



INSTITUTO DE EDUCACIÓN SECUNDARIA SERPIS

# Gestor de proyectos para desarrollo de aplicaciones informáticas con Scrum

Proyecto de Desarrollo de Aplicaciones Web

**Ciclo Formativo Desarrollo de Aplicaciones Web**  
**Departamento de informática**

**Autor:** Sanz López, Enrique

**Tutor:** Dolz Ferrer, Sergio

**Curso:** 2022/2023



## Portada

## Resumen

En este trabajo se busca crear una aplicación web que sirva para centralizar las tareas asociadas a la gestión de proyectos mediante la metodología ágil Scrum. En la actualidad, existen muchas herramientas online para la gestión de proyectos que utilizan metodologías ágiles, pero no de código abierto, lo cual impide una posible ampliación a futuro de los servicios de la aplicación según las necesidades de los usuarios. Por este motivo, el propósito de este proyecto es el desarrollo de una aplicación de código abierto que sea fácil de implementar y de ampliar en funcionalidades por parte de los usuarios. Para lograrlo, se ha implementado una aplicación web que emplea PHP, JavaScript, MySQL, HTML, y CSS capaz de realizar las funcionalidades de un gestor de proyectos, pero gracias al framework de Laravel, con una gran facilidad para ser ampliado. Además, como parte del proyecto se ha incluido un ejemplo de uso ilustrativo para mostrar el correcto funcionamiento de la herramienta. Por último cabe destacar, que para garantizar la mayor apertura al uso por parte de los usuarios, se ha buscado que la aplicación pueda ser empleada desde cualquier dispositivo garantizando que el diseño de la aplicación sea responsivo a los tipos de pantalla en los que se espera que pueda ser lanzada.

## Resum

En aquest treball es busca crear una aplicació web que serveixi per centralitzar les tasques associades a la gestió de projectes mitjançant la metodologia àgil Scrum. Actualment, hi ha moltes eines en línia per a la gestió de projectes que utilitzen metodologies àgils, però no de codi obert, cosa que impedeix una possible ampliació a futur dels serveis de l'aplicació segons les necessitats dels usuaris. Per aquest motiu, el propòsit d'aquest projecte és desenvolupar una aplicació de codi obert que sigui fàcil d'implementar i ampliar en funcionalitats per part dels usuaris. Per aconseguir-ho, s'ha implementat una aplicació web que utilitza PHP, JavaScript, MySQL, HTML i CSS capaç de realitzar les funcionalitats d'un gestor de projectes, però gràcies al framework de Laravel, amb una gran facilitat per ser ampliat. A més, com a part del projecte s'ha inclòs un exemple d'ús il·lustratiu per mostrar el funcionament correcte de l'eina. Finalment, cal destacar que per garantir la major obertura a l'ús per part dels usuaris, s'ha buscat que l'aplicació pugui ser utilitzada des de qualsevol dispositiu garantint que el disseny de l'aplicació sigui responsiu als tipus de pantalla en què s'espera que pugui ser llançada.

## Abstract

This work seeks to create a web application that serves to centralize the tasks associated with project management through the agile Scrum methodology. Currently, there are many online tools for project management that use agile methodologies, but not open source, which prevents a possible future extension of the application services according to the needs of the users. For this reason, the purpose of this project is the development of an open source application that is easy to implement and expand its functionality by users. To achieve this, a web application has been implemented that uses PHP, JavaScript, MySQL, HTML, and CSS capable of performing the functionalities of a project manager, but thanks to the Laravel framework, with great ease to be extended. In addition, as part of the project, an illustrative use example has been included to show the correct operation of the tool. Finally, it should be noted that in order to guarantee the greatest openness to use by users, it has been sought that the application can be used from any device, guaranteeing that the design of the application is responsive to the types of screens in which it is expected. that can be thrown.

## Índice

1.	Justificación.....	6
2.	Gestión del proyecto.....	6
3.	Herramientas utilizadas.....	9
4.	Descripción del proyecto.....	10
4.1.	Análisis.....	11
4.1.1.	Requisitos funcionales.....	11
4.1.2.	Requisitos no funcionales.....	12
4.1.3.	Criterios de aceptación.....	12
4.2.	Diseño.....	12
4.2.1.	Guía de estilos.....	13
4.2.2.	Iconografía y tono.....	14
4.2.3.	Diagrama de entidad relación.....	15
4.2.4.	Casos de uso.....	16
4.2.5.	Site map.....	18
4.3.	Desarrollo.....	19
4.4.	Pruebas.....	20
4.4.1.	Pruebas de usuario.....	20
4.4.2.	Pruebas de proyectos.....	23
4.4.3.	Control de acceso.....	27
4.4.4.	Pruebas de API's.....	28
4.5.	Documentación.....	29
5.	Trabajos futuros.....	29
6.	Conclusiones.....	30
7.	Webgrafía.....	31
8.	Glosario.....	32
9.	Anexos.....	35

## Índice de imágenes

Figura 1: modelo de organización de un proyecto Scrum.....	7
Figura 2: logotipos de las tecnologías empleadas en el desarrollo del proyecto.....	9
Figura 3: Prototipo de la página de login para pantallas de ordenador de sobremesa, tablet, y móvil.....	13
Figura 4: logotipo de la aplicación.....	14
Figura 5: Diagrama de entidad relación de la base de datos del proyecto.....	15
Figura 6: Diagrama de casos de uso de la aplicación.....	17
Figura 7: Site map de la aplicación.....	18
Figura 8: ejemplo de código de una plantilla multilingüe usando Laravel Blade.....	20
Figura 9: pantalla de inicio cuando no se ha iniciado sesión, plano centrado en los botones de navegación y el botón de registro.....	21
Figura 10: pantalla de registro con ejemplo de rellenado de datos.....	21
Figura 11: ejemplo de los registros de usuarios en la base de datos.....	22
Figura 12: formulario de inicio de sesión.....	22
Figura 13: página de inicio de la aplicación, pero con la sesión iniciada.....	22
Figura 14: página de proyectos, en estos momentos vacía.....	23
Figura 15: página de creación de proyectos.....	24
Figura 16: página de proyectos, ahora con un proyecto tras nuestra creación.....	24
Figura 17: página del proyecto, mostrando sus características.....	25
Figura 18: página de edición de proyecto.....	25
Figura 19: página del proyecto, mostrando sus características tras la edición.....	26
Figura 20: página de confirmación de eliminación del proyecto.....	26
Figura 21: página de usuarios con capacidad para eliminar usuarios y asignarles rol de usuario o administrador.....	27
Figura 22: página de usuarios al buscar un usuario que contenga las letras que seguidas en su nombre o correo.....	28

## 1. Justificación

En la actualidad la mayoría de los proyectos informáticos requieren la colaboración de equipos informáticos con personas que no saben de informática durante largos periodos de tiempo. Esto ha conllevado la necesidad de emplear metodologías de trabajo que fomenten la cooperación, la comunicación con los clientes, y la flexibilidad en el trabajo.

Para satisfacer estas necesidades en el entorno laboral se han desarrollado las metodologías ágiles, que centran el desarrollo del proyecto en la comunicación periódica con el cliente, la revisión del proyecto, y el trabajo en equipo.

Según la Innovation & Entrepreneurship Business School, las metodologías ágiles más empleadas son Extrema (XP), Scrum, y Kanban, siendo Scrum una metodología muy abierta y fácil de combinar con otras metodologías como lo demuestran metodologías como la Scrumban, mezcla de Scrum y Kanban.

Es por este motivo por el que se ha decidido optar por usar Scrum como metodología para la aplicación, ya que puede ser fácilmente ampliada usando elementos de las otras metodologías, y al utilizar Laravel y ser open source, es fácil ampliar la aplicación para que se adapte a las necesidades de cada equipo. Con esto se pretende desarrollar una aplicación escalable que permita una gestión simple y clara de proyectos mediante una metodología ágil.

## 2. Gestión del proyecto

Dado que este proyecto trata del desarrollo de un gestor de proyectos de metodología Scrum, para la gestión del desarrollo del mismo, se ha optado por el uso de la metodología Scrum para organizar el proyecto.

