

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

SISTEMA WEB DE GESTIÓN DE CONFERENCIAS CON PAGOS EN LÍNEA

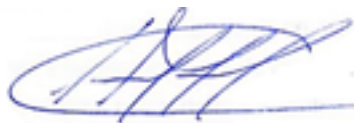
COMPONENTE FRONTEND

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE



JORDI SEBASTIÁN VALENCIA CARTAGENA

jordi.valencia@epn.edu.ec



DIRECTORA: ING. MAYRA ISABEL ALVAREZ JIMÉNEZ

mayra.alvarez@epn.edu.ec

DMQ, febrero 2023

CERTIFICACIONES

Yo, Jordi Sebastián Valencia Cartagena declaro que el trabajo de integración curricular 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.

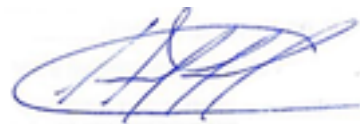


JORDI SEBASTIÁN VALENCIA CARTAGENA

jordi.valencia@epn.edu.ec

svalencia310@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Jordi Sebastián Valencia Cartagena, bajo mi supervisión.



ING. MAYRA ISABEL ALVAREZ JIMÉNEZ MSc.

mayra.alvarez@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



JORDI SEBASTIÁN VALENCIA CARTAGENA

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
ÍNDICE DE CONTENIDO.....	III
RESUMEN	V
ABSTRACT	VI
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos	1
1.3 Alcance	2
1.4 Marco Teórico	2
2 METODOLOGÍA	3
2.1 Metodología de Desarrollo	4
Metodologías ágiles	4
Roles	4
Artefactos.....	6
2.2 Diseño de interfaces (mockups).....	8
Herramientas para el diseño	8
Sistema Web	9
2.3 Diseño de la arquitectura	9
Patrón arquitectónico	10
2.4 Herramientas de desarrollo.....	11
3 RESULTADOS.....	13
Sprint 0. Configuración del ambiente de desarrollo	13
Sprint 1 Página inicio de sesión/registro.....	16
Sprint 2 Sección de ajustes y secciones.....	19
Sprint 3 Módulo de sección de pagos.	28
Sprint 4 Módulo de gestión.....	32
Sprint 5 Módulo de gestión usuarios.	39
Sprint 6. Pruebas del sistema	47
4 CONCLUSIONES	50
5 RECOMENDACIONES	51
6 REFERENCIAS BIBLIOGRÁFICAS.....	52

7 ANEXOS.....	54
ANEXO I.....	55
ANEXO II.....	56
ANEXO III.....	34
ANEXO IV	35

RESUMEN

Un sistema de gestión de conferencias en línea puede automatizar muchos de los procesos manuales que consumen tiempo y recursos, como la gestión de registro, el procesamiento de pagos y la comunicación con los participantes. Por tanto, como proyecto de titulación la propuesta es el desarrollo de un sistema *web* de gestión de conferencias con pagos en línea que utiliza *software Open Source* y la metodología ágil *Scrum*. El sistema maneja dos perfiles administrador y cliente, cuenta con módulos para la visualización de la página principal, panel de administración, registro, inicio de sesión y sección de pagos a través de PayPal. Con este fin, se proporcionan herramientas prácticas para ayudar a los organizadores de conferencias a planificar, administrar registros, establecer precios y administrar con éxito sus eventos. El objetivo del sistema es ahorrar tiempo y esfuerzo a los organizadores para que puedan concentrarse en el contenido y el entretenimiento de la reunión en lugar de en la gestión de la reunión.

Este proyecto de está organizado de la siguiente forma: en la primera sección se describen los antecedentes, objetivos, alcance del proyecto y el marco teórico. En la segunda sección se aborda la aplicación de la metodología *Scrum*, prototipos y las herramientas usadas para el respectivo desarrollo del sitio *web*. En la tercera sección se muestran las actividades realizadas en cada iteración y los resultados obtenidos en cada *Sprint* juntamente con las pruebas realizadas para verificar del correcto funcionamiento del componente *Frontend* del sistema *web*.

PALABRAS CLAVE: sistema web, servicios, interfaz, frontend.

ABSTRACT

An online conference management system can automate many of the manual processes that consume time and resources, such as registration management, payment processing, and communication with participants. Therefore, as a curricular integration project, the development of a conference management web system with online payments that uses Open-Source software, and the agile Scrum methodology is proposed. The system manages two administrator and client profiles, it has modules for viewing the main page, administration panel, registration, login and payment section through PayPal. To this end, practical tools are provided to help conference organizers successfully plan, manage registrations, price, and manage their events. The goal of the system is to save hosts time and effort so they can focus on meeting content and entertainment rather than meeting management.

This curricular integration project is organized as follows: the first section describes the background, objectives, scope of the project and the theoretical framework. The second section deals with the application of the Scrum methodology, prototypes and tools used for the development of the website. The third section shows the activities carried out in each iteration and the results obtained in each Sprint together with the tests carried out to verify the correct functioning of the Frontend component of the web system.

KEY WORDS: web system, services, interface, frontend.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La tecnología nos ha dado la capacidad de estar en varios lugares al mismo tiempo y tener experiencias de forma remota. Por tanto, las conferencias en línea se han vuelto una valiosa alternativa a las conferencias tradicionales, ya que el conocimiento se puede adquirir independientemente de la ubicación geográfica. La gestión de conferencias implica el uso tanto de *software* como de *hardware*, siendo el primero una aplicación desarrollada para sistemas operativos móviles y de escritorio que permitan llevar a cabo gestiones de eventos referentes a conferencias. Estas herramientas digitales vienen en diferentes presentaciones y formatos, pero su objetivo es siempre el mismo, que es brindar a los usuarios una experiencia de comunicación efectiva. La ventaja más importante de la gestión de conferencias *web* es que eliminan la barrera de la distancia y la comunicación efectiva, ya que es la herramienta, una plataforma que permite a las empresas, gerentes y empleados enfocarse en el contenido y el entretenimiento del evento y/o conferencia en lugar de en la gestión de esta, haciendo que el flujo de trabajo sea más productivo [1].

Un sistema de gestión de contenidos puede ser la solución a los problemas que surgen al planificar y organizar una conferencia. Esta herramienta puede ayudar a los organizadores a superar los desafíos técnicos y cualquier otra dificultad única que puedan enfrentar. Esto les permite a los organizadores gestionar mejor su trabajo [2].

Por tanto, el objetivo final es el desarrollo de un sitio web amigable para con el usuario, que brinde una experiencia intuitiva y fácil de usar, que en comparación con otros sistemas no es una aplicación como tal que se instale en el ordenador ya que estará disponible en internet, adicionalmente este sistema contará con una funcionalidad de pagos en línea a través de PayPal lo que brinda un valor agregado en comparación con otros sistemas de la misma temática de gestión de conferencias en línea.

1.1 Objetivo general

Desarrollar el *frontend* de un sistema *web* para la gestión de conferencias con pagos en línea.

1.2 Objetivos específicos

1. Definir los requerimientos necesarios para lograr la visualización integra del *frontend* del sistema *web*.
2. Diseñar prototipos e interfaz del *frontend* del sistema *web* de acuerdo con los requerimientos obtenidos.

3. Implementación módulos correspondientes para el *frontend* del sistema *web*.
4. Realizar pruebas para verificar el funcionamiento del *frontend* del sistema *web* en diferentes navegadores y dispositivos.

1.3 Alcance

El desarrollo de presente sitio incluye la creación de una interfaz de usuario para un sistema *web* que permita la administración de conferencias utilizando *software* de código abierto y aplicando la metodología *Scrum* para cumplir con los requisitos establecidos. El sistema *web* incluirá módulos para la página principal, el registro o inicio de sesión, secciones para servicios, pagos y gestiones. En cuanto a los usuarios, se ha previsto que se tenga dos roles en un principio: administrador (*admin*) y usuario base (*user*). El usuario administrador está en la capacidad de crear más roles y asignarlos a los usuarios registrados, además de opciones como: inicio de sesión, gestionar la información de las conferencias y proporcionar detalles sobre precios y pagos. Por otro lado, el cliente tiene la opción de registrarse, visualizar la información sobre las conferencias y realizar los pagos correspondientes en línea.

Este sistema brindará una herramienta práctica para ayudar a los organizadores de conferencias a planificar, administrar los registros, establecer precios y tener éxito en la administración de un evento. El objetivo es ahorrar tiempo y esfuerzo a los organizadores, permitiéndoles concentrarse en el contenido y el entretenimiento de la conferencia en lugar de en su gestión. Además, los usuarios podrán realizar pagos en línea a través del sistema utilizando PayPal como método de pago.

1.4 Marco Teórico

Un *framework* tiene como objetivo principal mejorar la eficiencia y calidad en el desarrollo de nuevos programas de computadora, lo que a su vez puede mejorar la calidad, la confiabilidad y la solidez del *software*. En particular, los *frameworks* pueden mejorar la productividad de los desarrolladores al permitirles enfocarse en las necesidades específicas de la aplicación en lugar de preocuparse por la infraestructura de la aplicación.

Las aplicaciones *web* son una opción popular para los usuarios debido a la facilidad de uso del navegador y la facilidad de actualización y mantenimiento sin necesidad de instalar programas adicionales. Ejemplos de aplicaciones *web* populares incluyen correos electrónicos, wikis, blogs y tiendas en línea [3].

En el desarrollo de *software*, una aplicación *web* está diseñada para poder ser utilizada mediante el uso de navegadores o de una red local, y se desarrolla en un lenguaje compatible con estos sistemas. En la actualidad, muchas empresas y negocios están adoptando esta tecnología para adaptarse a las necesidades y demandas de sus clientes, ofreciendo soluciones informáticas personalizadas y accesibles desde cualquier lugar con conexión a Internet. [4].

El desarrollo web es el proceso de construir y mantener sitios web funcionales en internet mediante el uso de diversos lenguajes de programación. Cada sitio web tiene una dirección web única que lo diferencia de otros en la red. Este proceso se divide en dos partes: frontend, que se encarga de la presentación visual y la funcionalidad que interactúa con el usuario, y backend, que se enfoca en la estructura del sitio web y su gestión, incluyendo la accesibilidad, la actualización, las bases de datos y los cambios [5]. Desde los comienzos del uso del Internet han surgido diversos requerimientos de usuarios y las respuestas han sido dictadas por lenguajes estáticos. Con el tiempo, las tecnologías han ido evolucionando y han surgido nuevos problemas a resolver. Esto condujo al desarrollo de los lenguajes de programación dinámicos dedicados a la *web* que permitirían a los usuarios interactuar y poder trabajar con bases de datos.

De acuerdo con el componente a desarrollar el *frontend* corresponde a la parte visible o interfaz de un sitio *web* o aplicación con la que interactúa el usuario y se denomina "lado del usuario". En este contexto es todo lo que vemos a través de nuestra pantalla y como usuarios podemos navegar e interactuar.

El *frontend* se compone de textos, colores, efectos visuales, objetos en movimiento, desplazamientos, imágenes y muchos otros elementos que el usuario puede disfrutar en un producto digital. Los desarrolladores crean bases en el *backend* que ayudan a que el *frontend* funcione de la mejor manera posible con un diseño centrado en el usuario [6].

2 METODOLOGÍA

Los métodos ágiles de desarrollo de *software* pretenden entregar aplicaciones funcionales en períodos cortos de tiempo para mejorar la satisfacción del cliente. Esta metodología prioriza la simplicidad en la documentación y permite cambios durante todo el ciclo que esté disponible el proyecto. El objetivo fundamental de la metodología ágil es proporcionar sistemas de *software* que estén siempre en ejecución y sean desarrollados con ciclos de iteración cortos [7].

2.1 Metodología de Desarrollo

Es un conjunto de prácticas, procesos, técnicas y herramientas que se utilizan para planificar, diseñar, desarrollar, verificar y dar mantenimiento el *software* de manera efectiva y organizada. La elección de una metodología depende de factores como el tamaño del proyecto, la naturaleza del *software* y los requisitos del cliente, entre otros. Por otro lado, los métodos también sirven para controlar el desarrollo del trabajo, lo que permite minimizar los márgenes de error y anticiparse a esta situación. Otro beneficio de usar una metodología es que le ahorra tiempo y permite administrar mejor los recursos disponibles [8].

Metodologías ágiles

La metodología ágil está transformando la forma en que muchas empresas, incluyendo Google, Amazon y Microsoft, abordan el desarrollo de proyectos. Este enfoque innovador ha ganado popularidad en todo el mundo y se considera que se ha establecido en la industria. La clave del éxito de Agile radica en la capacidad de organizar y distribuir el trabajo de manera rápida y flexible entre equipos multidisciplinarios.

La metodología ágil se utiliza comúnmente en el desarrollo de sistemas de *software* y proyectos de alto rendimiento, y se enfoca en la rápida implementación de equipos flexibles y eficientes para planificar el flujo de trabajo. El enfoque permite elegir cual sería la mejor opción para cada situación sin tener que comprometer el proyecto en su conjunto.

Las metodologías ágiles se basan en varias series de principios que deben aplicarse en todo el proyecto, y se caracteriza por la realización de iteraciones de pequeñas partes del proyecto para aumentar gradualmente el progreso general [8]. En resumen, las metodologías ágiles son flexibles y adaptables que permite a los equipos abordar los proyectos de manera eficiente y eficaz.

Scrum se define como marco de trabajo ágil para el desarrollo de *software*, productos y servicios, que se enfoca en la entrega de valores a los clientes y a la mejora continua del equipo. Fue desarrollado en el año 1993 por Jeff Sutherland [9].

Roles

Los roles conforman un pequeño equipo formado por un *Scrum Master*, *Product Owner* y los desarrolladores. No hay sub-equipos o jerarquías en este grupo, los profesionales son multifuncionales y tienen las habilidades para lograr el objetivo del producto final. También

deciden entre ellos quién hace qué, cuándo y cómo. Además, son responsables de todo lo relacionado con el producto, desde el trabajo con el cliente hasta la verificación de operaciones, mantenimiento, operaciones, investigación y desarrollo [9].

Product Owner

Es responsable de guiar al equipo a la mejor solución que cumpla con los requerimientos ya que de primera mano el producto final. Además, toma las decisiones y es el encargado de la gestión eficaz de la cartera de productos, el equipo y el esfuerzo [9].

Scrum Master

El *Scrum Master* es algo así como un facilitador y *coach* para el *Development Team*, ayudándoles a implementar la metodología *Scrum* de una manera efectiva, el que mueve al equipo encargado de ver soluciones para superar los posibles retrasos que puedan presentarse durante el proceso, pone en alerta sobre los riesgos para tratar de mitigarlos y alentar a cada integrante a buscar soluciones a cada dificultad. Esto incluye guiar al equipo durante la planificación y realización de los *Sprints*, asegurarse de que se respeten las reuniones diarias, reuniones de revisión y retrospectiva, y ayudar a resolver cualquier obstáculo que pueda surgir [9].

Development Team

Development Team es un grupo de personas responsables de cumplir con el trabajo de desarrollar el *software* y de crear el producto final. Este equipo está compuesto por desarrolladores, ingenieros de pruebas, diseñadores, entre otros, y todos trabajan juntos para lograr los objetivos del proyecto. En la TABLA I, se muestra los distintos roles del *Scrum Team*.

TABLA I. Asignación de roles

ROLES	NOMBRES
<i>Product Owner</i>	Ing. Mayra Isabel Alvarez Jiménez
<i>Scrum Master</i>	Ing. Mayra Isabel Alvarez Jiménez
<i>Development Team</i>	Sebastián Valencia

Artefactos

En *Scrum* tienen como finalidad principal maximizar la transparencia de la información clave. De esta forma, todas las personas involucradas en el proyecto tienen una base común de adaptación. Cada artefacto está comprometido a proporcionar información que mejore el enfoque en el progreso del proyecto. El *Product Backlog* se enfoca en el producto, el *Sprint Backlog* en el *Sprint* y el incremento en la definición de terminado [10]. En resumen, los artefactos de *Scrum* son elementos claves para mejorar el enfoque en el progreso del proyecto, y aseguran una mejor comunicación y alineación entre todos los *stakeholders*.

Recopilación de Requerimientos

La definición del alcance del proyecto es un proceso crucial en el que se identifican las necesidades y lo que se espera de los *stakeholders*. Estas se documentan formalmente para reflejar el entendimiento común de las partes involucradas y se gestionan durante todo el proyecto para lograr los objetivos establecidos.

Historias de Usuario

Son una técnica muy útil en la metodología ágil para describir las necesidades y expectativas de un *software* desde la perspectiva de un usuario. Estas historias ayudan a alinear las expectativas de los *stakeholders* y reducen la probabilidad de errores críticos en el futuro. Por lo general, estas historias se describen en un lenguaje común entre los *stakeholders* y son muy breves [11]. En resumen, las historias de usuario son una herramienta imprescindible en el proceso de desarrollo de *software* que permiten una comunicación efectiva entre los usuarios y el equipo de trabajo.

En la TABLA II, se presenta un ejemplo de cómo son las historias de usuario en cuestión como requerimiento del sistema. El resto de las historias de usuario se encuentran en el ANEXO II de este documento.

TABLA II. Historia de usuario Nro.1

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Administrador/Cliente
Nombre Historia: Mostrar página inicial	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Iteración Asignada: 1	

Responsable: Sebastián Valencia
Descripción: Tanto el usuario base como el usuario administrador al ingresar al sitio web podrá visualizar la página principal que es la misma para los dos roles.
Observación: la página mostrará información sobre conferencias y los respectivos secciones para la navegación a través de la página y el acceso al panel de administración.

