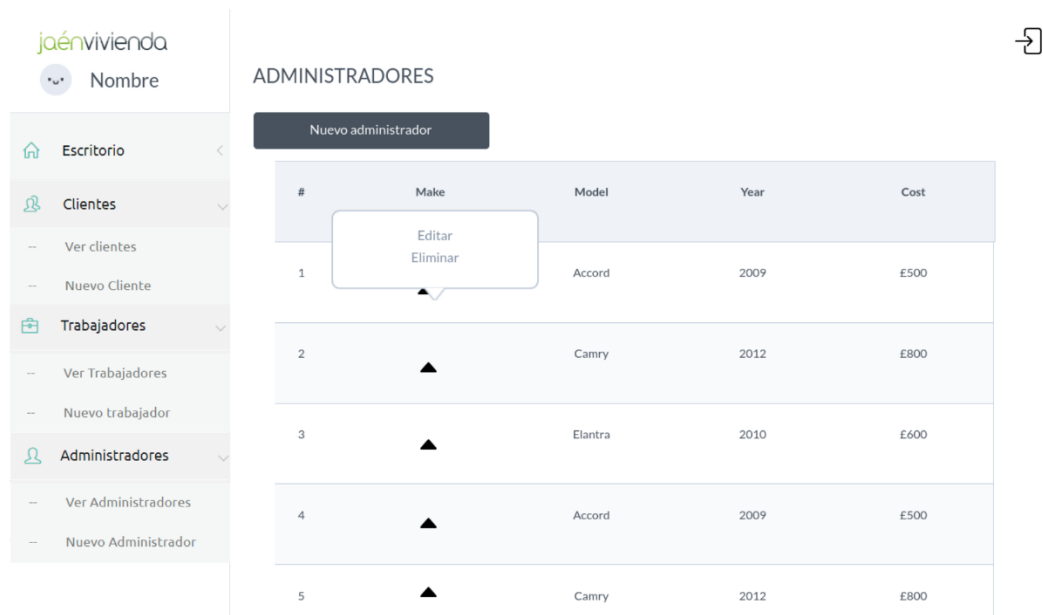
Tipo de archivo

Nómina
Protección de datos
Contrato
Option 4

Subir archivo

Guardar

13. Administradores:



jaénvivienda

Nombre

Escritorio

Cientes

Ver clientes

Nuevo Cliente

Trabajadores

Ver Trabajadores

Nuevo trabajador

Administradores

Ver Administradores

Nuevo Administrador

ADMINISTRADORES

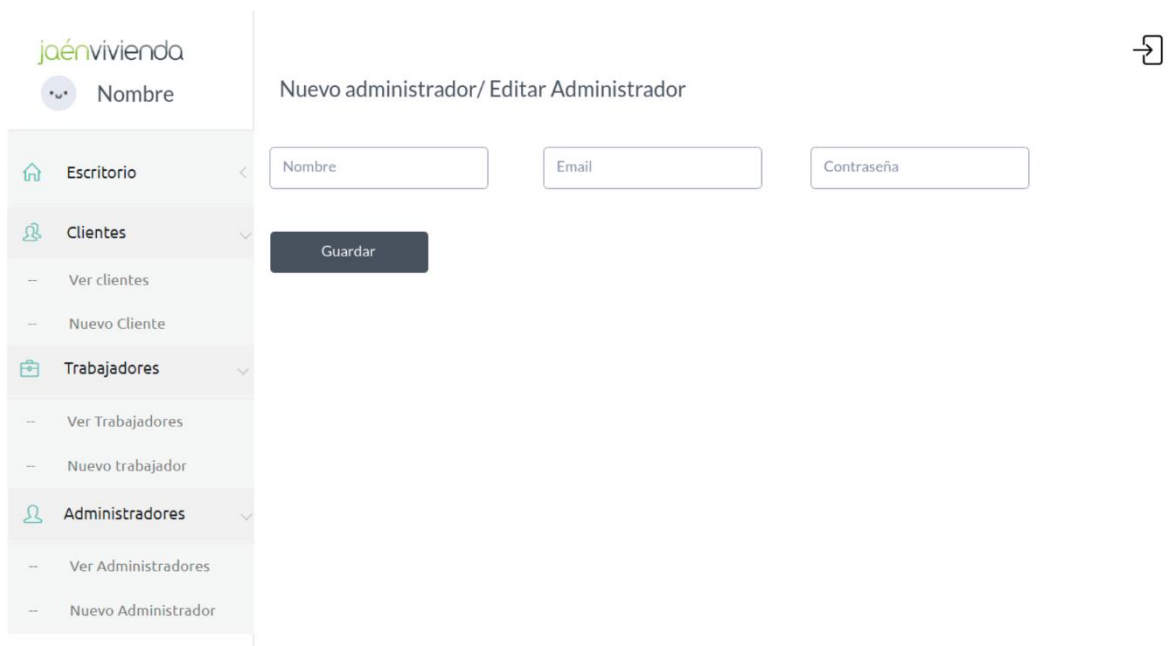
Nuevo administrador

#	Make	Model	Year	Cost
1		Accord	2009	£500
2		Camry	2012	£800
3		Elantra	2010	£600
4		Accord	2009	£500
5		Camry	2012	£800

