**ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN DE SOCIOS DE FUNDACIÓN “FUHPED”**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TÉCNOLOGO EN DESARROLLO DE SOFTWARE**

**Cristian Andres Mañay Cevallos**

[cristian.manay@epn.edu.ec](mailto:cristian.manay@epn.edu.ec)

**DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO, MSC**

juan.zaldumbide@epn.edu.ec

**CODIRECTOR: ING. MONICA DE LOURDES VINUEZA RHOR, MSC**

monica.vinueza@epn.edu.ec

**Quito, octubre 2021**

**CERTIFICACION**

Certificamos que el presente trabajo fue desarrollado por el Sr. Mañay Cevallos Cristian Andrés como requerimiento parcial a la obtención del título de TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE, bajo nuestra supervisión:

|  |  |  |
| --- | --- | --- |
| **Ing. Juan Pablo Zaldumbide**  DIRECTOR(A) DEL PROYECTO |  | **Ing. Mónica Vinueza Rhor**  CODIRECTOR(A) DEL PROYECTO |

**DECLARACIÓN**

Yo Mañay Cevallos Cristian Andrés con CI: 1751864685 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin prejuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

|  |
| --- |
| **Cristian Andrés Mañay Cevallos** |

**DEDICATORIA**

Este proyecto está dedicado a toda mi familia, especialmente a mis padres, quienes me apoyaron en todo momento para completar cada uno de los objetivos que me he propuesto.

**CRISTIAN ANDRES MAÑAY CEVALLOS**

**AGRADECIMIENTO**

Quiero expresar un sincero agradecimiento, a mis padres por su comprensión y apoyo incondicional en todo momento para que pueda culminar esta etapa de mi vida con éxito.

A cada uno de los ingenieros que han compartido su conocimiento a lo largo de mi formación profesional.

**CRISTIAN ANDRES MAÑAY CEVALLOS**

**ÍNDICE DE CONTENIDOS**

[1 Introducción 1](#_Toc86012265)

[1.1 Objetivo general 1](#_Toc86012266)

[1.2 Objetivos específicos 1](#_Toc86012267)

[1.3 Alcance 2](#_Toc86012268)

[2 Metodología 3](#_Toc86012269)

[2.1 Metodología de Desarrollo 3](#_Toc86012270)

[2.1.1 Roles 3](#_Toc86012271)

[2.1.2 Artefactos 4](#_Toc86012272)

[2.2 Diseño de interfaces 5](#_Toc86012273)

[2.2.1 Herramientas utilizadas para el diseño 5](#_Toc86012274)

[2.3 Diseño de la arquitectura 6](#_Toc86012275)

[2.3.1 Patrón arquitectónico 6](#_Toc86012276)

[2.4 Herramientas de desarrollo 7](#_Toc86012277)

[3 Resultados y Discusión 8](#_Toc86012278)

[3.1 Sprint 0. Configuración del ambiente de desarrollo 8](#_Toc86012279)

[3.1.1 Diseño de la base de datos 9](#_Toc86012280)

[3.1.2 Configuración del entorno Backend 9](#_Toc86012281)

[3.1.3 Configuración del entorno Frontend 9](#_Toc86012282)

[3.1.4 Definición de usuarios 10](#_Toc86012283)

[3.2 Sprint 1. Implementación del módulo de autenticación - Implementación del módulo de noticias. 11](#_Toc86012284)

[3.2.1 Controlador de usuario 11](#_Toc86012285)

[3.2.2 Autenticación del usuario a la Aplicación Web 11](#_Toc86012286)

[3.2.3 Página principal 12](#_Toc86012287)

[3.2.4 Controlador de noticias 15](#_Toc86012288)

[3.2.5 Visualizar Noticias 15](#_Toc86012289)

[3.2.6 Publicar Noticias 15](#_Toc86012290)

[3.2.7 Eliminar Noticias 16](#_Toc86012291)

[3.3 Sprint 2. Implementación del módulo de gestión de socios - Implementación del módulo de gestión de afiliados. 17](#_Toc86012292)

[3.3.1 Controlador de socio 17](#_Toc86012293)

[3.3.2 Registrar usuario socio 17](#_Toc86012294)

[3.3.3 Visualizar lista de socios registrados 19](#_Toc86012295)

[3.3.4 Eliminar socios 19](#_Toc86012296)

[3.3.5 Controlador de afiliado 19](#_Toc86012297)

[3.3.6 Registrar usuario afiliado 20](#_Toc86012298)

[3.3.7 Visualizar lista de afiliados 21](#_Toc86012299)

[3.3.8 Eliminar afiliado 21](#_Toc86012300)

[3.4 Sprint 3. Implementación del módulo de gestión de servicios 21](#_Toc86012301)

[3.4.1 Actualizar información de usuario 22](#_Toc86012302)

[3.4.2 Controlador de servicio 22](#_Toc86012303)

[3.4.3 Registrar Servicio 22](#_Toc86012304)

[3.4.4 Visualizar lista de servicios 23](#_Toc86012305)

[3.4.5 Eliminar Servicio 24](#_Toc86012306)

[3.5 Sprint 4. Implementación del módulo de administración de citas. 24](#_Toc86012307)

[3.5.1 Controlador de citas 25](#_Toc86012308)

[3.5.2 Registrar cita 25](#_Toc86012309)

[3.5.3 Visualizar lista de citas 26](#_Toc86012310)

[3.5.4 Eliminar cita 27](#_Toc86012311)

[3.5.5 Actualizar cita 28](#_Toc86012312)

[3.6 Sprint 5. Implementación módulo de donaciones - Implementación módulo de reportes. 29](#_Toc86012313)

[3.6.1 Funcionalidad de donaciones 29](#_Toc86012314)

[3.6.2 Controlador de notificaciones 31](#_Toc86012315)

[3.6.3 Visualizar notificaciones 31](#_Toc86012316)

[3.6.4 Visualizar reportes de socios 31](#_Toc86012317)

[3.6.5 Visualizar reportes de afiliados 32](#_Toc86012318)

[3.6.6 Visualizar reportes de citas 33](#_Toc86012319)

[3.7 Sprint 6. Pruebas y despliegue de la Aplicación Web. 34](#_Toc86012320)

[3.7.1 Pruebas Unitarias 34](#_Toc86012321)

[3.7.2 Pruebas de compatibilidad 35](#_Toc86012322)

[3.7.3 Pruebas de rendimiento 35](#_Toc86012323)

[3.7.4 Despliegue de la Aplicación Web 37](#_Toc86012324)

[4 Conclusiones y Recomendaciones 39](#_Toc86012325)

[4.1 Conclusiones 39](#_Toc86012326)

[4.2 Recomendaciones 40](#_Toc86012327)

[5 Bibliografía 41](#_Toc86012328)

[6 ANEXOS i](#_Toc86012329)

[6.1 Manual Técnico i](#_Toc86012330)

[6.2 Manual de Usuario i](#_Toc86012331)

[6.3 Manual de Instalación i](#_Toc86012332)

**ÍNDICE DE FIGURAS**

[**Fig. 1:** Página Principal 6](#_Toc86012375)

[**Fig. 2:** Arquitectura de la Aplicación Web 7](#_Toc86012376)

[**Fig. 3:** Configuración Backend 9](#_Toc86012377)

[**Fig. 4:** Configuración Frontend 10](#_Toc86012378)

[**Fig. 5**: Usuarios de la Aplicación Web 10](#_Toc86012379)

[**Fig. 6:** Formulario de Inicio de Sesión 11](#_Toc86012380)

[**Fig. 7:** Validación Inicio de Sesión 12](#_Toc86012381)

[**Fig. 8:** Página Principal 12](#_Toc86012382)

[**Fig. 9:** Página Principal 13](#_Toc86012383)

[**Fig. 10:** Página Principal 14](#_Toc86012384)

[**Fig. 11:** Página Principal 14](#_Toc86012385)

[**Fig. 12:** Lista de Noticias 15](#_Toc86012386)

[**Fig. 13:** Publicación de Noticias 15](#_Toc86012387)

[**Fig. 14:** Validación Publicación de Noticias 16](#_Toc86012388)

[**Fig. 15:** Correcto ingreso de noticias 16](#_Toc86012389)

[**Fig. 16:** Mensaje eliminación de noticia 16](#_Toc86012390)

[**Fig. 17:** Mensaje de éxito al eliminar una noticia 17](#_Toc86012391)

[**Fig. 18:** Formulario de registro de socio 18](#_Toc86012392)

[**Fig. 19:** Validación registro de socio 18](#_Toc86012393)

[**Fig. 20:** Lista de socios 19](#_Toc86012394)

[**Fig. 21:** Eliminar Socio 19](#_Toc86012395)

[**Fig. 22:** Registro de afiliado 20](#_Toc86012396)

[**Fig. 23:** Validación registro de afiliado 20](#_Toc86012397)

[**Fig. 24:** Lista de afiliados 21](#_Toc86012398)

[**Fig. 25:** Eliminar afiliado 21](#_Toc86012399)

[**Fig. 26:** Perfil de usuario 22](#_Toc86012400)

[**Fig. 27:** Formulario de servicios 23](#_Toc86012401)

[**Fig. 28:** Validación de registro de servicio 23](#_Toc86012402)

[**Fig. 29:** Lista de Servicios 24](#_Toc86012403)

[**Fig. 30:** Eliminar servicio 24](#_Toc86012404)

[**Fig. 31:** Formulario de registro de cita 25](#_Toc86012405)

[**Fig. 32:** Validación de registro de cita 26](#_Toc86012406)

[**Fig. 33:** Lista de citas vista por un socio 26](#_Toc86012407)

[**Fig. 34:** Lista de citas vista por un afiliado 27](#_Toc86012408)

[**Fig. 35:** Lista de citas vista por el administrador 27](#_Toc86012409)

[**Fig. 36:** Eliminar cita 28](#_Toc86012410)

[**Fig. 37:** Actualizar cita 28](#_Toc86012411)

[**Fig. 38:** Donaciones 29](#_Toc86012412)

[**Fig. 39:** Donación con PayPal 30](#_Toc86012413)

[**Fig. 40:** Donación con tarjeta de crédito 30](#_Toc86012414)

[**Fig. 41:** Notificaciones 31](#_Toc86012415)

[**Fig. 42:** Descargar reporte de Socios 31](#_Toc86012416)

[**Fig. 43:** Reporte en PDF de Socios 32](#_Toc86012417)

[**Fig. 44:** Descargar reporte de afiliados 32](#_Toc86012418)

