Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN SISTEMAS DE INFORMACIÓN

Desarrollo de una aplicación web para gestionar proyectos

Estudante: Fernández Suárez, Adrián
Dirección: Cabrero Souto, David

A Coruña, setembro de 2019.

Dedicado a la tozuded de mi madre, sin ella no hubiese llegado hasta aquí.

Agradecimientos

Después de un intenso período, hoy es el día: escribo este apartado de agradecimientos para finalizar mi trabajo de fin de grado. Ha sido un período de aprendizaje intenso, no solo en el campo de la tecnología sino también a nivel personal. Escribir este trabajo ha tenido un gran impacto en mi y es por eso que me gustaría agradecer a todas aquellas personas que me han ayudado y apoyado durante este proceso.

Primero de todo, me gustaría agradecer a mis compañeros de prácticas por su colaboración. Me habéis apoyado enormemente y siempre habéis estado ahí para ayudarme cuando lo necesitaba.

En segundo lugar quiero agradecer a mis compañeros de trabajo que poco a poco se han convertido en personas muy importantes para mi y me han ayudad más de lo que se puedan llegar a imaginar.

También me gustaría agradecer a mi familia por sus sabios consejos y su comprensión. Siempre han estado ahí para mi.

Finalmente, a mis amigos, por estar ahí cuando los he necesitado.

¡Muchas gracias a todos!

Resumen

El mercado de las aplicaciones para la gestión de proyectos y de credenciales es bastante amplio pero a pesar de ello, siguen existiendo necesidades que aún no han sido cubiertas.
Este proyecto aborda la problemática actual en la organización de equipos de trabajo y gestión de proyectos a la que se enfrentan las empresas que ofrecen servicios de diseño gráfico y web actualmente. Para ello, se aporta una solución tecnológica que permita solventar este problema.
Con el desarrollo de esta aplicación web se busca facilitar el trabajo inherente a la creación de nuevos proyectos y a la gestión de las credenciales generadas. La aplicación permite la automatización de estos procesos para que las empresas puedan gestionar sus proyectos desde cero, administrando todos los recursos disponibles y controlando los tiempos de un modo más eficaz.

Abstract

Application's market for project's and credential's management is quite huge, nevertheless there are still some needs that are not fulfilled yet.
This project focuses on the real problematic of team's organization on the workplace and project management that graphic and web design companies are now facing with. Within this framework, this project gives a technological solution that could be implemented for solving this issue.
Developing this project pretends to be the solution for creating new projects and for managing generated credentials. This app allows the automation of these processes for companies to develop this so-called processes from the very beginning, which means that companies could manage their available resources and controlling their timings in a more efficient way.

Palabras clave:

- PHP7
- Tareas
- Wordpress
- Laravel
- Javascript
- jQuery
- REST
- CRUD
- VestaCP
- Pusher
- Echo

- Vue
- Axios
- Gantt

Keywords:

- PHP7
- Tasks
- Wordpress
- Laravel
- Javascript
- jQuery
- REST
- CRUD
- VestaCP
- Pusher
- Echo
- Vue
- Axios
- Gantt

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introdución

La labor de un administrador de sistemas en un entorno de trabajo puede resultar monótona ya que existen multitud de pequeñas tareas repetitivas que lastran su actividad y acaban por reducir su rendimiento. Automatizar esas tareas puede ayudar no solo a incrementar el rendimiento del propio administrador de sistemas sino que, además, puede incrementar el rendimiento de todos los empleados incrementando así el rendimiento de la empresa.

Existe también otra tarea, inherente a toda empresa, que es la gestión del trabajo a realizar. Dicha tarea es, quizá, la más importante y la más difícil de todas ya que de ella depende el buen funcionamiento de la empresa.

Con esto en mente se puede entender que la creación de un nuevo proyecto en una empresa requiere seguir unos pasos ya establecidos y comunes a la creación de otros proyectos anteriores. También es fácil darse cuenta de que una vez creado este nuevo proyecto, es necesario planificarlo y hacer un seguimiento de sus tareas. Por este motivo se pensó en buscar una aplicación que automatizase todo el proceso pero al no dar con ninguna que cumpliese todos los requisitos, se propuso desarrollar una nueva aplicación que si lo hiciese.

El mercado de las aplicaciones para gestión de proyectos es un mercado que comienza a estar saturado y, sin embargo, siguen existiendo necesidades que aún no han sido cubiertas completamente. El desarrollo de esta aplicación ha sido motivado por esas carencias identificadas y la necesidad de solventarlas.

Una simple búsqueda en Internet nos arroja un sin fin de aplicaciones para la gestión de proyectos o para la gestión de credenciales. Pero las soluciones encontradas están lejos de cubrir los requisitos necesarios. Las aplicaciones encontradas más destacadas son:

- **Asana:** Asana es una herramienta de gestión de tareas y proyectos, permite a los equi-

pos compartir, planificar, organizar, y seguir el progreso de las tareas en las que cada miembro está trabajando. Asana es una aplicación web sencilla y fácil de utilizar.[1]

- **Bitrix24:** Bitrix24 es un espacio de trabajo unido que maneja los muchos aspectos de operaciones diarias y tareas. CRM, Project Management y una Plataforma de Colaboración para su gestión empresarial. Cloud o self-hosted.[2]
- **Slack:** Slack es una herramienta de colaboración en equipo. Slack permite abarcar todos los detalles, desde los primeros pasos de proyectos hasta conversaciones sobre presupuestos.[3]
- **LastPass:** LastPass es un gestor de contraseñas freemium que almacena las contraseñas encriptadas en la nube. El objetivo de LastPass es resolver el problema de la fatiga de contraseñas centralizando la gestión de usuario y contraseña en la nube. Las contraseñas en LastPass Password Manager son protegidas por una contraseña maestra, son cifradas localmente y son sincronizadas a cualquier otro navegador soportado. LastPass ofrece un complementador automático de formularios web para automatizar la autenticación y generar nuevas altas en sitios web, todo ello con generación automática de contraseñas.[4]

No es intención de este proyecto el competir con estas aplicaciones sino, más bien, combinar unas funcionalidades ya existentes para facilitar el trabajo al crear nuevos proyectos en la empresa y gestionar luego las credenciales generadas.

La aplicación debía aportar una solución tecnológica a unas demandas concretas en el marco de una pequeña empresa de diseño y desarrollo web. Por tanto la aplicación desarrollada está diseñada a medida para cubrir estas necesidades y se ajusta a ciertas limitaciones inherentes al entorno de trabajo.

En el caso que nos ocupa, los proyectos son gestionados mediante pequeñas tareas de duración determinada y asignados a una única persona. Dichas tareas pueden asignarse a otros usuarios siendo necesario un sistema de anotaciones para mantener el flujo de información. Además, en estos proyectos, se generan una documentación y unas credenciales que es necesario guardar y compartir con las personas implicadas en cada proyecto. Y por último debe controlarse que usuarios tienen acceso a las credenciales.

Este documento recoge la memoria del proyecto. En él se enumeran los objetivos fijados, se hará una descripción de cada una de las metodologías y tecnologías empleadas y una explicación final de la aplicación desarrollada.

Capítulo 2

Objetivos

ESTA es la lista de objetivos a cumplir en el proyecto:

- Gestionar proyectos
  - Crear proyectos
  - Mostrar información de proyectos
  - Modificar la información de los proyectos
  - Borrar proyectos
  - Gestionar tareas
    - Crear tareas
    - Asignar tareas a usuarios
    - Mostrar información y estado de las tareas
    - Modificar información de tareas
    - Comentar tareas
- Gestionar información de webs
  - Crear sitios web instalando Wordpress
  - Guardar credenciales
  - Mostrar credenciales a los usuarios
  - Modificar la información de los sitios web
  - Borrar sitios web
  - Gestionar repositorio de plugins
    - Añadir plugins
    - Borrar plugins

        * Modificar plugins
        * Eliminar plugins
        * Ver lista de plugins
        * Instalar plugins automáticamente al instalar Wordpress
    - Gestionar repositorio de temas
        * Añadir temas
        * Modificar temas
        * Ver lista de temas
        * Borrar temas
        * Instalar temas automáticamente al instalar Wordpress
- Gestionar usuarios
    - Crear usuarios
    - Borrar usuarios
    - Deshabilitar usuarios
    - Autenticar usuarios
    - Gestionar permisos
        * Crear roles
        * Asignar roles
        * Borrar roles
- Integrar con Producteev

Capítulo 3

Tecnologías y metodologías

A lo largo del proyecto se han empleado y combinado un conjunto de tecnologías y metodologías. No todas son ampliamente conocidas, por ese motivo se recopilan a continuación una lista de metodologías y tecnologías que pueden no ser de dominio público.

3.1 SCRUM

Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. SCRUM adopta plenamente los principios de los métodos de desarrollo ágiles y los incorpora a la gestión de proyectos. Partiendo de la idea de que todos los requisitos están inicialmente sin perfeccionar y son poco claros. SCRUM se centra en mejorar la capacidad del equipo de desarrollo para observar y adaptarse a las nuevas exigencias.

En SCRUM se siguen dos pasos esenciales: inspeccionar y adaptarse. Siguiendo esta filosofía SCRUM se centra en la realización de entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, los requisitos son cambiantes o poco definidos y la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.[5]

3.2 VestaCP

VestaCP es uno de los paneles de control web de código abierto más utilizado actualmente, gracias a la simplicidad de su funcionamiento, robustez y rapidez. Está pensado para hacer web hosting de manera sencilla ya que se basa en una serie de scripts que facilitan el trabajo habitual y al mismo tiempo permite que se pueda configurar manualmente cada una e las apli-

caciones. Al emplear aplicaciónes conocidas (MySQL, Apache, nginx...) hace que encontrar soluciones a posibles problemas sea más sencillo.[6]

3.3 Pusher channels

Pusher es un servicio en la nube que encapsula la implementación de websockets y la funcionalidad de la aplicación sin necesidad de tener que ejecutar un servidor de Websockets.[7]

3.4 Axios

Axios es un cliente HTTP basado en promesas que funciona tanto en el lado del cliente (navegador) como en el lado del servidor y que permite configurar y realizar sencillas solicitudes a un servidor y recibiendo respuestas fáciles de procesar.[8][9][10]

Axios ofrece las siguientes ventajas:

- La API es unificada para las solicitudes Ajax.
- Está optimizado el consumo de servicios web, API REST y que devuelvan una respuesta en formato JSON.
- Muy fácil de utilizar.
- Es una librería muy ligera.
- Compatibilidad con las versiones actuales de los navegadores.

3.5 Vue

Vue es un framework open source de JavaScript, creado por Evan You en el año 2014, el cual nos permite construir interfaces de usuarios de una forma muy sencilla.
Una de las características más importantes de Vue es el trabajo con componentes. Los componentes permiten desarrollar proyectos modularizados y fáciles de escalar.
Un componente Vue es un elemento el cual se encapsula código reutilizable. Dentro de un componente podremos encontrar etiquetas HTML, estilos de CSS y código JavaScript.[11]

3.6 ziggy

Ziggy es un módulo de Laravel que permite acceder a las rutas de la aplicación desde Javascript. La verdadera ventaja que ofrece este modulo es que permite acceder a las rutas empleando el nombrado de rutas igual que se haría en PHP mediante el comando route().[12]

3.7 Flysystem-SMB

Flysystem-smb es una librería que potencia el sistema de almacenamiento de archivos de Laravel al incluir una implementación más amigable para trabajar con el protocolo Samba. Flysystem-smb necesita la librería Icewind-smb para poder funcionar.