Editar

Eliminar

14. Formulario Administrador:



jaénvivienda

Nombre

Escritorio

Cientes

Ver clientes

Nuevo Cliente

Trabajadores

Ver Trabajadores

Nuevo trabajador

Administradores

Ver Administradores

Nuevo Administrador

Nuevo administrador/ Editar Administrador

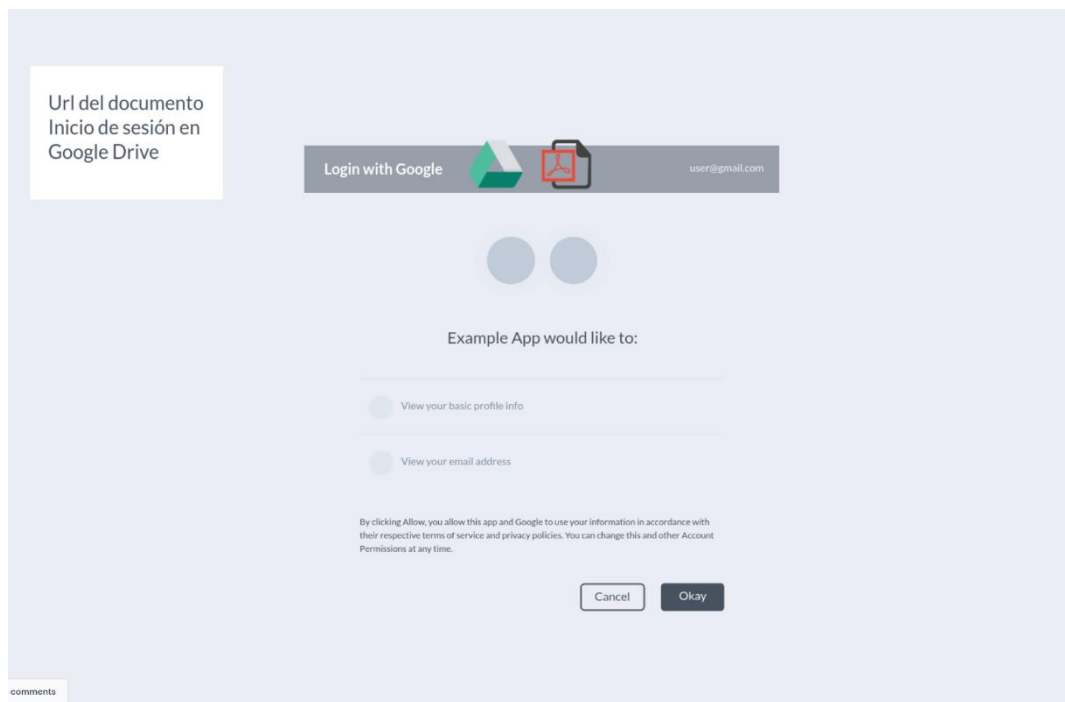
Nombre

Email

Contraseña

Guardar

15. Drive:



16. Alerta Borrado:



Anexo 3: Manual de puesta en marcha

Instalación de wampserver

Descargar e instalar wampserver desde el enlace <https://www.wampserver.com/en/>.

Creación la base de datos

Acceder a <http://localhost/phpmyadmin/> y crear una nueva base de datos desde la primera opción del menú.

phpMyAdmin:



Nombre: j_

Bases de datos

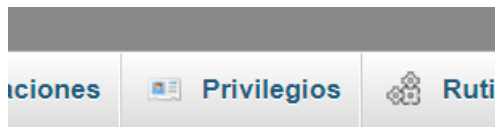
 Crear base de datos 

j_	latin1_spanish_ci	Crear
---------------------------------	--	--------------------------------------

Creación de usuario de la base de datos

Desde el menú, hacer clic en privilegios para crear el usuario de la base de datos.

Privilegios



Agregar cuenta de usuario

Agregar cuenta de usuario

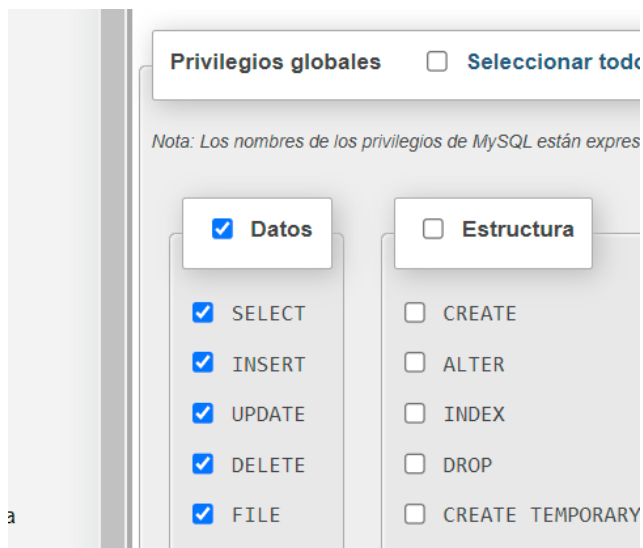
Información de la cuenta

Nombre de usuario:	Use el campo de texto	jaenvivienda
Nombre de Host:	Cualquier servidor	%
Contraseña:	Use el campo de texto	Strength:
Debe volver a escribir:		
Complemento de autenticación	Autenticación de MySQL nativo	
Generar contraseña:	Generar	

Base de datos para la cuenta de usuario

- ☐ Crear base de datos con el mismo nombre y otorgar todos los privilegios.
- ☐ Otorgar todos los privilegios al nombre que contiene comodín (username_%).
- ☒ Otorgar todos los privilegios para la base de datos j_.