[**Fig. 45:** Reporte en PDF de afiliados 33](#_Toc86012419)

[**Fig. 46:** Descargar reporte de citas 33](#_Toc86012420)

[**Fig. 47:** Reporte en PDF de citas 34](#_Toc86012421)

[**Fig. 48:** Prueba Unitaria: Registrar citas 35](#_Toc86012422)

[**Fig. 49:** Resultados de pruebas de carga al API 36](#_Toc86012423)

[**Fig. 50:** Prueba de Carga a la Aplicación Web 36](#_Toc86012424)

[**Fig. 51:** Resultado de pruebas de estrés al API 37](#_Toc86012425)

[**Fig. 52:** Prueba de estrés a la Aplicación Web 37](#_Toc86012426)

[**Fig. 53:** Configuración final de la Aplicación Web en Vercel 38](#_Toc86012427)

[**Fig. 54:** Despliegue de la Aplicación Web en Vercel 38](#_Toc86012428)

**ÍNDICE DE TABLAS**

[**TABLA I**: Equipo Scrum 4](#_Toc88204148)

[**TABLA II**: Historia de Usuario 4](#_Toc88204149)

[**TABLA III**: Herramientas para el desarrollo 7](#_Toc88204150)

[**TABLA IV:** Librerías usadas en el desarrollo de la aplicación 8](#_Toc88204151)

[**TABLA V**: Navegadores Compatibles con la Aplicación Web 35](#_Toc88204152)

**RESUMEN**

En los últimos años el internet se ha vuelto una herramienta de búsqueda de información indispensable en la vida cotidiana de las personas por tal motivo el presente proyecto se ha desarrollado con el fin de proporcionar una Aplicación Web para gestionar socios y afiliados a la fundación “FUHPED”, permitiendo agilizar procesos internos de la fundación, así también brindando mayor visibilidad en internet.

En la sección I, se especifica el motivo del desarrollo del proyecto, así como también se expone el objetivo general, los objetivos específicos y se determina el alcance del proyecto.

En la sección II, se especifica la metodología Scrum y los roles establecidos en el desarrollo del proyecto, asimismo se detallan los diseños de las interfaces y herramientas escogidas para la elaboración de la Aplicación Web.

En la sección III, se detalla el desarrollo de cada módulo de la Aplicación Web, asimismo se exponen los resultados obtenidos en las pruebas unitarias, de compatibilidad y de rendimiento.

Finalmente, en la sección IV, se explican las conclusiones y recomendaciones obtenidas en el desarrollo del presente proyecto

**PALABRAS CLAVE:** Scrum, Laravel, React, Aplicación Web

**ABSTRACT**

In recent years the internet has become an indispensable information search tool in people's daily lives, for this reason this project has been developed to provide a web application to manage partners and affiliates of the “FUHPED” foundation, allowing to streamline internal processes of the foundation, therefore also providing greater visibility on the internet.

In section I, the reason for the development of the project is specified, as well as the general objective, the specific objectives and the scope of the project is determined.

In section II, the Scrum methodology and the roles established in the development of the project are specified, as well as the designs of the interfaces and tools chosen for the development of the web application.

In section III, the development of each module of the web application is detailed, as well as the results obtained in the unit, compatibility, and performance tests.

Finally, in section IV, the conclusions and recommendations obtained in the development of this project are explained.

**KEYWORDS:** Scrum, Laravel, React, Web Application

# Introducción

Las nuevas herramientas tecnológicas y la aparición del internet han contribuido para mejorar en gran medida el desarrollo de las actividades diarias de la humanidad [1] [2]. Uno de los sucesos más importantes en la actualidad es el crecimiento de internet y la repercusión que lo conlleva a ser indispensable en la búsqueda de información [1] [3]. Este constante crecimiento ha convertido a internet en la principal herramienta colaborativa, por lo que es utilizado por estudiantes, profesores, médicos y científicos para interactuar entre sí e intercambiar información relevante [1]. En la actualidad el internet es sin duda alguna un mecanismo que permite solucionar necesidades de la vida cotidiana [2][4]. Esta manera de obtener información ha generado que organizaciones opten por un cambio de metodología para darse a conocer al público [2].

La Fundación Halcones por el deporte, es una institución dedicada a fomentar la actividad física en niños y adolescentes, también contribuye en la enseñanza de varios deportes, sin embargo, no cuenta con la visibilidad necesaria para llegar a más personas, además de realizar sus actividades de forma manual ya que no cuentan con un sistema informático que permita automatizar el proceso de registro de socios, citas y la captación de donaciones a través de internet, esto genera varias complicaciones durante el desarrollo de dichas actividades. El uso de plataformas informáticas permite agilizar la gestión de nuevos usuarios y facilita que la información llegue de mejor manera al público [4]. Por tal motivo, el presente proyecto se centra en el desarrollo de una Aplicación Web para gestionar socios y afiliados a la fundación con el objetivo de proporcionar a la fundación mayor visibilidad en internet, además de proveer a la administración de la fundación una Aplicación Web capaz de automatizar procesos que actualmente se los realiza de forma manual.

## Objetivo general

Desarrollar una Aplicación Web para la gestión de socios de fundación “FUHPED”.

## Objetivos específicos

* Levantar requerimientos mediante reuniones con el cliente.
* Diseñar la base de datos y arquitectura en base a la información obtenida.
* Diseñar interfaces de la aplicación.
* Implementar los módulos del sistema de gestión y los módulos del aplicativo web.
* Probar la aplicación.

## Alcance

En la actualidad el uso de aplicaciones web por parte de fundaciones sin fines de lucro es un factor fundamental que facilita el acercamiento con el público; además permite automatizar y controlar procesos que se realizan de forma manual.

Por lo tanto, por medio de esta Aplicación Web se puede dar a conocer las actividades realizadas por parte de la fundación FUHPED; además de gestionar los procesos internos realizados por la administración de la fundación.

A la vez, la Aplicación Web garantiza el acceso a la información por medio de un módulo de autenticación que cuenta con perfiles de administrador, socio y afiliado, los cuales se detallan a continuación:

**La Aplicación Web permite a los usuarios con perfil de administrador:**

* Registro y eliminación de noticias.
* Registro, actualización y eliminación de socios.
* Registro, actualización y eliminación de afiliados.

**La Aplicación Web permite a los usuarios con perfil de socio:**

* Registro, actualización y eliminación de servicios prestados.
* Registro, actualización y eliminación de citas.

**La Aplicación Web permite a los usuarios con perfil de afiliado:**

* Revisar citas con los socios.

# Metodología

A continuación, se detalla la metodología que se ha utilizado en el presente proyecto.

## Metodología de Desarrollo

Scrum es una metodología ágil de desarrollo la cual se basa en la creación de ciclos cortos denominados “*Sprint*” [5] [6]. Al inicio de cada *sprint* se realiza una pequeña reunión para definir las actividades a realizarse durante la semana y al final del sprint se presentará un pequeño avance al cliente con el fin de satisfacer dudas con respecto al desarrollo del proyecto [7]. Esta metodología es de fácil implementación en el desarrollo de software, además de ser de fácil adaptabilidad frente a cambios en los requerimientos por parte del cliente.

### Roles

Los roles son sumamente importantes dentro de la metodología *Scrum*, ya que establecen la responsabilidad de cada miembro del equipo [5].

**Propietario del Producto (*Product Owner*)**

La persona con este rol es la responsable de tomar las decisiones del negocio, además es la encargada de ser el mediador entre el cliente y el equipo de desarrollo. Es la única persona a cargo de gestionar el *Product Backlog* [5].

Este rol está representado por parte del Directivo de FUHPED, la cual, conoce los procesos que se llevan a cabo dentro de la fundación.

**Maestro Scrum (*Scrum Master*)**

La persona con este rol es la responsable de verificar que el modelo y el marco de trabajo funciona correctamente. Se encarga de mantener la visión del equipo de desarrollo en los objetivos planteados al inicio de cada reunión [5].

**Equipo de desarrollo (*Developer Team*)**

Las personas con este rol son las encargadas de organizar y tomar decisiones para alcanzar el objetivo planteado. Por lo general es un equipo de 5 a 9 personas [5].

Este rol está representado por el desarrollador encargado del proyecto integrador, el cual cuenta con los conocimientos necesarios para alcanzar el objetivo del proyecto.

El equipo Scrum está conformado como se presenta en la T*ABLA* I.

**TABLA I**: Equipo Scrum

|  |  |
| --- | --- |
| **Nombre** | **Rol** |
| Representante de FUHPED (Christian Chávez) | *Product Owner* |
| Ing. Juan Pablo Zaldumbide | *Scrum Master* |
| Cristian Mañay | Equipo de desarrollo |

### Artefactos

Los artefactos son elementos que se aseguran que todo el equipo tenga claridad y transparencia de la información dentro del proyecto.

**Recopilación de Requerimientos**

El levantamiento de requerimientos es una etapa fundamental para el desarrollo del producto, ya que en esta se detallan los requisitos por parte del cliente. La lista de requisitos especifica las características y funcionalidades del producto [7].

La obtención de requerimientos se ha realizado mediante reuniones con el dirigente de la fundación FUHPED y el desarrollador. En cada una de las reuniones se ha recolectado información que ayuda a definir la funcionalidad de la Aplicación Web.

**Historias de Usuario**

Las historias de usuario comprenden las funcionalidades que va a tener el producto, las cuales son consecuencia de las reuniones entre el cliente y el equipo [5]. Se redactan en forma de tarjeta empleando lenguaje natural evitando términos técnicos, de esta manera los requerimientos quedan claros tanto para el cliente como para el equipo [6].

Las historias de usuario se han elaborado a partir de los requerimientos recolectados y se plasmaron en forma de tarjeta, un ejemplo se detallada en la T*ABLA* II. Las 26 Historias de Usuario restantes se encuentran detalladas en el Manual Técnico – Sección Historias de Usuario.

**TABLA II**: Historia de Usuario

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | |
| **Identificador (ID):** HU001 | **Usuario:** Administrador |
| **Nombre de la historia:** Iniciar Sesión | |
| **Prioridad en negocio:** Alta | **Riesgo en desarrollo:** Alto |
| **Iteración Asignada:** 1 | |
| **Responsable:** Cristian Mañay | |
| **Descripción:**  La Aplicación Web permite al usuario iniciar sesión con las credenciales correctas como son correo y contraseña. | |
| **Observación:**  La Aplicación Web cuenta con 3 perfiles de usuario, los cuales serán seleccionados correspondiendo el usuario requerido. | |

***Product Backlog***

Es una lista ordenada con todas las necesidades del cliente, la cual cuenta con cada uno de los requisitos, características, funcionalidades y cambios a realizar a futuro en el producto [5].