Product Backlog

Es uno de los artefactos principales de *Scrum*, y consiste en una lista ordenada y emergente de las características, funcionalidades, mejoras y correcciones necesarias para el producto o servicio. Es una lista dinámica que cambia constantemente para adaptarse a lo que busca el mercado, y nunca se considera completa. Cada elemento en el *Product Backlog* tiene una descripción que lo identifica, un orden de prioridad y una estimación de esfuerzo, y se le llama *Product Backlog Item* (PBI). Es la fuente de requisitos principal para cualquier cambio que tenga que ver con el producto o servicio, y se actualiza continuamente según la retroalimentación del mercado, lo que lo convierte en un artefacto vivo. [12]. En la TABLA III, muestra un pequeño ejemplo con unos ítems del *Product backlog*. El resto de los ítems se encuentran en el **ANEXO II** de este documento.

TABLA III. Product Backlog

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>					
ID – PB	ID – HU	Historia de usuario	Iteración	Estado	Prioridad
HU002	HU002	Mostrar Página de inicio/registro	2	Finalizado	Alta
HU003	HU003	Mostrar Página de ajustes de usuario	3	Finalizado	Media

Sprint Backlog

En cada *Sprint* ocurren cambios en los *ítems* que vienen de *Product Backlog*, estos cambios los hace el equipo es el cual determina las tareas puntuales que se pueden completar [12]. En la TABLA IV, se muestra un ejemplo de las tareas a realizar. El resto de los ítems se encuentran en el **ANEXO II** de este documento.

TABLA IV. Sprint Backlog

Elaboración del <i>Sprint backlog</i>					
ID – SB	Nombre	ID-HU	Historia de usuario	Tareas	Tiempo estimado
SB00	Configuración del entorno de desarrollo	N/A	N/A	<ul style="list-style-type: none"> • Instalación de herramientas. • Creación del Proyecto. • Diseño del prototipo. 	40 H

2.2 Diseño de interfaces (mockups)

Una vez finalizada la recopilación de requerimientos, se debe realizar la elaboración del prototipado y del diseño de las respectivas interfaces que son una representación visual de una interfaz de usuario que se utiliza para mostrar cómo será el diseño del proeycto y la funcionalidad esperada de la aplicación.

Herramientas para el diseño

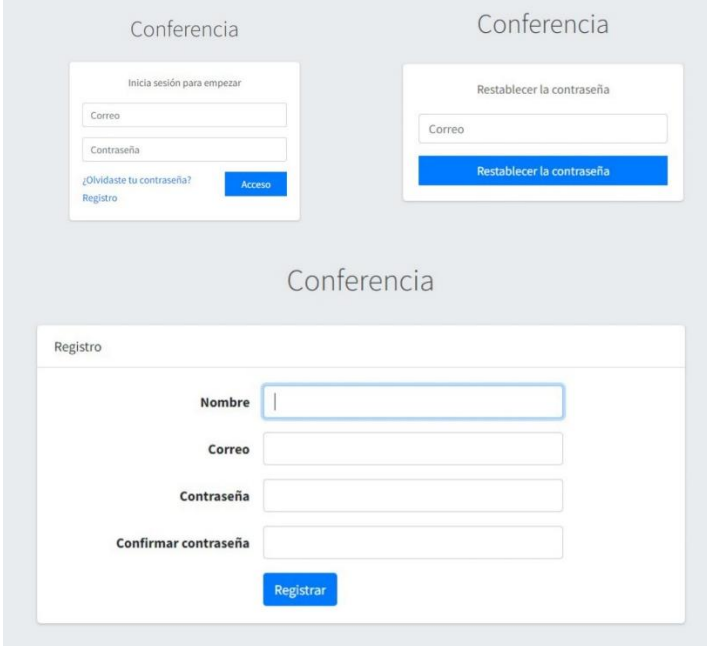
Las herramientas de desarrollo son programas que ayuda a los desarrolladores de *software* a crear y mantener aplicaciones y sistemas informáticos de manera más eficiente y de alta calidad. Estas herramientas incluyen editores de código, depuradores, control de versiones, entre otros, y están diseñadas para mejorar la funcionalidad y la calidad del *software* desarrollado.

Figma

Es una plataforma online de diseño gráfico y de interfaces que cuenta con la particularidad de ser colaborativa. Se utiliza para crear diversos tipos de diseños, desde páginas web y aplicaciones hasta publicaciones para redes sociales y presentaciones. Esta herramienta es muy apreciada por empresas y estudios de diseño debido a su capacidad de permitir la colaboración entre sus usuarios. Una de las características más destacadas de Figma es que todos los miembros del equipo pueden trabajar en el mismo archivo al mismo tiempo, involucrando a diseñadores gráficos, desarrolladores y redactores en las primeras fases del proyecto. [13].

Sistema Web

Un sistema *web* es un conjunto de componentes interconectados que trabajan juntos para permitir el acceso y uso de aplicaciones y servicios mediante de la *World Wide Web* (WWW). Un sistema *web* incluye servidores, aplicaciones y navegadores *web* [18]. La **Fig 1**, presenta las páginas correspondientes de *login*, registro y restablecimiento de contraseñas.



The image displays three web forms for a system named 'Conferencia'. The top-left form is for login, titled 'Inicio sesión para empezar', with fields for 'Correo' and 'Contraseña', a link for '¿Olvidaste tu contraseña?', and a 'Registro' link, with a blue 'Acceso' button. The top-right form is for password reset, titled 'Restablecer la contraseña', with a 'Correo' field and a blue 'Restablecer la contraseña' button. The bottom form is for registration, titled 'Registro', with fields for 'Nombre', 'Correo', 'Contraseña', and 'Confirmar contraseña', and a blue 'Registrar' button.

Fig 1. Prototipo de inicio de sesión, registro y restablecimiento de contraseñas

2.3 Diseño de la arquitectura

Es una técnica y el proceso de definir la estructura, las componentes, módulos, interfaces y datos de un sistema *software*, con el objetivo de satisfacer los requisitos técnicos y funcionales y mejorar la calidad y el rendimiento del *software*. La arquitectura de *software* es como un plan para el sistema que describe cómo se organizan sus partes y las directrices para su comportamiento y funcionalidad. Es esencial elegir la arquitectura adecuada para el proyecto, ya que esto puede afectar su calidad, rendimiento, escalabilidad y facilidad de mantenimiento. La estructuración de una arquitectura que optimice el entendimiento de sus componentes es fundamental para el éxito del proyecto [14].

Patrón arquitectónico

El Modelo-Vista-Controlador (MVC) se define como un patrón de diseño arquitectónico de *software* que se utiliza para desarrollar aplicaciones interactivas, especialmente aplicaciones de interfaz de usuario. Este patrón divide a la aplicación en tres componentes: modelo, vista y controlador [15].

Modelo

Nivel de manipulación de datos, por lo que contiene mecanismos para poder acceder a la información y modificar su estado. Los datos normalmente residen en la base de datos, por lo que el modelo debe tener todas las funciones para acceder a las tablas y realizar selecciones, actualizaciones, inserciones, entre otros [15].

Vistas

Como sugiere el nombre, las vistas contienen el código del proyecto que crea la interfaz o código que representa el estado de la aplicación en HTML. Sin embargo, las vistas, que normalmente manipulan datos, no tienen acceso directo a los datos. La vista solicita datos del modelo y produce la salida requerida por la aplicación [15].

Controlador

Se trata de una capa intermedia que conecta la vista y el modelo en una aplicación, y se encarga de implementar las funcionalidades necesarias para responder a las solicitudes de la aplicación como: visualizar artículos, realizar compras, buscar información, entre otras acciones [15].

El desarrollo del presente sistema *web* dispone de un patrón arquitectónico el cual se muestra en la **Fig 2**, donde se menciona las herramientas y *frameworks* utilizados en un sistema con patrón arquitectónico MVC. En este sistema a desarrollar, la visualización de datos se realiza mediante el acceso a un servidor privado que aloja el proyecto.

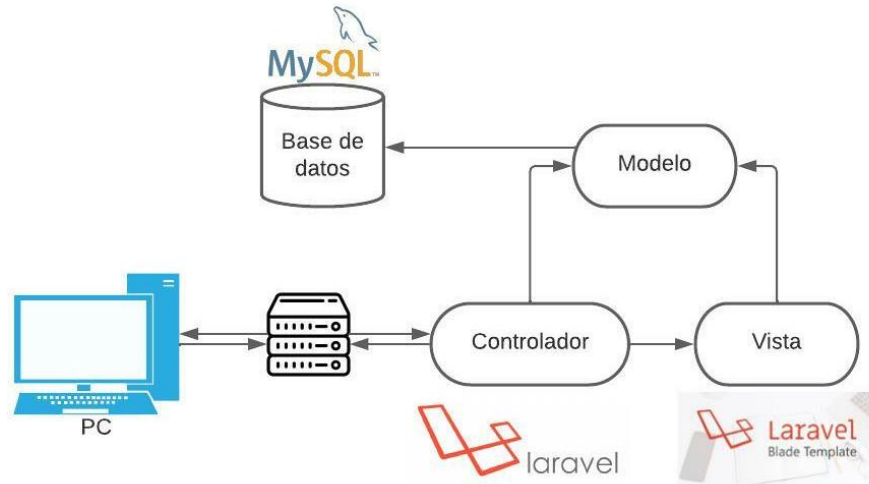


Fig 2. Patrón arquitectónico

2.4 Herramientas de desarrollo

En base al patrón arquitectónico definido, se hará uso de herramientas que cumplan con los requerimientos para el desarrollo del sistema *web*. En la TABLA V, se describen las herramientas de desarrollo a usar con su respetiva justificación.

TABLA V. Herramientas y librerías para el desarrollo del Sistema Web

Herramientas	Justificación
Laravel	Es un framework que ofrece una experiencia de desarrollo de alta calidad y características avanzadas como la integración profunda de las respectivas dependencias, una capa de abstracción de base de datos, colas, tareas programadas y pruebas unitarias [16].
Bootstrap	Bootstrap es una herramienta de desarrollo de interfaz de usuario gratuita que se utiliza para desarrollar sitios web y aplicaciones web. Su enfoque se centra en permitir la creación de estilos para sitios web y aplicaciones web con capacidad de respuesta para dispositivos móviles. Bootstrap proporciona plantillas y sintaxis de diseño que facilitan el diseño de interfaces de usuario atractivas y consistentes [17].
	API de PayPal es una interfaz de programación de aplicaciones que ayudan a los desarrolladores de <i>software</i> integrar las funcionalidades de PayPal en sus aplicaciones. Con la API de PayPal, los desarrolladores

API PayPal	pueden aceptar pagos en línea, gestionar transacciones y realizar operaciones financieras en línea [19].
Librerías	Justificación
Blade	Blade sirve para definir el modelo en la vista no solo le permite realizar operaciones en los datos, sino que también reemplaza secciones del modelo con otro contenido, hereda entre modelos, define el diseño y básicamente es una librería que puede crear varias estructuras HTML. Llenado por otras plantillas, subvistas, plantillas base, etc [20].
Artisan	Permite la creación de las vistas por medio comandos enviados por consola

3 RESULTADOS

A continuación, se describen los resultados de la implementación de los módulos con los diferentes componentes en base a los requerimientos del sistema documentados en las historias de usuario y artefactos de la metodología ágil *Scrum*.

Sprint 0. Configuración del ambiente de desarrollo

En este punto se realizan las tareas de configuración de todos los aspectos que permitan continuar el desarrollo del presente proyecto, lo que incluye la instalación y configuración de herramientas y plataformas de desarrollo.

Tareas asignadas para el *Sprint 0*:

- Instalación de herramientas.
- Creación del Proyecto.
- Diseño del prototipo.

Instalación de herramientas

Laravel es un *framework* basado en PHP, por lo que es necesario instalar una versión compatible de PHP. Composer es un gestor de paquetes para PHP que se utiliza para instalar y administrar las dependencias de *Laravel* y sus paquetes.

Git que es un sistema para el control de versiones que se utiliza para administrar y versionar el código fuente del proyecto, un servidor *web* para ejecutar la aplicación. *Laravel* incluye un servidor de desarrollo integrado para pruebas locales. Tal como se muestra en la **Fig 3, Fig 4 y Fig 5**.

```
C:\Users\HP\Desktop\trabajoitc>npm --version  
8.19.2
```

Fig 3. Versión de Node.js

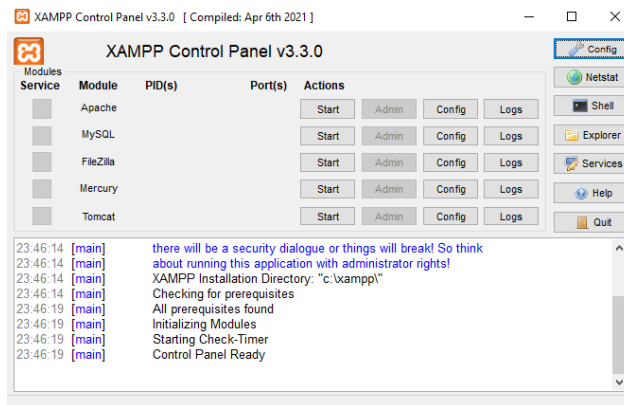


Fig 4. Panel de control instalado de xampp

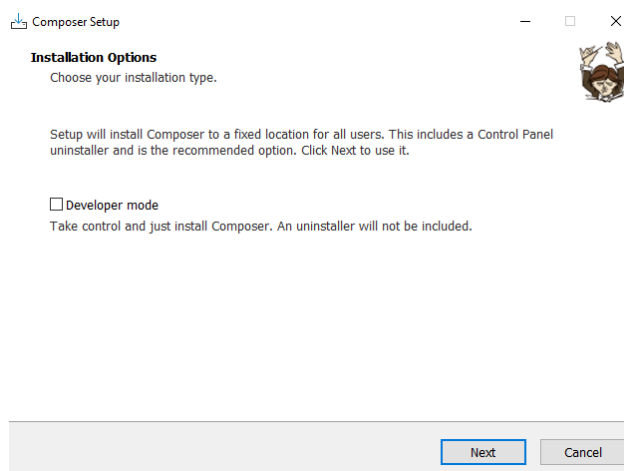


Fig 5. Instalación de composer

Creación del proyecto

La creación del proyecto se hizo mediante el símbolo del sistema usando composer ejecutando el siguiente comando: *composer create-project laravel/laravel proyecto* en la ruta designada para el proyecto tal como se muestra en la **Fig 6**, seguido de eso y ya creado el proyecto se revisa la estructura del proyecto para su modificación como muestra en la **Fig 7**.

```
C:\xampp\htdocs\trabajoITC>composer create-project laravel/laravel example-app
Creating a "laravel/laravel" project at "./example-app"
Installing laravel/laravel (v9.4.1)
- Downloading laravel/laravel (v9.4.1)
- Installing laravel/laravel (v9.4.1): Extracting archive
Created project in C:\xampp\htdocs\trabajoITC\example-app
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 107 installs, 0 updates, 0 removals
- Locking brick/math (0.10.2)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/instantiator (1.4.1)
- Locking doctrine/lexer (1.2.3)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (3.2.1)
- Locking fakerphp/faker (v1.21.0)
- Locking filp/whoops (2.14.6)
- Locking fruitcake/php-cors (v1.2.0)
- Locking graham-campbell/result-type (v1.1.0)
- Locking guzzlehttp/guzzle (7.5.0)
- Locking guzzlehttp/promises (1.5.2)
- Locking guzzlehttp/psr7 (2.4.3)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v9.45.0)
```

Fig 6. Creación del proyecto

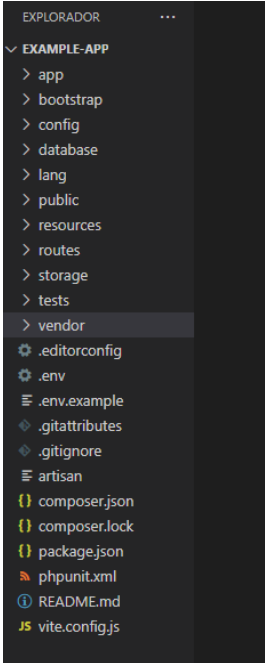


Fig 7. Estructura del proyecto

Diseño del prototipo

El prototipado es una parte crítica del proceso de desarrollo, permite el diseño y desarrollo crear y probar de manera eficiente las interfaces de usuario antes de la implementación. La sección de *login* se realizó con la herramienta Figma la cual se puede observar en la Fig 1.

Sprint 1. Página inicio de sesión/registro.

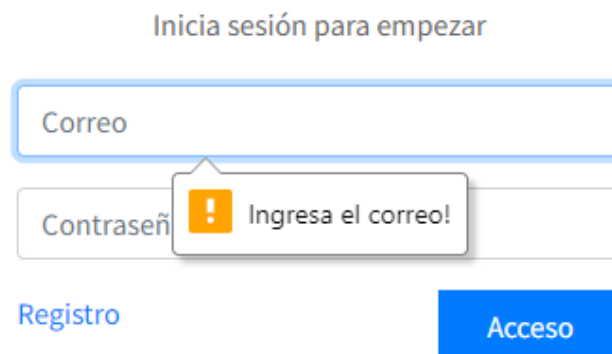
Es la página que los usuarios ven cuando quieren acceder al panel de administración del sistema *web*, se requiere la creación de componentes específicos que conformarán el inicio de sesión y el registro, como formularios de registro y login, la estructura de la página, y la implementación de las funcionalidades necesarias y validaciones para la página.

Tareas asignadas para el *Sprint 1*:

- Implementación página para inicio de sesión del sistema *web*.
- Implementación página para registro del sistema *web*.
- Implementación sección para restablecimiento de credenciales del sistema *web*.
- Prueba unitaria *Sprint 1*.

Implementación página para inicio de sesión del sistema *web*

El *Sprint 1*, corresponde al desarrollo de la respectiva sección del inicio de sesión que es una parte del sistema *web* que permite a los usuarios autenticarse mediante el ingreso del correo electrónico y la contraseña. Una vez autenticados, los usuarios pueden acceder a las funciones y datos restringidos del sitio *web*, tal como se muestra en la **Fig 8**.