3.8 Icewind-smb

Icewind-SMB es una encapsulación de PHP para smbclient y libsmbclient-php que permite hacer uso del protocolo Samba desde PHP.[13]

3.9 Boostrap Multiselect

Bootstrap Multiselect es un plugin basado en JQuery y Bootstrap que proporciona una interfaz intuitiva para el uso de select inputs con selección múltiple.[14]

3.10 jsGantt Improved

La librería jsGantt Improved es una librería de jQuery que permite crear diagramas de Gantt con multitud de funcionalidades[15] como:

- Tareas y grupos de tareas desplegables
- Múltiples dependencias
- Indicador de porcentaje de tarea completado
- Estilos propios para cada tipo de tarea
- Hitos
- Recursos empleados en la tarea

3.11 Laravel

Laravel es un framework PHP de código abierto para el desarrollo de aplicaciones y servicios web. Nace en 2011 de la mano de Taylor Otwell y desde entonces su comunidad ha ido creciendo haciendo de Laravel un framework robusto y confiable lleno de funcionalidades para suplir las demandas de los desarrolladores.[16]

La filosofía de Laravel se basa en el uso de una sintaxis elegante y expresiva haciendo del desarrollo una experiencia agradable y creativa. Siguiendo esta filosofía Laravel hace uso del patrón MVC (Modelo-Vista-Controlador) e implementa a su vez las mejores características de otros frameworks de desarrollo y la ventaja aportada de las últimas versiones de PHP.

El uso de Laravel facilita las tareas comunes utilizadas en la mayoría de los proyectos web, como la autenticación, enrutamiento, sesiones y almacenamiento en caché.

3.11.1 EloquentORM

Laravel emplea un ORM (Mapeo Objeto-Relacional) llamado Eloquent para mapear los datos que se encuentran en la base de datos almacenados en SQL a objetos de PHP y viceversa con la idea de tener un código portable sin tener que escribir las llamadas SQL directamente. En el patrón de diseño MVC Eloquent se encarga de gestionar la parte correspondiente al Modelo de la aplicación.

Como todo en Laravel, el uso de Eloquent es sencillo. Se centra en definir unas clases Modelo con los atributos que los componen y las relaciones que hay entre los modelos. Estas relaciones se corresponden con las relaciones que existen entre las distintas tablas de la base de datos. Así se tiene una representación del esquema relacional en PHP y, junto con las funciones que aporta Eloquent, se pueden realizar las distintas consultas.

Aunque el uso de Eloquent facilita el trabajo con los datos de la aplicación, al mismo tiempo entraña una serie de limitaciones impuestas por Eloquent para su uso eficaz como son el empleo de unas directrices concretas al nombrar los modelos y las tablas con las que se relacionan.[17]

3.11.2 Blade

Las vistas son la parte pública de la aplicación, es lo que el usuario vé y es la parte de la aplicación con la que interactúa. En Laravel, para gestionar las vistas, se emplea el motor de plantillas Blade.

Como ya se ha mencionado, una de las ideas detrás de Laravel es la de programar un código sencillo y limpio y en esto ayuda mucho la modulación, una de las principales característica de Blade. En Blade el código se divide en pequeños segmentos reutilizables e independientes entre sí llamados plantillas o templates. Estos pequeños segmentos se unen para formar estructuras más complejas conformando finalmente la vista que se le presenta al usuario. Un template puede estar compuesto por otros templates generando una estructura de jerarquía.

A la raíz de dicha estructura se le suele llamar "Layout" y en ella irán incluidos la cabecera de HTML, las importaciones de las hojas CSS y JavaScript.[17][16]

Los templates de Blade suelen estar escritos en HTML pero admiten además código PHP y una sintaxis propia de Blade que permite trabajar con la propia vista reduciendo el código necesario para generarla y añadiendo funcionalidades extra como estructuras de control, llamada a otros templates, etc. La vista finalmente se conformará en el motor Blade generando un código PHP que es guardado en caché de forma que acceder a las vistas no supone una carga extra para el servidor.

3.11.3 Controladores

Un controlador es una clase PHP que dispone de métodos públicos que son el punto de entrada final de una petición HTTP (Request PHP) a la aplicación. En los controladores suele encontrarse la lógica de la aplicación.

3.11.4 Rutas

Todas las peticiones HTTP se realizan a las de rutas y estas son las encargadas de llamar a los métodos para que realicen las acciones correspondientes. Las rutas se almacenan en archivos en función de su uso (web, api...) y dichos archivos pueden estructurarse jerárquicamente.

Una de las ventajas de Laravel es que es capaz de realizar pequeñas acciones para el usuario desde el nivel de rutas sin tener que llegar a los controladores.[16]

3.11.5 Middlewares

Los middlewares son mecanismos que permiten agregar capas adicionales de lógica para realizar un filtrado sobre las peticiones HTTP de la aplicación. Se ejecutan antes de realizar el enrutamiento. Su funcionalidad es muy variada siendo especialmente útiles para controlar la autenticación del usuario, realizar comprobaciones de seguridad y transformar la información recibida para que sea fácilmente manejable al llegar a la aplicación.[17]

3.11.6 Artisan

Desarrollar en Laravel es fácil e intuitivo pero a mayores Laravel aporta una herramienta muy poderosa que son los comandos artisan. Con estos comandos se puede hacer multitud de acciones, desde generar y poblar la base de datos hasta generar estructuras de código básicas para cualquier elemento empleado en Laravel (Modelos, controladores, Eventos, etc).

3.11.7 Laravel Echo

Laravel Echo es una librería JavaScript que facilita subscribirse a los canales de emisión y escuchar por eventos emitidos por Laravel.

Capítulo 4

Aplicación

La solución tecnológica desarrollada es una aplicación web creada empleando el framework de desarrollo Laravel y por tanto codificada principalmente en PHP 7. Al usar Laravel se explota enormemente el patrón de Modelo-Vista-Controlador. En las vistas se emplea principalmente los estilos de Bootstrap haciendo que toda la aplicación mantenga los mismos estilos de diseño.

La elección de un entorno web se debe a varios motivos:

- Facilidad de implementación.
- Facilidad de mantenimiento.
- Fácilmente escalable.
- Las plataformas con las que interactúa están basadas en entornos web (Wordpress, VestaCP, etc).
- Las tecnologías y lenguajes empleados son ampliamente conocidos.

4.1 Metodología de trabajo

Para el desarrollo de esta aplicación se ha empleado SCRUM como metodología de trabajo pero adaptando el proceso para una sola persona.

Esta aplicación se ha planteado desde un principio como una primera iteración enmarcada en un desarrollo iterativo incremental siguiendo tal como se sugiere en la metodología SCRUM. Así pues el resultado obtenido, si bien es una aplicación funcional que cubre las necesidades básicas, dista mucho de ser una aplicación totalmente terminada.

En la tabla de la figura 4.1 (página 13) se recopilan todas las tareas realizadas, el tiempo estimado que ha llevado completarlas, el tiempo real y el estado actual.

Nombre de tarea	Prioridad	Tiempo estimado	Tiempo total	Estado
Diseñar base de datos del servidor	10	5	6	Finalizado
Modelar UML de la aplicación	9	12	15	Finalizado
Crear tests unitarios de la aplicación	1	40	-	Pendiente
Crear base de datos de la aplicación	8	1	13	Finalizado
Crear modelos de la aplicación	9	12	11	Finalizado
Listar proyectos	6	6	7	Finalizado
Ver información del proyecto	5	12	12	Finalizado
Identificar usuario	7	6	2	Finalizado
Crear proyecto web	6	12	13	Finalizado
Crear un proyecto de diseño	6	12	5	Finalizado
Instalar Wordpress automáticamente	5	30	64	Finalizado
Crear tarea	4	20	33	Finalizado
Modificar tarea	4	20	26	Finalizado
Eliminar tarea	4	20	29	Finalizado
Añadir comentario	4	12	32	Finalizado
Guardar credenciales instalación	4	25	22	Finalizado
Mostrar credenciales proyecto	6	25	17	Finalizado
Añadir credenciales al proyecto	6	20	22	Finalizado
Borrar credenciales	6	20	12	Finalizado
Borrar proyecto	5	12	9	Finalizado
Editar proyecto	6	12	19	Finalizado
Editar credenciales	6	6	7	Finalizado
Crear árbol de directorios	6	6	34	Finalizado
Crear usuario	12	12	16	Finalizado
Editar información de usuario	12	12	15	Finalizado
Listar usuarios	6	6	5	Finalizado
Ver información del usuario	6	6	9	Finalizado
Borrar usuario	6	6	4	Finalizado
Deshabilitar usuario	12	12	14	Finalizado
Crear rol de usuario	20	20	26	Finalizado
Asignar rol de usuario	6	6	8	Finalizado

Continua en la siguiente página

Tabla 4.1 – *Continuado desde la página anterior*

Nombre de tarea	Prioridad	Tiempo estimado	Tiempo total	Estado
Revocar rol de usuario	6	6	5	Finalizado
Subir plugin Wordpress	12	12	16	Finalizado
Eliminar plugin Wordpress	3	3	8	Finalizado
Subir tema Wordpress	12	12	16	Finalizado
Eliminar tema Wordpress	3	3	8	Finalizado
Sincronizar tareas con Producteeve	20	20	-	Pendiente
Crear api REST	25	25	-	Pendiente

Tabla 4.1: Tabla con las tareas en las cuales figuran su estado actual, la prioridad, el tiempo estimado y el tiempo de ejecución

La tabla de la figura 4.1 (página 13) es una recopilación de las tareas que han sido planificadas para completar el proyecto. Cada fila corresponde a una de las tareas que se identifica mediante un "Nombre de Tarea" y recoge la información de "Prioridad", "Tiempo estimado", "Tiempo real" y "Estado". La prioridad describe la dependencia entre las tareas de modo que a mayor prioridad antes debe realizarse esa tarea.

El estado indica la situación actual de la tarea (pendiente o finalizada).
Como se puede observar la estimación de horas para cada tarea no se ajusta para nada con la realidad. En líneas generales se debe a que se subestimó la duración de cada tarea. Las causas son muy variadas pero los motivos más destacados son:

- Falta de conocimientos. Sobre todo al trabajar con Axios, Pusher y Vue.
- El proceso de instalación de Wordpress se alargó demasiado por diversos errores pero principalmente por problemas de conexión.
- Dificultad para trabajar con las tareas y el diagrama de Gantt.

En la tabla también se puede apreciar que existen tareas que estaban planificadas pero nunca llegaron a realizarse principalmente por la falta de tiempo o imposibles de realizar (este es el caso, por ejemplo, de la sincronización con Producteev).

4.1.1 Estimación de costes

La estimación del coste del proyecto es teorica, para su cálculo se tomó un sueldo de 1000€/mes brutos por empleado. Ese sueldo se divide entre las 40 horas laborales del més y

el resultado se multiplica por el total de horas estimadas (para obtener el coste estimado) y horas ejecutadas (para obtener el coste final).

En total el tiempo estimado es de 507 horas con un coste de 13.942,50€ pero el tiempo real de ejecución del proyecto asciende a 556 horas y un coste de 15.290,00€. Todo esto sin tener en cuenta el tiempo que habría que invertir en completar las tareas que faltan.

4.2 Modelos

La aplicación se centra en el manejo de diez recursos básicos (Comentarios, Credenciales, Permisos, Plugins, Proyectos, Roles, Tareas, Temas, Usuarios y Webs) pero solo ha sido necesario generar modelos para trabajar con siete de ellos.

4.2.1 Comentario

Es el modelo que gestiona la información de los comentarios de las tareas.

4.2.2 Credencial

Gestiona la información de las credenciales de una web. Está formado por los siguientes atributos:

- **Usuario:** Es el nombre del usuario de la credencial
- **Contraseña:** Corresponde con la contraseña de la credencial
- **Descripción:** Es un breve texto identificando cual es el uso de esa credencial.

4.2.3 Proyecto

Gestiona la información de un proyecto. Está formado por los siguientes atributos:

- **Nombre:** Es el nombre del proyecto
- **Tipo:** Indica el tipo de proyecto
- **Fecha de inicio:** Fecha en la que está programado que se inicie el proyecto
- **Fecha finalización:** Fecha en la que se fija el fin del proyecto.
- **Descripción:** Contiene una breve descripción del proyecto.

4.2.4 Tarea

Es el encargado de gestionar la información de la tarea de un proyecto.