Scrum es una metodología ágil que distribuye los proyectos en sprints, periodos de entre una y dos semanas, en los cuales se busca completar una funcionalidad específica del proyecto, para tener una nueva reunión con el equipo y el cliente, analizar el estado actual del proyecto, y en base a los resultados reorganizar los sprints o terminar el proyecto. Gracias a esto es una metodología rápida y efectiva. La organización de un proyecto sprint es algo que podemos ver más aclarado en el siguiente diagrama:

## SCRUM FRAMEWORK

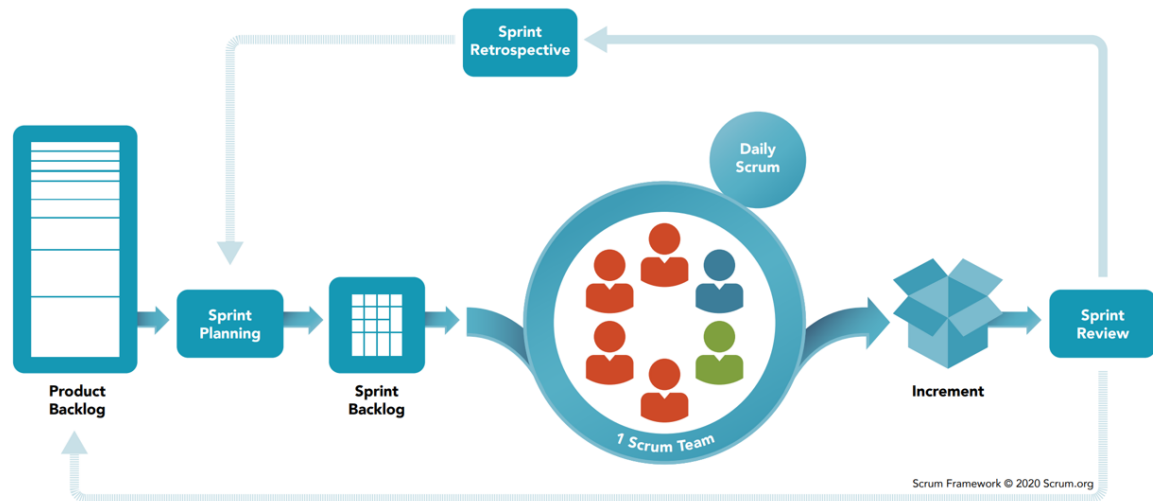


Figura 1: modelo de organización de un proyecto Scrum

Debido a esto, el proyecto se ha organizado en Sprints de una semana, tras los cuales se ha desarrollado al menos una funcionalidad del proyecto y se ha procedido a reestructurar la planificación de los Sprints futuros. Lo cual ha dado al proyecto un alto grado de organización y flexibilidad.

El objetivo de esta aplicación, es que además de poder gestionar los procesos de un proyecto Scrum, sea fácilmente expandible, tanto horizontal como verticalmente. Debido a esto, durante la implementación de la misma se ha buscado garantizar una buena división de las responsabilidades que cada apartado de la aplicación debe controlar.

Para el caso particular de este proyecto, dado que es un proyecto individual, no hay una división de tareas entre trabajadores, y se considera al tutor como el cliente que ha pedido la aplicación dado que es el que puede pedir o alterar los requisitos de la aplicación según vea el desarrollo de la misma. Por otra parte, los Sprints han tenido una duración de una semana para así poder garantizar la flexibilidad del desarrollo de este proyecto debido a los requisitos de duración del desarrollo especificados. Por último, cada una de las tareas incluidas en el desarrollo de los Sprints constituye su propia funcionalidad dentro de la aplicación, garantizando así, que al completarse las tareas, el proyecto gana funcionalidades nuevas, sin poder encontrarse en situaciones donde tenga algunas funciones a medias.

Debido a todo lo dicho anteriormente, para el desarrollo de cada scrum se ha optado por desarrollar una funcionalidad específica del proyecto.

Originalmente, la idea con la que se planteó el desarrollo del proyecto consistía en desarrollar la maquetación y la base de datos durante los dos primeros sprints, entonces dedicar un par de sprints a las vistas, y entonces dedicar el resto del tiempo a los controladores y modelos. No obstante, conforme se desarrolló la implementación de la base de datos se hizo evidente que incluso con la especificación de requisitos había elementos de la base de datos que requerirían modificaciones conforme se desarrollara el proyecto. Debido a esto se decidió cambiar a un sistema en el que en cada sprint se desarrollara una parte específica de la aplicación como se puede observar a continuación.

Durante el primer sprint, a finales de marzo, se procedió a desarrollar una maquetación del proyecto de la que ya se ha visto parte en el apartado de diseño, y se procedió a diseñar la base de datos. En retrospectiva de este sprint, se determinó que parte de la base de datos tendría que modificarse en futuros sprints en base al desarrollo del proyecto para satisfacer las necesidades que pudieran surgir durante la fase de desarrollo.

En el segundo sprint, a principios de abril, se ha procedido al despliegue del prototipo del proyecto, y la base de datos. Además, se desarrolló el sistema de sesiones de la aplicación, y varias plantillas de uso general para la aplicación.

En el segundo sprint, a mediados de abril, se desarrolló el aspecto visual de la aplicación para darle el aspecto desarrollado en la fase de maquetación, y se procedió a desarrollar las funcionalidades básicas de los proyectos, los sprints, y las tareas.

El tercer sprint, a finales de abril, se centró en la asociación de los usuarios a los proyectos, los sprints, y las tareas. Gracias a este sprint, se logró que los usuarios se pudieran introducir en las tareas. También fue cuando se desarrolló el sistema de seguridad contra accesos sin autorización.

Durante el cuarto sprint, a principios de mayo se desarrolló el sistema para hacer la aplicación multiidioma, y se desarrollaron las peticiones por medio de API Restful, dando mayores funcionalidades al entorno cliente.



Finalmente, durante el quinto sprint a mediados de mayo, se procedió a pulir detalles y hacer testeo de las funcionalidades de la aplicación.

### 3. Herramientas utilizadas



Figura 2: logotipos de las tecnologías empleadas en el desarrollo del proyecto

Para el desarrollo de este proyecto, se han empleado las siguientes tecnologías:

Como framework se ha empleado Laravel 9.52.5, un framework de PHP muy utilizado a día de hoy para desarrollar aplicaciones modernas y robustas. Este framework provee de una serie de herramientas y librerías muy útiles para el desarrollo de una aplicación escalable y segura.

Como motor de plantillas se ha empleado Blade, un motor de plantillas del propio framework de Laravel que provee de una sintaxis sencilla para la creación de vistas y plantillas para la aplicación.

Como sistema de mapeo de objetos se ha empleado Eloquent ORM, el cual forma parte de Laravel y provee de una forma simple e intuitiva de interactuar con la base de datos, permitiendo a los desarrolladores emplear consultas eficientes para obtener la información deseada.

Como gestor de dependencias y paquetes se ha empleado Composer, un gestor de PHP que facilita la instalación, desinstalación, y actualización de las librerías de la aplicación.

Para la base de datos se ha utilizado una base de datos MySQL, dado que MySQL es un sistema de manejo de datos open-source y muy popular que permite guardar y gestionar la información de la aplicación con muchas herramientas para la gestión y uso de la misma.

Como administrador de la base de datos se ha empleado phpMyAdmin 5.2.0, un gestor simple e intuitivo para el manejo de las bases de datos escrito en PHP.

Como sistema de control de versiones se ha empleado Git, el cual permite controlar los cambios de código en los archivos, colaborar con otros desarrolladores, e incluso revertir a versiones previas.

Como framework de front-end se ha empleado Bootstrap 5, un framework popular que se usa para desarrollar aplicaciones web responsivas que provee una serie de herramientas y componentes para facilitar un desarrollo de interfaces moderno y profesional.

## 4. Descripción del proyecto

El objetivo de este proyecto, es el desarrollo de una aplicación web capaz de centralizar las tareas asociadas a la gestión de proyectos. Para lograr esto, la aplicación ha distinguido entre los administradores de la aplicación, los administradores de los proyectos, los usuarios identificados, y los usuarios no identificados.

Los administradores de la aplicación tendrán acceso a la base de datos, a la información de los usuarios de la aplicación, y a todos los proyectos de esta.

Todos los usuarios identificados tendrán la capacidad de crear proyectos, pasando así a ser administradores de proyectos, mandar mensajes a otros usuarios de la aplicación, ver y editar los proyectos a los que se les haya invitado, puede ver los backlogs de los proyectos a los que se le ha invitado, ver aquellos que sean públicos, y eliminar sus cuentas.

Finalmente, los usuarios no identificados, podrán crear sus propias cuentas para utilizar la aplicación, iniciar sesión en la aplicación, y ver los proyectos públicos de la aplicación así como los backlogs de estos proyectos.

Todos los proyectos creados en la aplicación, permitirán la creación de sprints y tareas, indicando la fecha de inicio y la duración de los mismos, la creación de un backlog al que tendrán acceso al menos todos los usuarios con acceso al proyecto. Además, el proyecto mostrará una lista de las tareas pendientes en el sprint actual.

## 4.1. Análisis

Para lograr los objetivos especificados para este proyecto, es necesario cumplimentar una serie de requisitos para garantizar que la aplicación pueda ser usada en un entorno de trabajo para la gestión de proyectos. Estos requisitos se han indicado a continuación.

### 4.1.1. Requisitos funcionales

#### Autenticación de usuario

La aplicación tiene que tener un inicio de sesión seguro basándose en su dirección de correo y contraseña.

Los usuarios serán capaces de crear nuevas cuentas, y a partir de ellas crear, proyectos, sprints, y tareas, editarlas, y borrarlas si fuera necesario.

#### Administración de proyectos

Los administradores de proyectos deberán ser capaces de invitar o borrar del proyecto a otros usuarios.

Los usuarios serán capaces de marcar las tareas completadas.

Los usuarios deberán ser capaces de mandarse mensajes para poder comunicarse a través de la aplicación.

#### Interfaz de usuario

El diseño de la interfaz ha de ser responsivo y estar optimizado para diferentes pantallas.

Además, la interfaz de usuario debería ser intuitiva y fácil de navegar, con nombres y botones claros para las acciones comunes.