***Sprint* *Backlog***

Es una lista de tareas pendientes a realizar en cada *sprint*, las cuales se desarrollan en un intervalo de tiempo con el propósito de alcanzar la meta establecida en cada *sprint*. Cada tarea es asignada a un integrante del equipo, además se le asigna un tiempo para realizarla [5].

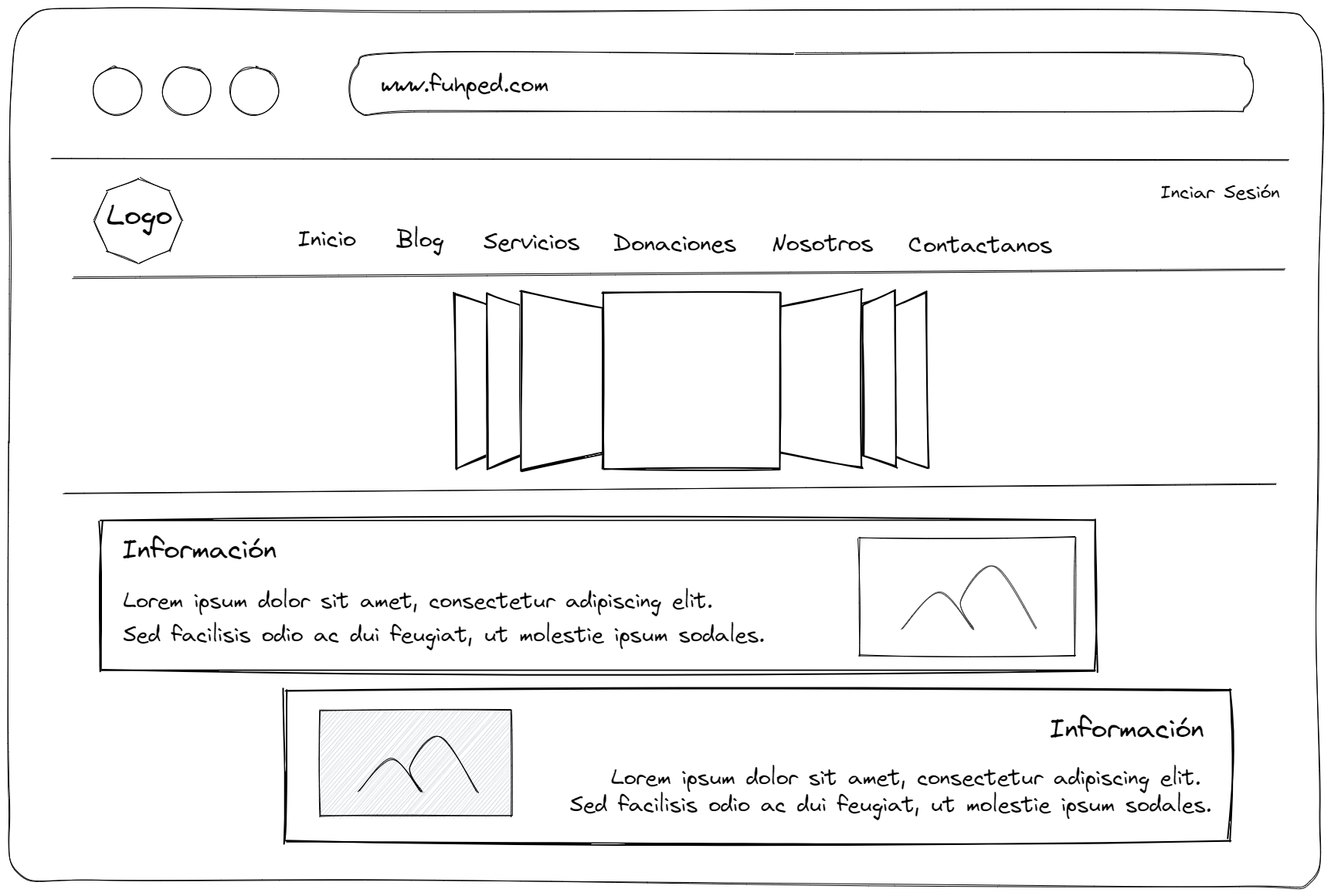
## Diseño de interfaces

Una vez establecidos los requerimientos se ha procedido a elegir la herramienta de diseño y posterior a diseñar las interfaces (*mockups*) de la Aplicación Web.

### Herramientas utilizadas para el diseño

La herramienta elegida para el diseño de las interfaces de la Aplicación Web es *excalidraw*, la cual es una herramienta similar a una pizarra que permite realizar diagramas y dibujos a mano en el navegador. Además, permite agregar plantillas que facilitan el diseño de interfaces, también permite el trabajo colaborativo con varias personas [8].

La **Fig. 1** muestra el diseño de la página principal de la Aplicación Web, mientras que las 14 interfaces diseñadas se encuentran en el Manual Técnico – Sección Diseño de interfaces.



**Fig. 1:** Página Principal

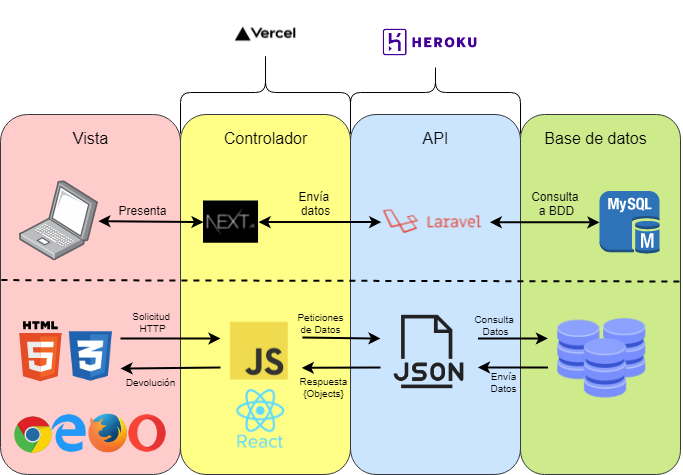
## Diseño de la arquitectura

A continuación, se detalla el patrón arquitectónico que ha sido utilizado en la Aplicación Web.

### Patrón arquitectónico

El Modelo Vista Controlador (MVC) tiene como objetivo reducir la carga de programación en los sistemas múltiples y sincronizados permitiendo la incorporación de cada componente por separado y de esta forma se garantiza la actualización del software de forma sencilla [9].

El diseño de la arquitectura facilita el desarrollo de la Aplicación Web, permitiendo organizar de mejor manera las funcionalidades, además de mejorar el tiempo de desarrollo entre cada actividad. La **Fig*.* *2*** muestra la arquitectura y herramientas a ser utilizadas en la Aplicación Web.



**Fig. 2:** Arquitectura de la Aplicación Web

## Herramientas de desarrollo

Las herramientas detalladas en la T*ABLA* III se establecieron una vez determinados los requerimientos y el patrón arquitectónico de la Aplicación Web.

**TABLA III**: Herramientas para el desarrollo

|  |  |
| --- | --- |
| **Herramienta** | **Justificación** |
| **Laravel** | Es un marco progresivo de trabajo creado para el desarrollo de aplicaciones web en leguaje PHP. Su implementación proporciona una estructura y un punto de partida para empezar con el desarrollo, además, permite crear aplicaciones web modernas y completas [10]. |
| **MySQL** | Es un sistema de gestión de base de datos relacional de código abierto basada en una arquitectura cliente servidor. Su implementación facilita la consulta, registro y actualización de datos [11]. |
| **Next.js** | Es un marco de trabajo construido sobre React. Su implementación permite realizar server rendering de aplicaciones desarrolladas en JavaScript, además facilita el despliegue de aplicaciones en producción [12]. |
| **Composer** | Es un gestor de dependencias para desarrollar aplicaciones en PHP. Su implementación permite mantener actualizados los paquetes del proyecto [13]. |
| **Postman** | Es una herramienta que permite realizar pruebas de una API REST. Su implementación proporciona la posibilidad de realizar pruebas automatizadas, monitorizarlas y documentarlas [14]. |
| **XAMPP** | Es una herramienta que contiene software libre (Apache, MySQL, PHP y Perl). Su implementación permite levantar un servidor local para establecer la conexión con la base de datos [15]. |

**Librerías**

A continuación, en la T*ABLA* IV se muestran las librerías usadas en el desarrollo de la Aplicación Web.

**TABLA IV:** Librerías usadas en el desarrollo de la Aplicación Web

|  |  |
| --- | --- |
| **Librería** | **Descripción** |
| **React** | Librería de JavaScript dedicada a crear fácilmente interfaces de usuario de forma interactiva. Se encarga de renderizar y actualizar componentes tomando en cuenta su estado y método de renderizado [16]. |

# 

# Resultados y Discusión

A continuación, se detallan los resultados conseguidos en cada uno de los siete Sprint definidos y las pruebas realizadas a la Aplicación Web.

## *Sprint* 0. Configuración del ambiente de desarrollo

De acuerdo con la planificación establecida en el *Sprint* *Backlog*. El *Sprint* 0 contiene las configuraciones en el entorno de desarrollo. A continuación, se muestran las actividades realizadas en este *Sprint*.

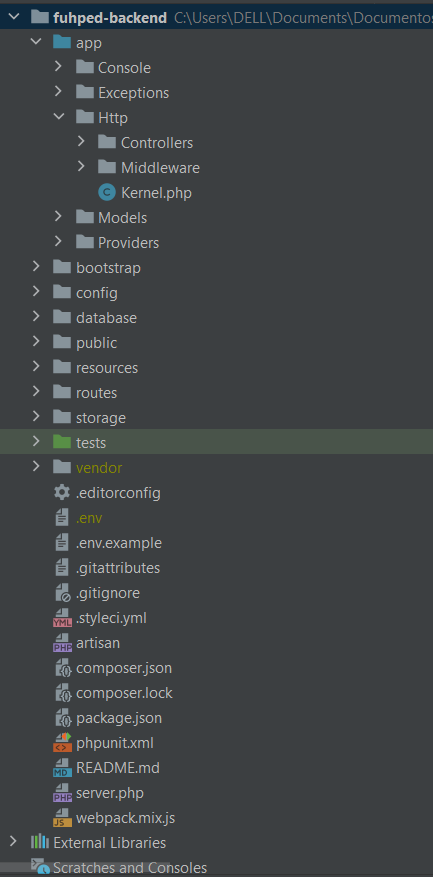
* Diseño de la base de datos
* Configuración del entorno *Backend*
* Configuración del entorno *Frontend*
* Definición de usuarios

### Diseño de la base de datos

De acuerdo con los requerimientos se ha seleccionado una base de datos relacional para ser implementada en la Aplicación Web con el SGBD MySQL. El diseño de la base de datos se encuentra detallada en el Manual Técnico – Sección Diseño de la Base de Datos (pág. 24).