Inicia sesión para empezar

Correo

Contraseña

Ingresa el correo!

Registro

Acceso

Fig 8. Inicio de sesión y validaciones

Implementación página para registro del sistema *web*

El registro que permite crear una cuenta y proporcionar información personal, como el nombre de usuario, el correo electrónico y la contraseña. Esta información es utilizada para autenticar al usuario, tal como se muestra en la **Fig 9**.

The image shows a registration form with four input fields and a button. The 'Nombre' field contains 'jordi'. The 'Correo' field contains '?.com|' and is highlighted with a blue border. A tooltip with an orange exclamation mark icon and the text 'No es un correo!' is positioned over the 'Correo' field. The 'Contraseña' and 'Confirmar contraseña' fields are empty. A blue 'Registrar' button is located below the fields.

Fig 9. Registro y validaciones

Implementación sección para restablecimiento de credenciales del sistema web

En caso de que los usuarios necesiten restablecer la contraseña para el ingreso al sistema web, se implementó la sección contáctanos, lo que permite la comunicación directamente con el responsable del soporte del sistema, que en este caso es el autor del proyecto, tal como se muestra en las Fig 10 y Fig 11.

The image shows a 'CONTACTANOS' section. At the top, the text 'Si tienes dudas o problemas comunícate con nuestro soporte' is centered above a green horizontal line. Below this, three columns of contact information are presented in a light blue box with rounded corners. The first column is labeled 'DIRECCIÓN' and contains 'N18-237, 15 de Julio, Quito 170204'. The second column is labeled 'NÚMERO DE TELÉFONO' and contains '0997562240'. The third column is labeled 'CORREO' and contains 'svalencia310@gmail.com'.

Fig 10. Contacto soporte

Restablecer la contraseña

jordictg312@gmail.com

Contraseña

Confirm ! Ingresa la contraseña!

Restablecer la contraseña

Fig 11. Restablecimiento de contraseña

Prueba unitaria *Sprint 1*

Este código define una función pública llamada *testLoginTest()*, que se utiliza para realizar una prueba automatizada. La prueba verifica si la ruta */login* devuelve una respuesta HTTP 200, que indica que la página se cargó correctamente. Para lograr esto, la función utiliza el método *get()* del objeto *\$this* (que hace referencia a una instancia de la clase *TestCase*), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método *assertStatus()* para verificar si la respuesta recibida tiene un estado HTTP 200, que indica que la solicitud se realizó correctamente, véase **Fig 12**.

```
public function testLoginTest()  
{  
    $response = $this->get('/login');  
    $response->assertStatus(200);  
}
```

Fig 12. Prueba unitaria #1. Acceso al login

La **Fig 13**, muestra el resultado que indica que la prueba automatizada *testLoginTest* del archivo *testLoginTest.php* ubicado en la carpeta *Tests\Unit* ha pasado exitosamente. La prueba verificó si la página de login se carga correctamente. La solicitud (Request) es una petición HTTP GET enviada a la URL <http://34.246.221.132/login>. El método utilizado para realizar la solicitud es GET. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 200 OK, lo que significa que la solicitud se realizó correctamente y la página correspondiente al *login* se cargó sin problemas

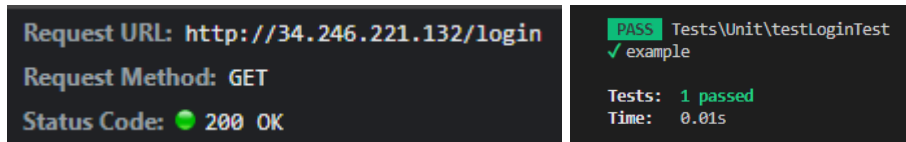


Fig 13. Resultados prueba unitaria #1

Sprint 2. Sección de ajustes y secciones.

El *Sprint 2*, se centra en la implementación de la sección de panel de administrador y las secciones principales del sistema *web*.

Tareas asignadas para el *Sprint 2*:

- Implementación de interfaz de administración.
- Implementación de interfaces de las secciones de la página principal.
- Prueba unitaria *Sprint 2*.

Implementación de interfaz de administración.

El *Sprint 2*, corresponde a la implementación de las secciones del panel de administrador o usuario que según el rol puede determinar qué acciones puede realizar dentro del sistema ya sea realizar todas las acciones, incluidas la gestión de usuarios, la configuración del sistema y la modificación de datos, o como usuario realizar acciones limitadas como ver contenido designado por el administrador, como se muestra en la **Fig 14**.

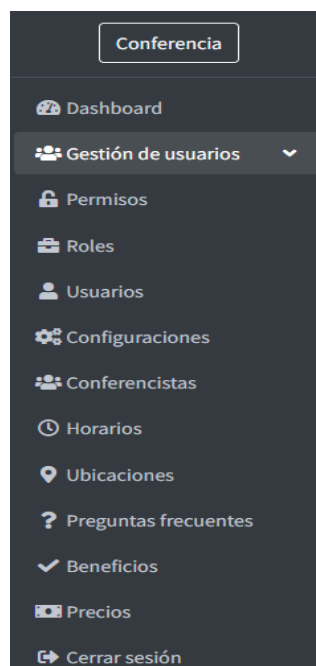


Fig 14. Panel de administración

Implementación de interfaces de las secciones de la página principal.

En lo que corresponde a la página principal está estructurada en secciones hacia abajo haciendo uso de *scroll* (desplazamiento con la barra lateral derecha). En la parte superior se encuentra la sección introductoria con un "nav" que es una etiqueta HTML que representa una sección de una página web que contiene enlaces a otras secciones. Por lo general, se utiliza para crear una barra de navegación en un sitio web el cual es el caso propuesto, como se muestra en la Fig 15.



Fig 15. Intro de la página

La sección "Acerca de" proporciona información de las conferencias que se compone de una descripción breve del evento, el lugar donde se va a realizar y las respectivas fechas, como se muestra en la Fig 16.



Fig 16. Sección Acerca de

La sección "Conferencistas" presenta a los oradores o conferencistas que participarán en las conferencias. Esta sección incluye información como el nombre, descripción breve y en botón para redirigir a la sección "Descripción completa" la cual muestra información más a detalle sobre los conferencistas y enlaces a sus redes sociales, como se muestra en las

Fig 17 y Fig 18. El objetivo de la sección "Conferencistas" es informar a los asistentes potenciales sobre quiénes son los oradores y qué pueden esperar de sus presentaciones.

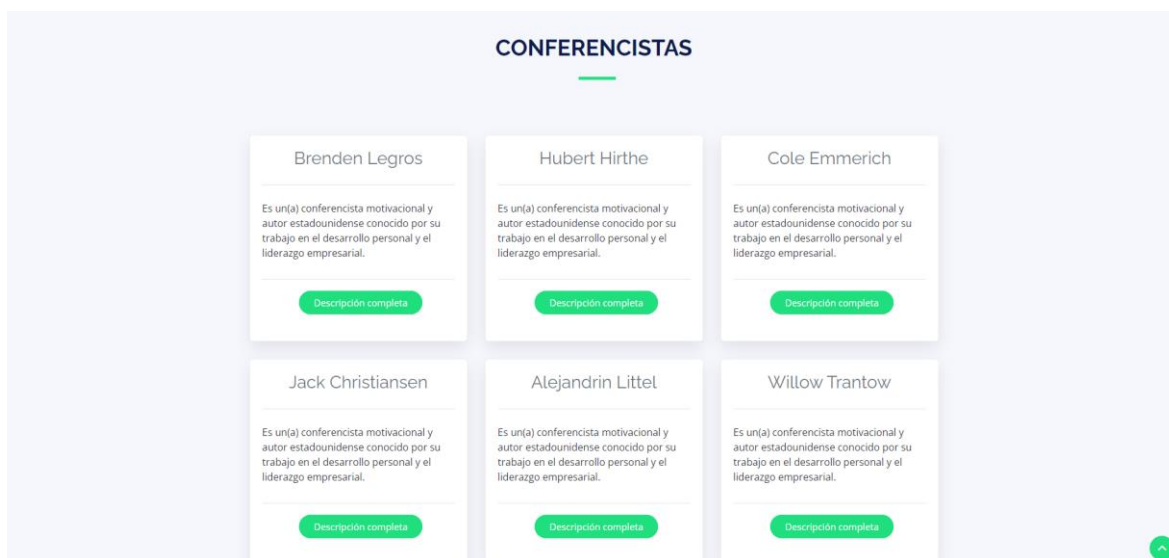


Fig 17. Sección de conferencistas



Fig 18. Sección de detalles de conferencista

La sección "Horarios" detalla el programa de actividades para la conferencia. El objetivo es proporcionar una visión general del programa de actividades para la conferencia y ayudar a los asistentes a planificar su tiempo. En general esta sección es una parte importante de la planificación de la conferencia y ayudará a asegurarse de que los asistentes tengan una experiencia satisfactoria y productiva, como se muestra en la **Fig 19**.

HORARIOS

Revisa el horario de las actividades que se ofrecen

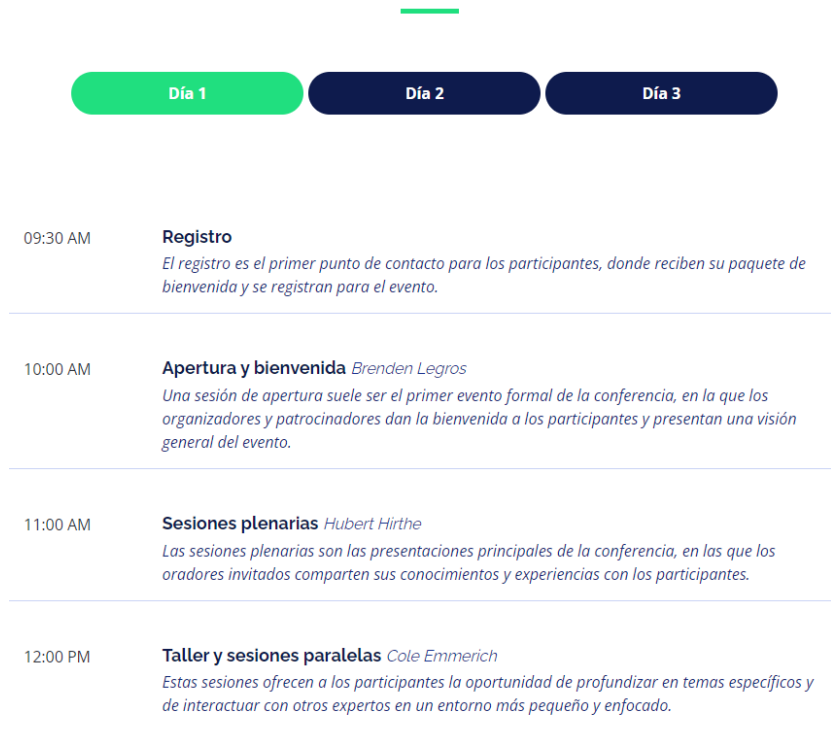


Fig 19. Sección de horarios

La sección "Ubicaciones" proporciona información sobre los lugares relevantes para las conferencias. Esta sección puede incluir información sobre la ubicación geográfica, la dirección, el transporte, el alojamiento, la comida, etc.

El objetivo de la sección es ayudar a los asistentes a planificar su viaje y asegurarse de que tengan toda la información necesaria para asistir. Además de mostrar la ubicación mediante la interacción con el mapa de Google el cual fue implementado por consumo de la API de Google *Maps*, como se muestra en las **Fig 20** y **Fig 21**.



Fig 20. Sección de ubicaciones

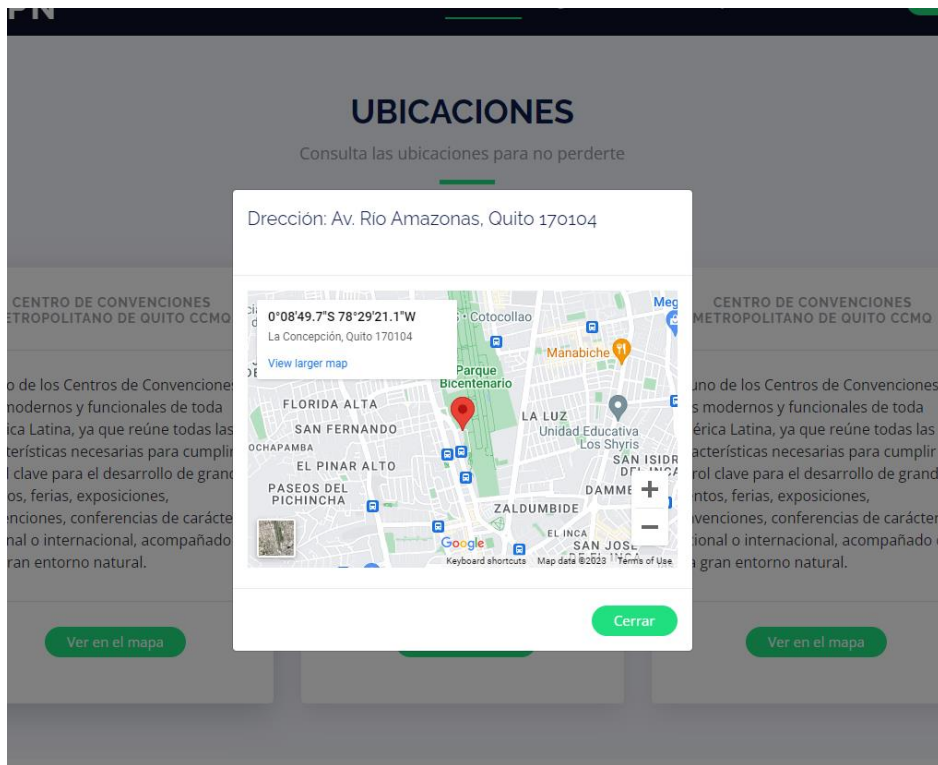
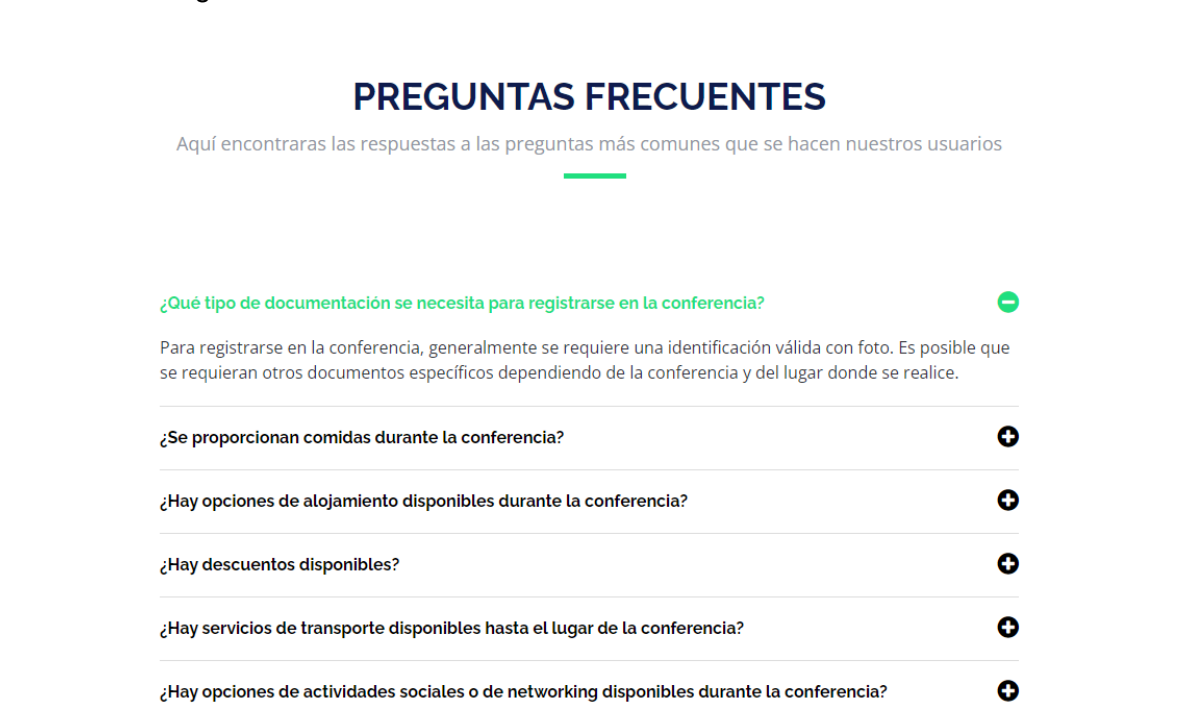


Fig 21. Modal del mapa

La sección "Preguntas Frecuentes" proporciona respuestas a preguntas comunes sobre un producto, servicio, conferencias o tema en general. La sección "Preguntas Frecuentes" es una forma efectiva de proporcionar información útil a los usuarios de manera rápida.

El principal objetivo de la sección "Preguntas Frecuentes" es mejorar la experiencia del usuario al responder a las preguntas más comunes de manera clara y concisa, y reducir la necesidad de contactar al servicio al cliente, como se muestra en las . No se encuentra el origen de la referencia..