#### Integración de servicios externos

La aplicación también integrará un servicio de calendario externo para que los usuarios puedan ver sus tareas pendientes en formato de calendario.

### 4.1.2. Requisitos no funcionales

#### Rendimiento

La aplicación deberá cargarse rápidamente, y no presentar retrasos a los eventos de usuario.

#### Seguridad

La aplicación debe ser diseñada teniendo la seguridad en cuenta, incluyendo protección contra ataques por inyección de código, y secuencias de comandos entre sitios.

La información de los usuarios deberá estar encriptada para impedir accesos sin autorización a la misma.

#### Compatibilidad

La aplicación debe ser compatible con navegadores modernos como Google Chrome, Mozilla Firefox, y Microsoft Edge.

### 4.1.3. Criterios de aceptación

La aplicación debe ser funcional tanto en los requisitos funcionales como en los no funcionales y haber sido testeada para garantizar el cumplimiento de todos los criterios ya indicados a un nivel en el que pueda ser empleada en un entorno laboral.

## 4.2. Diseño

Como ya se ha visto en el apartado anterior, es requisito de la aplicación garantizar que pueda ser usada con comodidad en cualquier tipo de pantalla, para lo cual es necesario garantizar un diseño responsivo para todas las páginas. Para lograr esto a lo largo de toda la aplicación, se ha hecho uso de flexbox, grid, media queries, y elementos de bootstrap, que controlan los anchos de los elementos en pantalla, y su capacidad para redistribuirse según convenga a la pantalla en la que se encuentren. A continuación se puede observar cómo se distribuye una misma página para diferentes pantallas para entender el funcionamiento del diseño.



Figura 3: Prototipo de la página de login para pantallas de ordenador de sobremesa, tablet, y móvil

### 4.2.1. Guía de estilos

El objetivo del diseño de la aplicación es transmitir un diseño enérgico pero profesional, centrándose en la accesibilidad y en remarcar los puntos de acción principales de la aplicación. Para lo cual se ha usado colores azules para la navegación y los botones de confirmación y edición, blanco para el texto de la navegación, el texto del footer y el fondo del resto de la página para que destaquen del resto de secciones, negro para el footer, rojo para los botones de eliminar, y gris para los botones de cancelación. Con esto se ha buscado dar una imagen formal pero simultáneamente animada, mientras se usa una paleta de colores que puede ser empleada fácilmente en futuras páginas.

La paleta de colores ha sido usada en la aplicación ha sido:

- #0d6efd para los botones de confirmación y edición.
- #7BC1ED para la navegación.
- #FFFFFF para el fondo general de la aplicación, los enlaces de la navegación, el texto de botones, y el fondo de los seleccionables.
- #212529 para el fondo del pie de página.
- #000000 para el texto general de la página.
- #DC3545 para los botones de eliminación.

Por otra parte, se ha usado las fuentes Segoe UI, Roboto, Helvetica Neue, Arial, Noto Sans, Liberation Sans, sans-serif, Apple Color Emoji, Segoe UI Emoji, Segoe UI Symbol, y Noto Color Emoji, por su facilidad de lectura, gracias a sus caracteres limpios, su buen espaciado, y su fuerte marcado de los caracteres. Además, se ha ampliado el tamaño de todas las letras y el espaciado entre elementos, para facilitar la lectura.

## 4.2.2. Iconografía y tono

Dado que el objetivo de la aplicación es poder ser usada para la gestión de proyectos en el entorno laboral, en toda la aplicación se ha optado por un tono neutro y formal, apropiado para el entorno laboral.

Por otra parte, con respecto a la iconografía se ha optado por emplear poca y clara, los botones por ejemplo, tienen escrito el uso de los mismos. Sin embargo, se han empleado iconos en la aplicación para representar instituciones, familias, borrados, y el logotipo de la aplicación.

Por un lado, se ha empleado el siguiente icono como representante de la aplicación:



Figura 4: logotipo de la aplicación

El color de fondo de este icono busca mantener una sensación de armonía con la aplicación mientras simultáneamente resalta con respecto al resto de la navegación, dado que se ubica en esta, y puede ser usado como enlace para retornar al inicio de la misma. Se usan letras blancas con el propósito de resaltar el nombre de la aplicación con respecto al resto de la misma, y hay una línea roja similar a una gráfica en crecimiento en el fondo con el propósito de dar una sensación de progreso a quienes ven la aplicación.

Además del logotipo, para el borrado de trabajadores en las tareas, se ha empleado una cruz roja, ya que hacer un botón entero de borrado de las tareas invadiría demasiado espacio visual en la aplicación, y a diferencia de los botones de borrado, el eliminado de trabajadores funciona mediante peticiones API Restful, y no requieren de ser un botones ni enlaces.

Por último, se han empleado los logotipos del IES Serpis, y de la familia de Informática y Comunicaciones en el footer para denotar el origen de este proyecto.

### 4.2.3. Diagrama de entidad relación

En el desarrollo de esta aplicación se ha empleado una base de datos relacional MySQL, por su fiabilidad, facilidad de uso y extensión en el mercado laboral. Las tablas empleadas en la base de datos se han visto dadas por la estructura de los proyectos Scrum, las tablas predeterminadas de Laravel, por las necesidades de acceso de los usuarios y por las necesidades de las API 's Restful empleadas. Dando como resultado el siguiente diagrama:

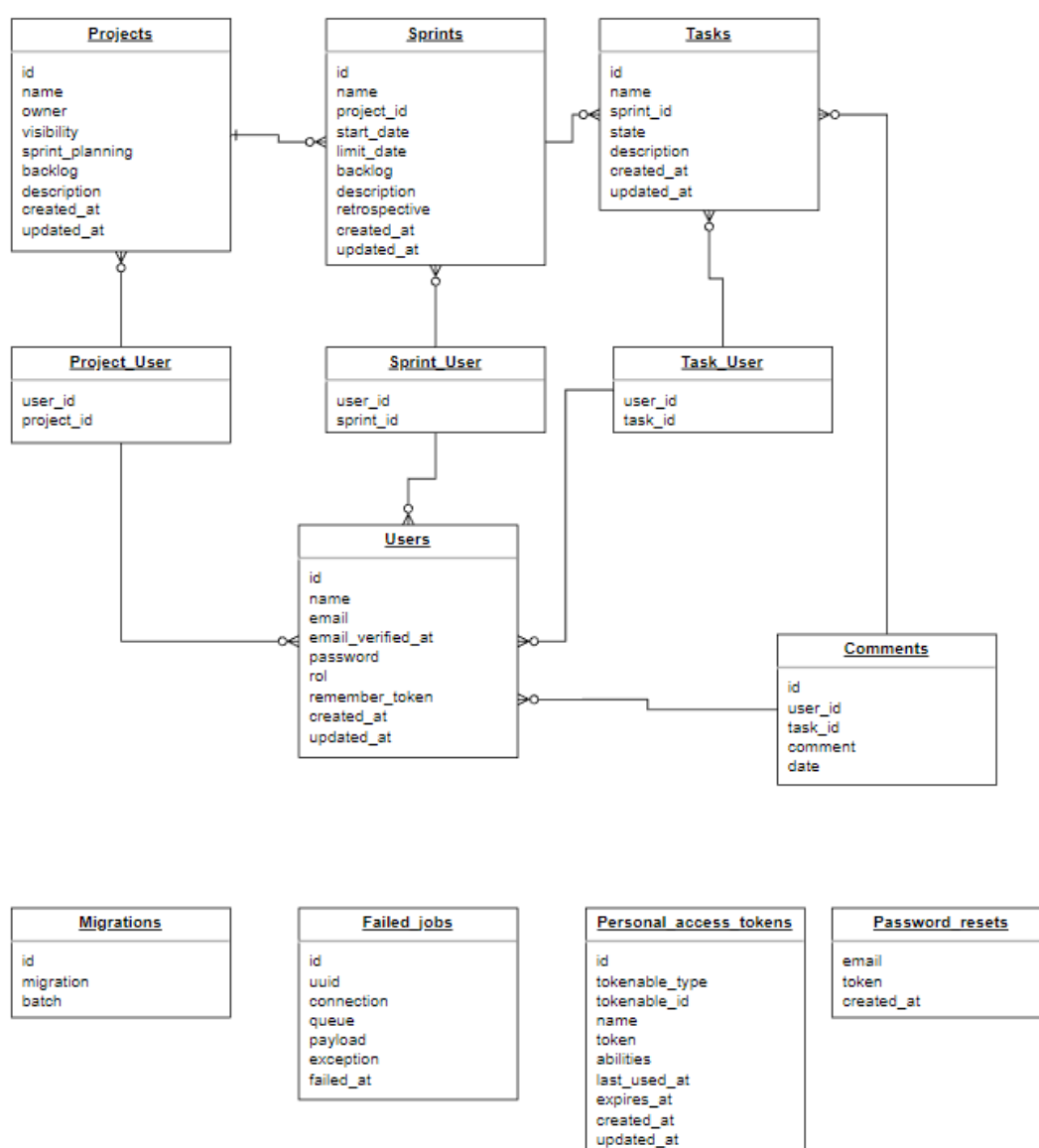


Figura 5: Diagrama de entidad relación de la base de datos del proyecto

Como se puede observar en la figura 5, existen varias tablas en la parte inferior del diagrama que no se unen a ninguna otra tabla. Estas son tablas que Laravel genera automáticamente para su gestión de la base de datos y los tokens de acceso.