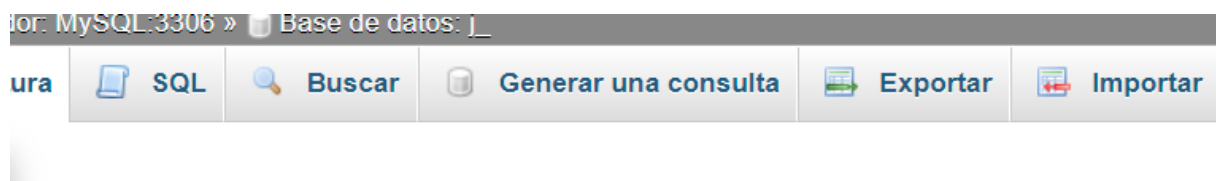
Privilegios



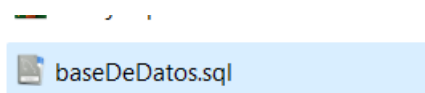
Importar la base de datos

Importar el archivo o copiar y pegar el texto en SQL.

Menú phpMyAdmin



Archivo sql de la base de datos



Instalar Nodejs

Instalar Node js desde el enlace <https://nodejs.org/es/download/>.

Instalación

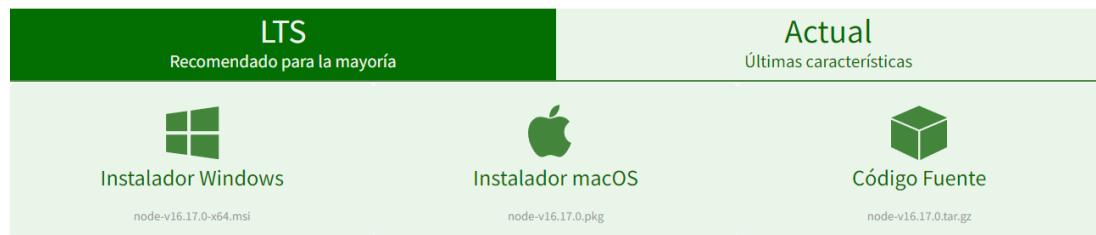
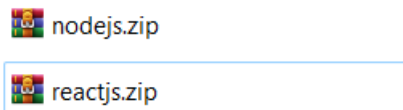


Ilustración 0.1

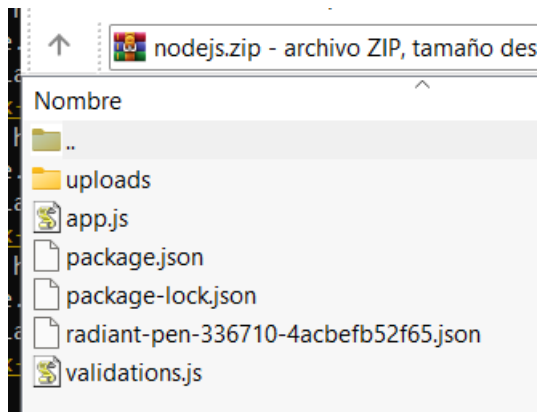
Descarga de aplicación

Descargar los archivos .zip y descomprimir los archivos:

Archivos zip



Nodejs.zip

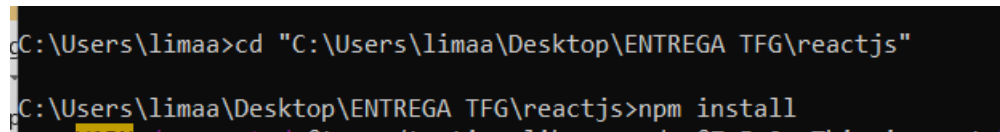


- Abrir el cmd, ejecutar `npm install --global yarn`
- Abrir el archivo nodejs y ejecutar: `yarn serve`
- Abrir el archivo reactjs y ejecutar: `npm install`

cmd

```
C:\Users\limaa\Desktop\ENTREGA TFG\nodejs>yarn install
yarn install v1.22.17
```

cmd



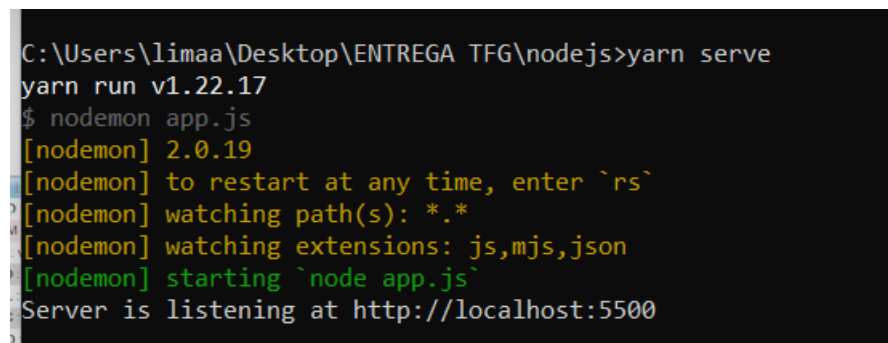
```
C:\Users\limaa>cd "C:\Users\limaa\Desktop\ENTREGA TFG\reactjs"  
C:\Users\limaa\Desktop\ENTREGA TFG\reactjs>npm install
```

Lanzamiento de aplicación

Abrir en un cmd un archivo y en otro cmd el otro.