PREGUNTAS FRECUENTES

Aquí encontraras las respuestas a las preguntas más comunes que se hacen nuestros usuarios

¿Qué tipo de documentación se necesita para registrarse en la conferencia?	⊖
Para registrarse en la conferencia, generalmente se requiere una identificación válida con foto. Es posible que se requieran otros documentos específicos dependiendo de la conferencia y del lugar donde se realice.	
¿Se proporcionan comidas durante la conferencia?	⊕
¿Hay opciones de alojamiento disponibles durante la conferencia?	⊕
¿Hay descuentos disponibles?	⊕
¿Hay servicios de transporte disponibles hasta el lugar de la conferencia?	⊕
¿Hay opciones de actividades sociales o de networking disponibles durante la conferencia?	⊕

Fig 22. Sección de preguntas frecuentes

La sección "Comprar *Tickets*" permite a los usuarios comprar entradas para las conferencias. Esta sección incluye información sobre el tipo de entradas disponibles y los precios. El objetivo de la sección "Comprar *Tickets*" es ofrecer una experiencia de compra sencilla y eficiente para los usuarios, permitiéndoles adquirir sus entradas en línea sin tener que recurrir a una taquilla física, como se muestra en la **Fig 23**. El método de pago definido para el sistema se realiza mediante *PayPal*.

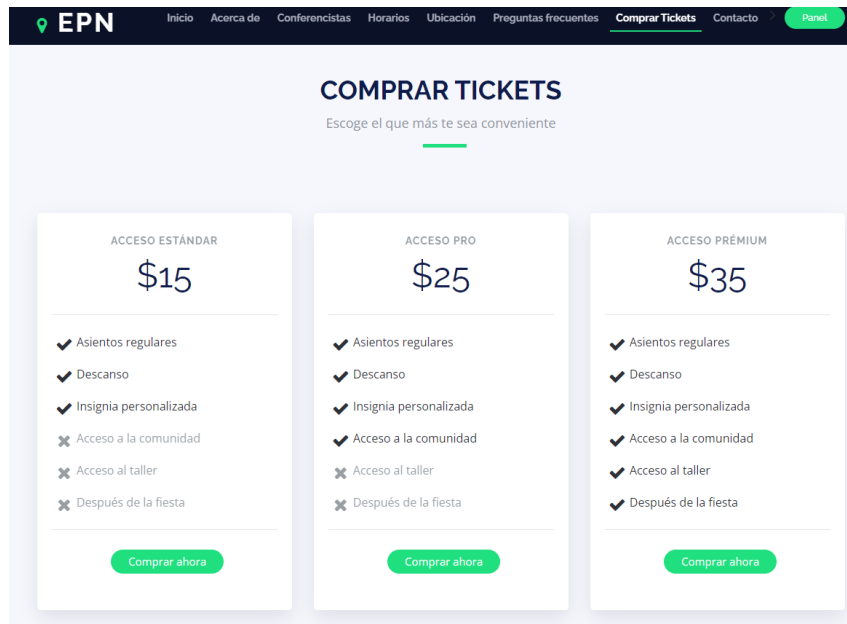


Fig 23. Sección de comprar *tickets*.

La sección "Contactos" proporciona información sobre cómo ponerse en contacto con la persona responsable del sitio web en este caso el administrador. Esta sección incluye información como el número de teléfono, el correo electrónico y un formulario de contacto en línea.

El objetivo de la sección "Contactos" es brindar a los usuarios una forma conveniente de comunicarse con el responsable del sitio web. Esta sección también puede ser utilizada para recopilar información de los usuarios y mejorar el servicio al usuario, y es una parte importante de la construcción de una relación de confianza con los clientes, como se muestra en la **Fig 24**.

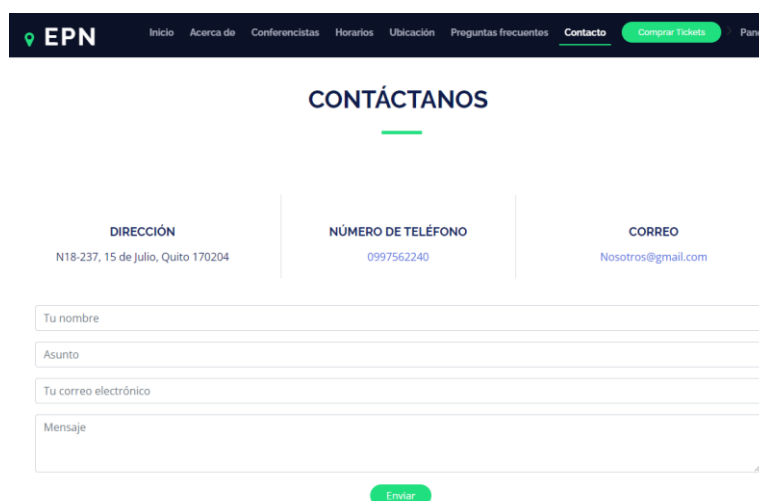


Fig 24. Sección de contactos

Un ejemplo del uso del formulario de contacto, las validaciones respectivas y la notificación de correo enviado, se muestran en las Fig 25, Fig 26 y Fig 27.

CONTÁCTANOS

DIRECCIÓN

N18-237, 15 de Julio, Quito 170204

NÚMERO DE TELÉFONO

0997562240

CORREO

Nosotros@gmail.com

Sebastián Valencia

correo prueba

svalencia310@gmail.com

Este sistema web de gestión de conferencias es una aplicación en línea diseñada para ayudar a los organizadores de conferencias a administrar y coordinar todos los aspectos relacionados con un evento. Este tipo de sistema incluye funcionalidades como la gestión de ubicaciones de eventos, la gestión de pagos, la gestión de horarios, la gestión de conferencistas.

Enviar

Fig 25. Sección de contactos.



Fig 26. Sección de contactos, notificación

CONTÁCTANOS

Si tienes dudas o problemas comunícate con nuestro soporte

DIRECCIÓN N18-237, 15 de Julio, Quito 170204	NÚMERO DE TELÉFONO 0997562240	CORREO svalencia310@gmail.com
--	---	---

Sebastián Valencia
Asunto
Tu correo electrónico
Mensaje

Enviar

Fig 27. Sección de contactos y validación de ingreso de datos.

Prueba unitaria Sprint 2

Este código define una función pública llamada *testAdminTest()*, que se utiliza para realizar una prueba automatizada. La prueba verifica si la ruta */admin* devuelve una respuesta HTTP 200, que indica que la página se cargó correctamente. Para lograr esto, la función utiliza el método *get()* del objeto *\$this* (que hace referencia a una instancia de la clase *TestCase*), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método *assertStatus()* para verificar si la respuesta recibida tiene un estado HTTP 200, que indica que la solicitud se realizó correctamente. Muestra en la Fig 28.

```
public function testAdminTest()  
{  
    $response = $this->get('/admin');  
    $response->assertStatus(200);  
}
```

Fig 28. Prueba unitaria #2. Acceso al panel de ajustes.

La Fig 29, muestra el resultado que indica que la prueba automatizada *testAdminTest* del archivo *testAdminTest.php* ubicado en la carpeta *Tests\Unit* ha pasado exitosamente. La prueba verificó si la página de login se carga correctamente. La solicitud (Request) es una petición HTTP GET enviada a la URL <http://34.246.221.132/admin>. El método utilizado para

realizar la solicitud es GET. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 200 OK, lo que significa que la solicitud se realizó correctamente y la página correspondiente al *admin* se cargó sin problemas.

```
Request URL: http://34.246.221.132/admin
Request Method: GET
Status Code: 200 OK

PASS Tests\Unit\testAdminTest
✓ admin
Tests: 1 passed
Time: 0.01s
```

Fig 29. Resultados prueba unitaria #2

Sprint 3. Módulo de sección de pagos


En el *Sprint 3*, se encuentran las secciones "Pagos con el uso de la API de *PayPal*" que permite a los usuarios realizar pagos en línea a través de la plataforma de pagos en línea. La API (*Application Programming Interface*) de *PayPal* son herramientas y recursos que permiten a los desarrolladores integrar las funciones de *PayPal* en su sitio web. La integración de la API de *PayPal* en la sección de pagos permite a los usuarios realizar pagos de manera segura y conveniente a través de su cuenta.

Tareas asignadas para el *Sprint 3*:

- Redirección a la página de *PayPal*.
- Implementar las interfaces para las transiciones entre páginas.
- Prueba unitaria *Sprint 3*.

Redirección a la página de *PayPal*

El objetivo de la sección "Pagos con el uso de la API de *PayPal*" es ofrecer una experiencia de pago fácil y segura para los usuarios, permitiéndoles realizar pagos en línea a través de su cuenta de *PayPal* sin tener que solicitarle información sensible, como el número de su tarjeta de crédito. El uso del servicio de se lo realiza a través de la sección "Comprar Tickets" la cual nos redirige a los respectivos formularios de *PayPal*, como se muestra en la Fig 30.



Pagar con PayPal

Con una cuenta de Paypal, usted reunirá los requisitos para gozar de la Devolución por Nuestra Cuenta, la Protección al Comprador y mucho más.

Correo electrónico o número de celular
sb-7dvlb24874634@business.example.com

Ingrese su contraseña
•••••••

[¿Ha olvidado su contraseña?](#)

[Iniciar sesión](#)

[Abrir una cuenta](#)

[Cancelar y volver a Test Store](#)

[Español](#) | [English](#) | [Français](#) | [中文](#)

Fig 30. Redirección al formulario de *PayPal*

Una vez ingresado a la cuenta de *PayPal* se encuentra la información del precio y las formas de pago, como se muestra en la **Fig 31**.

The screenshot shows a checkout interface. At the top, there's a header with 'ID', a PayPal logo, and '\$25.00'. Below that, the shipping address is 'Ship to John Doe, 1 Main St, San Jose, CA 95131' with a 'Change' link. The 'Pay with' section has a 'Split' toggle set to 'Off'. Three payment options are listed: 'CREDIT UNION 1' (selected) with a 'Make this my preferred way to pay' checkbox, 'Visa Credit', and 'PayPal Credit'. A '+ Add debit or credit card' link is at the bottom. A large blue button says 'Continue to Review Order'. Below it are links for 'Payment method rights' and 'Cancel and return to Test Store'.

Fig 31. Formulario de compra con el respectivo precio

Implementar las interfaces para las transiciones entre páginas

Después de confirmar la compra el usuario será redirigido a la página principal donde para comprobar si la compra se ha efectuado debe dirigirse a la sección de “Comprar Tickets” en donde aparecerá una notificación confirmando el pago o la cancelación de este, como se muestra en las Fig 32 y Fig 33.

The screenshot shows a notification titled 'COMPRAR TICKETS'. A green banner at the top says 'Transacción completada con éxito!'. Below are three ticket options: 'ACCESO ESTÁNDAR' for \$15, 'ACCESO PRO' for \$25, and 'ACCESO PREMIUM' for \$35. Each option lists features with checkmarks (Asientos regulares, Descanso, Insignia personalizada) and features with X marks (Acceso a la comunidad, Acceso al taller, Después de la fiesta). Each option has a green 'Comprar ahora' button.

Fig 32. Notificación de transacción completada

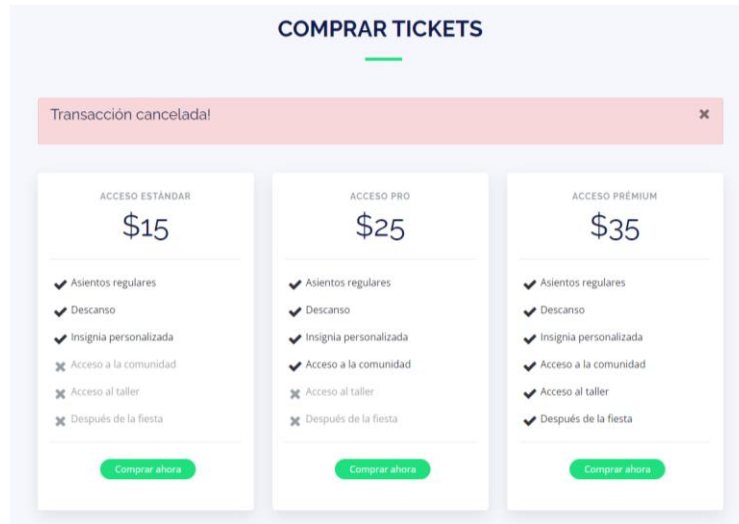


Fig 33. Notificación de transacción cancelada

Prueba unitaria Sprint 3

La prueba verifica si la ruta `/charge` devuelve una respuesta HTTP 302, que indica que la página se redirigió correctamente. Para lograr esto, la función utiliza el método `post()` del objeto `$this` (que hace referencia a una instancia de la clase `TestCase`), que realiza una solicitud POST a la ruta especificada. Luego, utiliza el método `assertStatus()` para verificar si la respuesta recibida tiene un estado HTTP 302, que indica que la solicitud se realizó correctamente. Muestra en la Fig 34.

```
public function testChargeTest()
{
    $response = $this->post('/charge');
    $response->assertStatus(302);
}
```

Fig 34. Prueba unitaria #3 - Redirección a los pagos

La Fig 35, muestra el resultado que indica que la prueba automatizada `testChargeTest` del archivo `testChargeTest.php` ubicado en la carpeta `Tests\Unit` ha pasado exitosamente. La prueba verificó si la página de `charge` se redirige correctamente. La solicitud (*Request*) es una petición HTTP POST enviada a la URL `http://34.246.221.132/charge`. El método utilizado para realizar la solicitud es POST. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 302 Found, lo que significa que la solicitud se realizó correctamente y la página correspondiente al `charge` se redirigió la solicitud a otra ubicación sin problemas.

```

Request URL: http://34.246.221.132/charge
Request Method: POST
Status Code: 302 Found

PASS Tests\Unit\testChargeTest
✓ charge

Tests: 1 passed
Time: 0.01s

```

Fig 35. Resultados prueba unitaria #3.

Sprint 4. Módulo de gestión.

La sección "Panel de Administración" es un área restringida que proporciona acceso a herramientas y recursos esenciales para administrar y mantener el sitio web. Este panel es accesible solo para usuarios autorizados, como los administradores o los desarrolladores del sitio.

Tareas asignadas para el *Sprint 4*:

- Implementación de interfaz de edición de configuraciones de la página.
- Implementación de validaciones de las secciones de gestión.
- Prueba unitaria *Sprint 4*.

Implementación de interfaz de edición de configuraciones de la página.

El panel de administración incluye funciones como la gestión de usuarios, la publicación y edición de contenido, la gestión de configuraciones del sitio, etc. El objetivo de la sección "Panel de Administración" es proporcionar una herramienta fácil de usar y accesible para los usuarios autorizados que permita la administración eficiente y efectiva del sitio web.

La sección de configuraciones corresponde a la gestión del contenido que se muestra en la página principal, como se muestra en la **Fig 36**.

ID	Key	Valor	Acción
1	title	Conferencias Anuales De Marketing	Ver Editar
2	subtitle	10-12 Diciembre, Cd Bicentenario	Ver Editar
3	youtube_link	https://www.youtube.com/watch?v=A7lclDQmkkv	Ver Editar
4	about_description	Está ubicado en un sector con un clima privilegiado. Tres rutas de acceso de primer orden. Cerca de centros escolares. Cuenta con líneas de buses y todos los servicios básicos. Estructura de hormigón, sísmorresistente.	Ver Editar
5	about_where	Cd Bicentenario	Ver Editar
6	about_when	Lunes a Miércoles 10-12 Diciembre	Ver Editar
7	about_description_2	Está ubicado en un sector con un clima privilegiado. Tres rutas de acceso de primer orden. Cerca de centros escolares. Cuenta con líneas de buses y todos los servicios básicos. Estructura de hormigón, sísmorresistente.	Ver Editar

Fig 36. Configuraciones

En cuanto a la visualización de la parte de creación y edición de todas las secciones de gestión tienen controles y validaciones como mensajes de confirmación de guardado, edición y eliminación. El ingreso de datos está controlado con validaciones y mensajes personalizados que ayudan a que los datos sean correctos y no causen conflicto al momento de ser visualizados en la página principal. La visualización de las diferentes secciones de gestión se encuentra a partir de la **Fig 37** hasta **Fig 39**.

Agregar Conferencista

Conferencista

ID	Nombre	Descripción	Twitter	Facebook	LinkedIn	Acción
1	Brenden Legros	Es un(a) conferencista motivacional y autor estadounidense conocido por su trabajo en el desarrollo personal y el liderazgo empresarial.	https://twitter.com/	https://www.facebook.com/	https://www.linkedin.com/	Ver Editar Borrar
2	Hubert Hirthe	Es un(a) conferencista motivacional y autor estadounidense conocido por su trabajo en el desarrollo personal y el liderazgo empresarial.	https://twitter.com/	https://www.facebook.com/	https://www.linkedin.com/	Ver Editar Borrar
3	Cole Emmerich	Es un(a) conferencista motivacional y autor estadounidense conocido por su trabajo en el desarrollo personal y el liderazgo empresarial.	https://twitter.com/	https://www.facebook.com/	https://www.linkedin.com/	Ver Editar Borrar
4	Jack Christiansen	Es un(a) conferencista motivacional y autor estadounidense conocido por su trabajo en el desarrollo personal y el liderazgo empresarial.	https://twitter.com/	https://www.facebook.com/	https://www.linkedin.com/	Ver Editar Borrar
5	Alejandrin Littel	Es un(a) conferencista motivacional y autor estadounidense conocido por su trabajo en el desarrollo personal y el liderazgo empresarial.	https://twitter.com/	https://www.facebook.com/	https://www.linkedin.com/	Ver Editar Borrar
6	Willow Trantow	Es un(a) conferencista motivacional y autor estadounidense conocido por su trabajo en el desarrollo personal y el liderazgo empresarial.	https://twitter.com/	https://www.facebook.com/	https://www.linkedin.com/	Ver Editar Borrar

Fig 37. Gestión de conferencistas

Agregar Horario

Horario

ID	Número de día	Hora de inicio	Título	Sutítulo	Conferencista	Acción
1	1	09:30:00	Registro	El registro es el primer punto de contacto para los participantes, donde reciben su paquete de bienvenida y se registran para el evento.		Ver Editar Borrar
2	1	10:00:00	Apertura y bienvenida	Una sesión de apertura suele ser el primer evento formal de la conferencia, en la que los organizadores y patrocinadores dan la bienvenida a los participantes y presentan una visión general del evento.	Brenden Legros	Ver Editar Borrar
3	1	11:00:00	Sesiones plenarias	Las sesiones plenarias son las presentaciones principales de la conferencia, en las que los oradores invitados comparten sus conocimientos y experiencias con los participantes.	Hubert Hirthe	Ver Editar Borrar
4	1	12:00:00	Taller y sesiones paralelas	Estas sesiones ofrecen a los participantes la oportunidad de profundizar en temas específicos y de interactuar con otros expertos en un entorno más pequeño y enfocado.	Cole Emmerich	Ver Editar Borrar
5	1	14:00:00	Pausas y networking	Las pausas y las oportunidades de networking son momentos clave para que los participantes conozcan a otros asistentes, establezcan contactos y discutan temas relacionados con la conferencia.	Jack Christiansen	Ver Editar Borrar

Fig 38. Gestión de horarios

Agregar Ubicación

Ubicación

ID	Nombre	Dirección	Latitud	Longitud	Acción
1	Centro de Convenciones Metropolitano de Quito CCMQ	Av. Río Amazonas, Quito 170104	-0.147137986463	-78.4891802094	Ver Editar Borrar
2	Centro de Convenciones Metropolitano de Quito CCMQ	Av. Río Amazonas, Quito 170104	-0.147137986463	-78.4891802094	Ver Editar Borrar
3	Centro de Convenciones Metropolitano de Quito CCMQ	Av. Río Amazonas, Quito 170104	-0.147137986463	-78.4891802094	Ver Editar Borrar

Fig 39. Gestión de ubicaciones

Las demás secciones de gestión cuentan con la misma lógica de crear, editar y eliminar. Estas acciones solo se pueden realizar si el usuario cuenta con los permisos ya que la gestión se maneja por un sistema de roles y permisos, el administrador es que tiene el acceso a todas las secciones de gestión mientras que los usuarios regulares están restringidos a la función de solo lectura, de ser el caso se agregan roles específicos para una gestión determinada. La visualización de las diferentes secciones de gestión se encuentra a partir de la **Fig 40** hasta **Fig 42**.