Por otra parte, la tabla Task\_User podría parecer innecesaria teniendo la tabla de Comments, pero cabe destacar, que un usuario puede estar incluido en una tarea pero no haber dejado un comentario, y un usuario puede dejar de trabajar en la tarea pero sus comentarios deben permanecer en la tarea.

Con respecto a la tabla Users, que guarda la información de los usuarios. Se ha optado por guardar el nombre, la contraseña, el rol, y el email, el resto de valores son valores que Laravel asigna automáticamente a los usuarios para controlar las sesiones. Por otra parte, se ha optado por guardar el email del usuario para en futuras expansiones lograr introducir un servicio de notificación a través del correo electrónico.

Por último, las relaciones de las tablas Projects, Sprints, y Tasks, con la tabla Users tienen todas una tabla relacional por que un usuario puede estar trabajando en varios proyectos, sprints, y tareas simultáneamente, pero al mismo tiempo cualquier de estos puede tener a varios trabajadores trabajando en un mismo proyecto, sprint o tarea.

#### 4.2.4. Casos de uso

Para poder realizar un correcto desarrollo de la aplicación ha sido necesario desarrollar un diagrama de casos de uso que explique de una forma esquematizada explique las acciones que el usuario pueda hacer desde la aplicación.

Para esto, ha habido que diferenciar las acciones que el usuario de la aplicación puede realizar sin haber abierto sesión, las que puede realizar habiendo abierto sesión pero sin ser un administrador, y las que puede realizar solo un administrador que ha abierto sesión. Todo esto, puede ser visto en el siguiente diagrama de casos de uso:



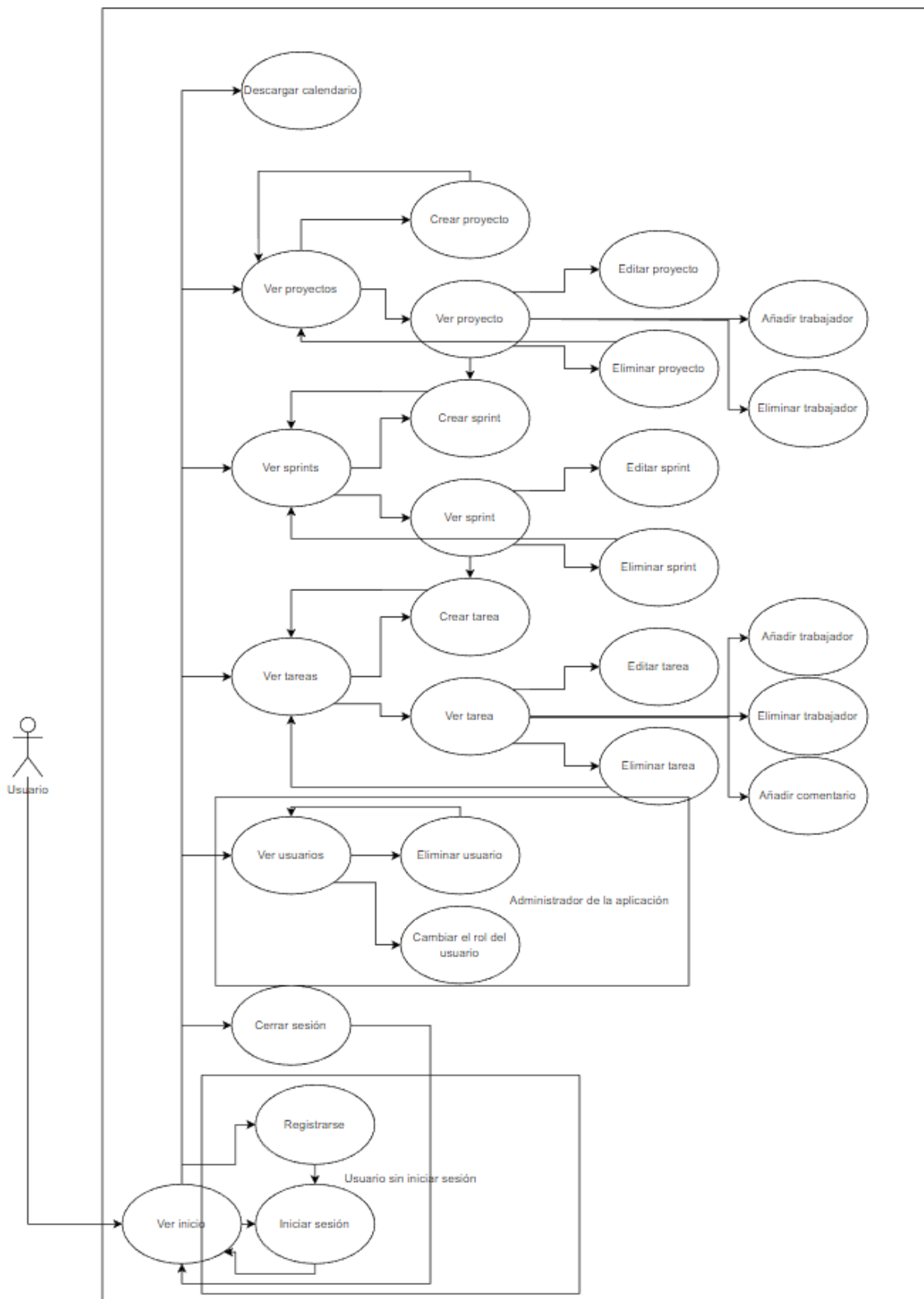


Figura 6: Diagrama de casos de uso de la aplicación

Como podemos ver, un usuario que no ha iniciado sesión solo puede entrar en la pantalla de inicio, registrarse, o iniciar sesión. Mientras que al iniciar sesión, cualquier usuario, independientemente de si es un administrador o no, puede realizar todas las funciones relacionadas a la gestión de proyectos siempre y cuando los proyectos sean propios. Por último solo los administradores pueden cambiar los permisos de otros usuarios y eliminar usuarios.

#### 4.2.5. Site map

Entendidas las acciones que puede realizar el usuario en la aplicación, se ha considerado necesaria la creación de un diagrama que explique la distribución de las páginas web de la aplicación para simplificar la navegación por la misma, cosa que podemos ver en el siguiente diagrama:

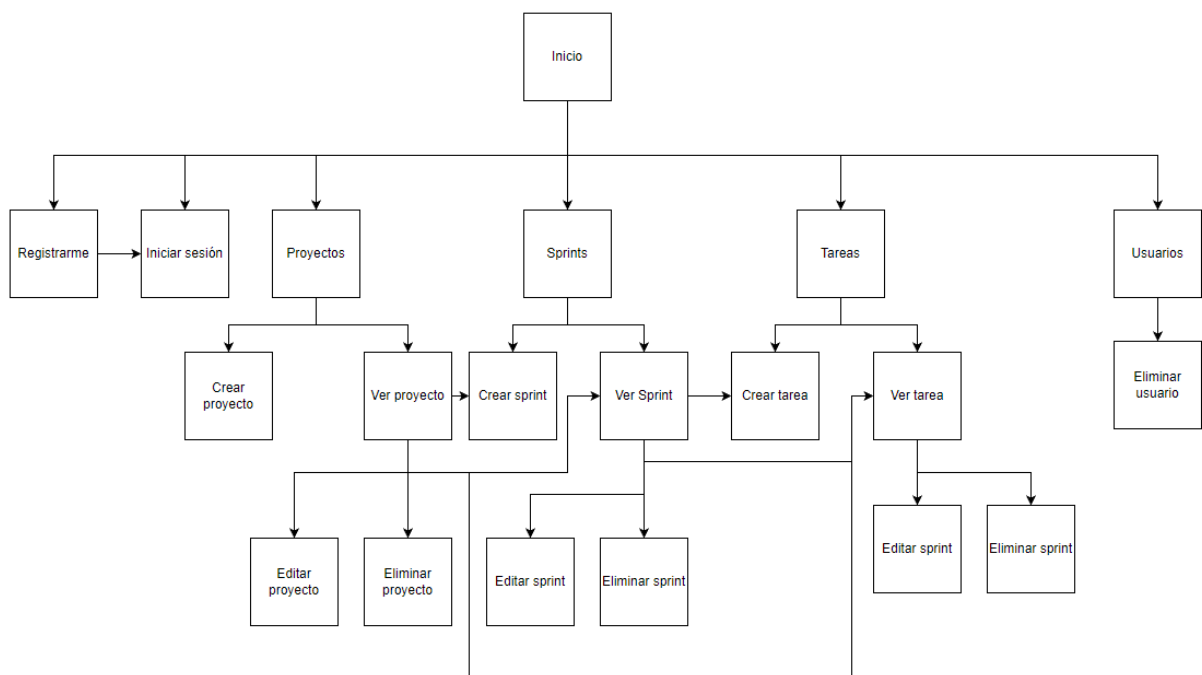


Figura 7: Site map de la aplicación

Como se puede observar, hay algunas funciones indicadas en el diagrama de casos de uso que no se contemplan en este diagrama, pero esto se debe a que algunas de esas funciones se ha optado por resolverlas mediante el uso de API Restful, lo cual implica que no generan nuevas páginas para la aplicación.