### Configuración del entorno *Backend*

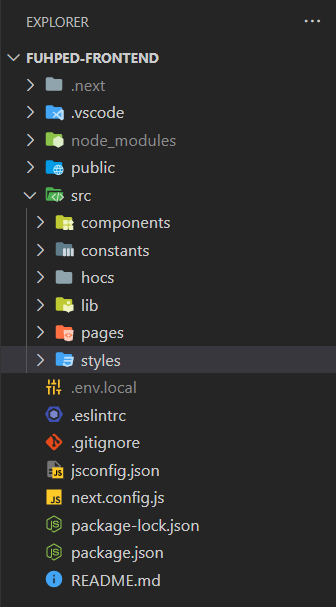
Para la configuración del entorno *Backend* se instala las herramientas de XAMPP, PHPStorm, Composer, WebStorm y Postman. A partir de instalar las herramientas se procede a crear el proyecto como se muestra en la **Fig. 3.**



**Fig. 3:** Configuración Backend

### Configuración del entorno *Frontend*

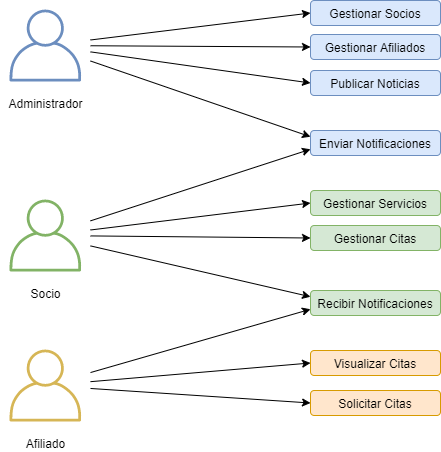
Para la configuración del entorno *Frontend* se instala las herramientas Node Js, npm y Visual Studio Code. A partir de la instalación de las herramientas se procede a crear el proyecto y configurar la estructura de directorios como se muestra en la **Fig. 4.**



**Fig. 4:** Configuración Frontend

### Definición de usuarios

En la **Fig. 5** se muestran los roles de usuarios que pueden autenticarse en la Aplicación Web, además se muestran los módulos a los que tienen acceso según corresponda el rol asignado.



**Fig. 5**: Usuarios de la Aplicación Web

## *Sprint* 1. Implementación del módulo de autenticación - Implementación del módulo de noticias.

De acuerdo con la planificación establecida en el *Sprint* *Backlog*. El *Sprint* 1 contiene las siguientes actividades.

* Controlador de usuario
* Autenticación del usuario en la Aplicación Web
* Página principal
* Controlador de noticias
* Visualizar noticias
* Publicar Noticias
* Eliminar noticias

### Controlador de usuario

En el controlador del usuario se crea las funciones de autenticación, registro y actualización del usuario, además se han establecido validaciones en el caso que el usuario ingrese información incorrecta, si esto sucede la aplicación retornará un mensaje con detalles del error.

### Autenticación del usuario a la Aplicación Web

La Aplicación Web puede ser accedida por tres usuarios administrador, socio y afiliado, los cuales pueden autenticarse por medio de un formulario, como se indica en la **Fig. 6**, en el cual se debe ingresar las credenciales correctas para ingresar a la aplicación, en el caso de ingresar mal las credenciales se muestra un mensaje marcando los datos incorrectos, esto se muestra en la **Fig. 7** Si el usuario ingresa las credenciales correctas muestra el nombre su nombre en la barra de menú y redirige a la página principal.



**Fig. 6:** Formulario de Inicio de Sesión

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fig. 7:** Validación Inicio de Sesión

### Página principal

De la **Fig. 8** a la **Fig. 11** muestran la página principal de la aplicación, en la que se detalla información relevante de la fundación. Además, se presenta un formulario de contacto en el que cualquier usuario va a poder llenarlo y enviar al correo de la fundación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fig. 8:** Página Principal