Agregar Preguntas frecuentes

Preguntas frecuentes

ID	Pregunta	Respuesta	Acción
1	¿Qué tipo de documentación se necesita para registrarse en la conferencia?	Para registrarse en la conferencia, generalmente se requiere una identificación válida con foto. Es posible que se requieran otros documentos específicos dependiendo de la conferencia y del lugar donde se realice.	Ver Editar Borrar
2	¿Se proporcionan comidas durante la conferencia?	La disponibilidad de comidas durante la conferencia puede variar dependiendo del organizador y del lugar donde se realice el evento.	Ver Editar Borrar
3	¿Hay opciones de alojamiento disponibles durante la conferencia?	La disponibilidad de opciones de alojamiento durante la conferencia puede variar dependiendo del organizador y del lugar donde se realice el evento.	Ver Editar Borrar
4	¿Hay descuentos disponibles?	NO	Ver Editar Borrar

Fig 40. Gestión de preguntas frecuentes

Agregar Beneficios

Beneficios

ID	Nombre	Acción
1	Asientos regulares	Ver Editar Borrar
2	Descanso	Ver Editar Borrar
3	Insignia personalizada	Ver Editar Borrar
4	Acceso a la comunidad	Ver Editar Borrar
5	Acceso al taller	Ver Editar Borrar
6	Después de la fiesta	Ver Editar Borrar

Fig 41. Gestión de beneficios

Agregar Precio

Precio

ID	Nombre	Precio	Beneficios	Acción
1	Acceso estándar	15.00	Asientos regulares Descanso Insignia personalizada	Ver Editar Borrar
2	Acceso Pro	25.00	Asientos regulares Descanso Insignia personalizada Acceso a la comunidad	Ver Editar Borrar
3	Acceso Prémium	35.00	Asientos regulares Descanso Insignia personalizada Acceso a la comunidad Acceso al taller Después de la fiesta	Ver Editar Borrar

Fig 42. Gestión de precios con beneficios

Implementación de validaciones de las secciones de gestión

Las validaciones son una parte fundamental para garantizar la integridad de todos los datos y la funcionalidad del sistema en su conjunto. Las validaciones son procesos automatizados que se encargan de verificar que los datos ingresados en el sistema cumplan con ciertos criterios de calidad y coherencia, lo que permite evitar errores, inconsistencias y problemas de seguridad en el sistema.

A continuación, se mostrarán los controles y las validaciones de la sección de gestión de ubicaciones para no extender el documento con todas las demás secciones ya que la lógica de los controles y validaciones es la misma y sería redundar innecesariamente, como se muestra desde la **Fig 43** hasta la **Fig 45**.

The screenshot shows a confirmation dialog box at the top with the text "34.246.221.132 dice" and "Está seguro?". Below the dialog is a green button labeled "Agregar Ubicación" and two buttons labeled "Aceptar" and "Cancelar". Below this is a table titled "Ubicación" with the following data:

ID	Nombre	Dirección	Latitud	Longitud	Acción
1	Centro de Convenciones Metropolitano de Quito CCMQ	Av. Río Amazonas, Quito 170104	-0.147137986463	-78.4891802094	Ver Editar Borrar
2	Centro de Convenciones Metropolitano de Quito CCMQ	Av. Río Amazonas, Quito 170104	-0.147137986463	-78.4891802094	Ver Editar Borrar
3	Centro de Convenciones Metropolitano de Quito CCMQ	Av. Río Amazonas, Quito 170104	-0.147137986463	-78.4891802094	Ver Editar Borrar

Fig 43. Mensaje de confirmación de eliminación

The screenshot shows a form titled "Crear Ubicación" with the following fields and a validation error:

- Nombre***: A text input field that is empty and highlighted with a blue border.
- Dirección***: A text input field with a validation error message: "Ingresa el nombre!".
- Latitud***: A text input field.
- Longitud***: A text input field.
- Descripción**: A text area.

At the bottom of the form are two buttons: "Guardar" (red) and "volver a la lista" (grey).

Fig 44. Validación de campos

Crear Ubicación

Nombre*
Cd Bicentenario

Dirección*
|

Latitud*
|

Longitud*
|

Descripción

Guardar

volver a la lista

Ingresar la dirección!

Fig 45. Validación de campos

Al validar los campos, se asegura que los datos ingresados por el usuario cumplan con ciertos requisitos o restricciones antes de que sean procesados y almacenados en una base de datos, como se muestra desde la **Fig 46** hasta la **Fig 48**.

Crear Ubicación

Nombre*
Cd Bicentenario

Dirección*
N18-237, 15 de Julio, Quito 170204

Latitud*
|

Longitud*
|

Descripción

Guardar

volver a la lista

Ingresar la latitud!

Fig 46. Validación de campos

34,246,221.132 dice
Desea guardar los cambios?

Crear Ubicación

Nombre*
Cd Bicentenario

Dirección*
N18-237, 15 de Julio, Quito 170204

Latitud*
-0.0634395

Longitud*
-78.4517485

Descripción
N18-237, 15 de Julio, Quito 170204

Guardar

volver a la lista

Fig 47. Confirmación de guardar

34,246,221.132 dice
Está seguro de regresar sin guardar?

Crear Ubicación

Nombre*
Cd Bicentenario

Dirección*
N18-237, 15 de Julio, Quito 170204

Latitud*
-0.0634395

Longitud*
-78.4517485

Descripción
N18-237, 15 de Julio, Quito 170204

Guardar

volver a la lista

Fig 48. Confirmación de regresar sin guardar

Prueba unitaria Sprint 4

Este código define una función pública llamada `testSettingsTest()`, que se utiliza para realizar una prueba automatizada. La prueba verifica si la ruta `/settings` devuelve una respuesta HTTP 200, que indica que la página se cargó correctamente. Para lograr esto, la función utiliza el método `get()` del objeto `$this` (que hace referencia a una instancia de la clase `TestCase`), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método `assertStatus()` para verificar si la respuesta recibida tiene un estado HTTP 200, que indica que la solicitud se realizó correctamente. Muestra en la **Fig 49**.

```
public function testSettingsTest()
{
    $response = $this->get('/admin/settings');
    $response->assertStatus(200);
}
```

Fig 49. Prueba unitaria #4. Acceso a configuraciones

La **Fig 50**, muestra el resultado que indica que la prueba automatizada `testSettingsTest` del archivo `testSettingsTest.php` ubicado en la carpeta `Tests\Unit` ha pasado exitosamente. La prueba verificó si la página de `admin/settings` se carga correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a la URL `http://34.246.221.132/admin/settings`. El método utilizado para realizar la solicitud es GET. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 200 OK, lo que significa que la solicitud se realizó correctamente y la página correspondiente al `admin/settings` se cargó sin problemas.

```
Request URL: http://34.246.221.132/admin/settings
Request Method: GET
Status Code: 200 OK

PASS Tests\Unit\testSettingsTest
✓ settings

Tests: 1 passed
Time: 0.01s
```

Fig 50. Prueba unitaria #4

Sprint 5. Módulo de gestión usuarios.

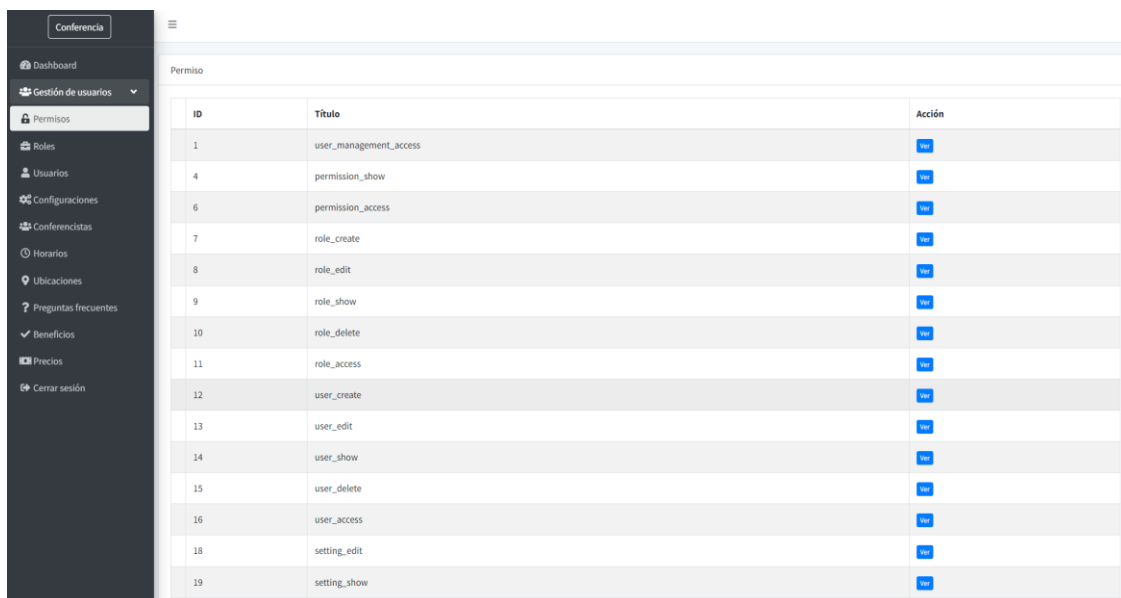
La sección "Gestión de Usuarios" es una parte de un panel de administración que permite a los administradores controlar y gestionar los usuarios registrados. Esta sección incluye funciones para agregar, editar y eliminar usuarios, así como para asignar roles y permisos a los usuarios.

Tareas asignadas para el Sprint 5:

- Implementar la interfaz gestión de roles y permisos.
- Implementar la interfaz gestión de usuarios.
- Prueba unitaria *Sprint 5*.

Implementar la interfaz gestión de roles y permisos

La gestión de usuarios es importante para el sitio web, ya que permite controlar quiénes pueden acceder a áreas restringidas y realizar determinadas acciones en el sitio. Por ejemplo, los administradores pueden asignar roles de editor o moderador a los usuarios, permitiéndoles publicar o aprobar contenido en el sitio. También les permite controlar y gestionar los permisos y roles de los usuarios, asegurándose de que los usuarios estén autorizados para realizar las acciones que desean realizar. En las **Fig 51** y **Fig 52**.



ID	Título	Acción
1	user_management_access	ver
4	permission_show	ver
6	permission_access	ver
7	role_create	ver
8	role_edit	ver
9	role_show	ver
10	role_delete	ver
11	role_access	ver
12	user_create	ver
13	user_edit	ver
14	user_show	ver
15	user_delete	ver
16	user_access	ver
18	setting_edit	ver
19	setting_show	ver

Fig 51. Visualización de permisos

Agregar Rol

Rol

ID	Título	Permisos	Acción
1	Admin	user_management_access permission_show permission_access role_create role_edit role_show role_delete role_access user_create user_edit user_show user_delete user_access setting_edit setting_show setting_access speaker_create speaker_edit speaker_show speaker_delete speaker_access schedule_create schedule_edit schedule_show schedule_delete schedule_access venue_create venue_edit venue_show venue_delete venue_access faq_create faq_edit faq_show faq_delete faq_access amenity_create amenity_edit amenity_show amenity_delete amenity_access price_create price_edit price_show price_delete price_access	Ver Editar Borrar
2	User	speaker_show speaker_access schedule_show schedule_access venue_show venue_access faq_show faq_access amenity_show amenity_access price_show price_access user_management_access role_show role_access user_show user_access	Ver Editar Borrar

Fig 52. Gestión de roles

Implementar la interfaz gestión de usuarios

En la creación de usuarios se aplican los mismos controles que en las otras secciones, que los campos que lo requieran estén llenados, el nombre no debe contener números, el correo debe tener el formato correcto y no debe ser igual a otro correo ya registrado, para la edición de usuarios se aplican los mismos controles de validación, lo que se muestra en desde la Fig 53 hasta la Fig 56.

≡

Agregar Usuario

Usuario

ID	Nombre	Correo	Roles	Acción
1	Admin	admin@admin.com	Admin	Ver Editar Borrar
2	Sebastián	svalencia310@gmail.com	User	Ver Editar Borrar
3	jordi	jordictg312@gmail.com	User	Ver Editar Borrar

Fig 53. Gestión de usuarios

Crear Usuario

Nombre*

Correo*

Contraseña

Roles* [Seleccionar todo](#) [Deseleccionar todo](#)

Guardar

volver a la lista

Fig 54. Gestión de usuarios

Crear Usuario

Nombre*

Correo*

Contraseña

Roles* [Seleccionar todo](#) [Deseleccionar todo](#)

Guardar

volver a la lista

! Ingresa el nombre!

Fig 55. Validación de los campos

Este correo ya está registrado!

Crear Usuario

Nombre*

Correo*

Contraseña

Roles* [Seleccionar todo](#) [Deseleccionar todo](#)

[Guardar](#)

[volver a la lista](#)

Fig 56. Validación de correo existente

La asignación de roles consta del nombre del rol y los permisos de este, de la misma forma que las otras secciones esta cuenta con los controles de validación, como se muestra en las Fig 57 y Fig 58.

Crear Rol

Título*

Permisos* [Seleccionar todo](#) [Deseleccionar todo](#)

Ingresa el nombre del rol!

[Guardar](#)

[volver a la lista](#)

Fig 57. Validación de campos

Crear Rol

Título*

tester

Permisos* [Seleccionar todo](#) [Deseleccionar todo](#)

- user_management_access
- permission_show
- permission_access
- role_create
- role_edit
- role_show

Fig 58. Creación de roles

Se ha creado un rol de “tester” el cual solo tiene acceso a las secciones de beneficios y precios, el cual se muestra en la **Fig 59**.

Agregar Rol

Rol creado!

Rol

ID	Título	Permisos	Acción
1	Admin	user_management_access permission_show permission_access role_create role_edit role_show role_delete role_access user_create user_edit user_show user_delete user_access setting_edit setting_show setting_access speaker_create speaker_edit speaker_show speaker_delete speaker_access schedule_create schedule_edit schedule_show schedule_delete schedule_access venue_create venue_edit venue_show venue_delete venue_access faq_create faq_edit faq_show faq_delete faq_access amenity_create amenity_edit amenity_show amenity_delete amenity_access price_create price_edit price_show price_delete price_access	Ver Editar Borrar
2	User	speaker_show speaker_access schedule_show schedule_access venue_show venue_access faq_show faq_access amenity_show amenity_access price_show price_access user_management_access role_show role_access user_show user_access	Ver Editar Borrar
3	tester	amenity_create amenity_edit amenity_show amenity_delete amenity_access price_create price_edit price_show price_delete price_access	Ver Editar Borrar

Fig 59. Rol “tester” en el listado

Una vez creado el rol ya puede ser asignado al usuario en cuestión que al momento de ingresar al panel de administración solo tendrá acceso a las funciones designadas por el rol "tester", como se muestra en las Fig 60, Fig 61 y Fig 62.