## 4.3. Desarrollo

Para la división de responsabilidades en la aplicación, se ha empleado el modelo vista controlador, dado que es el sistema de desarrollo más extendido en el desarrollo de las aplicaciones web, y facilita la extensibilidad y el control de errores de las aplicaciones.

Además de esto, se ha tenido en cuenta que no todas las interacciones con la base de datos se van a hacer pasando directamente por el entorno servidor, sino que algunas es posible que se realicen por medio de peticiones API Restful.

Por último se ha tenido en cuenta la división de las diferentes partes de un proyecto según la metodología Scrum.

Esto ha llevado a que el desarrollo del proyecto se haya repartido en varias fases.

Al principio se crearon el modelo, las vistas, el controlador, y las peticiones de sesión, más tarde se hizo lo mismo con el apartado de proyectos, posteriormente con sprints, y luego con las tareas.

Una vez funcionaban las secciones básicas de la aplicación, el siguiente apartado fue el de controlar los permisos de usuario, en el que se creó una nueva vista, y se comenzó a trabajar con las peticiones API. Fue en esta fase donde se introdujo la seguridad en las rutas, dado que al poder configurarse desde la aplicación los permisos de cada usuario, ya se podía experimentar con los mismos para controlar su buen funcionamiento y las redirecciones.

Cuando ya funcionaba el control de accesos, se procedió a hacer la aplicación multilingüe, a través del uso de cookies. En este apartado se empleó una mayor cantidad de tiempo debido a la necesidad de preparar traducciones para todas las frases utilizadas por la aplicación en tres idiomas. Además se fue preparando el uso de las API's para el siguiente sprint.

## Gestor de proyectos para desarrollo de aplicaciones informáticas con Scrum

```
@extends('layout')
@section('titulo', __('inicio'))
@section('contenido')
    @if (!isset(Auth::user()->name))
        <div class="card" style="width: 18rem;">
            <div class="card-body">
                <h5 class="card-title">Scrum Admin</h5>
                <p class="card-text">{{ __('desarrolla') }}<br><br>
                    {{ __('facil') }}</p>
                <a href="{{ route('registro') }}" class="btn btn-primary">{{ __('registrarme') }}</a>
            </div>
        </div>
    @endif
    <section class="novedades">
        {{-- Aquí irán las novedades --}}
        <h1>
            @if (!isset(Auth::user()->name))
                {{ __('crea-desarrolla') }}
            @else
                {{ __('bienvenido') }} {{ Auth::user()->name }}
            <br>
            <a href="{{ route('calendar') }}" class="btn btn-primary">{{ __('imprimir-calendario') }}</a>
        @endif
        </h1>
    </section>
@endsection
```

Figura 8: ejemplo de código de una plantilla multilingüe usando Laravel Blade

Por último, se implementó el sistema para añadir usuarios a tareas y proyectos a través de peticiones API Restful, y la búsqueda por medio de JQuery de usuarios. Además, se logró crear un botón para descargar el calendario que el usuario tendría que seguir de acuerdo a los requisitos de la aplicación. Al lograr esto la aplicación llegó a un punto en el que ya se podía emplear en el entorno laboral, y no había plazo para desarrollar más la aplicación antes de presentarla para la exposición.

## 4.4. Pruebas

Para demostrar el correcto funcionamiento de la aplicación, ha sido necesario realizar una serie de pruebas para comprobar que la aplicación se encuentra en una fase en la que puede ser empleada en el desarrollo de un proyecto. Sin embargo, debido a la naturaleza de las múltiples funcionalidades del proyecto, se ha optado por dividir las pruebas de funcionalidad en varios apartados: pruebas de usuario, pruebas de proyectos, control de acceso, y API 's.

### 4.4.1. Pruebas de usuario

Las pruebas de usuario se refieren a las pruebas de creación de cuenta, inicio de sesión, y cierre de sesión. Es necesario comprobar que cualquier usuario puede crear una cuenta en la aplicación, puede abrir sesión en ella correctamente, y puede cerrar sesión con esta.

## Gestor de proyectos para desarrollo de aplicaciones informáticas con Scrum

Con respecto a la creación de cuenta en la aplicación, se puede entrar en la página de registro desde el botón de registro en el inicio de la aplicación, o desde el apartado de registro de la aplicación como se puede ver en la siguiente imagen.

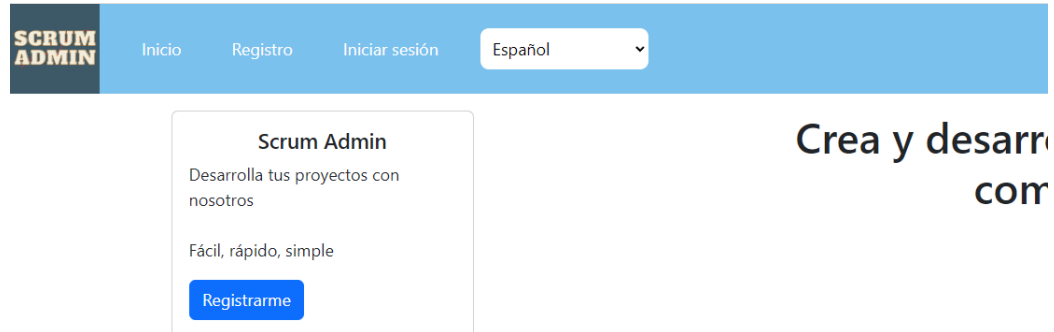


Figura 9: pantalla de inicio cuando no se ha iniciado sesión, plano centrado en los botones de navegación y el botón de registro

Al pulsar cualquiera de estos botones, se traslada al usuario a una nueva página con un formulario de registro como podemos ver a continuación:

Figura 10: pantalla de registro con ejemplo de rellenado de datos

Como se puede observar, en caso de que el usuario intente enviar un formulario incompleto o con datos incorrectos, el formulario se volverá a mostrar con mensajes de error para indicar al usuario los cambios que debe realizar.

Tras enviar un formulario de registro cumplimentado, se le redirige a la página de inicio de sesión, desde la que podrá iniciar sesión. Además, si entramos en la base de datos, podremos ver el registro del usuario en la tabla de usuarios como vemos en la siguiente imagen:

## Gestor de proyectos para desarrollo de aplicaciones informáticas con Scrum

id	name	email	email_verified_at	password	rol	re
1	sa	iesserpis@gmail.com	NULL	\$2y\$10\$7N5hJnBg8oL6EwHgl4XXduluVbldQKCZv6L6ovuuyOa...	admin	NL
2	enrique	enriquesanzlopez@gmail.com	NULL	\$2y\$10\$U.lrd6MJdkbEqRd0/D2zkObDQaK1n4OMop3HnOmmXTZ...	user	NL

Figura 11: ejemplo de los registros de usuarios en la base de datos

Como podemos observar, no solo se guarda correctamente la cuenta, sino que además se codifica la contraseña, y de forma predeterminada se le añade el rol de usuario a la cuenta.

Por otra parte, con respecto al inicio de sesión, como ya se ha explicado, en caso de registrarse, el usuario es enviado a la página de inicio de sesión, pero además, el usuario tiene acceso a la página de inicio de sesión desde la navegación como se puede ver en la figura 4. Una vez en esta página, el usuario verá un formulario de inicio de sesión como podemos ver a continuación:

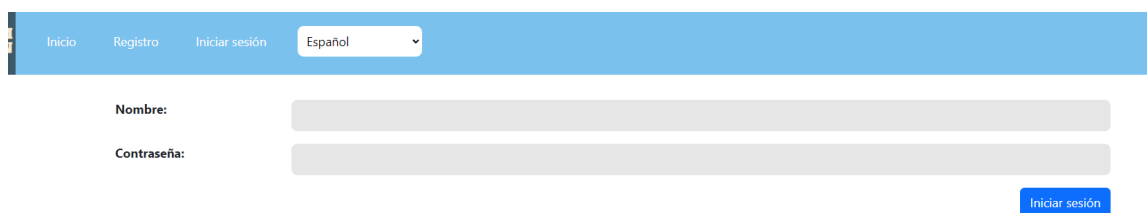
El formulario de inicio de sesión se encuentra en una barra de navegación azul. En la barra, hay enlaces para 'Inicio', 'Registro' y 'Iniciar sesión', seguidos de un menú desplegable con la opción 'Español'. Debajo de la barra, hay dos campos de entrada: 'Nombre:' y 'Contraseña:'. A la derecha de los campos, hay un botón azul que dice 'Iniciar sesión'.