Interfaz de usuario gráfica, Aplicación

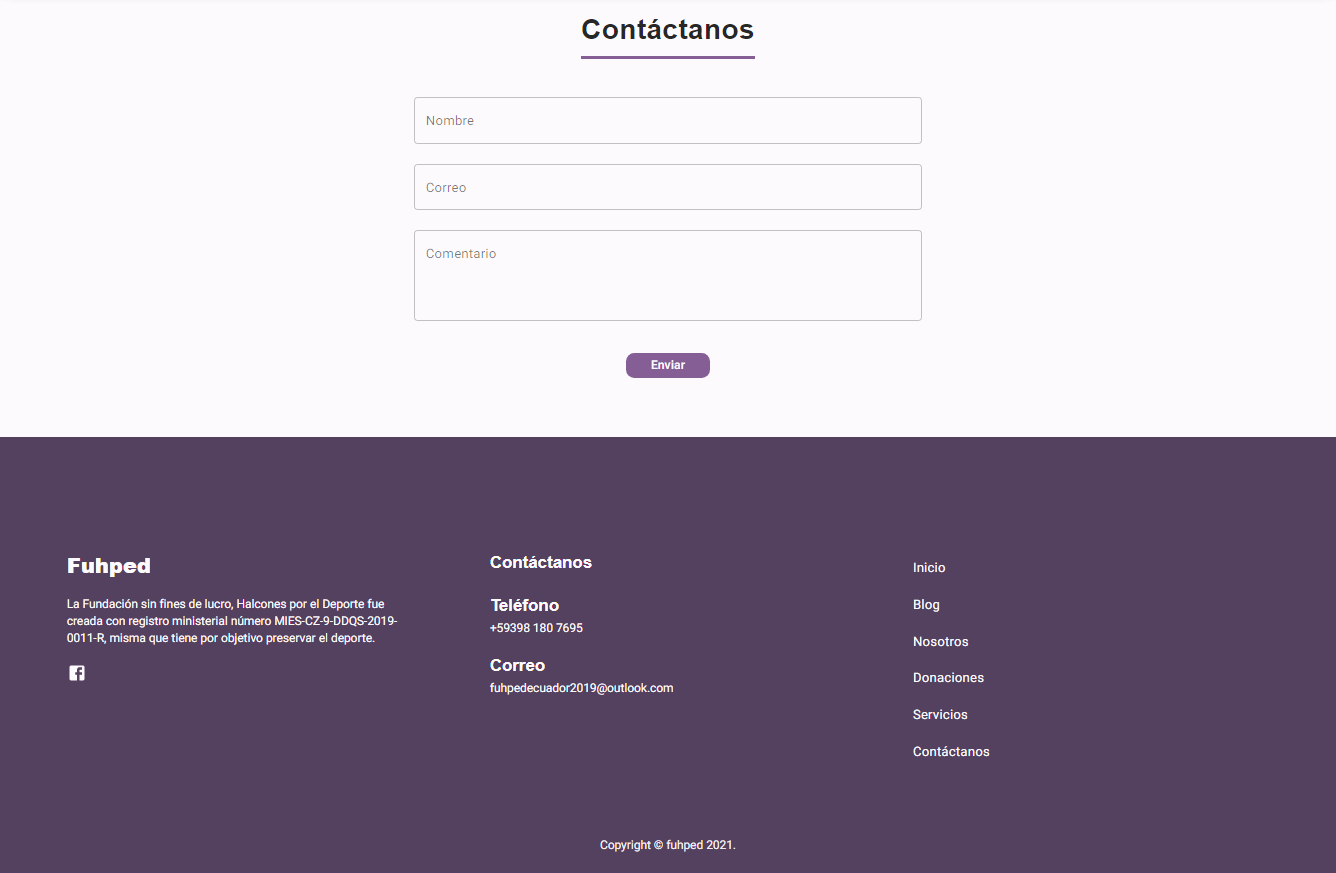
Descripción generada automáticamente

**Fig. 9:** Página Principal

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

**Fig. 10:** Página Principal



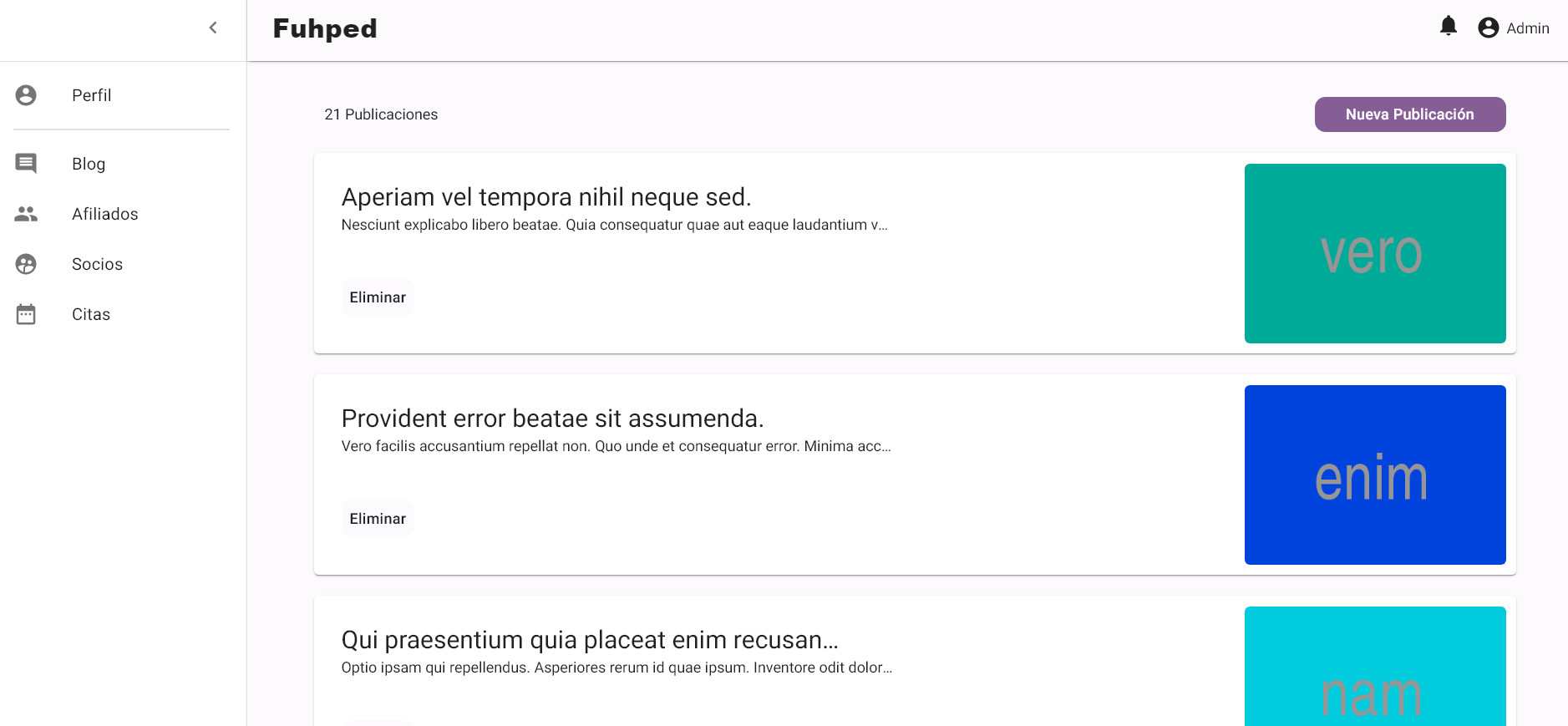
**Fig. 11:** Página Principal

### Controlador de noticias

En el controlador de noticias se crean las funciones de visualizar y eliminar noticias, además se han establecido validaciones que retornan un mensaje con el detalle del error en el caso que el usuario ingrese información incorrecta.

### Visualizar Noticias

En la **Fig. 12** se muestra una lista de todas las noticias publicadas en forma de tarjetas con título, descripción y una imagen. Además, se muestra un botón de eliminar.



**Fig. 12:** Lista de Noticias

### Publicar Noticias

La **Fig. 13** muestra el formulario en el cual el usuario administrador puede ingresar datos de la publicación. Si el usuario ingresa mal los datos se muestra un mensaje de error marcando el dato incorrecto como se ilustra en la **Fig. 14.** Si el usuario ingresa una imagen el botón cambia a color verde con un icono de visto como se ilustra en la **Fig. 15.** Una vez guardada la publicación se puede ver en la lista de noticias.



**Fig. 13:** Publicación de Noticias



**Fig. 14:** Validación Publicación de Noticias



**Fig. 15:** Correcto ingreso de noticias

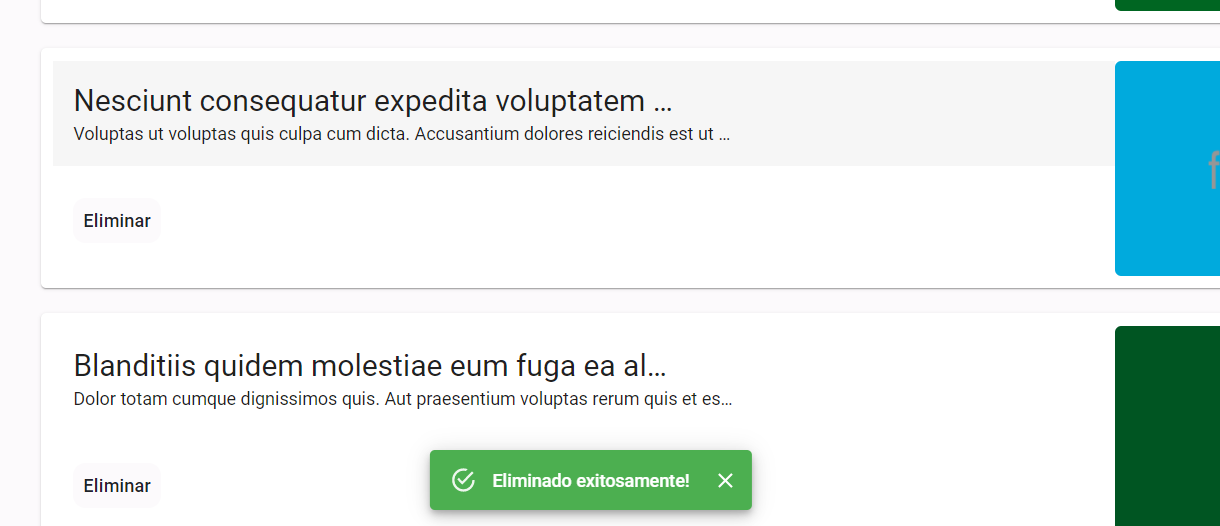
### Eliminar Noticias

La **Fig. 16** muestra un mensaje de confirmación que se genera cuando el usuario administrador presiona el botón eliminar, al momento de eliminar una noticia se muestra un mensaje de éxito al eliminar la noticia como se ilustra en la **Fig. 17**

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**Fig. 16:** Mensaje eliminación de noticia



**Fig. 17:** Mensaje de éxito al eliminar una noticia

## *Sprint* 2. Implementación del módulo de gestión de socios - Implementación del módulo de gestión de afiliados.

De acuerdo con la planificación establecida en el *Sprint* *Backlog*. El *Sprint* 2 contiene las siguientes actividades.

* Controlador de socio
* Registrar usuario socio
* Visualizar lista de socios registrados
* Eliminar socio
* Controlador de afiliado
* Registrar usuario afiliado
* Visualizar lista de afiliados
* Eliminar afiliados

### Controlador de socio

En el controlador de socios se crean las funciones de visualizar y actualizar socio, además se han establecido validaciones, en el caso que el usuario ingrese información incorrecta el API retorna un mensaje con el error.

### Registrar usuario socio

La **Fig. 18** se muestra el formulario en el que el usuario administrador puede ingresar información del socio la cual es validada por la aplicación para posterior ser almacenada en la base de datos. Si el usuario ingresa información incorrecta se muestra un mensaje marcando los datos incorrectos como se ilustra en la **Fig. 19**. Una vez registrado el socio puede iniciar sesión en la aplicación.

Interfaz de usuario gráfica

Descripción generada automáticamente

**Fig. 18:** Formulario de registro de socio

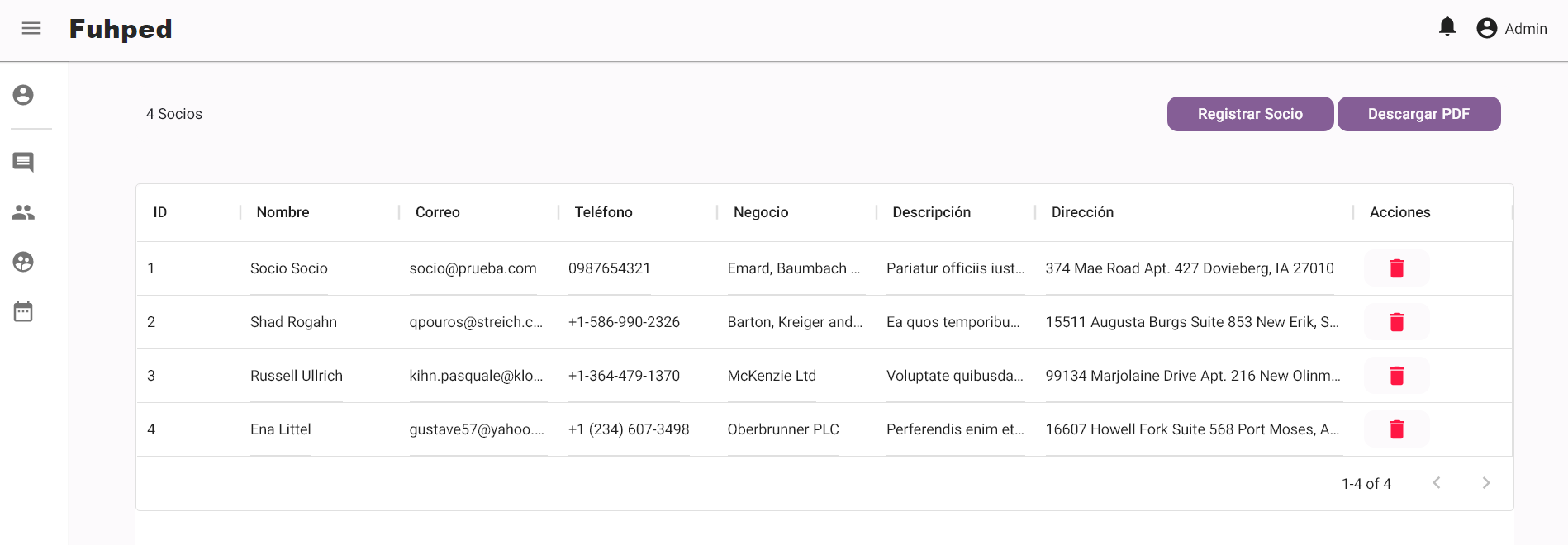
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fig. 19:** Validación registro de socio

### Visualizar lista de socios registrados

La **Fig. 20** muestra una tabla con información detallada de todos los socios registrados, además se muestra un botón de eliminar, el cual solo puede ser ejecutado por el usuario administrador.



**Fig. 20:** Lista de socios

### Eliminar socios

En la **Fig. 21** se muestra un mensaje de confirmación al momento de que el usuario elimine un socio, una vez eliminado se muestra un mensaje de éxito y se verá reflejado el cambio en la tabla.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 21:** Eliminar Socio

### Controlador de afiliado

En el controlador se han creado las funciones de visualizar y actualizar afiliado, además se han establecido validaciones para evitar que el usuario ingrese información incorrecta, si esto sucede el API retorna un mensaje con el error que se genera.

### Registrar usuario afiliado

En la **Fig. *22*** se muestra el formulario de registro de afiliado en el que el usuario administrador puede registrar la información de un nuevo afiliado, la cual es validada por la aplicación para posterior ser almacenada en la base de datos. Si el usuario ingresa información incorrecta se muestra un mensaje marcando los datos incorrectos como se ilustra en la **Fig. *23*.**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 22:** Registro de afiliado

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fig. 23:** Validación registro de afiliado

### Visualizar lista de afiliados

En la **Fig. 24** se muestra una tabla con información detallada de todos los afiliados registrados, además se muestra un botón de eliminar, el cual solo el usuario administrador está autorizado a ejecutarlo.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 24:** Lista de afiliados

### Eliminar afiliado

En la **Fig. 25** se muestra un mensaje de confirmación al momento de que el usuario elimine un afiliado, una vez eliminado se muestra un mensaje de éxito y se ve reflejado el cambio en la tabla.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 25:** Eliminar afiliado

## *Sprint* 3. Implementación del módulo de gestión de servicios

De acuerdo con la planificación establecida en el *Sprint* *Backlog*. El *Sprint* 3 contiene las siguientes actividades.