- En el archivo nodejs, ejecutar : yarn serve
- Ejecutar rs para volver a lanzarla

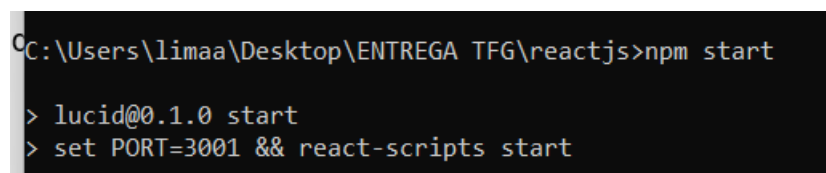
Cmd



```
C:\Users\limaa\Desktop\ENTREGA TFG\nodejs>yarn serve  
yarn run v1.22.17  
$ nodemon app.js  
[nodemon] 2.0.19  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,json  
[nodemon] starting `node app.js`  
Server is listening at http://localhost:5500
```

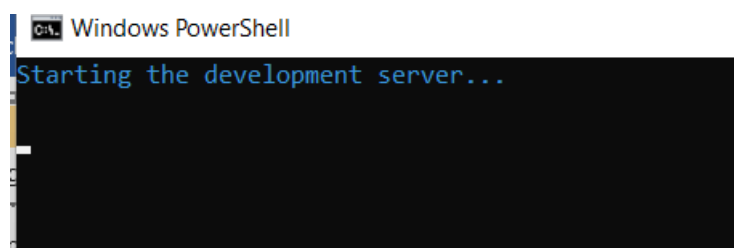
- En el archivo reactjs, ejecutar : npm start

Cmd



```
C:\Users\limaa\Desktop\ENTREGA TFG\reactjs>npm start  
  
> lucid@0.1.0 start  
> set PORT=3001 && react-scripts start
```

Cmd



```
Windows PowerShell  
Starting the development server...
```

Ilustración 0.2

- En el primer lanzamiento, tarda un par de minutos en ejecutarse.
- Automáticamente se abrirá una ventana del navegador con el inicio de sesión

Inicio de aplicación localhost

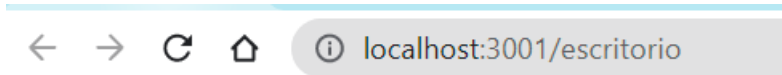


Ilustración 0.3

Iniciar sesión

Iniciar sesión como:

- Administrador
 - Usuario: celia.macias008@gmail.com
 - Contraseña: gm1234
- Cliente
 - Usuario: mcstrs9@gmail.com
 - Contraseña: gm1234
- Trabajador
 - Usuario: limaalimon1@gmail.com
 - Contraseña: gm1234

Detener ejecución

Pulsar Ctrl + C para detener la ejecución en el cmd

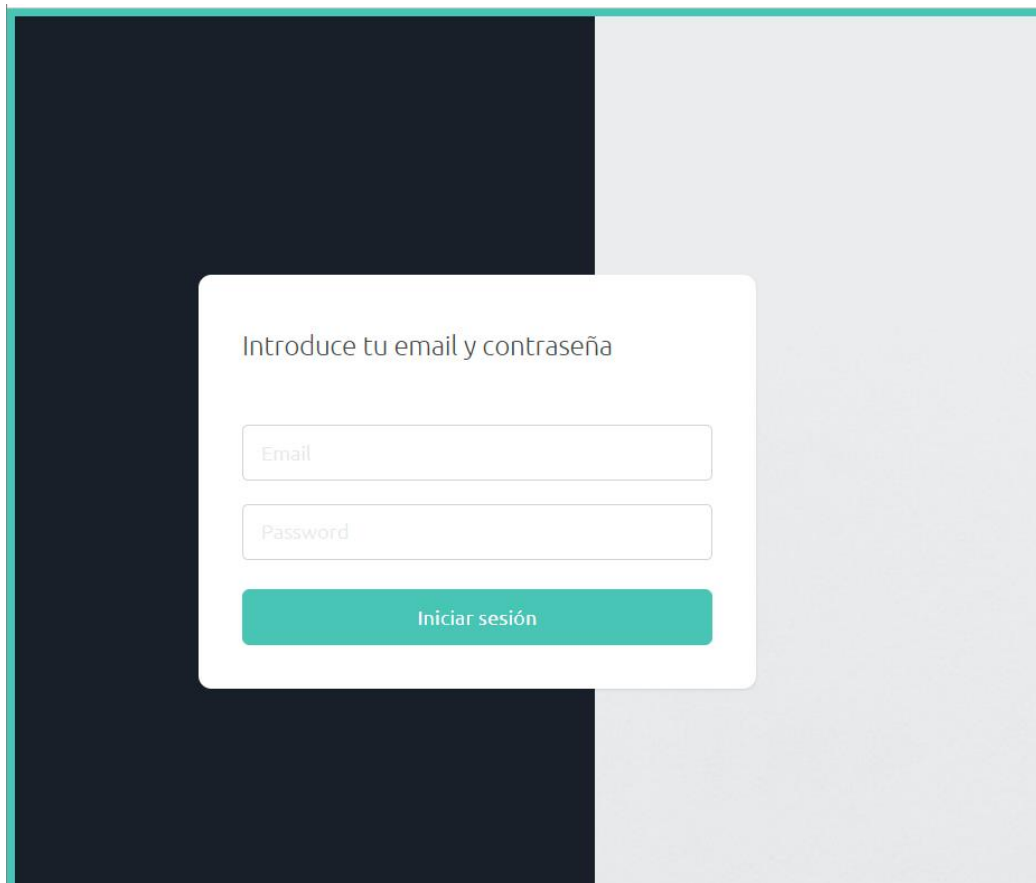
Anexo 4. Manual de usuario

Inicio de sesión

Creamos un administrador de ejemplo desde la base de datos, para realizar las pruebas correctamente.

El sistema de inicio de sesión funcionará mediante cookies. El sistema guardará el estado del usuario si está identificado en la plataforma, y, en caso contrario, lo redirigirá al inicio de sesión, impidiendo el acceso a otras pantallas por usuarios que no estén registrados.