Figura 12: formulario de inicio de sesión

Si este formulario se envía con datos incorrectos, no se permite el inicio de sesión, pero no se le indica al usuario el motivo del fallo para evitar el filtrado de datos. En caso de cumplimentar el formulario con datos correctos, el usuario iniciará sesión y se le devolverá al inicio, pero cambiado para devolver dar más opciones al usuario e indicarle que ha iniciado sesión, como se puede ver a continuación:

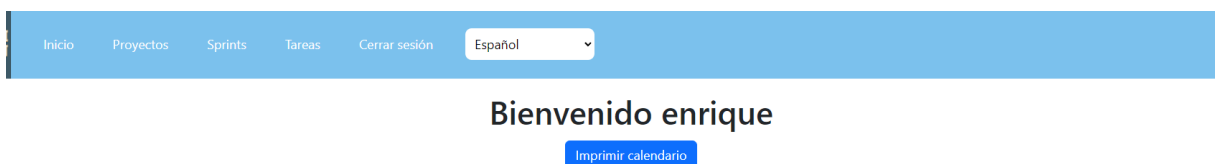
La página de inicio de la aplicación muestra una barra de navegación azul con los enlaces 'Inicio', 'Proyectos', 'Sprints', 'Tareas' y 'Cerrar sesión', seguidos de un menú desplegable con la opción 'Español'. Debajo de la barra, hay un mensaje de bienvenida que dice 'Bienvenido enrique'. A la derecha del mensaje, hay un botón azul que dice 'Imprimir calendario'.

Figura 13: página de inicio de la aplicación, pero con la sesión iniciada

Como se puede observar, la página de inicio ha cambiado, y ahora contiene más opciones para el usuario.

Por último, para comprobar el cierre de sesión, una vez iniciada sesión, uno solo tiene que hacer click en el apartado de la navegación de cerrar sesión y la aplicación devolverá al usuario al inicio pero con la sesión cerrada.

#### 4.4.2. Pruebas de proyectos

Las capacidades de crear e iniciar sesión de esta aplicación no tendrían uso alguno si esta aplicación no fuera capaz de gestionar proyectos y para ello, se comprueba en este apartado su capacidad para crear, mostrar, editar y eliminar proyectos, sprints, y tareas.

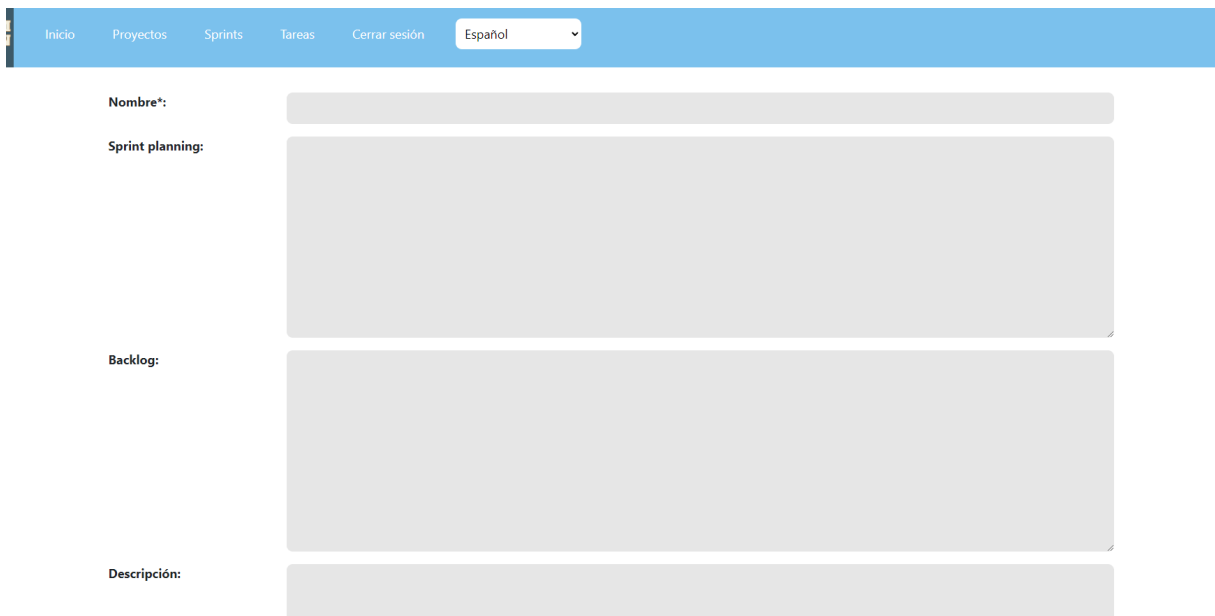
Con respecto a los proyectos, podemos acceder a ellos desde el link de proyectos en navegación como podemos ver en la figura 8.



Figura 14: página de proyectos, en estos momentos vacía

Como podemos observar, en estos momentos no existe ningún proyecto creado, esto se debe a que la cuenta que estamos empleando es la del usuario que acabamos de crear, y por tanto todavía no ha creado ni se le ha asignado ningún proyecto. Para crear un proyecto basta con pulsar el botón que pone crear proyecto. El cual nos enviará al siguiente formulario:

## Gestor de proyectos para desarrollo de aplicaciones informáticas con Scrum



Formulario de creación de proyectos. El formulario está dividido en secciones con etiquetas a la izquierda y campos de entrada a la derecha:

- Nombre\*:** Campo de texto único.
- Sprint planning:** Campo de texto de mayor altura.
- Backlog:** Campo de texto de mayor altura.
- Descripción:** Campo de texto único.

Figura 15: página de creación de proyectos

Este formulario, al igual que el resto de formularios indicados en páginas previas, también comprueba que se han dado valores apropiados a los proyectos. Si llenamos los campos del formulario con datos válidos y enviamos el formulario mediante el botón de crear, se nos devolverá a la lista de proyectos donde comprobaremos que existe un proyecto nuevo con el nombre y la descripción que le hayamos dado como podemos ver a continuación:



Página de proyectos. En la barra superior hay un menú con: Inicio, Proyectos, Sprints, Tareas, Cerrar sesión y un selector de idioma (Español). En el cuerpo de la página:

- A la izquierda, un botón azul que dice "Crear proyecto".
- A la derecha, una tarjeta blanca que muestra el título "Implementación del ERP" y un botón azul "Entrar" debajo.

Figura 16: página de proyectos, ahora con un proyecto tras nuestra creación

En caso de querer ver los detalles del proyecto, basta con pulsar el botón de entrar que tenemos en la propia ficha del proyecto en el listado que estamos viendo, y nos llevará a una página con los detalles del proyecto como vemos a continuación:



## Gestor de proyectos para desarrollo de aplicaciones informáticas con Scrum

The screenshot shows a web application interface for a Scrum project management tool. At the top, there is a navigation bar with links: Inicio, Proyectos, Sprints, Tareas, Cerrar sesión, and Usuarios. A language dropdown menu is set to 'Español'. Below the navigation bar, the main heading is 'Implementación del ERP'. To the right of this heading are three buttons: 'Editar' (blue), 'Crear Sprint' (blue), and 'Eliminar' (red). On the left side, there is a sidebar with the following sections: 'Descripción', 'Backlog', 'Sprint planning', 'Listado de Sprints', and 'Listado de tareas'. The 'Listado de Sprints' section shows 'No hay Sprints actualmente'. The 'Listado de tareas' section shows 'No hay tareas en este Sprint'. On the right side, there is a section titled 'Trabajadores' which lists a user named 'enrique' with the email 'enriquesandlopez@gmail.com'. Below this list is a search bar with the placeholder text 'Añadir trabajadores...' and a red 'x' icon.

Figura 17: página del proyecto, mostrando sus características

Como podemos ver, los detalles de proyecto se muestran apropiadamente, pero, en caso de que el usuario desee cambiar los datos del mismo, si es el creador del proyecto tiene acceso a un botón en la parte superior llamado editar, el cual le llevará a un formulario con los datos del proyecto que puede editar como vemos a continuación:

The screenshot shows the 'Editar' (Edit) page for the project 'Implementación del ERP'. The page has a similar navigation bar at the top. The main content area is divided into four sections: 'Nombre\*', 'Sprint planning', 'Backlog', and 'Descripción'. The 'Nombre\*' section has a text input field containing 'Implementación del ERP - con cambios'. The 'Sprint planning' section has a list of items: 'Bases de datos' and 'Vistas de impresión'. The 'Backlog' section is currently empty. The 'Descripción' section has a text area containing the text 'Vamos a implementar el ERP de la empresa para una empresa agroalimentaria.'.