* Actualizar información de usuario
* Controlador de servicio
* Registrar servicio
* Visualizar lista de servicios
* Eliminar servicios

### Actualizar información de usuario

En la **Fig. 26** se muestra el formulario de perfil de usuario con información del usuario autenticado, además se muestra un botón de guardar, el usuario puede actualizar su información y guardarla.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

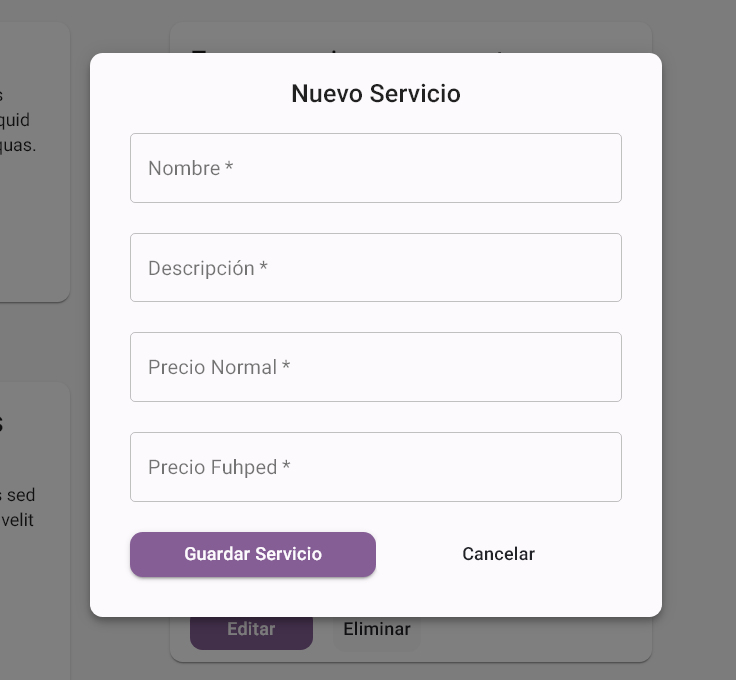
**Fig. 26:** Perfil de usuario

### Controlador de servicio

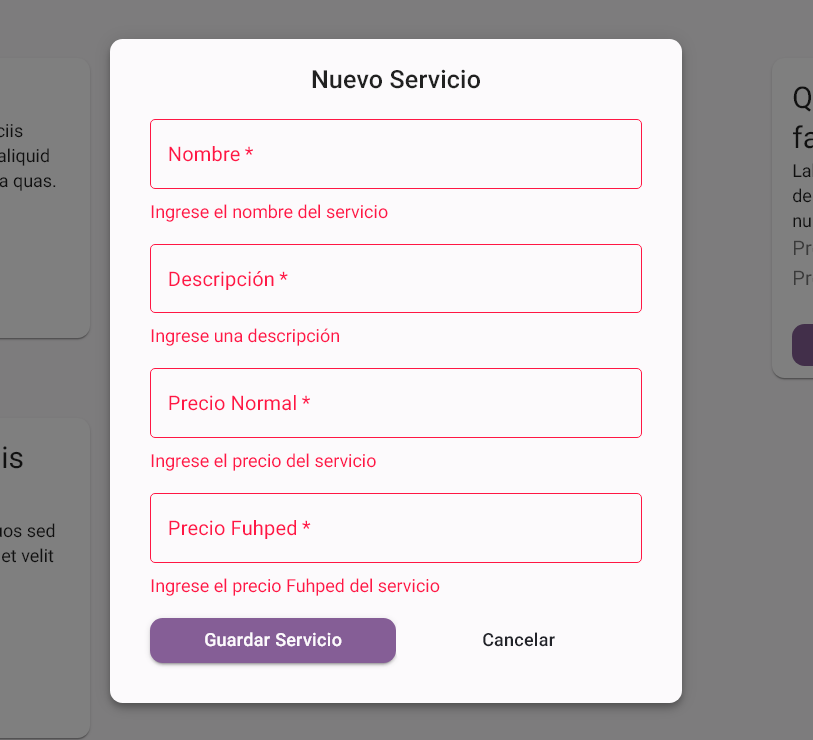
En el controlador se han creado las funciones de registrar, visualizar y actualizar servicio, además se han establecido validaciones para evitar que el usuario ingrese información incorrecta, si esto sucede el API retorna un mensaje con el error que se genera.

### Registrar Servicio

En la **Fig. 27** se muestra el formulario de registro de servicios en el que el usuario socio puede registrar la información de un nuevo servicio, el cual es validado por la aplicación para posterior ser almacenada en la base de datos. Si el usuario ingresa información incorrecta se muestra un mensaje marcando los datos incorrectos como se ilustra en la **Fig. 28.**



**Fig. 27:** Formulario de servicios



**Fig. 28:** Validación de registro de servicio

### Visualizar lista de servicios

En la **Fig. 29** se muestra la lista de servicios de cada socio en forma de carta con información de cada servicio.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fig. 29:** Lista de Servicios

### Eliminar Servicio

En la **Fig. 30** se muestra un mensaje de confirmación al momento de que el usuario elimine un servicio, una vez eliminado se muestra un mensaje de éxito y se ve reflejado el cambio en la lista de servicios.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fig. 30:** Eliminar servicio

## *Sprint* 4. Implementación del módulo de administración de citas.

De acuerdo con la planificación establecida en el *Sprint* *Backlog*. El *Sprint* 6 contiene las siguientes actividades.

* Controlador de citas
* Registrar cita
* Visualizar lista de citas
* Eliminar cita
* Actualizar cita

### Controlador de citas

En el controlador se han creado las funciones de registrar, visualizar y actualizar cita, además se han establecido validaciones para evitar que el usuario ingrese información incorrecta, si esto sucede el API retorna un mensaje con el error que se genera.

### Registrar cita

En la **Fig. 31** se muestra el formulario de registro de cita en el que el usuario socio puede registrar la información de una nueva cita, la cual es validada por la aplicación para posterior ser almacenada en la base de datos. Si el usuario ingresa información incorrecta se muestra un mensaje marcando los datos incorrectos como se ilustra en la **Fig. 32.**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Fig. 31:** Formulario de registro de cita



**Fig. 32:** Validación de registro de cita

### Visualizar lista de citas

En la **Fig. 33** se muestra una tabla con la lista de citas que tiene un socio. En la **Fig. 34** se muestra la lista de citas que tiene un afiliado, esta puede ser vista desde el perfil de afiliado. En la **Fig. 35** se muestra la lista de citas que tienen todos los socios, esta puede ser vista solo por el usuario administrador.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 33:** Lista de citas vista por un socio

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 34:** Lista de citas vista por un afiliado

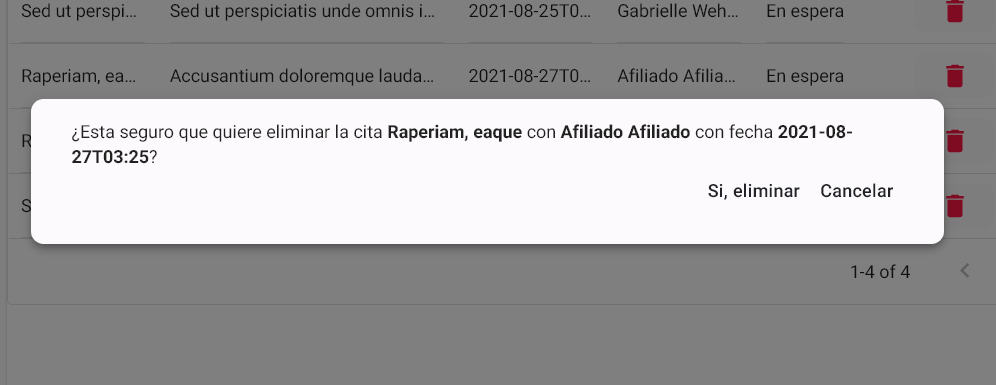
Tabla

Descripción generada automáticamente

**Fig. 35:** Lista de citas vista por el administrador

### Eliminar cita

En la **Fig. 36** se muestra un mensaje de confirmación al momento de que el usuario elimine una cita, una vez eliminado se muestra un mensaje de éxito y se ve reflejado el cambio en la tabla de citas.



**Fig. 36:** Eliminar cita

### Actualizar cita

En la **Fig. 37** se muestra un formulario con información de la cita seleccionada, el usuario socio puede actualizar la información de la cita y guardarla.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Fig. 37:** Actualizar cita