Inicio de sesión:

A login form overlay is centered on a dark blue background. The form has a white background and rounded corners. At the top, it says "Introduce tu email y contraseña". Below this, there are two input fields: "Email" and "Password". At the bottom of the form is a teal button labeled "Iniciar sesión". The background is split into a dark blue left half and a light gray right half.

Si el correo no existe en la base de datos o la contraseña no es correcta, aparecerá un mensaje de error:

Error inicio de sesión:



Introduce tu email y contraseña

correo@error.com

Password

Iniciar sesión

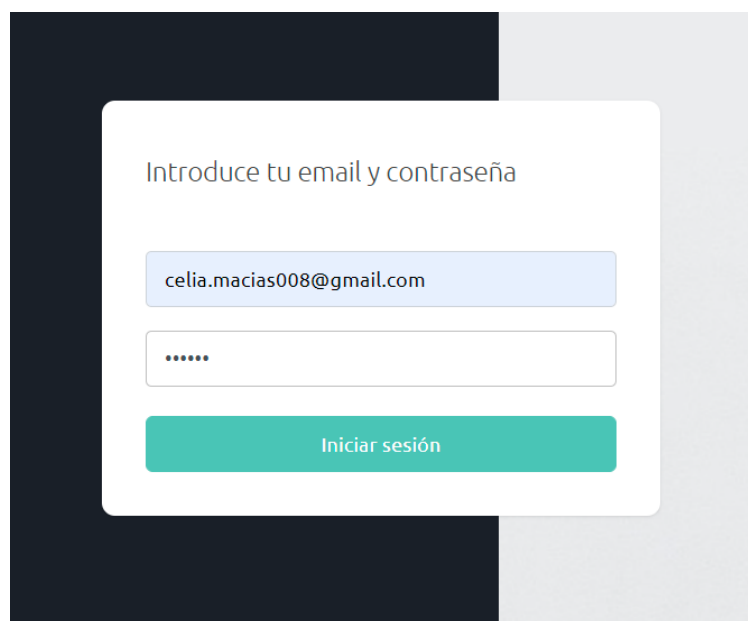
No existe usuario o contraseña

This image shows a login form with a title 'Introduce tu email y contraseña'. It has two input fields: the first contains 'correo@error.com' and the second is labeled 'Password'. Below the fields is a green 'Iniciar sesión' button. A red error message box at the bottom states 'No existe usuario o contraseña'.

En caso contrario, la app redirigirá a la vista principal.

Manual del Administrador

Inicia sesión con tu usuario y contraseña



Introduce tu email y contraseña

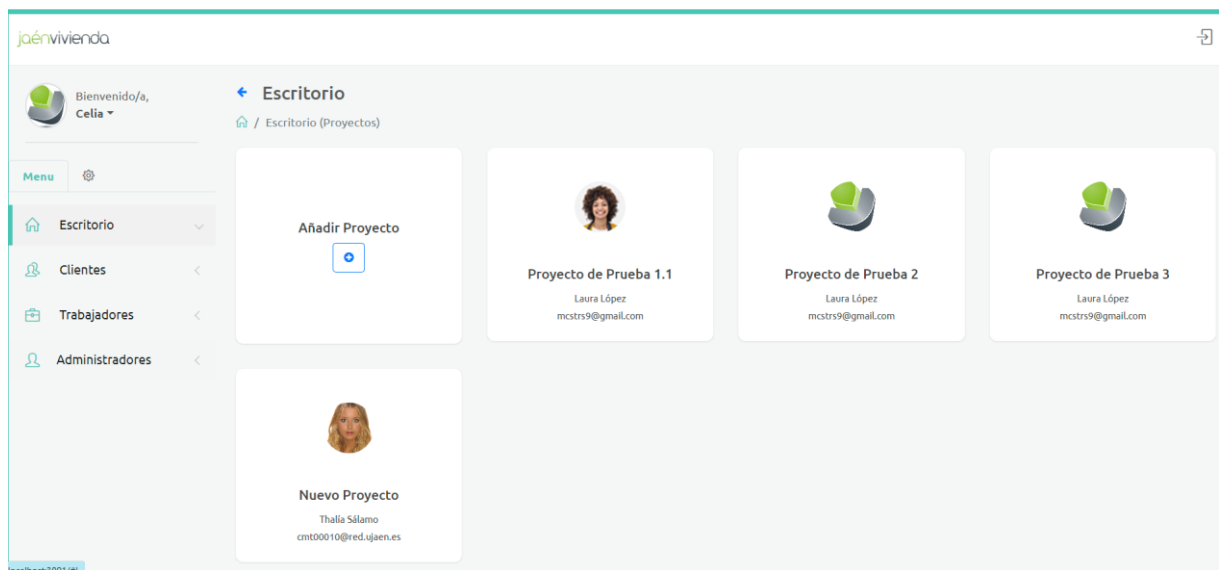
celia.macias008@gmail.com

.....