4.2.5 Archivo subido

Este es un caso especial. Mediante un pequeño truco de Laravel, este modelo es el encargado de representar la información de un plugin o de un tema. Internamente se les trata a los dos de la misma manera ya que los dos son muy similares en cuanto a las operaciones a realizar y la información que representan. Ambos se pueden abstraer como un tipo genérico para representar la información de un archivo subido a un repositorio. Lo único que los diferencia es el tipo de descarga.

En Laravel este tipo de modelos facilitan el trabajo del programador y hacen mucho más limpio el código. Para usarlos simplemente se le indica al modelo de qué tipo es (plugin o tema) y ya se encarga Laravel de gestionarlo todo internamente. Dado que los atributos y los métodos son los mismos en ambos casos no es necesario crear una clase específica para cada tipo de dato. Los atributos que componen este modelo son:

- **Nombre:** Es el nombre del archivo subido
- **Versión:** Es un código que indica la versión del archivo subido.
- **Tipo:** Este campo lo emplea Laravel para saber que clase de recurso es (plugin o tema).
- **Archivo:** Es la ruta del archivo subido.

4.2.6 Usuario

Por defecto este es un modelo que Laravel ya tiene creado junto con todo el sistema de registro y autenticación. Sin embargo durante el desarrollo de la aplicación este es un modelo que ha sido alterado para ajustarse a las necesidades de la misma. Está formado por los siguientes atributos:

- **Nombre:** Es el nombre de usuario.
- **Nombre completo:** Nombre completo del usuario (nombre y apellidos).
- **Correo electrónico:** Es la dirección de correo electrónico del usuario. Se emplea principalmente para autenticar al usuario.
- **Contraseña:** Clave secreta del usuario para acceder a la aplicación. También presenta las siguientes relaciones con los otros modelos:

- **Webs:** Relación 1 - N en la que el usuario es propietario de una web o una lista de webs.
- **Projects:** Relación 1 - N en la que el usuario puede ser el encargado de varias webs.
- **Tasks:** Relación N - 1 en la que el usuario puede estar asignado a varias tareas
- **Comments:** Relación 1 - N en la que un usuario es autor de varios comentarios.

4.2.7 Web

Es el modelo que gestiona la información de las webs almacenada en la aplicación. Está formado por los siguientes atributos:

- **Nombre:** es el nombre de la web.
- **Url:** es el dominio de la web.

4.2.8 Otros Modelos

Existen otros modelos con los que se trabaja en la aplicación pero que ya vienen creados y configurados con el módulo **Spatie Permissions**. Estos son el modelo de **Role** y **Permission** y ambos están relacionados con el modelo **Usuario**.

4.3 Controladores y vistas

Las operaciones realizadas con los modelos están programadas en los controladores. Cada modelo de la aplicación cuenta con su propio controlador y en cada acción se crea una ”guarda” para que se tenga un permiso asociado a cada la acción realizada. De esta forma todas las acciones que se pueden realizar en la aplicación, están guardadas por un permiso de tal forma que si se quiere realizar la acción se debe poseer el permiso correspondiente.

La aplicación está dividida en tres grandes áreas:

- Gestión de usuarios y permisos
- Gestión de proyectos
- Gestión web

4.3.1 Gestión de usuarios y permisos

Pese a que la aplicación está desinada para uso interno, cuenta con un sistema de autenticación de usuario para controlar el acceso a la misma. Este sistema no permite al usuario registrarse por sí mismo y requiere que un usuario con permisos suficientes (el administrador) lo añada a la lista de usuarios. Esto se ha pensado así para que el administrador sea el encargado de las altas y las bajas de usuarios y controlar en todo momento los usuarios activos.

El sistema de autenticación no ha sido necesario implementarlo, Laravel por defecto ya lo trae implementado. Sin embargo sí que fue necesario modificarlo ya que el sistema de autenticación de usuarios permite que un usuario no registrado pueda crear su propio usuario y es algo que, como ya se ha mencionado, no es necesario. Además, la primera vista que se le debe mostrar a un usuario no registrado es la "vista de acceso" para solicitar las credenciales del usuario y por defecto Laravel permite a un usuario no autenticado acceder directamente a la página principal.

La información guardada del usuario es muy básica y se emplea dentro de la aplicación para, entre otras razones, relacionar al usuario con su homólogo de la empresa. Así, un usuario posee un nombre completo, un nombre de usuario (más amigable para interactuar con el propio usuario o con otros usuarios dentro de la empresa), un correo electrónico (necesario para la autenticación y por si en un futuro se necesita implementar un sistema de notificaciones al correo) y una contraseña. Laravel a mayores genera también otros campos de información (**created_at**, **updated_at**, **deleted_at**) empleados principalmente para gestión interna de Laravel.

Operaciones básicas con usuarios

- **Listar usuarios activos:** Devuelve una vista con la lista paginada de los usuarios activos.
- **Añadir nuevo usuario:** Crea un nuevo usuario a partir de la información aportada y lo activa para su uso.
- **Ver usuario:** Muestra la información del usuario.
- **Editar usuario:** Modifica la información del usuario empleando la información aportada a la aplicación.
- **Buscar usuario:** Busca los usuarios cuyo nombre coincide con las palabras clave.

- **Deshabilitar usuarios:** Esta acción permite bloquear a un usuario sin necesidad de borrarlo permanentemente de la aplicación. Está pensado como una acción de borrado ligero (como enviarlo a la papelera) pudiendo en un futuro recuperarlo de nuevo. El único usuario que el administrador no puede deshabilitar o borrar es a sí mismo. Esta medida de seguridad se creó para evitar el supuesto caso de que un administrador pudiese dejar a la aplicación sin usuarios activos con el permiso de administración. De esta forma, solo los administradores pueden eliminar o deshabilitar usuarios (incluidos otros administradores) pero nunca podrán eliminarse o bloquearse a sí mismos.

- **Habilitar usuarios:** Es la acción inversa a deshabilitar usuario. Solo está disponible para usuarios deshabilitados previamente y lo que hace es restaurar el usuario a su estado previo (activo) recuperando todos los privilegios que poseía.

- **Listar usuarios deshabilitados:** Muestra una lista paginada con los usuarios deshabilitados.

- **Eliminar usuarios:** Elimina por completo la información de un usuario. Esta acción es irreversible.

Vistas de usuario

Las vistas de usuario son las vistas que muestran la información del usuario y los permisos. En principio solo son accesibles para el administrador que es el encargado de gestionar los usuarios y permisos.

- **Lista de usuarios:** Es la vista principal de administración de usuarios y desde ella se puede acceder a las demás vistas de gestión de usuarios. Cuenta con una barra de herramientas superior compuesta por un botón para añadir nuevos usuarios (que lleva a la vista de crear usuario), un botón para ver la lista de usuarios deshabilitados y un buscador para facilitar la tarea de administración.
  La parte central está formada por una tabla en la que cada fila contiene información básica de un usuario (id, nombre, fecha de creación…) y una lista de botones con las llamadas a las acciones que se pueden efectuar sobre ese usuario como pueden ser el ver la información en detalle, editar dicha información o deshabilitar el usuario.
  La lista de usuarios está limitada a 10 usuarios por página. Si es necesario moverse entre páginas en la parte inferior se encuentran los enlaces para acceder a ellas.

- **Ver usuario:** En esta vista se muestra la información del usuario empleando una disposición en tabla con dos columnas. En la columna de la derecha figura un título descriptivo y en la columna izquierda su valor. Se optó por esta disposición de los datos

pensando en posibles añadidos futuros. Una tabla con un ancho muy grande hubiese supuesto un problema para visualizar correctamente la información y este problema se hace mayor si se piensa en dispositivos móviles donde el tamaño de la pantalla es mucho más limitado. Con esta disposición, si en un futuro se añaden más campos de información, solo varía el alto de la tabla con lo cual le resulta mucho más intuitivo al usuario navegar por ella.
En el margen derecho de la vista están disponibles para el administrador una lista de acciones a realizar sobre ese usuario (editar la información y deshabilitar usuario).

- **Crear usuario:** Esta vista solo está disponible para usuarios con permisos suficientes (en principio solo el administrador puede verla) y está compuesta por un formulario formado por cinco campos: nombre completo, nombre de usuario, email y contraseña. Todos los campos son obligatorios y siendo los más importantes los de email y contraseña porque estos son los datos con los que accederá el usuario a la aplicación (el nombre de usuario solo se emplea para referirse a dicho usuario dentro de la aplicación).
  Una vez enviado el formulario se inicia todo el proceso de creación del usuario con la información aportada. En dicho proceso se comprobará que los campos no contienen errores y que el usuario que inició el proceso de creación de usuario cuenta con los suficientes permisos para llevar a cabo la acción. Como respuesta se redirige al usuario a la vista de lista de usuarios y se muestra un mensaje de éxito confirmando que se ha añadido un nuevo usuario. Si, por el contrario, ha ocurrido un error se pueden dar dos casos:

  - El usuario no posee permisos suficientes para añadir un nuevo usuario: En este caso se envia al usuario a la lista de usuarios y se le muestra un mensaje de error.
  - Los datos introducidos no son válidos: En este caso el usuario regresa al formulario con un mensaje indicando qué campos no están correctos.

- **Editar usuario:** En la vista de editar formulario se reutiliza la vista de crear usuario. La diferencia entre una y otra, a parte del título, es que, en el caso de la vista de editar usuario, el formulario ya está rellenado con la información del usuario.
  Al modificar la información del usuario se pueden dar las mismas respuestas que al crear un usuario: usuario sin permisos, datos inválidos o datos cambiados correctamente.

- **Lista de usuarios deshabilitados:** La vista de usuarios deshabilitados es la misma que la empleada en la vista de lista de usuarios. La única variación es la información mostrada que se corresponde con los usuarios que han sido borrados. Las acciones que se pueden hacer sobre estos usuarios también cambian limitándose a ver la información del usuario, borrar definitivamente el usuario o recuperar al usuario.

4.3.2 Permisos

La autenticación de usuarios solo controla el acceso a la aplicación, no es capaz de controlar que puede hacer el usuario una vez ha accedido. Para ejercer este tipo de control la aplicación cuenta con un sistema de permisos empleando el módulo ”laravel-permission” de Spatie. El primer paso es deducir todas las posibles acciones que podría realizar cualquier usuario. Así obtenemos la lista completa de capacidades. El siguiente paso es definir los roles de la aplicación. Al tratarse de una empresa de diseño web pequeña, los roles están asociados con los puestos de trabajo.

Los roles definidos no se espera que varíen con el tiempo. Por esta razón se tomó la decisión de añadirlos a la base de datos al instalar la aplicación. Para esto se creó una nueva ”Migración” de Laravel que añade a la base de datos la lista completa de roles y las capacidades que posee cada rol. Este archivo se ejecuta al instalar la aplicación por primera vez o cada vez que se regenera la base de datos (siempre que se le indique que se quiera poblar la base de datos).

Todas las vistas de administración están protegidas del usuario común y solamente el administrador puede acceder a ellas. Esta limitación, a diferencia de las demás, se realiza mediante el Middleware para bloquear el acceso a nivel de enrutamiento. El resto de limitaciones se aplican en los controladores para controlar cada acción independientemente. De esta forma es más fácil definir qué acciones solo puede realizar un administrador y cuáles de ellas pueden estar compartidas entre varios roles. Además la respuesta de la aplicación varía siendo menos amigable para el usuario tratar de ejecutar una acción de administración.

Operaciones básicas con permisos

Al igual que ocurre con los usuarios, los roles también cuentan con una serie de acciones que se pueden realizar:

- Listar Roles: Muestra la vista con la lista de todos los roles de la aplicación.
- Añadir nuevo rol: Añade un nuevo rol a la aplicación definiendo también que usuarios forman parte de ese rol.
- Modificar rol: Modifica la información de los roles. Es un acción importante porque permite añadir o revocar permisos de los usuarios y es aquí donde se seleccionan los usuarios que pertenecen al rol.
- Eliminar rol: Elimina toda la información del rol revocando al usuario los permisos relacionados con ese rol. Esta acción no se puede deshacer y no existe la opción deshabilitar

temporalmente.