Figura 18: página de edición de proyecto

Una vez realizados los cambios a los datos del proyecto, el usuario solo debe pulsar el botón que pone guardar los cambios, este le devolverá a la página que muestra el proyecto, con los datos ya editados como se muestra a continuación:

## Gestor de proyectos para desarrollo de aplicaciones informáticas con Scrum

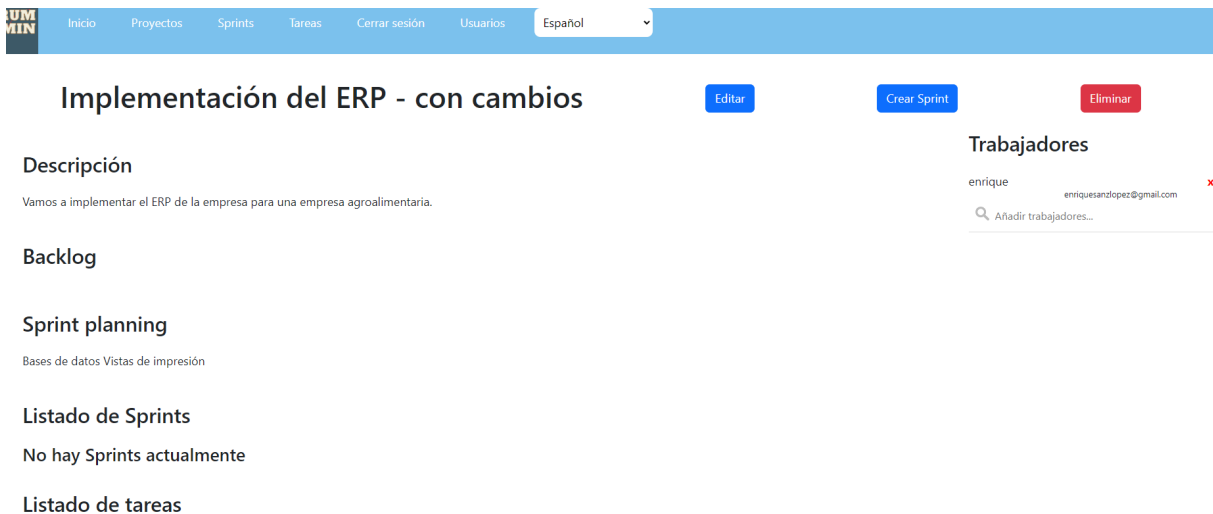


Figura 19: página del proyecto, mostrando sus características tras la edición

Con todo lo probado para este proyecto, solo queda probar la eliminación del mismo, para lo cual hay que pulsar el botón rojo en el que dice eliminar. Al hacerlo, se nos enviará a un formulario de confirmación como se muestra continuación:



Figura 20: página de confirmación de eliminación del proyecto

Si se pulsa en el botón rojo que dice eliminar, se procederá a borrar el proyecto, y todos los sprints y tareas asociados al mismo.

Las mismas funcionalidades que hemos podido ver se aplican a sprints y a tareas, con una salvedad. Un usuario no puede crear un sprint sin haber creado previamente un proyecto, esto se debe a que todo sprint debe ser parte de un proyecto, y al no haber creado uno, el usuario no puede asignarle un proyecto al sprint. Para las tareas pasa algo similar, donde toda tarea debe pertenecer a un sprint, y por tanto, para poder crear una tarea, el usuario debe tener algún sprint creado previamente.

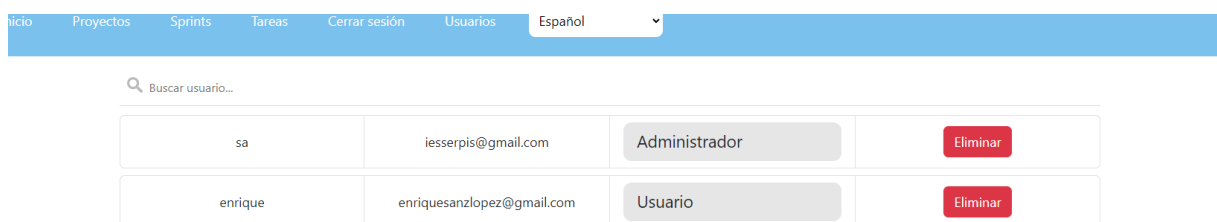
### 4.4.3. Control de acceso

Para el correcto funcionamiento de la aplicación, es necesario comprobar que el usuario solo puede entrar a las páginas de la aplicación a las que tiene acceso. En el primero de los casos, es cuando el usuario desea entrar en el panel de administrador. Si el usuario no es un administrador, no solo no se le muestra la opción de la navegación, sino que si intenta forzar su entrada por medio de la url, se le redirige a la página de inicio de la aplicación.

Del mismo modo, el usuario intenta entrar en las páginas de proyectos, sprints, o tareas, sin haber iniciado sesión, se le redirige a la página de inicio. En caso de haber iniciado sesión, solo se le permite ver los proyectos, sprints, y tareas en los que participa o ha creado.

Además, en caso de querer modificar un proyecto, sprint, o tarea sin ser el creador del proyecto, también te redirige a la página de inicio de la aplicación. De forma que el usuario no puede editar los proyectos si no son suyos. Evitando de esta forma problemas de seguridad cuando se trabaja con varios equipos.

Por otra parte, en caso de tener permisos de administrador de la aplicación, el usuario tiene un enlace adicional para llegar a la página de usuarios, en la que el usuario puede cambiar los roles de otros usuarios y eliminarlos como se puede ver a continuación.



Inicio	Proyectos	Sprints	Tareas	Cerrar sesión	Usuarios	Español
Buscar usuario...						
sa	iesserpis@gmail.com	Administrador	Eliminar			
enrique	enriquesanzlopez@gmail.com	Usuario	Eliminar			

Figura 21: página de usuarios con capacidad para eliminar usuarios y asignarles rol de usuario o administrador

Por último, si el usuario es administrador también gana la capacidad de editar y eliminar proyectos, sprints y tareas aunque no sean suyos.

#### 4.4.4. Pruebas de API 's

Por último, solo queda testear las API's restful empleadas en la implementación de esta aplicación.

En esta aplicación existen cuatro tipos de API distintas. La API de búsqueda de usuarios la cual solo busca a los usuarios de acuerdo a lo introducido por una input y lo devuelve por pantalla para que el usuario pueda verlo como vemos en la siguiente imagen:

Nombre	Correo
enrique	enriquesanzlopez@gmail.com

Figura 21: página de usuarios al buscar un usuario que contenga las letras que seguidas en su nombre o correo

La API de selección de rol, la cual permite cambiar el rol del usuario seleccionado dando a ese usuario autoridad de administrador, o de usuario normal.

La API de inclusión y eliminación de usuario en tarea o proyecto, la cual hace exactamente eso y les da acceso al proyecto para ver los detalles de la tarea y del proyecto. La cual podemos observar en funcionamiento en un proyecto al buscar un usuario en el buscador de usuarios del proyecto, y seleccionarlo, ya que se añadirá a la lista de usuarios en el proyecto.

Finalmente la API de comentarios en tareas, la cual permite a los usuarios escribir comentarios, pulsar un botón de envío, y el comentario se guarda en el servidor y se muestra en la tarea.

#### 4.5. Documentación

Junto a este archivo, el usuario puede la siguiente documentación relativa a la aplicación:

La carpeta TFG, que contiene el código fuente de la aplicación, así como los archivos necesarios para ser implementada en local siempre y cuando se despliegue

la base de datos y se tenga instalado composer para una versión compatible con Laravel 9.

Dentro de la carpeta TFG se encontrará a su vez el archivo bbdd.sql para desplegar la base de datos en caso de necesidad.

El archivo Instrucciones de instalación.pdf, que contendrá las instrucciones para la instalación y ejecución de la aplicación.

El archivo local.zip, que contendrá una versión de XAMPP con la base de datos implementada para una distribución en local de la aplicación.

La presentación de la exposición, para dar una mayor cantidad de información visual al usuario en caso de necesitarla.

## 5. Trabajos futuros

Si bien esta aplicación ha logrado cumplir la mayor parte de los requisitos indicados en la especificación, como se ha podido comprobar, no deja de ser cierto, que todavía hay cosas en las que la aplicación puede mejorar y que están previstas como futuras actualizaciones de la aplicación.

En este plazo de tiempo no ha resultado posible introducir el servicio de mensajería interno de la aplicación, y será la siguiente expansión que el proyecto abarque.

Si bien el sistema de calendario permite descargar el calendario con las fechas y títulos de los sprints del usuario, en un futuro se busca lograr que el calendario se muestre desde la propia aplicación.

Por último, sería un gran beneficio para la aplicación el poder enviar notificaciones por correo a los usuarios para que así estos supieran cuando han sido apuntados a una tarea o proyecto, cosa que esperamos poder lograr en un futuro.

## 6. Conclusiones

Este trabajo de final de ciclo ha consistido en el desarrollo de una aplicación de gestión de proyectos Scrum, la cual da acceso a una diversa cantidad de usuarios.

Conforme se ha realizado el análisis y el diseño de la aplicación, se nos ha hecho ver que el proyecto es mucho más grande y abierto de lo que se puede llegar a cubrir en un espacio de cuarenta horas con la mano de obra que poseemos.

Por ello, se ha tomado la decisión de limitar el desarrollo del proyecto a los elementos del mismo que se espera que sean empleados en el cualquier proyecto que utilice la metodología Scrum y garantizar que todas las funciones implementadas sean funcionales.

Al seguir la metodología Scrum, se han podido encontrar problemas de diseño y errores en el código más rápido que si se hubiera empleado una metodología de trabajo más tradicional gracias a la flexibilidad que esta metodología ofrece.

Durante las fases de análisis, diseño, desarrollo, y pruebas, se han puesto en práctica los conocimientos aprendidos a lo largo de todo el ciclo de Desarrollo de Aplicaciones Web, como el análisis de requisitos, la creación de maquetas, el análisis de fallos, y el diseño de páginas web y algoritmos entre otros, lo cual ha demostrado lo importantes que son las materias aprendidas a lo largo del ciclo para el trabajo en el entorno laboral.

Así mismo, el desarrollo de este proyecto ha servido para aprender materias asociadas al ciclo como:

- La forma de unir las peticiones API Restful con Laravel 9.
- La ampliación de conocimientos de lenguajes de programación y propiedades de CSS.
- Aprender a combinar el conocimiento de todas las materias que se han impartido durante el ciclo en un solo proyecto funcional.
- Aprender a gestionar el tiempo y repartir el trabajo en base a los requerimientos de un proyecto.