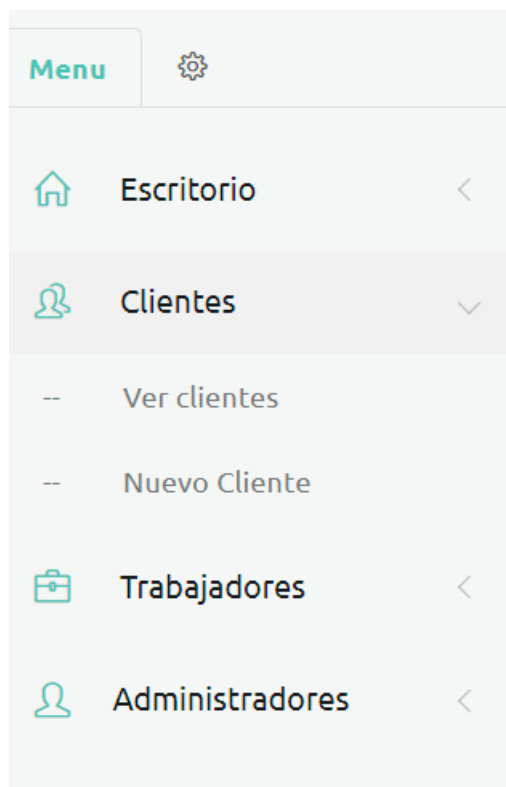
Iniciar sesión

This image shows a login form with a title 'Introduce tu email y contraseña'. It has two input fields: the first contains 'celia.macias008@gmail.com' and the second is filled with dots. Below the fields is a green 'Iniciar sesión' button. The form is centered on a dark blue and light gray background.

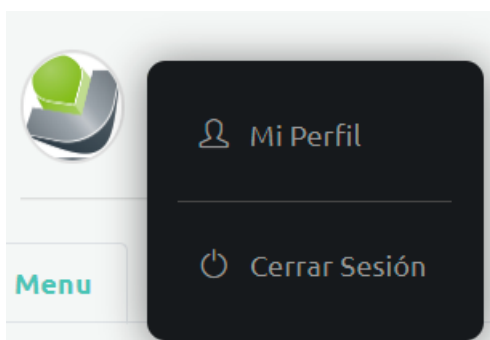
Accede al escritorio, donde podrás ver los proyectos que has creado y el cliente al que está asociado cada proyecto.



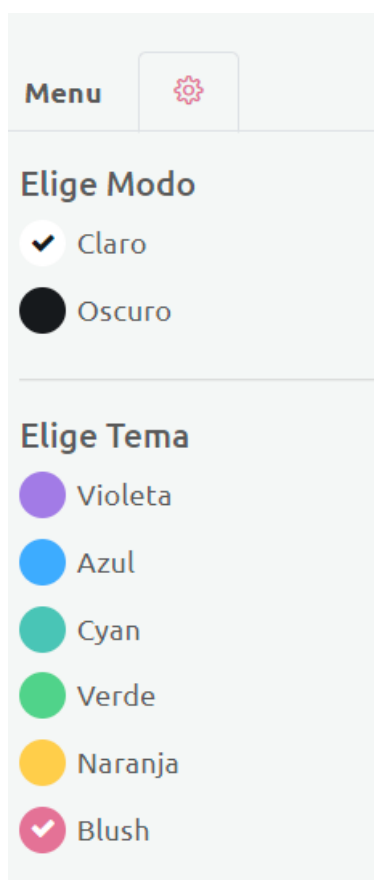
Desde el menú puedes dirigirte a los clientes, crear nuevos, ver a los trabajadores, crear nuevos, ver a los administradores, crear nuevos o volver al escritorio.



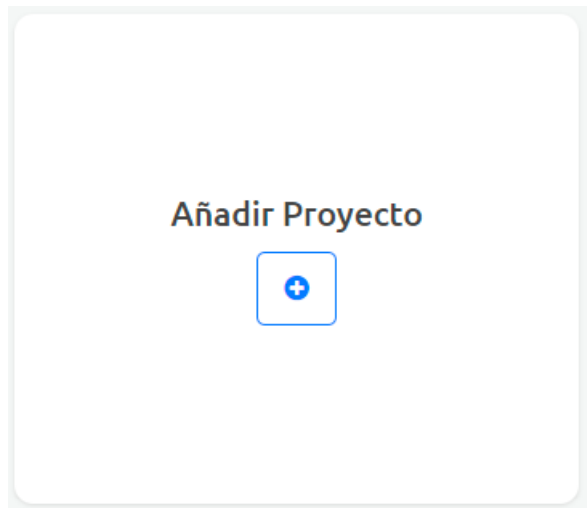
Puedes acceder a tu perfil o cerrar session desde el icono que ves al lado de tu nombre.



Puedes cambiar el tema de la aplicación desde el símbolo de ajustes.