4.4 Gestión de proyectos

La gestión de proyectos es la parte central de la aplicación y es la encargada de unificar a todas las demás partes que componen la aplicación.

Un proyecto es una planificación compuesta por una serie de actividades relacionadas entre sí y coordinadas con la finalidad de obtener un resultado y hacerlo de la forma más eficiente posible. Debe tener un nombre que lo identifique, un responsable que gestione el proyecto, unas tareas definidas para alcanzar el objetivo y unos recursos (los usuarios) que realizan las tareas.

Las tareas son actividades que deben ser completadas en un periodo de tiempo específico para finalizar el proyecto. Una tarea puede estar asignada a uno o varios usuarios pero en esta aplicación una tarea sólo se asigna a un usuario. Esta limitación viene de la forma de trabajar internamente en la empresa donde cada tarea solo es asignada a una persona y esta es luego revisada por otra persona para comprobar que la tarea se ha realizado correctamente.

4.4.1 Operaciones básicas con los proyectos

- **Listar proyectos:** Devuelve una lista paginada de los proyectos activos a los que el usuario tiene acceso.
- **Buscar proyecto:** Filtra la lista de proyectos para devolver solamente el proyecto buscado por nombre.
- Crear proyecto: Crea un nuevo proyecto dentro de la aplicación. Esta acción requiere que el usuario tenga permiso y solo los administradores y los gestores de proyectos poseen ese permiso. Al crearse un proyecto se debe indicar de qué tipo de proyecto se trata. Dependiendo del tipo indicado se generan distintos árboles de directorios en el sistema de archivos compartido. Esta funcionalidad se ha implementado empleando el sistema de archivos de Laravel.
  Al crear un proyecto se inicia un proceso en el que la información relativa al proyecto se guarda en la base de datos al mismo tiempo que se genera toda la estructura de datos en el directorio de proyectos del disco compartido en red.
- **Ver proyecto:** Devuelve la información básica del proyecto.
- **Editar proyecto:** Modifica los datos básicos del proyecto.

- **Eliminar proyecto:** Marca el proyecto como borrado. Esto quiere decir que no se borra la información del proyecto pero al mismo tiempo el proyecto no aparece como disponible y sólo es accesible a la información a través de la acciones específicas para ver los proyectos eliminados. Esto se consigue modificando el valor de **deleted_at** indicando la hora actual (valdría con cualquier anterior pero de esta forma se sabe cuando se guarda cuando un usuario ha sido eliminado). Para

- **Recuperar proyecto:** Esta acción permite recuperar un proyecto que previamente había sido borrado. Para lograr esto se edita el atributo **deleted_at** cambiando su valor a nulo.

- **Destruir proyecto:** Elimina el proyecto de la aplicación definitivamente impidiendo recuperar su información. Esta acción no se puede revertir y consiste en eliminar la información del proyecto de la base de datos.

Vistas de proyecto

- **Lista de proyectos:** Es la vista principal de la aplicación. Muestra la lista paginada de proyectos a los que el usuario tiene acceso. En la parte superior hay una barra de herramientas con unos botones para añadir un nuevo proyecto y ver la lista de proyectos borrados. También hay un buscador para facilitar la búsqueda de un proyecto concreto conociendo su nombre. La lista está formada por una tabla de 10 filas por página y en cada fila se muestra información básica del proyecto y una lista de acciones que se pueden realizar sobre ese proyecto (ver más información, editar el proyecto o borrarlo).

- **Lista de proyectos eliminados:** Esta vista es la misma que la empleada al listar los proyectos. La única diferencia es que se muestran los proyectos eliminados. También cuenta con un buscador para filtrar la lista de proyectos y encontrar un proyecto buscando por nombre. En la vista de listar proyectos eliminados cada proyecto cuenta con dos acciones: restaurar y eliminar definitivamente.

- **Añadir proyecto:** La vista de añadir proyecto está formada por un formulario con los campos de información necesarios para crear un proyecto en la aplicación. Una vez cubiertos todos los campos, si todo es correcto, se añade un nuevo proyecto a la aplicación y se muestra un mensaje de éxito. Si el usuario no tiene permisos para añadir un nuevo proyecto, se le muestra un mensaje de error y se le redirige a la lista de proyectos. En caso de que el usuario tenga permisos pero los datos al crear un proyecto no son correctos o no están completos, se le devuelve al formulario y se le muestra un mensaje de error indicando cuál ha sido el error cometido.

- **Editar proyecto:** La vista de editar proyecto es igual a la vista de añadir proyecto. Simplemente los datos del proyecto ya están cargados en el formulario.

- **Información de proyecto:** La vista de información de proyecto está dividida en dos partes bien diferenciadas. Por un lado tenemos la información del proyecto y por otro las tareas que componen ese proyecto. Para mostrar esta dualidad se optó por una estructura de "pestañas". La decisión de separar estas dos partes es debido a la cantidad de información a mostrar y la diferencia entre el tipo de información.

  - La **pestaña de información** ofrece información genérica del proyecto como por ejemplo el nombre, cuando se inicio el proyecto, quién es el encargado de ese proyecto, etc.
    Esta información, al igual que en el caso de los usuarios, se presenta al usuario mediante una tabla.
    Al margen derecho de la tabla de información se halla una lista con las acciones básicas que se pueden realizar sobre el proyecto (borrar proyecto y modificar la información del proyecto).
    En la parte inferior se muestra la lista de webs que pertenecen al proyecto así como sus credenciales. Desde aquí se pueden añadir nuevas webs a la aplicación para vincularlas al proyecto.

  - La **pestaña de tareas** muestra un diagrama de Gantt con las tareas del proyecto. En la parte superior se sitúa la barra de herramientas que permiten añadir o modificar la información de una tarea.
    El diagrama de Gantt se generó empleando la librería de Javascript jsGanttImproved. Esta librería muestra las tareas del proyecto dividiendo la vista en dos partes. En la derecha está la lista de tareas con información la información más importante (nombre de la tarea, estado, duración, etc) y a la izquierda está el diagrama de Gantt. La librería ya se encarga de añadir las dependencias y dibujar el diagrama. Solamente es necesario hacerle llegar la información en un formato que pueda entender. El método que se emplea en este caso es un Array de Javascript que se escribe en la vista con el JSON ya transformado por Laravel.
    Este JSON es leído por una función de Javascript que lo transforma en un objeto que pueda ser usado por la librería de jsGanttImproved.
    El diagrama y la tabla de jsGanttImproved han tenido que ser modificadas para adecuarlas a las necesidades del proyecto. Para poder interaccionar de forma más amigable con las tareas se añadieron una serie de botones a cada una de las tareas de la tabla para poder trabajar con dichas tareas. Así por ejemplo se pueden eliminar tareas, editar su información o ir a la vista de tarea con la información de

la tarea más detallada.
Las acciones de añadir o modificar una tarea abren una ventana modal con un formulario (similar en ambos casos) para crear o editar la tarea. Cualquier cambio realizado se guarda en la base de datos mediante una llamada Ajax al mismo tiempo que se modifica la lista de tareas para que figuren los nuevos cambios. Después de cada cambio en la lista de tareas se realiza una llamada a la API de jsGanttImproved para que vuelva a dibujar el diagrama actualizado.

- **Vista de tarea:** En esta vista se puede ver la información detallada de una tarea y los comentarios realizados sobre dicha tarea.
  La vista está dividida en dos partes. La información se muestra a la izquierda, en forma de tabla de dos columnas y siguiendo el mismo estilo empleado hasta ahora para la visualización de información. En el margen derecho de la tabla está la lista de acciones que se pueden realizar sobre la tarea. A la derecha se sitúa una ventana donde se visualizan los comentarios que realiza cada usuario sobre la tarea. Estos comentarios se muestran imitando la visualización de un chat en donde los comentarios del propio usuario se sitúan a la derecha mientras que las aportaciones realizadas por otros usuarios se sitúan a la izquierda (imitando el funcionamiento de Whatsapp).
  La funcionalidad del chat se implementó empleando VUE y Axios combinándolo con el servicio Pusher y el manejo de eventos de Laravel. Para que esto funcionase se crearon nuevas rutas para manejar la entrada de eventos, se modificó el Middleware correspondiente para autenticar las notificaciones recibidas sobre un evento y, a mayores, fue necesario crear un evento de Laravel para poder emitir notificaciones.
  El resultado de combinar estas tecnologías es una widget de chat sincronizado permitiendo una comunicación directa entre los distintos usuarios que hagan uso del chat. El chat está configurado empleando un canal privado para cada tarea. Esto implica que solo el chat de cada tarea recibirá los mensajes que le corresponden a esa tarea (ver figura A.8 (página 44)).

4.5 Gestión web

La gestión de sitios web que se realiza en la aplicación es otro de los pilares básicos de la aplicación. Como ya se ha explicado anteriormente los proyectos pueden no tener ninguna web asignada o tener varias webs. Sin embargo no todas las webs, de las cuales se guarda información en la aplicación, están relacionadas con un proyecto de la aplicación. Para resolver este problema la gestión web tiene su propia vista con la lista de todas las webs cuya información se gestiona desde la aplicación.

4.5.1 Operaciones básicas con web

- **Listar webs:** Devuelve una lista de webs paginada.
- **Añadir web:** Añade una nueva web a la lista de webs.
- **Eliminar web:** Elimina de la aplicación la información de una web.
- **Editar web:** Modifica la información de una web.
- **Listar credenciales:** Devuelve una lista de credenciales de una web.
- **Añadir credenciales:** Añade una nueva credencial para la web.
- **Eliminar credenciales:** Elimina la credencial de la lista de credenciales de la web.

4.5.2 Vistas de web

- **Listar webs:** Esta es la vista principal de la gestión web. En ella se muestra una lista de webs paginada y al pulsar en el nombre de cada una aparece la dirección URL de la web, los botones para editar o borrar la información y la lista de credenciales de la web. En la parte superior de la vista se encuentra el buscador y los botones para añadir una nueva web o para crear un nuevo sitio de Wordpress.
- **Añadir web:** Esta vista es un sencillo formulario para añadir la información básica de la web.
- **Editar información web:** Esta vista es similar a la vista de añadir web pero desde aquí también se pueden añadir o borrar las credenciales (ver figura A.2 (página 41)).

4.5.3 Repositorio de recursos Wordpress

El repositorio de recursos es la parte de la aplicación que guarda y gestiona los recursos de Wordpress (plugins o temas) que se suben a la aplicación para su posterior uso. El objetivo es mantener un banco de recursos fácilmente gestionable desde la aplicación.

En la figura 4.1 (página 30 se vé como se emplea dos veces el patrón MVC sobre un mismo modelo para realizar dos interpretaciones distintas, plugins y temas, sobre un mismo recurso.

Operaciones básicas con recursos

A continuación se muestra una lista con todas las operaciones que se pueden realizar sobre plugins y temas:

- **Añadir tema:** Añade un nuevo tema al repositorio de temas y añade la información correspondiente al tema a la base de datos de la aplicación.

- **Editar información de tema:** Modifica la información de un tema almacenada en la aplicación.

- **Listar temas:** Devuelve la lista paginada con los temas guardados en el repositorio.

- **Eliminar temas:** Elimina un tema del repositorio y borra sus datos de la base de datos de la aplicación.

- **Credenciales del Wordpress:** Estas son las credenciales del administrador de Wordpress Añade un plugin al repositorio e ingresa la información correspondiente en la aplicación.

- textbfListar plugins: Devuelve una lista paginada con todos los plugins que hay en el repositorio.

- **Editar plugins:** Modifica la información almacenada en la aplicación correspondiente a un plugin.

- **Eliminar plugins:** Elimina el plugin del repositorio y borra la información de la base de datos de la aplicación.

Vistas de recursos

Las vistas de recursos son muy similares entre los dos tipos de recursos (plugins y temas) ya que los atributos en ambos casos son los mismos.