No obstante a lo dicho anteriormente, como se ha comentado en el apartado anterior, la aplicación todavía tiene margen para expandir sus funcionalidades debido a la falta de mano de obra. Es por todo esto por lo que se considera que si bien el proyecto ha alcanzado unos niveles de funcionalidad aceptables, todavía tiene margen de mejora.

## 7. Bibliografía y Webgrafía

Para el desarrollo de este proyecto, se han empleado las siguientes páginas como referencia:

*Blade Templates - Laravel - The PHP Framework For Web Artisans*

<<https://laravel.com/docs/9.x/blade>> [Consulta: 30 de marzo de 2023]

*Capítulo 10. Vistas y motor de plantillas Blade · laravel-5*

<<https://richos.gitbooks.io/laravel-5/content/capitulos/chapter10.html>> [Consulta: 18 de mayo de 2023]

*Composer*

<<https://getcomposer.org/doc/>> [Consulta: 30 de marzo de 2023]

*Eloquent: Getting Started - Laravel - The PHP Framework For Web Artisans*

<<https://laravel.com/docs/9.x/eloquent>> [Consulta: 30 de marzo de 2023]

*Git - Documentation*

<<https://git-scm.com/doc>> [Consulta: 30 de marzo de 2023]

*IEBS - La Escuela de Negocios de la Innovación y los Emprendedores*

<<https://www.iebschool.com/>> [Consulta: 18 de marzo de 2023]

*Installation - Laravel - The PHP Framework For Web Artisans*

<<https://laravel.com/docs/9.x>> [Consulta: 30 de marzo de 2023]

*Introduction · Bootstrap v5.0*

<<https://getbootstrap.com/docs/5.0/getting-started/introduction/>> [Consulta: 30 de marzo de 2023]

*jQuery API Documentation*

<<https://api.jquery.com/>> [Consulta: 6 de mayo de 2023]

*jQuery - Wikipedia, la enciclopedia libre*

<<https://es.wikipedia.org/wiki/JQuery>> [Consulta: 18 de mayo de 2023]

MySQL :: MySQL Documentation

<<https://dev.mysql.com/doc/>> [Consulta: 7 de abril de 2023]

Qué es MVC

<<https://desarrolloweb.com/articulos/que-es-mvc.html>> [Consulta: 18 de mayo de 2023]

¿Qué es un diagrama entidad-relación? | Lucidchart

<<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>>  
[Consulta: 18 de mayo de 2023]

▷ ¿Qué es un Sitemap en diseño UX?

<<https://formiux.com/sitemaps-en-ux/>> [Consulta: 18 de mayo de 2023]

¿Qué es una API de RESTful? - Explicación de API de RESTful - AWS

<<https://aws.amazon.com/es/what-is/restful-api/>> [Consulta: 18 de mayo de 2023]

Scrum: qué es, cómo funciona y por qué es excelente

<<https://www.atlassian.com/es/agile/scrum>> [Consulta: 18 de mayo de 2023]

Todo lo que necesitas saber sobre los sprints de scrum

<<https://www.atlassian.com/es/agile/scrum/sprints>> [Consulta: 18 de mayo de 2023]

## 8. Glosario

API RESTful: “La API RESTful es una interfaz que dos sistemas de computación utilizan para intercambiar información de manera segura a través de Internet. La mayoría de las aplicaciones para empresas deben comunicarse con otras aplicaciones internas o de terceros para llevar a cabo varias tareas. Por ejemplo, para generar nóminas mensuales, su sistema interno de cuentas debe compartir datos con el sistema bancario de su cliente para automatizar la facturación y comunicarse con una aplicación interna de planillas de horarios. Las API RESTful admiten este intercambio de información porque siguen estándares de comunicación de software seguros, confiables y eficientes.” ¿Qué es una API de RESTful? - Explicación de API de RESTful - AWS

<<https://aws.amazon.com/es/what-is/restful-api/>> [Consulta: 18 de mayo de 2023]

Backlog del producto: “es la lista principal del trabajo que debe realizar el propietario o el responsable del producto. Se trata de una lista dinámica de funciones,



requisitos, mejoras y correcciones que actúa como la base del backlog de sprint. Básicamente, se trata de la lista de "tareas que hay que hacer" del equipo. El propietario del producto está constantemente revisando, cambiando las prioridades y realizando el mantenimiento del backlog del producto, ya que, a medida que sabemos más o que cambia el mercado, es posible que haya elementos que ya no sean relevantes o que los problemas se solucionen de otras maneras." *Scrum: qué es, cómo funciona y por qué es excelente*

<<https://www.atlassian.com/es/agile/scrum>> [Consulta: 18 de mayo de 2023]

Backlog de sprint: "se trata de la lista de elementos, historias de usuario o correcciones de errores, seleccionadas por el equipo de desarrollo, para su implementación en el ciclo actual de sprint. Antes de cada sprint, en la reunión de planificación de sprint (que analizaremos más adelante en el artículo), el equipo elige los elementos en los que trabajará para el sprint del backlog del producto. El backlog de sprint puede ser flexible y puede evolucionar durante un sprint. No obstante, no se puede poner en peligro el objetivo fundamental del sprint, lo que el equipo quiere lograr con el sprint actual." *Scrum: qué es, cómo funciona y por qué es excelente*

<<https://www.atlassian.com/es/agile/scrum>> [Consulta: 18 de mayo de 2023]

Diagrama entidad relación: "Un diagrama entidad-relación, también conocido como modelo entidad relación o ERD, es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas ER se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación. También conocidos como los ERD o modelos ER, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos. Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos." *¿Qué es un diagrama entidad-relación? | Lucidchart*

<<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>>

[Consulta: 18 de mayo de 2023]

jQuery: "jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. De acuerdo a un análisis de la Web (realizado en 2017)

JQuery es la biblioteca de JavaScript más utilizada, por un amplio margen.” *jQuery - Wikipedia, la enciclopedia libre*

<<https://es.wikipedia.org/wiki/JQuery>> [Consulta: 18 de mayo de 2023]

Modelo vista controlador (MVC): “En líneas generales, MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.” *Qué es MVC*

<<https://desarrolloweb.com/articulos/que-es-mvc.html>> [Consulta: 18 de mayo de 2023]

Plantilla: “Laravel usa unos archivos que se llaman plantillas o templates que suelen ser nuestros archivos principales, que tienen los segmentos de código que se repiten en más de una vista, como por ejemplo la barra de navegación, un menú de opciones, la estructura del acomodo de nuestro proyecto, etc. y como deben de estar prácticamente presentes en todos lados, no tiene sentido estarlos repitiendo en todas las vistas. Por defecto Laravel contiene un template llamado app.blade.php, usualmente los templates contienen el head del HTML, las ligas del CSS del sistema y una sección exclusiva para los archivos Javascript.” *Capítulo 10. Vistas y motor de plantillas Blade · laravel-5*

<<https://richos.gitbooks.io/laravel-5/content/capitulos/chapter10.html>> [Consulta: 18 de mayo de 2023]

Scrum: “Scrum es un marco de gestión de proyectos de metodología ágil que ayuda a los equipos a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. Al igual que un equipo de rugby (de donde proviene su nombre) cuando entrena para un gran partido, scrum anima a los equipos a aprender a través de las experiencias, a autoorganizarse mientras aborda un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente.”

*Scrum: qué es, cómo funciona y por qué es excelente*

<<https://www.atlassian.com/es/agile/scrum>> [Consulta: 18 de mayo de 2023]

Scrum diario: “se trata de una reunión diaria de muy corta duración que tiene lugar siempre a la misma hora (normalmente, por las mañanas) y en el mismo sitio para simplificarla. Muchos equipos tratan de finalizar la reunión en 15 minutos, pero eso es solo una guía. Esta reunión también se denomina "reunión rápida diaria" y con ello se hace hincapié en que debe ser rápida. El objetivo del scrum diario es que todos los miembros del equipo estén en sintonía, se coordinen en torno al objetivo

del sprint y cuenten con un plan para las próximas 24 horas.” *Scrum: qué es, cómo funciona y por qué es excelente*

<<https://www.atlassian.com/es/agile/scrum>> [Consulta: 18 de mayo de 2023]

Sitemap: “Un Sitemap es un diagrama jerárquico que muestra la estructura de un sitio web, aplicación o producto. Los diseñadores UX y los arquitectos de información lo utilizan para agrupar contenido similar y así definir la taxonomía del sitio.” ▷ *¿Qué es un Sitemap en diseño UX?*

<<https://formiux.com/sitemaps-en-ux/>> [Consulta: 18 de mayo de 2023]

Sprint: “Un sprint es un período breve de tiempo fijo en el que un equipo de scrum trabaja para completar una cantidad de trabajo establecida. Los sprints se encuentran en el corazón de las metodologías scrum y ágil, y hacer bien los sprints ayudará a tu equipo ágil a lanzar mejor software con menos quebraderos de cabeza.” *Todo lo que necesitas saber sobre los sprints de scrum*

<<https://www.atlassian.com/es/agile/scrum/sprints>> [Consulta: 18 de mayo de 2023]

## 9. Anexos