Pulsa el icono para añadir un nuevo proyecto



← **Nuevo proyecto**

🏠 / Nuevo proyecto

Nuevo Proyecto

Título*

Proyecto Tal

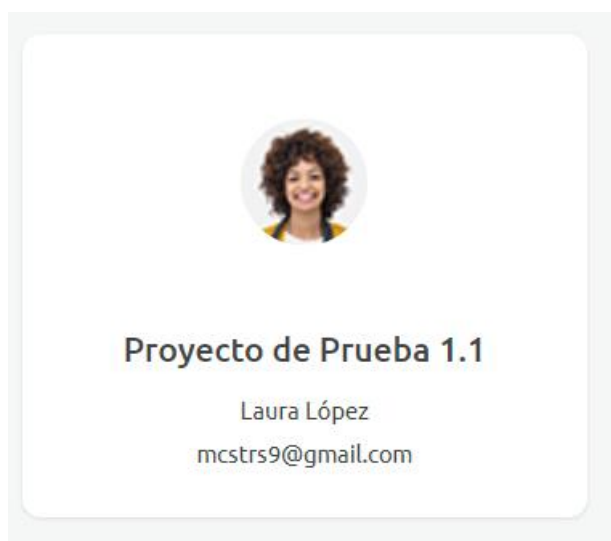
Selecciona.... ↕ Cliente Asociado

Descripción

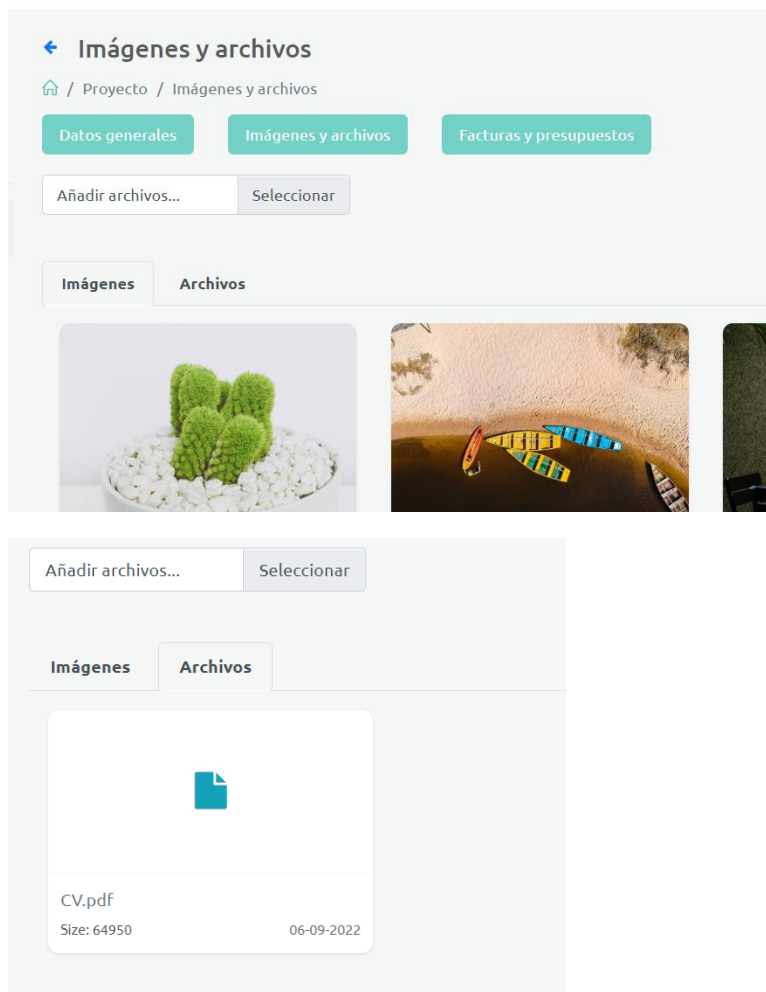
Descripción del proyecto

Guardar

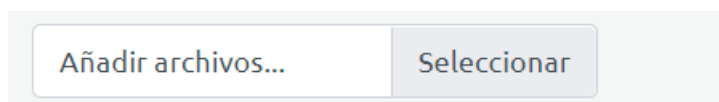
Haz clic en cualquier proyecto para dirigirte a él.



Accederás a la vista principal del proyecto, donde verás la pestaña de imágenes y de archivos



Puedes hacer una inserción masiva de archivos haciendo clic en Seleccionar.



Puedes editar los datos del proyecto en Datos generales

Datos Generales

🏠 / Proyecto / Datos generales

Datos generales Imágenes y archivos Facturas y presupuestos

Editar Proyecto

Título*

Proyecto de Prueba 1.1

Laura López Caño Cliente Asociado

Descripción

Este proyecto es una prueba editada

Guardar

Puedes añadir o editar los datos de las facturas y presupuestos. En el menú acciones encontrarás las posibilidades que ofrece cada tabla.

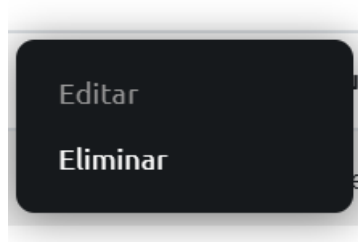
Facturas y Presupuestos

🏠 / Proyecto / Facturas y Presupuestos

Datos generales Imágenes y archivos Facturas y presupuestos

Nueva factura o presupuesto

#	Acciones	Título	Tipo	Fecha	Ver	Descargar
12	Editar Eliminar	Pruebaa	Presupuesto	04-09-2022		



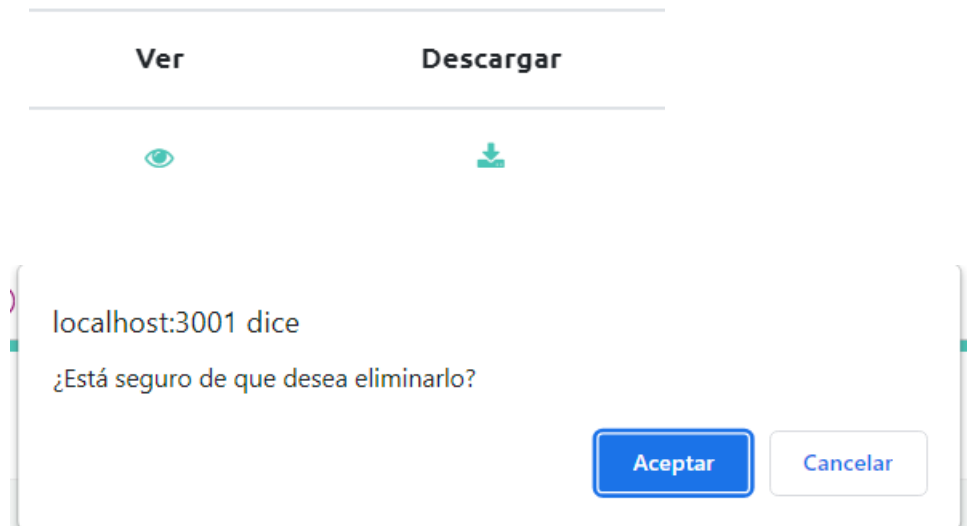
Antes de eliminar un archivo, siempre habrá un aviso, por seguridad.

localhost:3001 dice

¿Está seguro de que desea eliminarlo?