- **Listar recurso:** Lista todos los recursos del tipo indicado (plugins o temas) y los muestra como una tabla de tres columnas. En la parte superior figura una barra de herramientas con un botón para añadir nuevos recursos.
  De cada recurso se muestra su nombre, la versión actual y dos acciones (editar y borrar).

- **Añadir recurso:** es una vista en la que figuran dos campos de texto para el nombre y la versión y un selector de archivos para subir el archivo del recurso.

- **Editar recurso:** Esta vista es la misma que para añadir recurso, la diferencia es que los campos ya vienen precargados con la información del recurso.

4.5.4 Crear un sitio Wordpress

La acción de crear un sitio Wordpress es de los procesos más destacados de la aplicación y por eso se ha decidido crear un capítulo aparte dedicado a ello.

El proceso de instalación está dividido en tres etapas:

1. Generar las contraseñas y parámetros básicos (ver figura A.3 (página 41)).
2. Seleccionar los plugins o temas a instalar (ver figura A.4 (página 42)).
3. Iniciar el proceso de instalación (ver figura A.5 (página 30)).

Generar contraseñas y parámetros básicos

Es la primera parte del proceso, en esta etapa se muestra una vista con un formulario con multitud de campos. Los campos están divididos en las siguientes secciones:

1. **Información general:** En esta parte se selecciona un nombre para el sitio web. La aplicación está automatizada mediante una función Javascript para rellenar, a partir de ese nombre, todos los campos del formulario. La idea es automatizar lo máximo posible esta tarea y que el usuario no tenga que ir campo por campo cubriendo el formulario. De esta forma el usuario ya tiene cubierto el formulario con unas credenciales seguras, unos nombres de usuario descriptivos y acordes a su futuro uso y unos valores por defecto válidos.
2. **Credenciales del alojamiento:** Estas credenciales se corresponden con la cuenta del panel de control del servidor web (VestaCP, Cpanel, Plesk, etc) y con las credenciales de la cuenta de FTP.
3. **Credenciales del Wordpress:** Estas son las credenciales del administrador de Wordpress.
4. **Otros campos de interés:** Por defecto la versión de Wordpress que se toma es siempre la última disponible. Si por alguna razón se quisiese emplear una versión antigua, el sistema ya tiene disponible una lista a seleccionar con las últimas versiones existentes en orden descendente. Esta funcionalidad se logra realizando una llamada a la propia página de Wordpress consultando las versiones disponibles para descarga.
   Otros parámetros importantes es indicar el idioma de la futura web (por defecto se elige el inglés pero existen otros idiomas disponibles), indicar si la web estará en un subdirectorio, indicar se quiere indexar ya la web, el prefijo de las tablas de la base de datos, etc.

Una vez cubierto, se envía el formulario al servidor. El servidor comprobará que está todo correcto. Si los datos son correctos y no se ha elegido un nombre que ya existe entonces se pasa a la siguiente etapa.

Seleccionar plugins y temas

Esta vista está compuesta por otro formulario, solo se pueden realizar dos acciones y las dos son opcionales.

En primer lugar seleccionar un tema de la lista de temas que hay en el repositorio. Si no hubiese ningún tema en el repositorio este campo estaría oculto.

En segundo lugar se pueden elegir los plugins. Al igual que con los temas, los plugins se seleccionan de una lista que se obtiene de la lista de plugins del repositorio. E igualmente si esta lista está vacía el campo estaría oculto. Para las listas de selección múltiple se añadió una librería llamada **Bootstrap-multiselect**. Esta librería crea selectores múltiples con un aspecto atractivo y mucho más intuitivo para el usuario.

Una vez seleccionados los plugins y el tema deseados se envía el formulario para pasar al siguiente y último paso.

Iniciar proceso de instalación

La vista del último paso está compuesta por varios paneles donde se muestran las credenciales que se habían seleccionado en el primer paso y un panel donde irá mostrándose el progreso a modo de log de información.

El usuario no tiene que hacer nada hasta que el proceso finalice ya que el proceso se inicia automáticamente a los dos segundos de llegar a esta vista. Toda la instalación está controlada por una cadena de funciones de Javascript en las que se realizan peticiones Ajax a la aplicación y esta devuelve un mensaje indicando el éxito o el fracaso del proceso. En caso de éxito se invoca el siguiente paso hasta acabar el proceso. Cada una de estas funciones encadenadas supone un paso más realizado por la aplicación y va mostrando su progreso al usuario de tal forma que si algo falla el usuario sabe en que punto está fallando el proceso.

Lo primero que se hace es obtener toda la información recopilada en las etapas anteriores y con esta información se crea un array de Javascript. Este array es pasado a la función de Javascript para iniciar el proceso. Esta función solicitará a la aplicación que genere un nuevo sitio en el servidor remoto. Para estas acciones la aplicación cuenta con una API REST que se

encarga de atender las llamadas referentes al proceso de creación del sitio web y el proceso de instalación de Wordpress.

La aplicación no puede generar un sitio web por si misma. Depende del panel de control para hacerlo. Cada panel de control dispone de una API para realizar este tipo de tareas. El problema es que no están estandarizadas, cada panel usa su propia implementación con sus propios requisitos y ofreciendo sus propios mensajes de respuesta. Además para realizar peticiones de este tipo a la API de un panel de control necesitas las credenciales de un usuario administrador de dicho panel. Por estos motivos la llamada a la API del panel de control no las realizan las funciones de Javascript, se da la orden a la aplicación y es esta la que realiza dichas llamadas.

Para que la aplicación pueda comunicarse con los paneles de control se creó una librería específica para este trabajo. Esta librería implementa el patrón factoría para seleccionar la implementación en función del tipo de panel (ver figura 4.2 (página 30)). Actualmente solo se ha creado el código para el panel de VestaCP pero se ha dejado preparado para añadir el módulo correspondiente a Cpanel y en un futuro sería fácil añadir el correspondiente a Plesk.

Como se ha mencionado anteriormente, las APIs de los paneles de control requieren de unas credenciales y unos parámetros base de configuración. Esto en Laravel se solventa empleando las variables de entorno y el archivo ".ENV".

Una vez que se ha generado el sitio web (se ha creado la cuenta en el panel de control), el siguiente paso es añadir el dominio. Como se trata de un entorno de desarrollo el dominio que se elige es para uso interno. Lo que configurar el servidor como servidor DNS local indicando que la dirección del dominio (y los subdominios de este) están alojados en el servidor de desarrollo. De esta forma toda proyecto web se crea como un subdominio y el servidor al instante da a conocer esa información al resto de la red.

Con el dominio creado en el servidor es necesario añadir una nueva base de datos que alojará el Wordpress. Para esto volvemos a hacer otra llamada al servidor pasando los datos que tendrá la futura base de datos.

En cuanto el servidor responda afirmativamente, desde el javascript se iniciará la siguiente fase que es instalar el Wordpress.

Esta fase puede llevar varios minutos y depende mucho de la conexión del servidor.

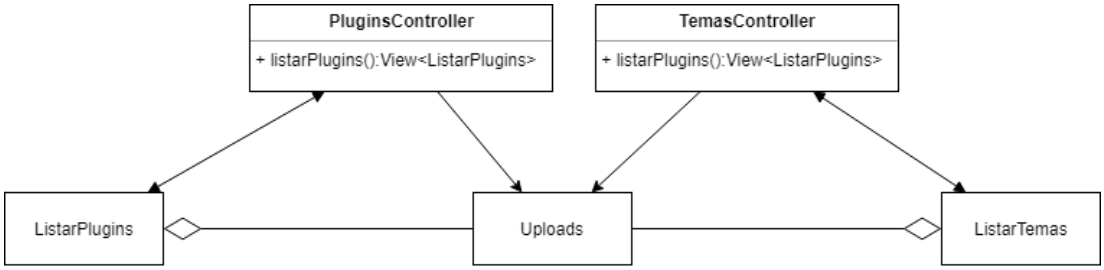

Figura 4.1: Modelo UML de plugins y temas, ambos comparten mismo modelo pero cada uno posee sus propias vistas y controladores

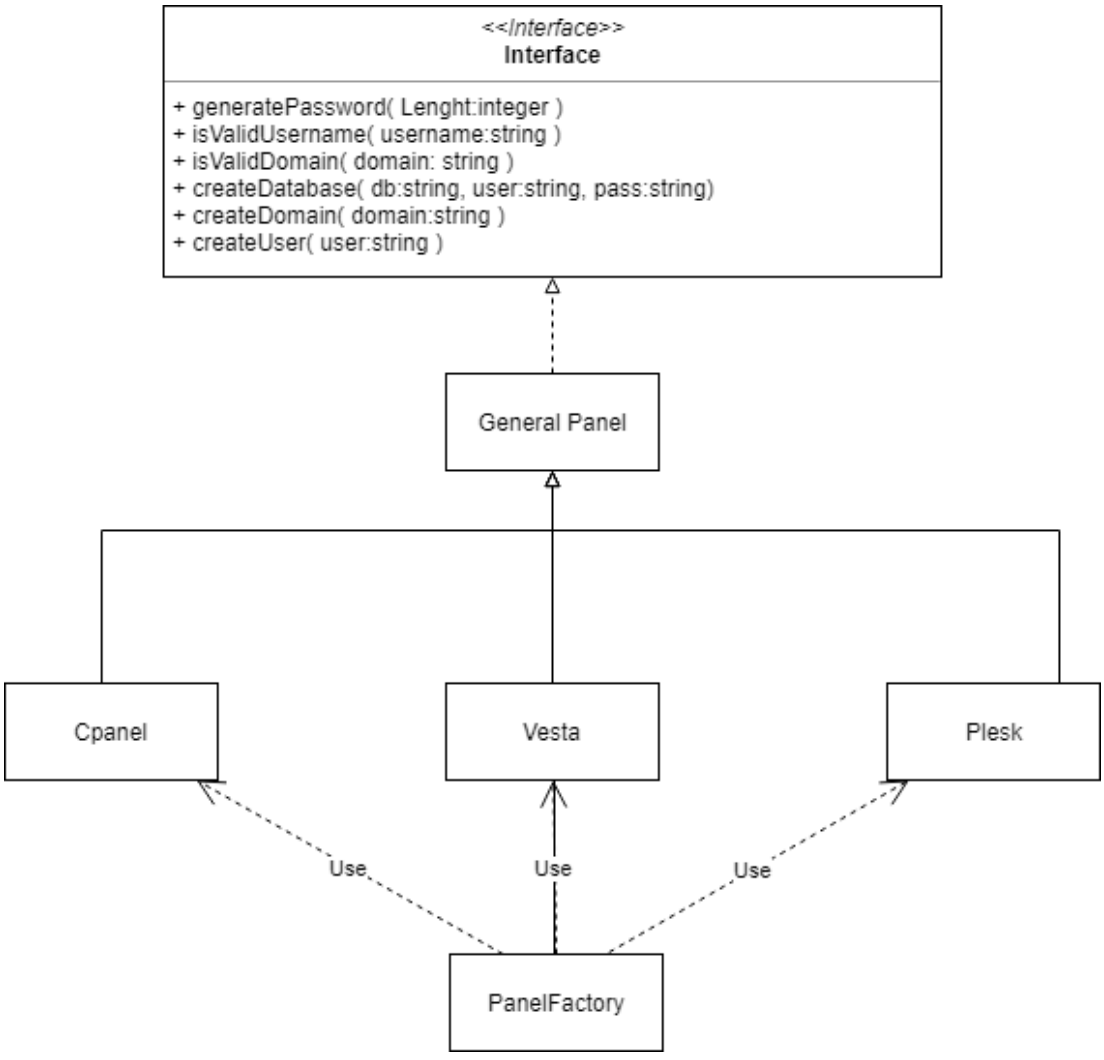

Figura 4.2: Diagrama UML de la implementación del patrón factoría en la librería de instalación de Wordpress

El proceso de instalación sería el siguiente:

1. En la aplicación se reciben todos los datos recopilados hasta ahora y con ellos se completan unas plantillas base para generar un script de PHP para la instalación y un archivo **wp-config.php** con la configuración básica de Wordpress. Este script se envía al servidor de desarrollo mediante una conexión FTP empleando el usuario y la contraseña que se generaron previamente. El script de PHP actuará como Payload en el servidor remoto ejecutando las órdenes de la aplicación.
2. Una vez enviado el script se lanza una llamada a la ruta donde, si todo a ido bien, debería estar el script para que este se inicie.
3. Si se ha indicado un subdirectorio se generará en este momento.
4. El script se encargará de descargar la versión de Wordpress indicada.
5. El Wordpress se descarga comprimido y por tanto el siguiente paso es descomprimirlo.
6. Luego se mueve todo el contenido del directorio Wordpress (creado al descomprimir el archivo) a la raíz de la web(o al subdirectorio indicado)
7. El último paso es el renombrado de los archivos ”index.php” y la destrucción del propio script de instalación.