Editar Usuario

Nombre*

Roles* [Seleccionar todo](#) [Deseleccionar todo](#)

Admin

User

tester

Fig 60. Asignación de rol

Agregar Usuario

Usuario actualizado! ✕

Usuario

ID	Nombre	Correo	Roles	Acción
1	Admin	admin@admin.com	Admin	Ver Editar Borrar
2	Sebastián	svalencia310@gmail.com	tester	Ver Editar Borrar
3	jordi	jordictg312@gmail.com	User	Ver Editar Borrar

Fig 61. Actualización de rol

ID	Nombre	Acción
1	Asientos regulares	Ver Editar Borrar
2	Descanso	Ver Editar Borrar
3	Insignia personalizada	Ver Editar Borrar
4	Acceso a la comunidad	Ver Editar Borrar
5	Acceso al taller	Ver Editar Borrar
6	Después de la fiesta	Ver Editar Borrar

Fig 62. Gestiones disponibles para el rol “tester”

Prueba unitaria Sprint 5

La prueba verifica si la ruta `/admin/users` devuelve una respuesta HTTP 200, que indica que la página se cargó correctamente. Para lograr esto, la función utiliza el método `get()` del objeto `$this` (que hace referencia a una instancia de la clase `TestCase`), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método `assertStatus()` para verificar si la respuesta recibida tiene un estado HTTP 200, que indica que la solicitud se realizó correctamente. Muestra en la **Fig 63**.

```
public function testUsersTest()
{
    $response = $this->get('/admin/users');
    $response->assertStatus(200);
}
```

Fig 63. Prueba unitaria #5. Acceso a manejo de usuarios

La **Fig 64**, muestra el resultado que indica que la prueba automatizada `testUsersTest` del archivo `testUsersTest.php` ubicado en la carpeta `Tests\Unit` ha pasado exitosamente. La prueba verificó si la página de `admin/users` se carga correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a la URL `http://34.246.221.132/admin/users`. El método utilizado para realizar la solicitud es GET. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 200 OK, lo que significa que la solicitud se realizó correctamente y la página correspondiente al `admin/users` se cargó sin problemas.

Request URL: <code>http://34.246.221.132/admin/users</code>	PASS Tests\Unit\testUsersTest
Request Method: GET	✓ users
Status Code: 200 OK	Tests: 1 passed
	Time: 0.01s

Fig 64. Resultado prueba unitaria #5

Sprint 6. Pruebas del sistema

Las pruebas de usabilidad y compatibilidad son dos tipos de pruebas necesarias en el desarrollo de un software. La prueba de usabilidad se enfoca en buscar como es su facilidad de uso y la experiencia del usuario, mientras que la prueba de compatibilidad se enfoca en garantizar que la aplicación funcione correctamente en diferentes plataformas y dispositivos. Ambas pruebas son importantes para asegurar la calidad y el rendimiento del sistema y mejorar la satisfacción del usuario.

Tareas asignadas para el Sprint 6:

- Pruebas unitarias.
- Pruebas compatibilidad.
- Pruebas usabilidad.
- Despliegue en servidor privado.

Pruebas unitarias

Estas pruebas deben ser capaces de ejecutarse sin la necesidad de intervención manual y pueden ser automatizadas. La rapidez de ejecución es importante para que las pruebas se puedan repetir tantas veces como sea necesario. Además, es fundamental que las pruebas cubran la mayoría del código de la aplicación y puedan ejecutarse en cualquier entorno sin depender del estado de este. De esta manera, las pruebas unitarias serán efectivas y valiosas en el proceso de desarrollo de software. [21].

La **Fig 65** y **Fig 66** presentan el resultado que indica que la prueba automatizada *test_abrirTest* del archivo *test_abrirTest.php* ubicado en la carpeta *Tests\Unit* ha pasado exitosamente. La prueba verificó si la página raíz ('/') se carga correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a la URL <http://34.246.221.132/>. El método utilizado para realizar la solicitud es GET. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 200 OK, lo que significa que la solicitud se realizó correctamente y la página correspondiente a la página raíz se cargó sin problemas. El resto de pruebas unitarias se encuentran en el **ANEXO II** de este documento.

```

14
15 public function test_abirTest()
16 {
17     $response = $this->get('/');
18
19     $response->assertStatus(200);
20 }
21 }
22

```

Fig 65. Prueba unitaria #6. Apertura de página raíz del sistema

The screenshot shows a test runner interface. On the left, it displays request details: 'Request URL: http://34.246.221.132/', 'Request Method: GET', and 'Status Code: 200 OK'. On the right, it shows the test result: 'PASS Tests\Unit\abirTest' with a green checkmark, 'abir test', 'Tests: 1 passed', and 'Time: 0.01s'.

Fig 66. Resultado prueba unitaria #6

Pruebas de compatibilidad

Estas pruebas ayudan a garantizar que la aplicación sea compatible con los diferentes entornos en los que se ejecutará, y a detectar errores o problemas que puedan surgir en un entorno específico. En la TABLA VI, se muestran los diferentes navegadores utilizados para verificar la adaptabilidad del contenido a cualquier dispositivo. Los detalles sobre las pruebas de compatibilidad realizadas y los resultados obtenidos se encuentran en el ANEXO II de este documento.

TABLA VI. Navegadores utilizados para pruebas de compatibilidad

NOMBRE	VERSIÓN	OBSERVACIÓN
<i>Brave</i>	1.47.186	Compatible
<i>Microsoft Edge</i>	109.0.1518.78	Compatible
<i>Google Chrome</i>	109.0.5414.120	Compatible

Pruebas de usabilidad

Estas pruebas implican la observación directa de los usuarios mientras realizan tareas típicas en el sistema *web* y la recopilación de datos para identificar la experiencia de usuario. Las pruebas de usabilidad de un sistema *web* pueden mejorar la eficiencia y la satisfacción del usuario, aumentar la retención de usuarios y mejorar la reputación del sitio *web*. Para este caso se realizó una encuesta con el fin de conocer la experiencia de usuario, como se muestra en la Fig 67.

SISTEMA WEB DE GESTIÓN DE CONFERENCIAS CON PAGOS EN LÍNEA

Los datos que proporcionen en este formulario serán utilizados para evaluar la facilidad de uso del sistema en cuestión. El estudio de usabilidad tiene como objetivo identificar problemas que puedan afectar negativamente la experiencia del usuario y mejorar la calidad de la aplicación. Es importante participar en este estudio para que tus comentarios y sugerencias puedan ser considerados y tomados en cuenta en la mejora del sistema.

svalencia310@gmail.com [Cambiar cuenta](#)

*Obligatorio

Correo electrónico *

Tu dirección de correo electrónico _____

¿Te resultó fácil registrarte en el sistema web?

Muy fácil

Ni fácil ni difícil

Muy difícil

Fig 67. Encuesta de usabilidad del sistema

Después de analizar las respuestas proporcionadas por los usuarios encuestados, podemos concluir que en general, la mayoría de los usuarios encontraron que el proceso de registro y la navegación por el sistema fue fácil. Sin embargo, algunos usuarios tuvieron dificultades para encontrar la información que necesitaban y para gestionar los roles y permisos de los usuarios en el panel de administración del sistema *web*. En general, la mayoría de los usuarios no tuvo complicaciones en gestionar los conferencistas, ubicaciones, preguntas frecuentes y precios y beneficios en el panel de administración del sistema. Los detalles sobre las pruebas de usabilidad realizadas y sus resultados se encuentran en el **ANEXO II** de este documento.

Despliegue en servidor privado

El despliegue del proyecto se realizó en un servidor con una dirección IP pública y se tuvo que realizar algunas consideraciones para su implementación. Es importante destacar que el proceso de despliegue en un servidor con dirección IP pública puede variar según el servidor y la configuración específica que se utilice. Los detalles del despliegue se encuentran en el **ANEXO II** de este documento.

4 CONCLUSIONES

Conclusiones obtenidas a lo largo del desarrollo del trabajo de integración curricular.

- El sistema web cumple con los objetivos y el alcance establecidos, ya que permite a los usuarios acceder a sistema web de gestión de conferencias y a las funcionalidades acorde a los requerimientos documentados en historias de usuario y artefactos de *Scrum*.
- En base a las pruebas de usabilidad se deduce que la interfaz de la página principal resulta fácil de usar mientras que el panel de administración no resulta tan sencillo de usar para algunos usuarios, por lo que en un futuro se debe trabajar en este aspecto.
- Se integra correctamente con otras aplicaciones y servicios en línea, como la implementación de un sistema de pagos mediante PayPal.
- El sistema *web* cumple con los requisitos de compatibilidad al ser usado desde diferentes navegadores inclusive se comprobó el correcto funcionamiento y adaptación desde dispositivos móviles.
- *Scrum* permitió un desarrollo ordenado y eficiente del sitio *web*. La metodología ágil se utilizó en el trabajo de integración curricular, permitiendo completar el sistema mediante etapas y validar las funcionalidades en los períodos de tiempo planificados.

5 RECOMENDACIONES

recomendaciones que surgieron durante del desarrollo del trabajo de integración curricular.

- Se recomienda para el sistema *web* desarrollado, tenga una segunda recopilación de requerimientos con la finalidad de mejorar la experiencia del usuario.
- La función de restablecimiento de contraseña podría automatizarse en lugar de contactar directamente al administrador del sitio *web*, por tanto, es un apartado que podría ser implementado en futuras actualizaciones del sistema.
- La asignación de roles no debería ser para muchos usuarios ya que el sistema y las funciones a gestionar no lo requieren por lo que tener demasiados usuarios con roles puede causar conflicto al momento de realizar las diferentes gestiones.
- Para mantener la integridad del sistema tanto los permisos como las configuraciones están restringidas en cuanto a acciones por lo que se recomienda no solicitar acceso completo a estas funciones.
- El sistema *web* se encuentra alojado en un servidor con IP pública debido a que de momento no cuenta con un dominio por lo que se recomienda hacer uso de la dirección proporcionada en el **ANEXO IV**, del documento.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Ati, «La importancia de las videoconferencias» 12 de mayo de 2022 [En línea]. Available: <https://tecnologiaintegrada.com.mx/2022/05/12/la-importancia-de-las-videoconferencias/> [Último acceso: 2022 noviembre 9].
- [2] smartspeakers, «Conferencias en línea, una tendencia en crecimiento » 17 de marzo de 2020. [En línea]. Available: <https://www.smartspeakersweb.com/item/23-conferencias-en-linea-una-tendencia-en-crecimiento> [Último acceso: 2022 noviembre 9].
- [3] L. Alegsa, « Definición de Framework de desarrollo (informática)» 6 de julio de 2016 [En línea]. Available: <http://www.alegsa.com.ar/Dic/framework.php> [Último acceso: 2022 noviembre 11].
- [4] Strapp Inc, «¿Qué es una Aplicación Web?» 30 de julio de 2019 [En línea]. Available: <https://www.strappinc.com/blog/strapp-datos/que-es-una-aplicacion-web> [Último acceso: 2022 noviembre 12].
- [5] M. Coppola, «Desarrollo web: qué es, etapas y principales lenguajes» 20 de julio de 2022. [En línea]. Available: <https://blog.hubspot.es/website/que-es-desarrollo-web> [Último acceso: 2022 noviembre 15].
- [6] Epitech, «Qué es el Frontend y para qué sirve en diseño web,» , 16 de junio de 2021. [En línea]. Available: <https://www.epitech-it.es/frontend/>. [Último acceso: 2022 noviembre 17].
- [7] RedHat «¿Qué es la metodología ágil? » 19 de julio de 2022 [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology> [Último acceso: 2022 noviembre 19].
- [8] S. Pursell, « Metodología Agile: qué es y cómo aplicarla a tu proyecto » 1 de diciembre de 2021 [En línea]. Available: <https://blog.hubspot.es/marketing/metodologia-agile> [Último acceso: 2022 noviembre 22].
- [9] CGFGlobal « Roles en Scrum » [En línea]. Available: <https://edu.cgfglobal.org/es/scrum/roles-en-scrum/1/> [Último acceso: 2022 noviembre 27].
- [10] R. Chito, « Artefactos de Scrum » 8 de abril de 2022 [En línea]. Available: <https://blog.buhoos.com/artefactos-de-scrum/> [Último acceso: 2022 noviembre 28]
- [11] S. vergara, « ¿Qué son las historias de usuario? » 10 de agosto de 2021 [En línea]. Available: <https://www.itdo.com/blog/que-son-las-historias-de-usuario/> [Último acceso: 2022 noviembre 30].
- [12] M. Garcia, « ¿Qué es el PRODUCT BACKLOG?» 20 de abril de 2020 [En línea]. Available: <https://ittude.com.ar/b/scrum/que-es-el-product-backlog/> [Último acceso: 2022 diciembre 3].
- [13] G. Blandino « Figma: Figma: qué es y cómo funciona» [En línea]. Available: <https://www.pixartprinting.es/blog/figma-que-es/> [Último acceso: 2022 diciembre 5].
- [14] E. Palmero, « Arquitectura de software: descubre en qué consiste y cuáles son las mejores prácticas» 9 de julio de 2021 [En línea]. Available: <https://www.eykkon.com/blog/arquitectura-de-software/> [Último acceso: 2022 diciembre 9].

- [15] desarrolloweb « Qué es MVC » 28 Julio de 2020 [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-mvc.html> [Último acceso: 2022 diciembre 10].
- [16] Documentación de laravel « Meet Laravel » (s.f.). [En línea]. Available: <https://laravel.com/docs/9.x> [Último acceso: 2022 diciembre 11].
- [17] A. Zola, « Definition Bootstrap » (s.f.). [En línea]. Available: <https://www.techtarget.com/whatis/definition/bootstrap> [Último acceso: 2022 diciembre 11].
- [18] World Wide Web Consortium (W3C), « Web technologies » 21 Diciembre 2021 [En línea]. Available: <https://www.w3.org/standards/webdesign/> [Último acceso: 2022 diciembre 11].
- [19] Crosshero « Paypal - ¿qué es y cómo configurar? » (s.f.). [En línea]. Available: <https://support.crosshero.com/es/articles/5397051-paypal-que-es-y-como-configurar#:~:text=La%20API%20REST%20de%20PayPal,autom%C3%A1tica%22%20cuando%20creamos%20su%20suscripci%C3%B3n.> [Último acceso: 2022 diciembre 18].
- [20] Ajgalleo.gitbook.io « Plantillas mediante Blade » (s.f.). [En línea]. Available: https://ajgalleo.gitbook.io/laravel-5/capitulo_1/capitulo_1_plantillas [Último acceso: 2022 diciembre 18].
- [21] I. Landa « Características de una buena prueba unitaria » 18 de febrero de 2009. [En línea]. Available: <https://geeks.ms/ilanda/2009/02/18/caractersticas-de-una-buena-prueba-unitaria/> [Último acceso: 2023 enero 10].

7 ANEXOS

A continuación, se muestran los elementos que forman parte de los anexos del presente documento:

ANEXO I. Certificado de originalidad.

ANEXO II. Manual técnico.

ANEXO III. Manual de usuario.

ANEXO IV. Manual de instalación.

ANEXO I

Este anexo consiste en el certificado de originalidad emitido por el director del presente trabajo de integración curricular.



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 28 de 02 de 2023

De mi consideración:

Yo, MAYRA ISABEL ALVAREZ JIMÉNEZ, en calidad de Directora del Trabajo de Integración Curricular componente FRONTEND, titulado "SISTEMA WEB DE GESTIÓN DE CONFERENCIAS CON PAGOS EN LÍNEA", elaborado por el estudiante JORDI SEBASTIÁN VALENCIA CARTAGENA de la carrera en DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



Ing. Mayra Alvarez MSc.
Técnico docente EPN - ESFOT

ANEXO II

En este anexo se incluye una descripción detallada de cada uno de los requerimientos del sistema, así como las historias de usuario que corresponden a cada uno de ellos. Además, se presentan las pruebas necesarias para validar el correcto funcionamiento del sistema y asegurar que cumple con todos los requerimientos establecidos.

Recopilación de requerimientos

En la TABLA VII, se muestra los requerimientos que se han obtenido al inicio del proyecto de acuerdo con lo solicitado.

TABLA VII. Recopilación de requerimientos

Recopilación de requerimientos		
Tipo de sistema	ID-RR	Enunciado del Ítem
sistema web	RR001	Como usuario cliente, necesito visualizar la página informativa que me muestre los servicios del sistema web. Con la finalidad de navegar y conocer la página.
	RR002	Como usuario cliente, necesito registrarme en el sistema mediante el respectivo formulario de registro. Con la finalidad de ingresar a mi perfil con mi correo y contraseña.
	RR003	Como usuario cliente, necesito que al ingresar con mi cuenta ya registrada muestre la página de ajustes según el rol asignado por el administrador. Con la finalidad de conocer sobre qué información puedo usar.
	RR004	Como usuario cliente, necesito dirigirme a las diferentes secciones través de la propia página. Con la finalidad de conocer a las personas implicadas.
	RR005	Como usuario cliente, necesito dirigirme a la sección de pagos a través de la propia página. Con la finalidad de realizar los pagos respectivos.
	RR006	Como usuario cliente, necesito salir de la página con la opción de cerrar mi sesión.

		Con la finalidad de salir del sistema sin dejar mi perfil activo.
	RR007	Como usuario administrador, necesito ingresar a la página y visualizar mi perfil con la información de la conferencia, asignación de precios y editar características de la página. Con la finalidad de tener información sobre el estado del sistema web.
	RR008	Como usuario administrador, necesito publicar y actualizar información en la página. Con la finalidad de realizar los cambios según sea necesario.
	RR009	Como usuario administrador, necesito visualizar y gestionar los clientes registrados. Con la finalidad de conocer la información de los usuarios y asignar roles con los respectivos permisos según sea necesario.
	RR010	Como desarrollador, se requiere realizar las respectivas pruebas unitarias, de usabilidad, de compatibilidad y despliegue del sistema.

Historias de usuario

En esta sección se encuentran las historias de usuario una vez concluida la recopilación de requerimientos a partir de la TABLA VIII.

TABLA VIII. Historia de usuario Nro.2

HISTORIA DE USUARIO	
Identificador (ID): HU002	Usuario: Cliente
Nombre Historia: Iniciar sesión y registrar usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Iteración Asignada: 2	
Responsable: Sebastián Valencia	
Descripción: El usuario cliente dispone de la opción de registro/inicio de sesión/recuperación de contraseña para acceder al sitio web mediante correo electrónico.	

Observación: El registro deberá incluir los campos. Nombre, apellido, correo electrónico y contraseña.