Aceptar Cancelar

Puedes ver y descargar los archivos haciendo clic en el icono de visión o de descarga



Desde la ventana de clientes, puedes ver y editar a tus clientes.

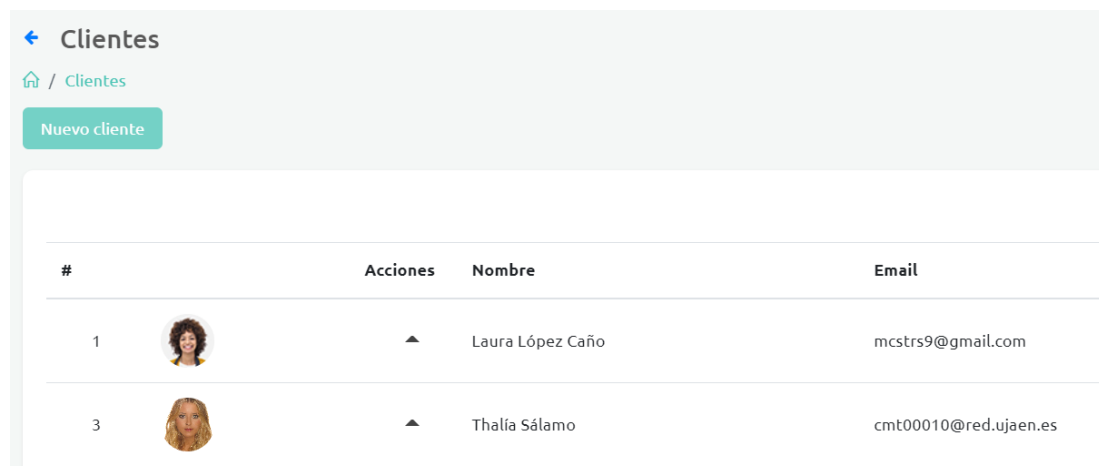


Ilustración 0.4

Desde la ventana de trabajadores, puedes verlos, editarlos, borrarlos, y acceder a sus documentos.

← **Trabajadores**

🏠 / Trabajadores

Nuevo trabajador

#	Acciones	Nombre	Email	DNI	
44	- Documentos - Editar - Eliminar	Celia Macías Torres	limaalimon1@gmail.com	77367311A	65
45	▲	Paqui Torres Díaz	paquitorres65@gmail.com	12345678N	15
46	▲	David Mérida Mesa	david@gmail.com	12344567A	
55	▲	Prueba Prueba PRueba	prueba@prueba.com		

Ilustración 0.5

Puedes crear nuevos administradores y editar o eliminar los existentes.

← **Administradores**

🏠 / Administradores

Nuevo administrador

#	Acciones	Nombre	Email
1	▲	Celia	celia.macias008@gmail.com

Ilustración 0.6

9.1.1. Manual del Cliente

Inicia sesión con tu usuario y contraseña. Puedes acceder a tu perfil y editar tus datos, cerrar sesión y ver toda la información referente a tus proyectos.



Ilustración 0.7



Ilustración 0.8



Ilustración 0.9



Ilustración 0.10

9.1.2. Manual del trabajador

Inicia sesión con tu usuario y contraseña. Puedes acceder a tu perfil y editar tus datos, cerrar sesión y ver toda la información referente a tus documentos.

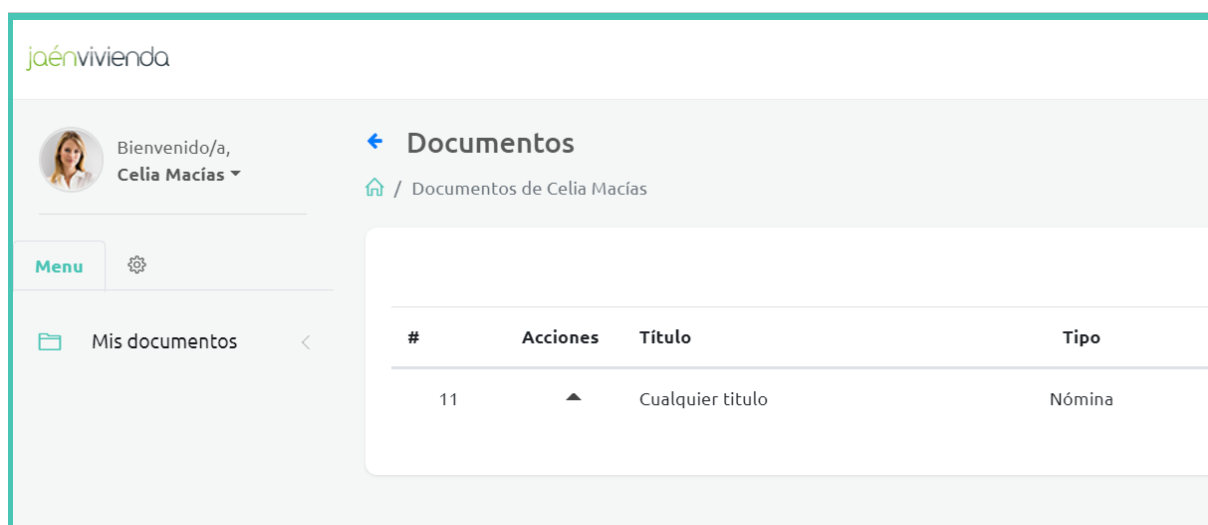


Ilustración 0.11