En la figura 4.3 (página 32) se puede ver el diagrama de secuencia con el proceso de instalación de Wordpress.

Una vez finalizado todo el proceso de instalación de Wordpress, si se ha indicado la instalación de temas o plugins se vuelve a enviar otro script para los temas y otro para los plugins para iniciar su proceso de instalación. El proceso seguido por estos plugins es el mismo que el seguido por el script de instalación de Wordpress pero en este caso se ejecuta en el directorio ”wp-content”.

Si la instalación de plugins y temas es correcta se le indicará al usuario que el proceso ha finalizado y se activará el botón para que el usuario confirme que se finaliza el proceso. Al hacerlo se activa definitivamente el Wordpress y se abre en una nueva pestaña del navegador. Al mismo tiempo las credenciales generadas durante todo el proceso se guardan en la base de datos y se da por finalizado el proceso.

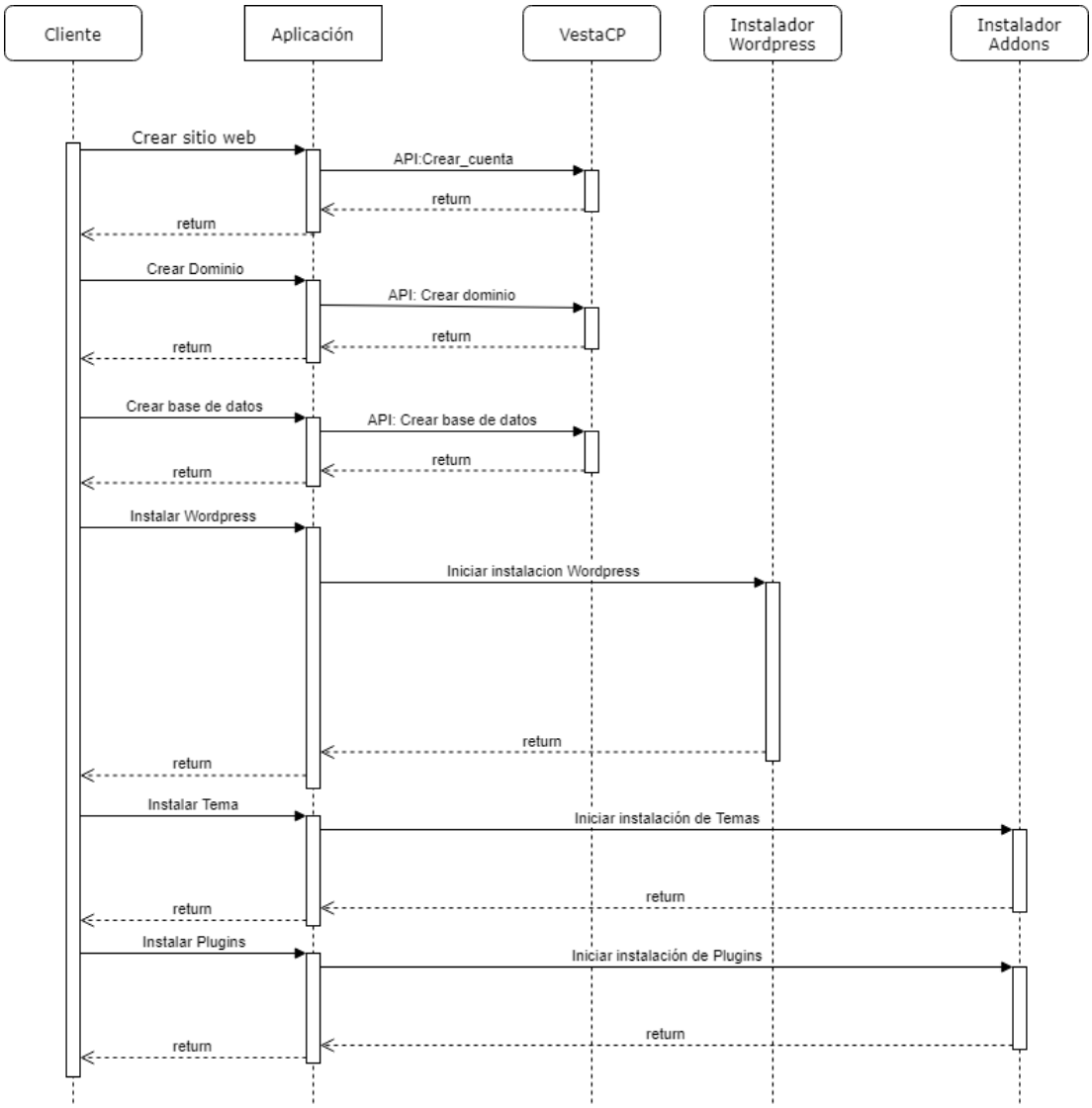

Figura 4.3: Diagrama de secuencia que muestra los pasos de ejecución de la instalación de Wordpress

Capítulo 5

Evaluación

En general la aplicación responde correctamente en un entorno de red local y con una buena velocidad de conexión. Si la velocidad de transmisión de la red no es muy buena el proceso de instalación de Wordpress se queda bloqueado sin advertir al usuario. Esto es debido a que descargar e instalar el Wordpress le lleva más tiempo del esperado y el servidor cierra la conexión. Hasta ahora no ha sido posible hallar una solución a este problema.

Como se ha expuesto en la tabla de planificación, aún no se han realizado los test unitarios por lo tanto solo se han realizado pruebas empíricas de su funcionamiento.

Actualmente la aplicación se encuentra desplegada en la empresa en un servidor de pruebas en fase de testeo.

Capítulo 6

Conclusiones

Como se ha expuesto a lo largo de este documento, la aplicación desarrollada aporta una solución a los problemas de gestión de proyectos y automatización de tareas comunes en una empresa pequeña de desarrollo web y diseño gráfico. Con esta herramienta una empresa es capaz de crear nuevos proyectos, gestionarlos, inicializar sitios web en el servidor de desarrollo, añadir los plugins y temas necesarios para comenzar trabajar en su desarrollo y gestionar las credenciales de las webs creadas.

La aplicación cumple con todos los objetivos básicos exceptuando uno que es la integración con el gestor de tareas Producteev. La idea original consistía en hacer uso de la API de Producteev para sincronizar los dos gestores de tareas pero este objetivo no se pudo alcanzar porque en el transcurso del desarrollo de la aplicación la empresa que posee los derechos de Producteev decidió cerrar la aplicación.

En referencia a los objetivos logrados hay multitud de cambios que se pueden realizar para que la aplicación mejore y que no se han llevado a cabo por falta de tiempo:

- **Sustituir todas las vistas de creación y edición de recursos por modales:** Esto es algo que se comenzó a hacer con la gestión de tareas y que convendría trasladar al resto de la aplicación. Los modales suponen una interacción más amigable con el usuario al eliminar la necesidad de cambiar de página en el momento de crear o editar información de los recursos.

- **Emplear más Vue y Axios:** La combinación de estas dos tecnologías es muy buena para un desarrollo de este tipo en el que la interacción se realiza contra un servicio REST. En la aplicación solamente se emplea a modo de prueba en el widget de comentarios de tareas pero se podría emplear para sustituir las vistas de listas paginadas o las vistas de información de recursos.

- **Usar Vue, Axios y Pusher en el proceso de instalación de Wordpress:** Actualmente el proceso de instalación de Wordpress junto con la instalación de plugins y temas se realiza mediante una serie encadenada de llamadas Ajax al servidor. Esto se puede cambiar por un componente Vue que interacciona con el servidor de Laravel mediante Axios mientras se queda escuchando a que el servidor lance un evento indicando el paso en el que se encuentra durante el proceso de instalación.
- **Convertir el payload en un servicio REST que realice todo el proceso de instalación:** Actualmente los payload que realizan la instalación en el servidor remoto es un simple script de php con unas tareas muy básicas. Esto puede convertirse en un Payload con un servicio REST que realice todas las tareas en función de las órdenes recibidas desde la aplicación.
- **Emplear la funcionalidad del sistema de archivos de Laravel para enviar los archivos al servidor remoto:** Las funciones que envían los archivos al servidor remoto son funciones nativas de PHP. Esto se puede modificar para emplear el sistema de gestión de archivos de Laravel y hacer mucho más limpio el código empleado.

Tal como se ha mencionado, la aplicación desarrollada es una primera iteración funcional dentro de un marco de desarrollo incremental. Existen muchos objetivos y posibilidades de evolucionar el código de la aplicación que, si bien se han pensado, no se han planteado para este proyecto:

- Crear API REST con autenticación Oauth2.
- Crear clientes para navegador que se conecten a la API REST e ingresen las credenciales web al acceder a una de las webs cuya credencial esté alojada en la aplicación.
- Añadir los módulos para otras implementaciones de paneles de control como Cpanel, Plesk, etc.
- Añadir funcionalidades para gestión de archivos
- Mejorar la gestión de tareas añadiendo otras funcionalidades como dependencias, cálculo de camino crítico, etc.
- Añadir sincronización con otros gestores de proyectos como Asana o Brietix24.

Apéndices

Apéndice A

Material adicional

A.1 Modelo Entidad-Relación

En la figura A.1 (página 40) se puede ver como está estructurada la base de datos y con los atributos de los modelos y las relaciones que se establecen entre ellos. La tabla ”Uploads” no guarda relación con ninguna otra tabla y solo se limita a almacenar la información de los archivos subidos.