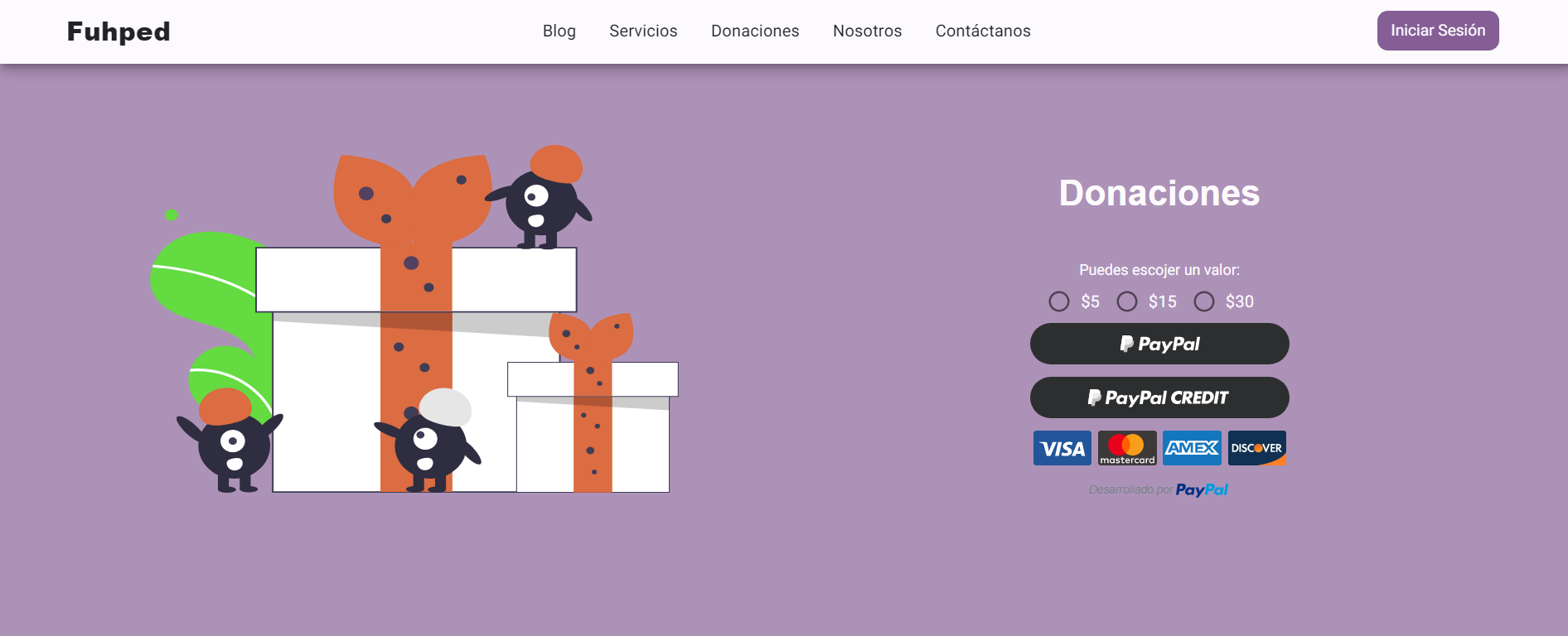
## *Sprint* 5. Implementación módulo de donaciones - Implementación módulo de reportes.

De acuerdo con la planificación establecida en el *Sprint* *Backlog*. El *Sprint* 5 contiene las siguientes actividades.

* Funcionalidad de donaciones
* Controlador de notificaciones
* Visualizar notificaciones
* Visualizar reporte de socios
* Visualizar reporte de afiliados
* Visualizar reporte de citas

### Funcionalidad de donaciones

En la **Fig. 38** se muestra la interfaz de donaciones ubicada en la página principal de la Aplicación Web, cualquier visitante de la página puede elegir el valor e iniciar sesión en PayPal para donar como se muestra en la **Fig. 39,** también se puede elegir la opción de tarjeta de crédito para donar como se muestra en la **Fig. 40.**

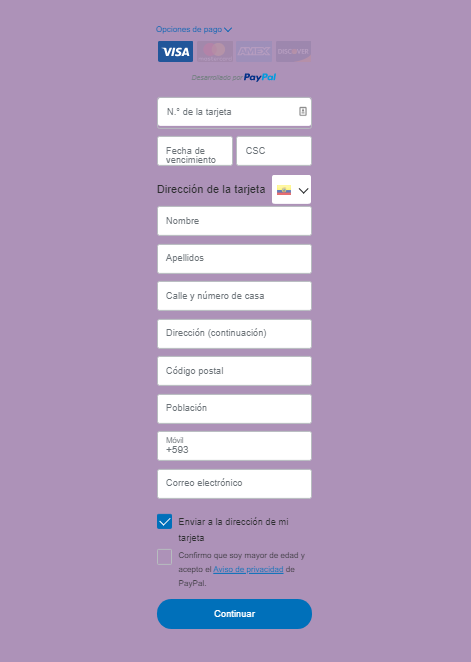


**Fig. 38:** Donaciones

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Fig. 39:** Donación con PayPal



**Fig. 40:** Donación con tarjeta de crédito

### Controlador de notificaciones

En el controlador se han creado las funciones de registrar y visualizar notificaciones, de acuerdo con el usuario autenticado el API retorna un JSON con la lista de notificaciones que le corresponden.

### Visualizar notificaciones

En la **Fig. 41** se muestra la lista de notificaciones, la cual se presenta en un menú desplegable dando click al icono en forma de campana ubicado en la barra de navegación. Las notificaciones se registran cuando un usuario socio registra una cita a un afiliado.

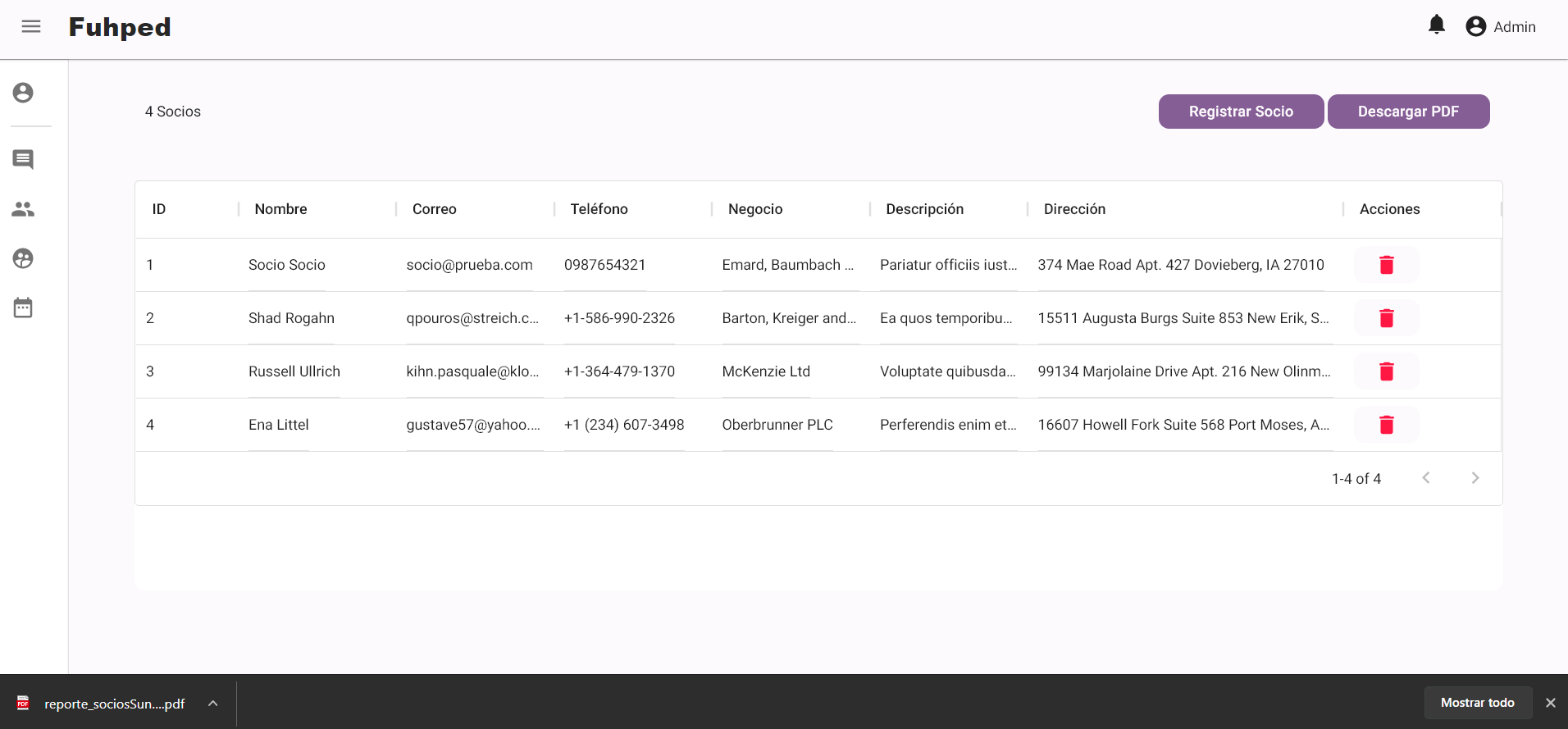
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 41:** Notificaciones

### Visualizar reportes de socios

En la **Fig. 42** se muestra el botón para descargar un reporte en PDF de todos los socios registrados en la Aplicación Web. En a **Fig. 43** se muestra el PDF en el cual se evidencia la lista de socios con sus datos correspondientes.



**Fig. 42:** Descargar reporte de Socios

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Fig. 43:** Reporte en PDF de Socios

### Visualizar reportes de afiliados

En la **Fig. 44** se muestra el botón para descargar un reporte en PDF de todos los afiliados registrados en la Aplicación Web. En a **Fig. 45** se muestra el PDF en el cual se evidencia la lista de afiliados con sus datos correspondientes.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 44:** Descargar reporte de afiliados

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

**Fig. 45:** Reporte en PDF de afiliados

### Visualizar reportes de citas

En la **Fig. 46** se muestra el botón para descargar un reporte en PDF de todas las citas registradas de todos los socios de la Aplicación Web. En la **Fig. 47** se muestra el PDF en el cual se evidencia la lista de citas con sus respectivos datos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 46:** Descargar reporte de citas

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig. 47:** Reporte en PDF de citas

## *Sprint* 6. Pruebas y despliegue de la Aplicación Web.

De acuerdo con la planificación establecida en el *Sprint Backlog*. El *Sprint* 6 contiene las siguientes actividades.

* Pruebas Unitarias
* Pruebas de compatibilidad
* Pruebas de rendimiento
* Despliegue de la Aplicación Web

### Pruebas Unitarias

Las pruebas unitarias forman parte de las actividades que se deben realizar internamente en las fases de las metodologías ágiles. Consisten en comprobar que pequeñas partes de código cumplan con el propósito para el que son escritas. Las pruebas se las realiza con el objetivo de verificar que los resultados obtenidos cumplan con los requisitos de código establecidos [17].

La **Fig. 48** muestra la prueba unitaria que se ha realizado para registrar una cita en la Aplicación Web. Las pruebas unitarias restantes se encuentran en el Manual Técnico – Sección Pruebas unitarias.

Captura de pantalla de un celular

Descripción generada automáticamente

**Fig. 48:** Prueba Unitaria: Registrar citas

### Pruebas de compatibilidad

Las pruebas de compatibilidad se encargan de comprobar que una Aplicación Web funcione correctamente en una cantidad pasable de navegadores web [18].

Las pruebas se realizaron en cuatro navegadores, el resultado obtenido se muestra se muestra en el Manual Técnico – Sección de Pruebas de compatibilidad.

En la T*ABLA* V se muestra los navegadores compatibles con la Aplicación Web.