TABLA IX. Historia de usuario Nro.3

HISTORIA DE USUARIO	
Identificador (ID): HU003	Usuario: Cliente
Nombre Historia: Mostrar página de ajustes del usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Iteración Asignada: 3	
Responsable: Sebastián Valencia	
Descripción: El usuario cliente una vez registrado tendrá a disposición las diferentes funcionalidades designadas por el administrador.	
Observación: La página principal se mostrará a al usuario toda la información disponible que haya sido asignada por el administrador.	

TABLA X. Historia de usuario Nro.4

HISTORIA DE USUARIO	
Identificador (ID): HU004	Usuario: Cliente
Nombre Historia: Visualizar secciones de la página	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Iteración Asignada: 4	
Responsable: Sebastián Valencia	
Descripción: El usuario cliente dispone de ver las diferentes secciones que conforman la página de las cuales están el personal designado para la conferencia, así como información personal y redes sociales.	
Observación: El usuario navegará a través de la interfaz y podrá ver información de todas las secciones.	

TABLA XI. Historia de usuario Nro.5

HISTORIA DE USUARIO	
Identificador (ID): HU005	Usuario: Cliente
Nombre Historia: Mostrar sección de pagos	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Iteración Asignada: 5	
Responsable: Sebastián Valencia	
Descripción: Si el usuario cliente necesita de realizar los pagos tendrá la opción de redirigirse a la parte con los pagos.	
Observación: El método de pago se realizará a través de PayPal.	

TABLA XII. Historia de usuario Nro.6

HISTORIA DE USUARIO	
Identificador (ID): HU006	Usuario: Administrador
Nombre Historia: Mostrar interfaz de administrador	
Prioridad en negocio: alta	Riesgo en desarrollo: Bajo
Iteración Asignada: 6	
Responsable: Sebastián Valencia	
Descripción: El usuario administrador requiere de visualizar el estado de la página y opciones de gestión.	
Observación: Las opciones a mostrar son la edición de la página como crear, editar, eliminar, publicar información, apartado de pagos, usuarios registrados y asignación de roles y permisos.	

TABLA XIII. Historia de usuario Nro.7

HISTORIA DE USUARIO	
Identificador (ID): HU007	Usuario: Administrador
Nombre Historia: Mostrar gestión de información	
Prioridad en negocio: alta	Riesgo en desarrollo: Bajo
Iteración Asignada: 7	

Responsable: Sebastián Valencia
Descripción: El usuario administrador requiere gestionar información referente a secciones de la página, contenido y pagos en la página.
Observación: Las opciones a mostrar son la edición de horarios, ubicaciones, conferencistas y añadir precios entre otros.

TABLA XIV. Historia de usuario Nro.8

HISTORIA DE USUARIO	
Identificador (ID): HU008	Usuario: Administrador
Nombre Historia: Mostrar gestión de usuarios	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Iteración Asignada: 8	
Responsable: Sebastián Valencia	
Descripción: El usuario administrador requiere gestionar a los usuarios clientes que estén registrados.	
Observación: Las opciones a mostrar son la ver información de usuario cliente, crear usuarios, asignar roles y permisos, eliminar usuario cliente entre otras.	

TABLA XV. Historia de usuario Nro.9

HISTORIA DE USUARIO	
Identificador (ID): HU009	Usuario: Administrador
Nombre Historia: Realizar pruebas y despliegue	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Iteración Asignada: 9	
Responsable: Sebastián Valencia	
Descripción: Realizar las pruebas unitarias, usabilidad y compatibilidad.	
Observación: Determinar el correcto funcionamiento del sistema web.	

Product Backlog

En esta sección se encuentra un listado a largo plazo del proyecto, los requisitos que se deben realizar junto con los grados prioridades según su función en el sistema, el objetivo del *Product Backlog* es mantener el enfoque del desarrollo y que el sistema tenga la calidad suficiente para satisfacer las necesidades del usuario.

La TABLA XVI, TABLA XVI enumera la prioridad de cada requisito que se ha implementado en el sistema *web*.

TABLA XVI. Product Backlog

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>					
ID – PB	ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU002	HU002	Iniciar sesión y registrar usuarios	2	Finalizado	Alta
HU003	HU003	Mostrar página de ajustes del usuario	3	Finalizado	Media
HU004	HU004	Visualizar secciones de la página	4	Finalizado	Media
HU005	HU005	Mostrar sección de pagos	5	Finalizado	Media
HU006	HU006	Mostrar interfaz de administrador	6	Finalizado	Alta
HU007	HU007	Mostrar gestión de información	7	Finalizado	Alta
HU008	HU008	Mostrar gestión de usuarios clientes	8	Finalizado	Media
HU009	HU009	Realizar pruebas y despliegue	9	Finalizado	Media

SPRINT BACKLOG

El *Sprint Backlog* es una lista de elementos de trabajo específicos que se ha identificado como las tareas más importantes que deben completarse durante el sprint. Estos elementos de trabajo se derivan de los elementos del Product Backlog.

La **TABLA XVII**, muestra las tareas específicas a realizar durante el desarrollo del proyecto.

TABLA XVII. SPRINT BACKLOG

ELABORACIÓN DEL SPRINT BACKLOG					
ID-SB	NOMBRE	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB00	Configuración del ambiente de desarrollo	N/A	N/A	<ul style="list-style-type: none"> • Instalación de herramientas. • Creación del Proyecto. • Diseño del prototipo. 	40 H
SB01	Página inicio de sesión/registro	HU001	Mostrar Página inicial	<ul style="list-style-type: none"> • Implementar página inicial informativa al ingresar al sistema web. • Implementación de opciones para registro y/o inicio de sesión. 	30 H
		HU002	Mostrar Inicio Sesión Registrar usuario	<ul style="list-style-type: none"> • Implementación de interfaz de registro de usuarios. • Implementación de interfaz de inicio de sesión. • Validación que todos los campos sean requeridos. • Validación del tipo de información ingresada en los campos del formulario. • Guardar el registro. 	

SB02	Sección de ajustes y secciones.	HU003	Mostrar Página de ajustes del usuario	<ul style="list-style-type: none"> Implementación de interfaz para mostrar información sobre las secciones que ofrece la página al usuario con el rol registrado. 	40 H
		HU004	Visualizar secciones de la página	<ul style="list-style-type: none"> Implementación de interfaz de las secciones de conferencistas, información de contacto, horarios, ubicaciones y preguntas frecuentes. 	
SB03	Módulo de sección de pagos.	HU005	Mostrar Sección de pagos	<ul style="list-style-type: none"> Implementar las interfaces para las transiciones entre páginas. Redirección a la página de PayPal. 	40 H
SB04	Módulo de gestión	HU006	Mostrar Interfaz administrador	<ul style="list-style-type: none"> Mostrar la información del pago que el usuario puede realizar. Implementar una interfaz para la sección de gestión de pagos. 	40 H
		HU007	Mostrar Gestión de información	<ul style="list-style-type: none"> Mostrar información del usuario registrado. Implementación de interfaz con las respectivas funciones de edición de configuraciones de la página, conferencistas, horarios, ubicaciones, preguntas frecuentes, beneficios y precios. Implementación de interfaz con los precios correspondientes que se pueden editar y/o agregar. 	

SB05	Módulo de gestión usuarios	HU008	Mostrar Gestión de usuarios clientes	<ul style="list-style-type: none"> • Mostrar información de usuarios clientes registrados • Implementar la interfaz para realizar la edición del apartado usuarios, roles y permisos. 	30H
SB06	Pruebas y despliegue	HU009	Realizar Pruebas y despliegue	<ul style="list-style-type: none"> • Pruebas de compatibilidad, usabilidad y unitarias • Despliegue en servidor privado 	20H
TOTAL					240 H

Diseño de interfaces

El diseño del sistema *web* se basó en una plantilla prediseñada que sirvió como punto de partida para el diseño del sistema, fue modificada para adaptarse a los requerimientos específicos del proyecto. La plantilla que se eligió tenía una estructura general de diseño, como los colores y las secciones, pero fue necesario personalizarla para agregar nuevos elementos de diseño y funcionalidades, cambiar la disposición de algunos elementos en la página y ajustar las dimensiones y tamaños de ciertos componentes en las diferentes secciones. En esta sección se encuentran todos los mockups en base a la plantilla a partir de la **Fig 68** y la **Fig 69**.

En lo que corresponde a la página principal está estructurada en secciones hacia abajo haciendo uso de scroll (desplazamiento con la barra lateral derecha).



Fig 68. Intro



Fig 69. Acerca de

La sección "Conferencistas" presenta a los oradores o conferencistas que participarán en las conferencias. Tal como se muestra en las Fig 70 y Fig 71.

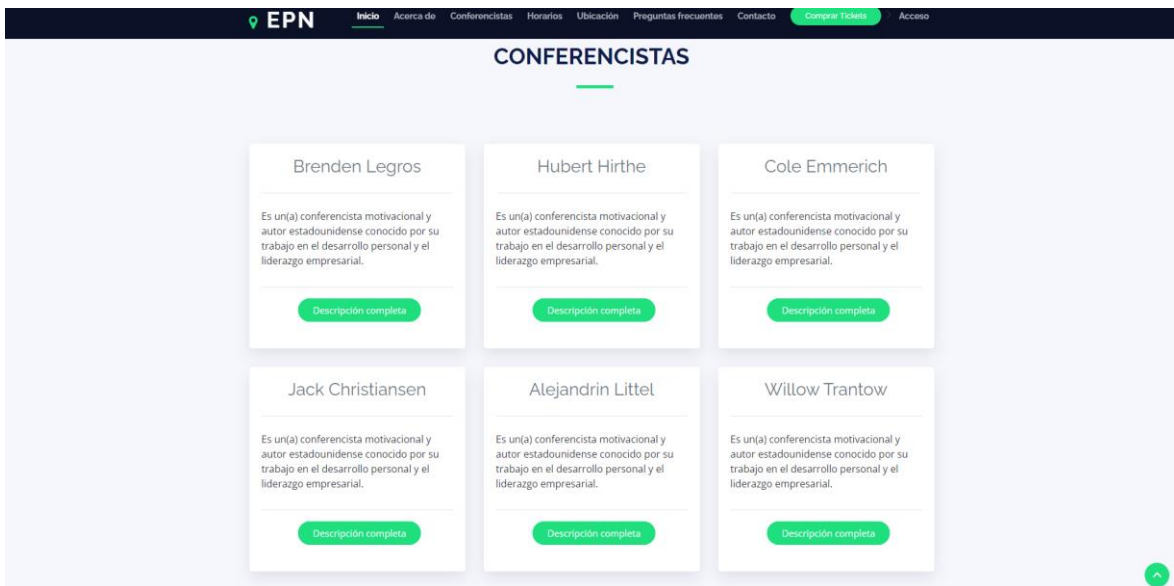


Fig 70. Conferencistas

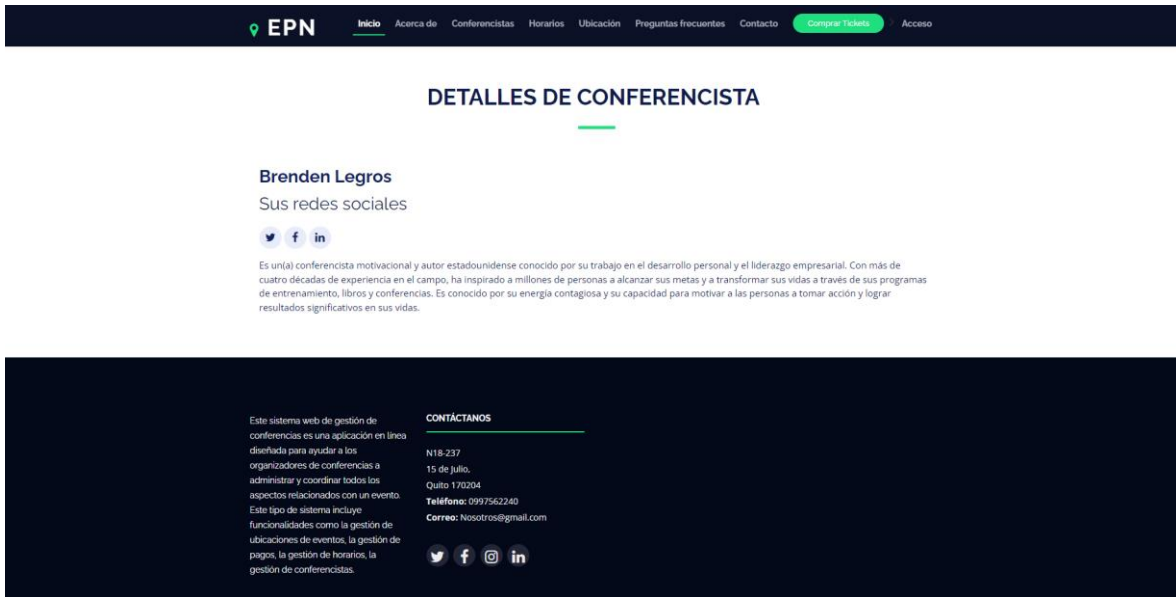


Fig 71. Detalles de conferencista

La sección "Horarios" detalla el programa de actividades para la conferencia. Tal como se muestra en la Fig 72.

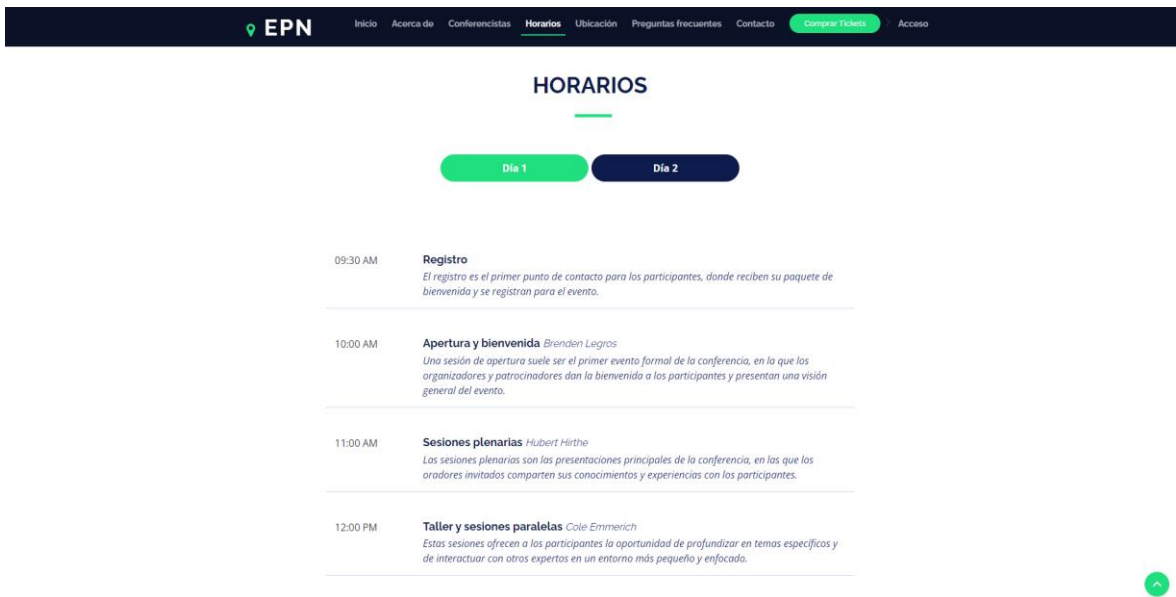


Fig 72. Horarios

La sección "Ubicaciones" proporciona información sobre los lugares relevantes para las conferencias. Esta sección puede incluir información sobre la ubicación geográfica, la dirección, el transporte, el alojamiento, la comida, etc. Tal como se muestra en las Fig 73 y Fig 74.



Fig 73. Ubicaciones

Drección: N18-237, 15 de Julio, Quito 170204

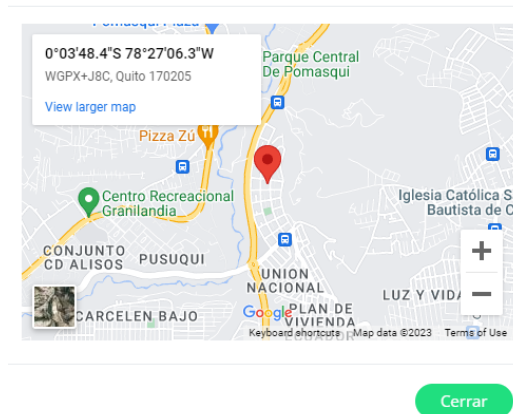


Fig 74. Mapa

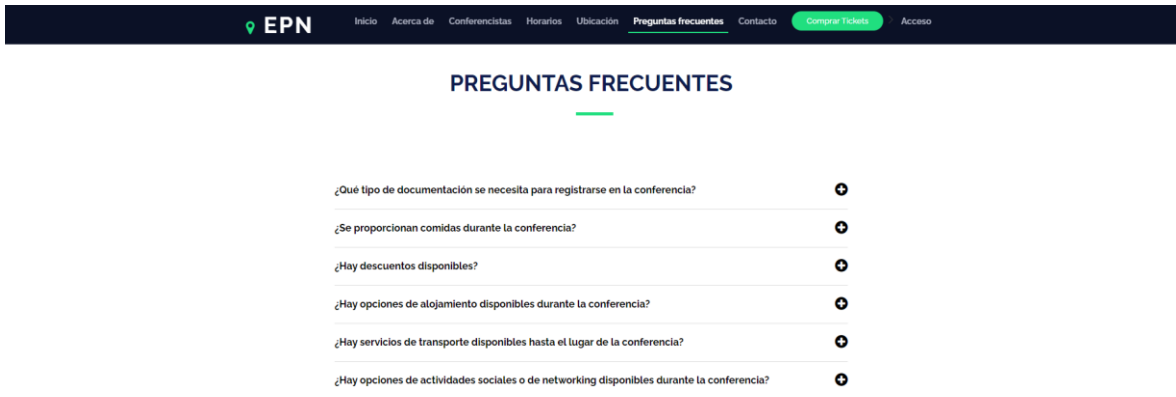


Fig 75. Preguntas frecuentes

La sección "Comprar *Tickets*" permite a los usuarios comprar entradas para las conferencias. Esta sección incluye información sobre el tipo de entradas disponibles y los precios. Tal como se muestra en la **Fig 76**.