A.2 Imagenes de Vistas más destacadas

A continuación se muestra una serie de vistas (sólo las más relevantes) tomadas de la aplicación

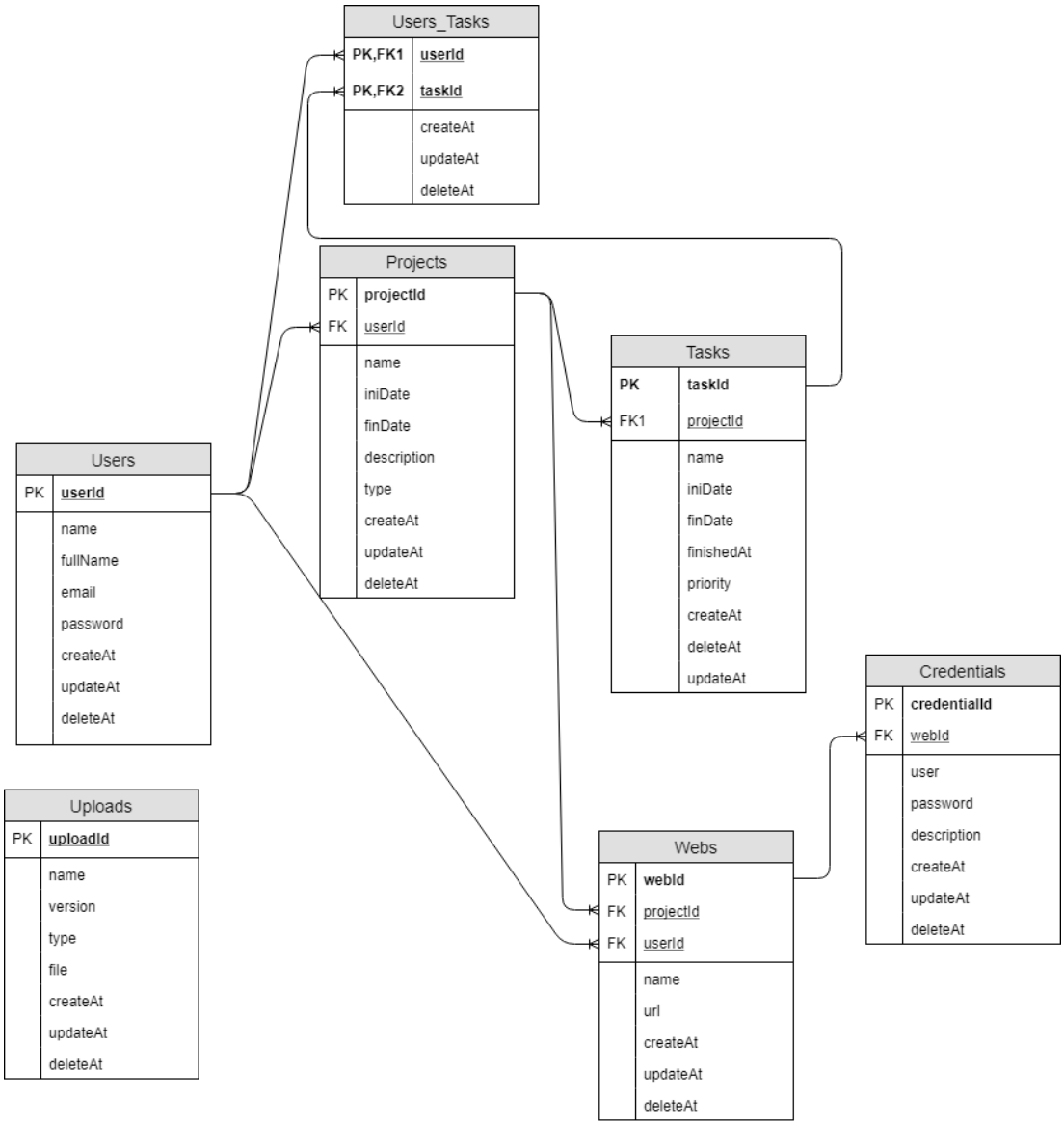

Figura A.1: Modelo Entidad-Relación que muestra la estructura de la base de datos de la aplicación

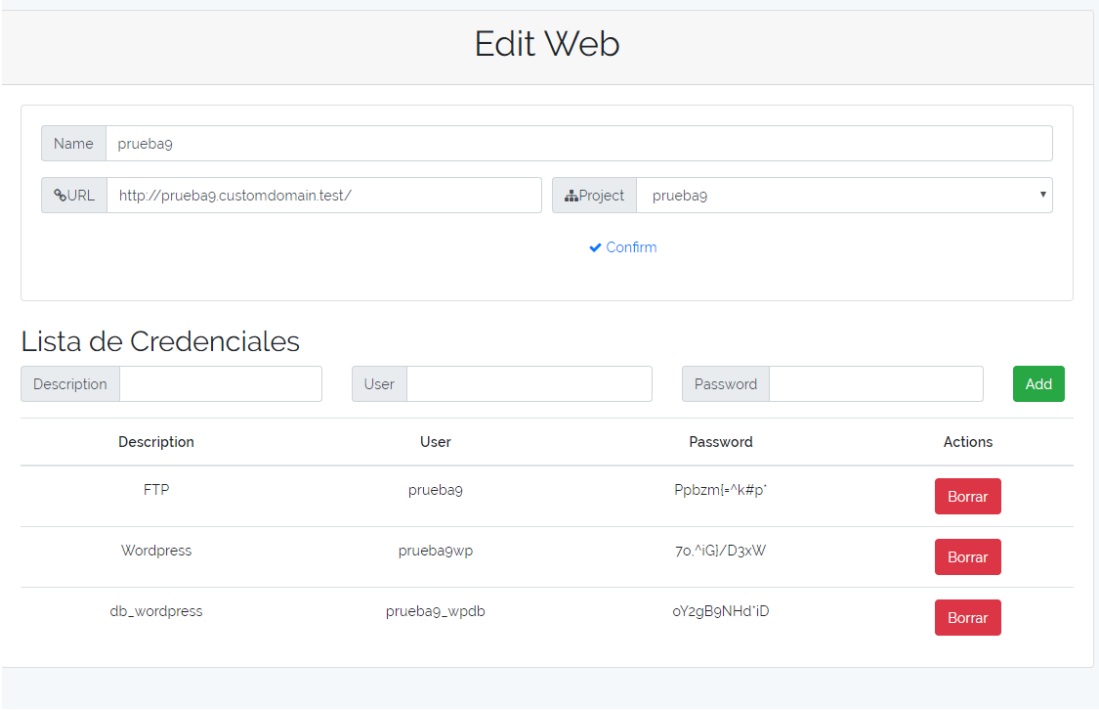

Figura A.2: Vista de edición de la información de la web

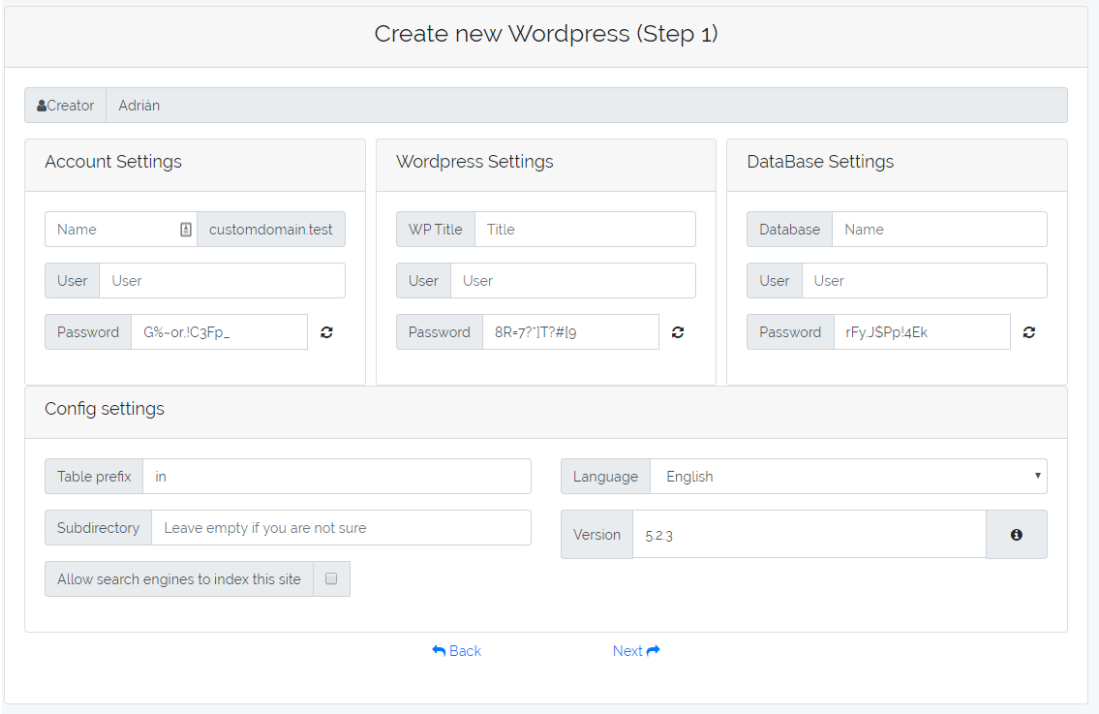

Figura A.3: Vista del primer paso del proceso de instalación

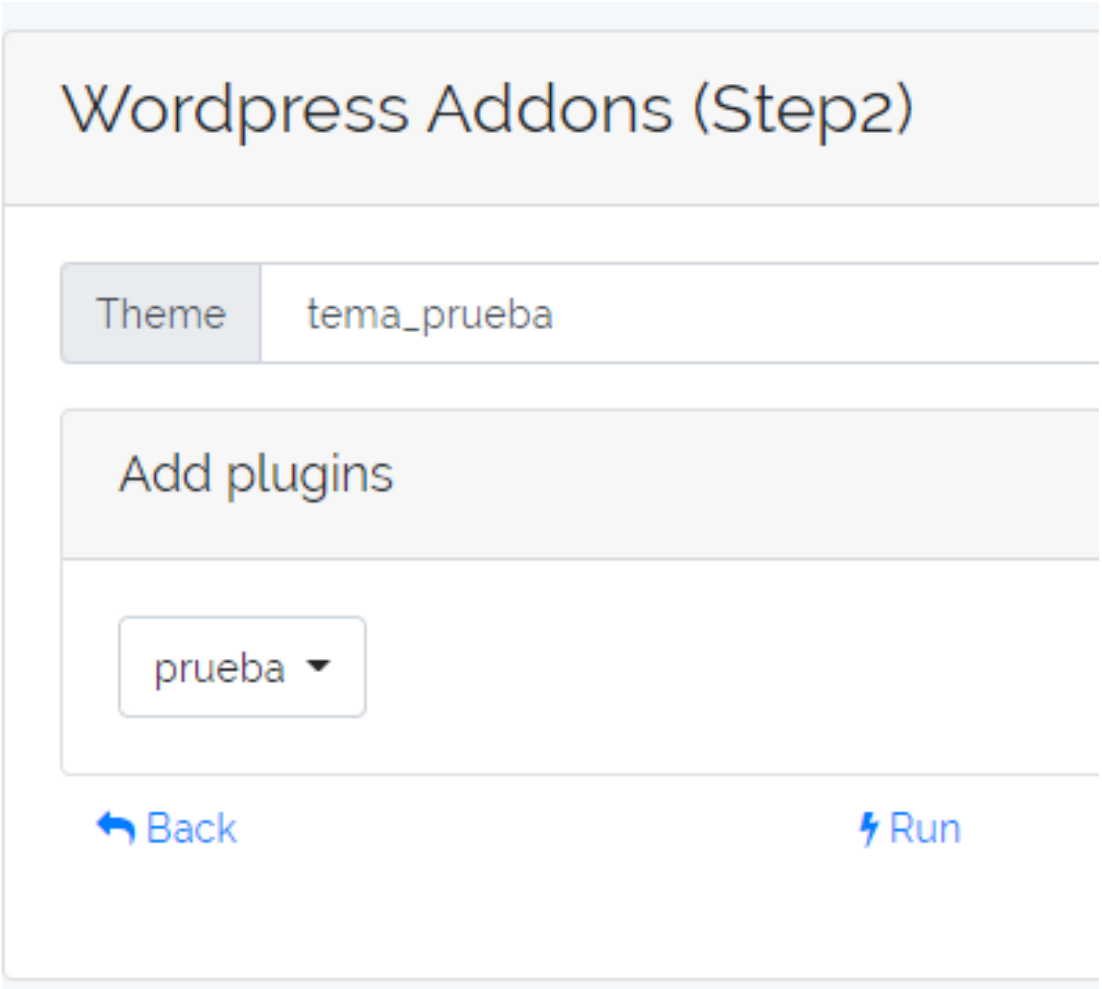

Figura A.4: Vista de la selección de plugins y temas durante el proceso de instalación