**TABLA V**: Navegadores Compatibles con la Aplicación Web

|  |  |
| --- | --- |
| Navegador | Versión |
| Google Chrome | 92.0.4515.159 (Build oficial) (64 bits) |
| Mozilla Firefox | 91.0.2 (64-bit) |
| Microsoft Edge | Versión 92.0.902.78 (64 bits) |
| Opera | 2019-04-09 |

### Pruebas de rendimiento

Las pruebas de rendimiento se encargan de determinar la capacidad de respuesta de un software bajo una carga de trabajo especifica [19]. Para realizar las pruebas de rendimiento se ha utilizado las herramientas de código abierto K6 para realizar pruebas al API y la herramienta JMeter para realizar pruebas a la Aplicación Web, con las cuales se puede evaluar el rendimiento y detectar fallas en el software permitiendo desarrollar software robusto con la menor cantidad de fallas [20].

#### Pruebas de carga al API

La **Fig. 49** muestra los resultados obtenidos al ejecutar el *script* de pruebas de carga al API. Se ha realizado 960 peticiones en total, de las cuales se ha obtenido el 100% de peticiones aceptadas y un promedio de 16.41 segundos en cada petición.

Texto

Descripción generada automáticamente

**Fig. 49:** Resultados de pruebas de carga al API

#### Pruebas de carga a la Aplicación Web

La **Fig. 50** muestra los resultados de ejecutar pruebas de carga a la Aplicación Web. Se ha realizado 200 peticiones a cada ruta dando un total de 800 peticiones con un error de 0% en todas las rutas y con un promedio de 163 milisegundos en la ejecución de todas las peticiones. El proceso detallado sobre las pruebas de carga se encuentra en el Manual Técnico – Sección pruebas de rendimiento.

Captura de pantalla de computadora

Descripción generada automáticamente

**Fig. 50:** Prueba de Carga a la Aplicación Web

#### Pruebas de estrés al API

La **Fig. 51** muestra los resultados obtenidos al ejecutar el *script*, se ha realizado un total de 4920 peticiones, de las cuales se ha obtenido el 99.47% de peticiones aceptadas con un promedio de 37.94 segundos en cada petición.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Fig. 51:** Resultado de pruebas de estrés al API

#### Pruebas de estrés a la Aplicación Web

La **Fig. 52** muestra los resultados de ejecutar pruebas de estrés a la Aplicación Web. Se ha realizado 1000 peticiones a cada ruta dando un total de 4000 peticiones. En las peticiones realizadas a la página principal se evidencia un error de 0.70%, en la página de inicio de sesión se ha obtenido un error de 0.10%, en la página de publicaciones se tiene un error de 0% y en la página de servicios se evidencia un error de 0% con un promedio de 260 milisegundos en la ejecución de todas las peticiones. El proceso detallado sobre las pruebas de estrés se encuentra en el Manual Técnico – Sección pruebas de rendimiento.



**Fig. 52:** Prueba de estrés a la Aplicación Web

### Despliegue de la Aplicación Web

Para el despliegue de la Aplicación Web se utiliza la plataforma Vercel, la cual permite alojar sitios web escalables que no requieran supervisión, además Vercel ofrece un mejor rendimiento para el usuario final.

En la **Fig. 53** se muestra el proceso final de despliegue de la primera versión de la Aplicación Web en la plataforma de Vercel.

Captura de pantalla de un celular

Descripción generada automáticamente

**Fig. 53:** Configuración final de la Aplicación Web en Vercel

En la **Fig. 54** se muestra el correcto funcionamiento de la Aplicación Web, además se muestra el enlace de acceso donde se encuentra disponible. El proceso detallado sobre el despliegue de la Aplicación Web se encuentra en el Manual de Instalación – Sección Despliegue de la Aplicación Web.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

**Fig. 54:** Despliegue de la Aplicación Web en Vercel

# Conclusiones y Recomendaciones

A continuación, se presentan las conclusiones y recomendaciones que se han obtenido durante el desarrollo de la Aplicación Web.

## Conclusiones

* La implementación de la Aplicación Web para la gestión de socios y afiliados de la fundación “FUHPED”, permite a los administradores tener un registro de las actividades realizadas por los usuarios de la Aplicación Web llevando un control entre la interacción de socios y afiliados de la fundación.
* La apropiada recopilación de requerimientos mediante reuniones con el representante de la fundación “FUHPED” en la primera fase del proyecto hace posible el desarrollo de la Aplicación Web, debido a que gracias a ello se ha logrado comprender la información obtenida y de esta manera poder establecer las herramientas necesarias para el desarrollo del proyecto.
* El diseño de la base de datos en base a la información obtenida permite comprender de mejor manera la interacción entre cada entidad, además de conocer qué tipos de datos utilizar.
* El diseñar la arquitectura de la Aplicación Web previamente al desarrollo del proyecto permite tener un enfoque claro de los requerimientos funcionales y no funcionales determinados por el cliente.
* El diseño de las interfaces de la Aplicación Web antes de la etapa de codificación ha proporcionado una visión del producto final definiendo la estructura y posición de cada elemento de la Aplicación Web.
* El uso de Laravel y Next JS como *Frameworks* para el desarrollo de la Aplicación Web ha facilitado la implementación de cada módulo de forma sencilla con una mejor organización de código, posibilitando la reutilización de componentes.
* Las pruebas unitarias realizadas a cada componente de la Aplicación Web, mediante la ejecución de consultas a todas las rutas de la API, han permitido comprobar la correcta devolución de los datos.
* Las pruebas de compatibilidad han permitido evaluar el funcionamiento en varios navegadores web como lo son Chrome, Firefox, Microsoft Edge y Opera, en los cuales se evidencia un correcto funcionamiento.
* Las pruebas de rendimiento realizadas a la Aplicación Web, mediante pruebas de carga y de estrés tanto al API como al *Frontend* demostraron que la Aplicación Web soporta gran cantidad de solicitudes y la entrada de diferentes usuarios simultáneos.

## Recomendaciones

* Para futuras actualizaciones se recomienda verificar las versiones de las librerías y *Frameworks* utilizados en la Aplicación Web, con el objetivo de evitar conflictos que puedan ocasionar fallas en el software.
* Se recomienda actualizar la página en caso de que se produzca un error al regresar a la página principal.
* No se recomienda usar el navegador Brave, ya que este no guarda el token generado por la API y dificulta usar las funcionalidades que posee un usuario autenticado.

# Bibliografía

|  |  |
| --- | --- |
| [1] | J. Sanz, «Internet y la búsqueda de información en salud pública: desde la relevancia hacia la «revelancia»,» 16 Marzo 2005. [En línea]. Available: https://www.scielosp.org/article/gs/2006.v20n2/159-160/es/. [Último acceso: 3 06 2021]. |
| [2] | L. V. M. León, «MARKETING DIGITAL: REFLEXIONES SOBRE SU RELEVANCIA PARA EL CRECIMIENTO DE LAS EMPRESAS EN LA ACTUALIDAD,» Bogotá, 2020. |
| [3] | M. Castells, «INTERNET Y LA SOCIEDAD RED,» Catalunya, 2000. |
| [4] | G. C. Pita, «Las TICs en las empresas: evolución de la tecnología y cambio estructural en las organizaciones,» *Revista Cientifica Dominio de las Ciencias,* p. 12, 2018. |
| [5] | M. T. Gallegos, «Metodología Scrum». |
| [6] | K. Schwaber y J. Sutherland, La guía de Scrum, 2013. |
| [7] | C. Rodríguez y R. Dorado, ¿Por qué implementar Scrum?, 2015. |
| [8] | «Victorhck in the free world,» Victorhck , 30 4 2020. [En línea]. Available: https://victorhckinthefreeworld.com/2020/04/30/excalidraw-es-una-pizarra-en-blanco-en-la-que-realizar-diagramas-dibujados-a-mano/#:~:text=the%20free%20world-,Excalidraw%20es%20una%20pizarra%20en%20blanco%20en,realizar%20diagramas%20dibujados%20a%20mano&te. [Último acceso: 2 7 2021]. |
| [9] | Y. F. R. Yanette Díaz González, «Patrón Modelo-Vista-Controlador,» *Telemática,* vol. 11, nº 1, p. 11, 2012. |
| [10] | Laravel, «Laravel,» [En línea]. Available: https://laravel.com/docs/8.x. [Último acceso: 05 07 2021]. |
| [11] | A. Robledano, «OpenWebinars,» 24 09 2019. [En línea]. Available: https://openwebinars.net/blog/que-es-mysql/. [Último acceso: 05 07 2021]. |
| [12] | M. Ruiz, «OpenWebinars,» 19 09 2019. [En línea]. Available: https://openwebinars.net/blog/que-es-nextjs/. [Último acceso: 05 07 2021]. |
| [13] | DesarrolloWeb6, «DesarrolloWeb6,» 24 01 2020. [En línea]. Available: https://desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html. [Último acceso: 05 07 2021]. |
| [14] | A. López, «OpenWebinars,» 03 06 2019. [En línea]. Available: https://openwebinars.net/blog/que-es-postman/. [Último acceso: 05 07 2021]. |
| [15] | M. García, «Nettix,» 30 05 2020. [En línea]. Available: https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo. [Último acceso: 05 07 2021]. |
| [16] | React, «React,» [En línea]. Available: https://es.reactjs.org/. [Último acceso: 06 07 2021]. |
| [17] | Apiumhub, «Apiumhub,» Apiumhub, 21 04 2018. [En línea]. Available: https://apiumhub.com/es/tech-blog-barcelona/beneficios-de-las-pruebas-unitarias/. [Último acceso: 27 08 2021]. |
| [18] | M. A. M. C. L. G. E. I. Leandro N. Sabaren, «Una Revisión Sistemática de la Literatura en Pruebas de Compatibilidad Web,» Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura, La Plata, 2017. |
| [19] | M. Focus, «Micro Focus,» [En línea]. Available: https://www.microfocus.com/es-es/what-is/performance-testing. [Último acceso: 28 08 2021]. |
| [20] | J. d. Andalucía, «Junta de Andalucía,» [En línea]. Available: http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/388. [Último acceso: 28 08 2021]. |

# ANEXOS

## Manual Técnico

* Recopilación de requerimientos
* Historias de usuario
* Product Backlog
* Sprint Backlog
* Diseño de Interfaces
* Diseño de la base de datos
* Pruebas Unitarias
* Pruebas de compatibilidad
* Pruebas de rendimiento

## Manual de Usuario

<https://youtu.be/Cf2KJgaz1eM>

<https://github.com/MCris29/fuhped-frontend>

## Manual de Instalación

* Despliegue de la API en Heroku
* Despliegue de la Aplicación Web en Vercel