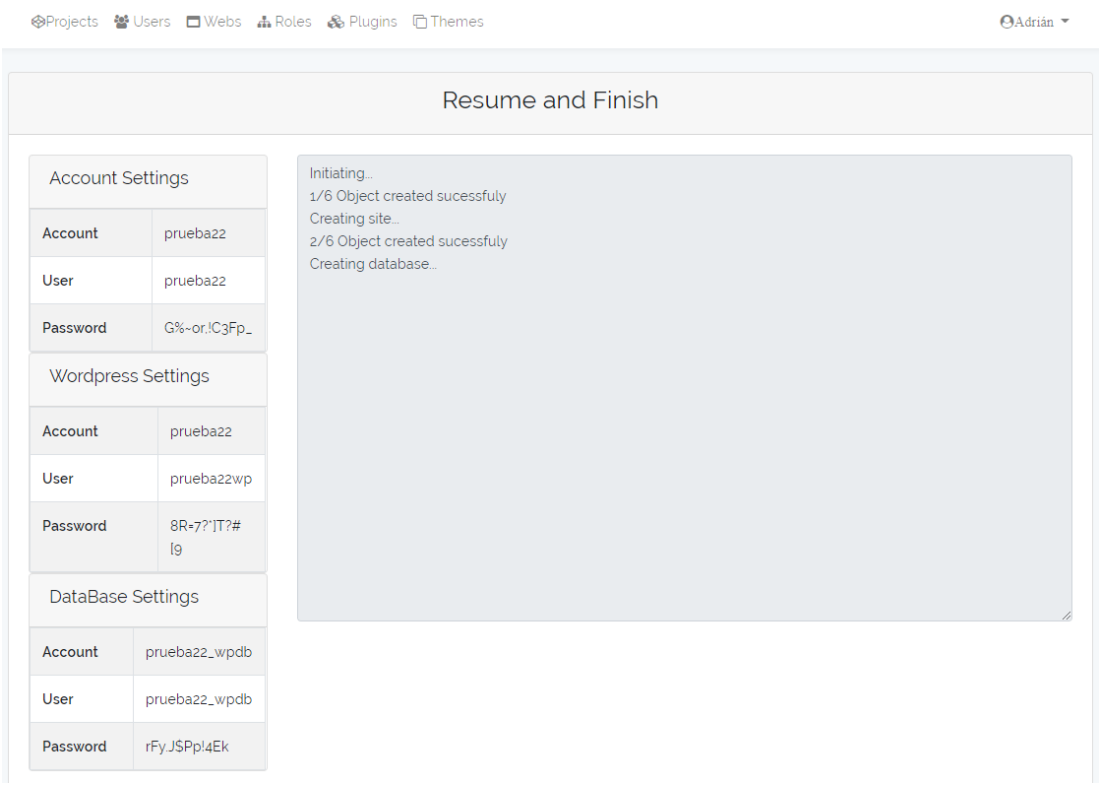

Figura A.5: Vista del ultimo paso del proceso de instalación de Wordpress

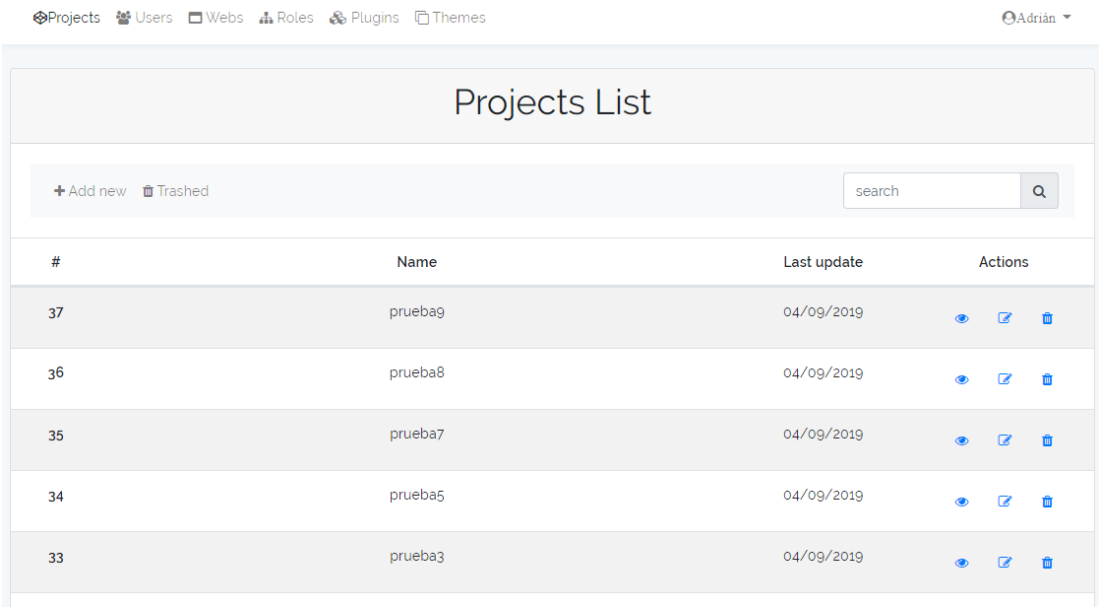

Figura A.6: Vista con la lista de proyectos

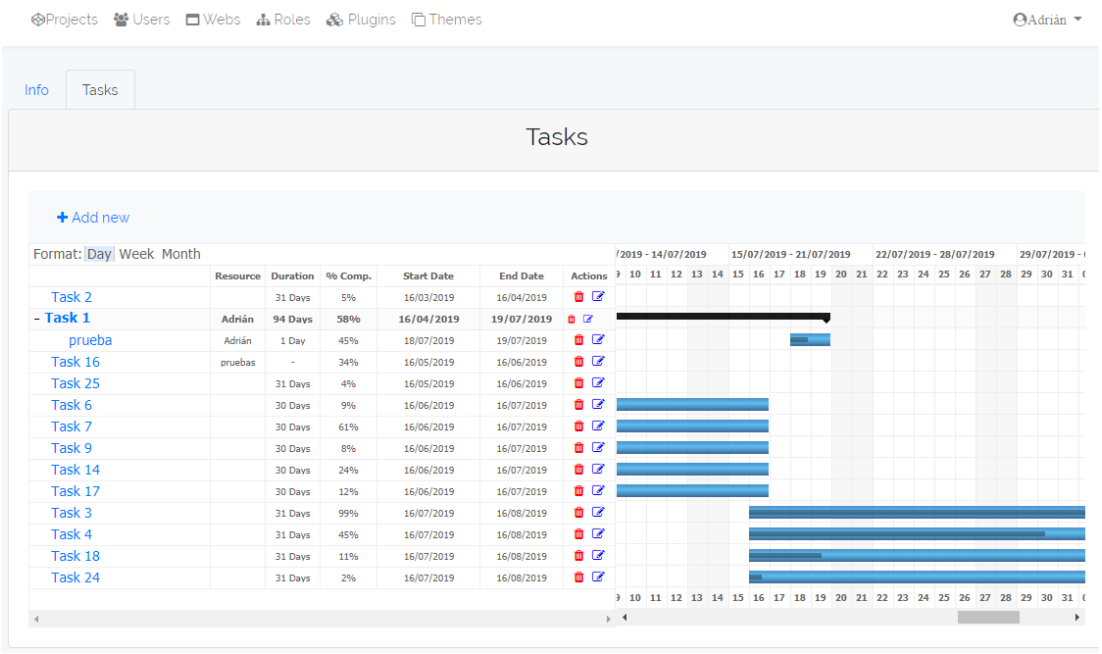

Figura A.7: Vista del proyecto mostrando la lista de tareas y el diagrama de Gantt

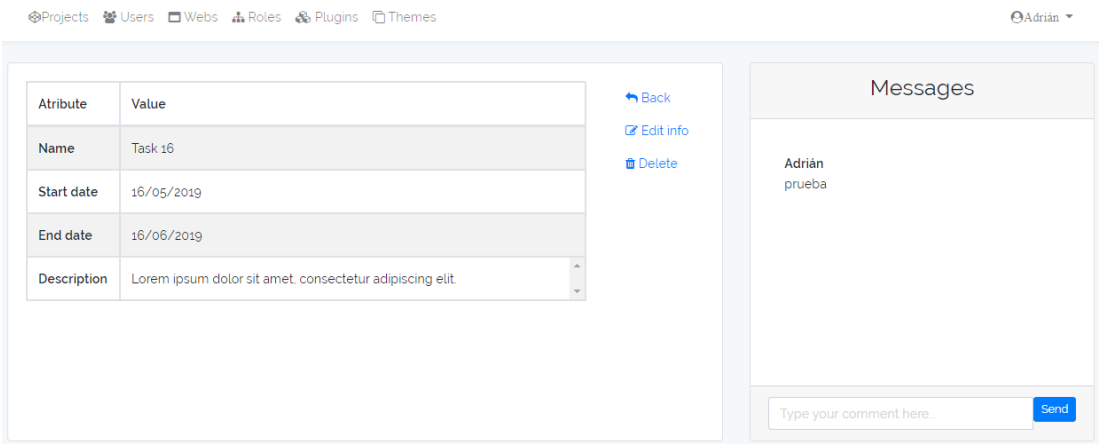

Figura A.8: Muestra la información en detalle de la taréa y el widget de chat para compartir anotaciónes.

Lista de acrónimos

ERLANG/OTP *Erlang Open Telecom Platform.*

smb *Server Message Block*

CRUD *Create Read Update Delete*

REST *Representational State Transfer*

JSON *JavaScript Object Notation*

Ajax *Asynchronous JavaScript And XML*

API *Application Programming Interface*

Glosario

widget Pequeño elemento o herramienta con funcionalidad limitada que tiene como objetivo dar fácil acceso a la funcionalidad aportando también información visual.

plugin Aplicación modular que depende de de un sistema y que aporta una funcionalidad concreta al sistema en el que se integra.

Bibliografía

[1] Marco, "¿qué es asana y por qué utilizarla en tu organización?" 2012. [En línea]. Disponible en: https://mvkoen.com/que-es-asana/

[2] V. Bondaruk, "¿qué es bitrix24?" 2018. [En línea]. Disponible en: https://helpdesk.bitrix24.es/open/6102783/

[3] "Primeros pasos." [En línea]. Disponible en: https://get.slack.help/hc/es/articles/115004071768--Qué-es-Slack-

[4] "Lastpass." [En línea]. Disponible en: https://es.wikipedia.org/wiki/LastPass/

[5] T. Blokehead and R. Durán, *Scrum - ¡Guía definitiva de prácticas ágiles esenciales de Scrum!* Babelcube, 2016.

[6] L. F. de Armas, "Vestacp: un panel de control práctico y potente," 2012. [En línea]. Disponible en: https://soluciones4hosting.com/vestacp-un-panel-de-control-practico-y-potente/

[7] A. L. A. Alcalde, "Pusher, sevicio cloud para gestión de conexiones y mensajes mediante websockets," 2013. [En línea]. Disponible en: https://unpocodejava.com/2013/03/31/pusher-sevicio-cloud-para-gestion-de-conexiones-y-mensajes-mediante-websockets/

[8] J. M. B. García, "Analizamos las características de la librería axios, un ligero cliente http para javascript," 2018. [En línea]. Disponible en: https://www.arsys.es/blog/programacion/axios/

[9] M. A. Alvarez, "Librería axios: cliente http para javascript," 2019. [En línea]. Disponible en: https://desarrolloweb.com/articulos/axios-ajax-cliente-http-javascript.html

[10] D. Palacios, "Solicitudes http con axios," 2018. [En línea]. Disponible en: https://styde.net/solicitudes-http-con-axios/

[11] "Qué es vue.js." [En línea]. Disponible en: https://codigofacilito.com/articulos/que-es-vue

[12] Tighten, "What is ziggy?" [En línea]. Disponible en: https://tightenco.github.io/ziggy/0.6/?ref=madewithlaravel.com

[13] R. Appelman, "Smb." [En línea]. Disponible en: https://github.com/icewind1991/SMB

[14] D. Stutz, "bootstrap-multiselect." [En línea]. Disponible en: https://github.com/davidstutz/bootstrap-multiselect

[15] E. Rodrigues, "jsganttimproved." [En línea]. Disponible en: https://jsganttimproved.github.io/jsgantt-improved/

[16] C. Pecoraro, *Mastering Laravel*, 1st ed. Packt Publishing, 2015.

[17] O. Aburto and R. Cuevas, "Laravel 5," 2016. [En línea]. Disponible en: https://richos.gitbooks.io/laravel-5/