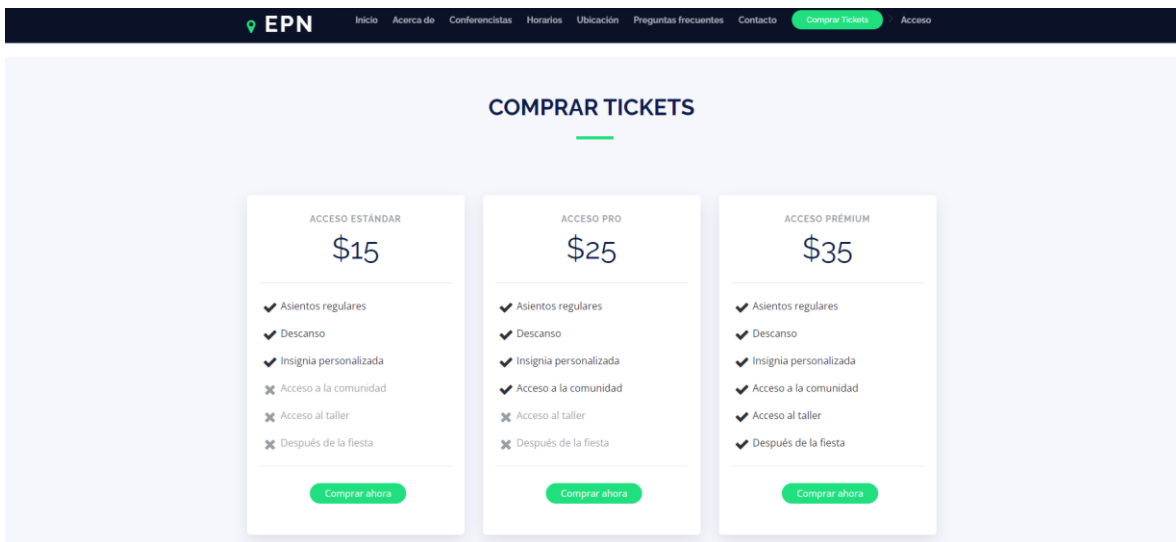


Fig 76. Comprar Tickets

EPN Inicio Acerca de Conferencistas Horarios Ubicación Preguntas frecuentes Contacto Comprar Tickets Acceso

CONTACTANOS

DIRECCIÓN
N18-237, 15 de Julio, Quito 170204

NÚMERO DE TELÉFONO
0997562240

CORREO
Nosotros@gmail.com

Tu nombre
Asunto
Tu correo electrónico
Mensaje

Enviar

Fig 77. Contacto

Pruebas unitarias

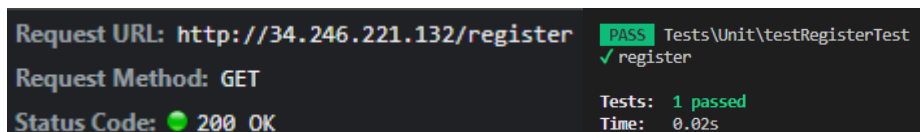
Las pruebas unitarias son un tipo de prueba de software que se enfoca en verificar que cada unidad o componente individual de un sistema o programa funcione correctamente de manera aislada. Se realizan escribiendo código específico para verificar el comportamiento esperado de cada unidad, y se ejecutan automáticamente durante el proceso de desarrollo para asegurarse de que no se introduzcan errores o problemas en el código. El objetivo principal de las pruebas unitarias es asegurarse de que cada unidad de código funciona correctamente antes de integrarla con otras partes del sistema, lo que puede ayudar a prevenir errores y mejorar la calidad general del software.

Este código define una función pública llamada `testRegisterTest()`, que se utiliza para realizar una prueba automatizada. La prueba verifica si la ruta `/register` devuelve una respuesta HTTP 200, que indica que la página se cargó correctamente. Para lograr esto, la función utiliza el método `get()` del objeto `$this` (que hace referencia a una instancia de la clase `TestCase`), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método `assertStatus()` para verificar si la respuesta recibida tiene un estado HTTP 200, que indica que la solicitud se realizó correctamente. Muestra en la **Fig 78**.

```
public function testRegisterTest()
{
    $response = $this->get('/register');
    $response->assertStatus(200);
}
```

Fig 78. Prueba unitaria #7. Acceso a registro de usuarios

La **Fig 79** muestra el resultado que indica que la prueba automatizada *testRegisterTest* del archivo *testRegisterTest.php* ubicado en la carpeta *Tests\Unit* ha pasado exitosamente. La prueba verificó si la página de */register* se carga correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a la URL <http://34.246.221.132/register>. El método utilizado para realizar la solicitud es GET. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 200 OK, lo que significa que la solicitud se realizó correctamente y la página correspondiente al */register* se cargó sin problemas.



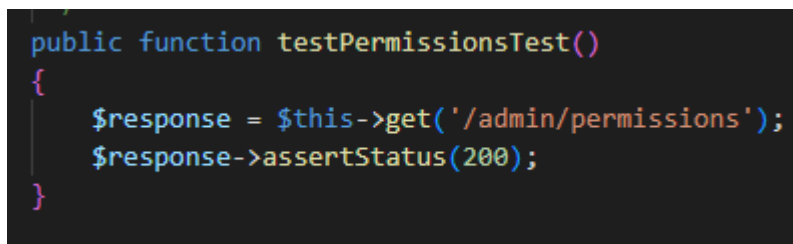
```
Request URL: http://34.246.221.132/register
Request Method: GET
Status Code: 200 OK

PASS Tests\Unit\testRegisterTest
✓ register

Tests: 1 passed
Time: 0.02s
```

Fig 79. Resultado prueba unitaria #7

Este código define una función pública llamada *testPermissionsTest()*, que se utiliza para realizar una prueba automatizada. La prueba verifica si la ruta */admin/permissions* devuelve una respuesta HTTP 200, que indica que la página se cargó correctamente. Para lograr esto, la función utiliza el método *get()* del objeto *\$this* (que hace referencia a una instancia de la clase *TestCase*), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método *assertStatus()* para verificar si la respuesta recibida tiene un estado HTTP 200, que indica que la solicitud se realizó correctamente. Muestra en la **Fig 80**.



```
public function testPermissionsTest()
{
    $response = $this->get('/admin/permissions');
    $response->assertStatus(200);
}
```

Fig 80. Prueba unitaria #8. Acceso a permisos de usuarios

La **Fig 81** muestra el resultado que indica que la prueba automatizada *testPermissionsTest* del archivo *testPermissionsTest.php* ubicado en la carpeta *Tests\Unit* ha pasado exitosamente. La prueba verificó si la página de */admin/permissions* se carga correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a la URL <http://34.246.221.132/admin/permissions>. El método utilizado para realizar la solicitud es GET. La respuesta (*Response*) recibida del servidor fue un código de estado HTTP 200 OK, lo que significa que la solicitud se realizó correctamente y la página correspondiente a */admin/permissions* se cargó sin problemas.

```
Request URL: http://34.246.221.132/admin/permissions
Request Method: GET
Status Code: 200 OK

PASS Tests\Unit\testPermissionsTest
permissions
Tests: 1 passed
Time: 0.01s
```

Fig 81. Resultado prueba unitaria #8

Estos códigos definen unas funciones públicas llamadas *testRolesTest()*, *testRoles_createTest()* y *testRoles_editTest()*, que se utilizan para realizar unan pruebas automatizadas. Las pruebas verifican si las rutas */admin/roles*, */admin/roles/create*, */admin/roles/1/edit* devuelven una respuesta HTTP 200, que indican que las páginas se cargaron correctamente. Para lograr esto, la función utiliza el método *get()* del objeto *\$this* (que hace referencia a una instancia de la clase *TestCase*), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método *assertStatus()* para verificar si las respuestas recibidas tienen un estado HTTP 200, que indica que las solicitudes se realizaron correctamente. Muestra en la **Fig 82**.

```
public function testRolesTest()
{
    $response = $this->get('/admin/roles');
    $response->assertStatus(200);
}

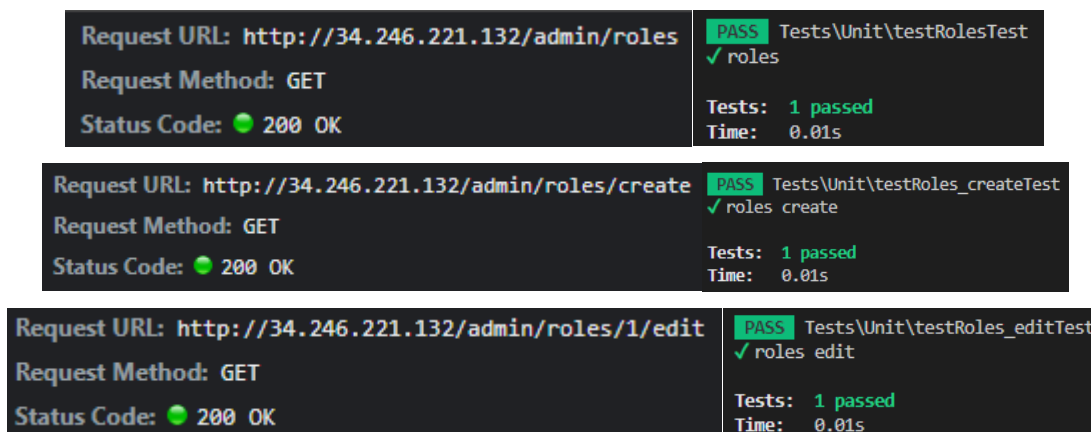
public function testRoles_createTest()
{
    $response = $this->get('/admin/roles/create');
    $response->assertStatus(200);
}

public function testRoles_editTest()
{
    $response = $this->get('/admin/roles/1/edit');
    $response->assertStatus(200);
}
```

Fig 82. Pruebas unitarias #9. Acceso a roles de usuarios(edición y creación)

La **Fig 83** muestra los resultados que indican que las pruebas automatizadas *testRolesTest()*, *testRoles_createTest()* y *testRoles_editTest()*, de los archivos *testRolesTest.php*, *testRoles_createTest.php* y *testRoles_editTest.php* ubicados en la carpeta *Tests\Unit* han pasado exitosamente. La prueba verificó si las páginas de */admin/roles*, */admin/roles/creáte* y */admin/roles/1/edit* se cargaron correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a las URL <http://34.246.221.132/admin/roles>, <http://34.246.221.132/admin/roles/create> y <http://34.246.221.132/admin/roles/1/edit>. El método utilizado para realizar las solicitudes es GET. Las respuestas (*Response*) recibidas del servidor fueron un código de estado HTTP

200 OK, lo que significa que las solicitudes se realizaron correctamente y las páginas correspondientes a `/admin/permissions`, `/admin/roles/create` y `/admin/roles/1/edit` se cargaron sin problemas.



```
Request URL: http://34.246.221.132/admin/roles
Request Method: GET
Status Code: 200 OK
PASS Tests\Unit\testRolesTest
✓ roles
Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/roles/create
Request Method: GET
Status Code: 200 OK
PASS Tests\Unit\testRoles_createTest
✓ roles create
Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/roles/1/edit
Request Method: GET
Status Code: 200 OK
PASS Tests\Unit\testRoles_editTest
✓ roles edit
Tests: 1 passed
Time: 0.01s
```

Fig 83. Resultado pruebas unitarias #9

Estos códigos definen unas funciones públicas llamadas `testSpeakersTest()`, `testSpeakers_createTest()` y `testSpeakers_editTest()`, que se utilizan para realizar unan pruebas automatizadas. Las pruebas verifican si las rutas `/admin/speakers`, `/admin/speakers/create` y `/admin/speakers/1/edit` devuelven una respuesta HTTP 200, que indican que las páginas se cargaron correctamente. Para lograr esto, la función utiliza el método `get()` del objeto `$this` (que hace referencia a una instancia de la clase `TestCase`), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método `assertStatus()` para verificar si las respuestas recibidas tienen un estado HTTP 200, que indica que las solicitudes se realizaron correctamente. Muestra en la **Fig 84**.


```

public function testSpeakersTest()
{
    $response = $this->get('/admin/speakers');
    $response->assertStatus(200);
}

public function testSpeakers_createTest()
{
    $response = $this->get('/admin/speakers/create');
    $response->assertStatus(200);
}

public function testSpeakers_editTest()
{
    $response = $this->get('/admin/speakers/1/edit');
    $response->assertStatus(200);
}

```

Fig 84. Pruebas unitarias #10. Acceso a conferencistas (edición y creación)

la **Fig 85** muestra los resultados que indican que las pruebas automatizadas *testSpeakersTest()*, *testSpeakers_createTest()* y *testSpeakers_editTest()* de los archivos *testSpeakersTest.php*, *testSpeakers_createTest.php* y *testSpeakers_editTest.php* ubicados en la carpeta *Tests\Unit* han pasado exitosamente. La prueba verificó si las páginas de */admin/speakers*, */admin/speakers/creáe* y */admin/speakers/1/edit* se cargaron correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a las URL <http://34.246.221.132/admin/speakers>, <http://34.246.221.132/admin/speakers/create> y <http://34.246.221.132/admin/speakers/1/edit>. El método utilizado para realizar las solicitudes es GET. Las respuestas (*Response*) recibidas del servidor fueron un código de estado HTTP 200 OK, lo que significa que las solicitudes se realizaron correctamente y las páginas correspondientes a */admin/speakers*, */admin/speakers/creáe* y */admin/speakers/1/edit* se cargaron sin problemas.

```

Request URL: http://34.246.221.132/admin/speakers PASS Tests\Unit\testSpeakersTest
Request Method: GET ✓ speakers
Status Code: 200 OK Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/speakers/create PASS Tests\Unit\testSpeakers_createTest
Request Method: GET ✓ speakers create
Status Code: 200 OK Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/speakers/1/edit PASS Tests\Unit\testSpeakers_editTest
Request Method: GET ✓ speakers edit
Status Code: 200 OK Tests: 1 passed
Time: 0.01s

```

Fig 85. Resultado pruebas unitarias #10

Estos códigos definen unas funciones públicas llamadas `testSchedulesTest()`, `testSchedules_createTest()` y `testSchedules_editTest()`, que se utilizan para realizar unan pruebas automatizadas. Las pruebas verifican si las rutas `/admin/schedules`, `/admin/schedules/create` y `/admin/schedules/1/edit` devuelven una respuesta HTTP 200, que indican que las páginas se cargaron correctamente. Para lograr esto, la función utiliza el método `get()` del objeto `$this` (que hace referencia a una instancia de la clase `TestCase`), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método `assertStatus()` para verificar si las respuestas recibidas tienen un estado HTTP 200, que indica que las solicitudes se realizaron correctamente. Muestra en la **Fig 86**.

```
public function testSchedulesTest()
{
    $response = $this->get('/admin/schedules');
    $response->assertStatus(200);
}

public function testSchedules_createTest()
{
    $response = $this->get('/admin/schedules/create');
    $response->assertStatus(200);
}

public function testSchedules_editTest()
{
    $response = $this->get('/admin/schedules/1/edit');
    $response->assertStatus(200);
}
```

Fig 86. Pruebas unitarias #11. Acceso a horarios (edición y creación)

la **Fig 87** muestra los resultados que indican que las pruebas automatizadas `testSchedulesTest()`, `testSchedules_createTest()` y `testSchedules_editTest()` de los archivos `testSchedulesTest.php`, `testSchedules_createTest.php` y `testSchedules_editTest.php` ubicados en la carpeta `Tests\Unit` han pasado exitosamente. La prueba verificó si las páginas de `/admin/schedules`, `/admin/schedules/create`, `/admin/schedules/1/edit` se cargaron correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a las URL `http://34.246.221.132/admin/schedules`, `http://34.246.221.132/admin/schedules/create` y `http://34.246.221.132/admin/schedules/1/edit`. El método utilizado para realizar las solicitudes es GET. Las respuestas (*Response*) recibidas del servidor fueron un código de

estado HTTP 200 OK, lo que significa que las solicitudes se realizaron correctamente y las páginas correspondientes a `/admin/schedules`, `/admin/schedules/create`, `/admin/schedules/1/edit` se cargaron sin problemas

```
Request URL: http://34.246.221.132/admin/schedules PASS Tests\Unit\testSchedulesTest
Request Method: GET ✓ schedules
Status Code: 200 OK Tests: 1 passed
Time: 0.22s

Request URL: http://34.246.221.132/admin/schedules/create PASS Tests\Unit\testSchedules_createTest
Request Method: GET ✓ schedules create
Status Code: 200 OK Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/schedules/1/edit PASS Tests\Unit\testSchedules_editTest
Request Method: GET ✓ schedules edit
Status Code: 200 OK Tests: 1 passed
Time: 0.01s
```

Fig 87. Resultado pruebas unitarias #11

Estos códigos definen unas funciones públicas llamadas `testVenuesTest()`, `testVenues_createTest()` y `testVenues_editTest()`, que se utilizan para realizar unan pruebas automatizadas. Las pruebas verifican si las rutas `/admin/venues`, `/admin/venues/create` y `/admin/venues/1/edit` devuelven una respuesta HTTP 200, que indican que las páginas se cargaron correctamente. Para lograr esto, la función utiliza el método `get()` del objeto `$this` (que hace referencia a una instancia de la clase `TestCase`), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método `assertStatus()` para verificar si las respuestas recibidas tienen un estado HTTP 200, que indica que las solicitudes se realizaron correctamente. Muestra en la Fig 88.

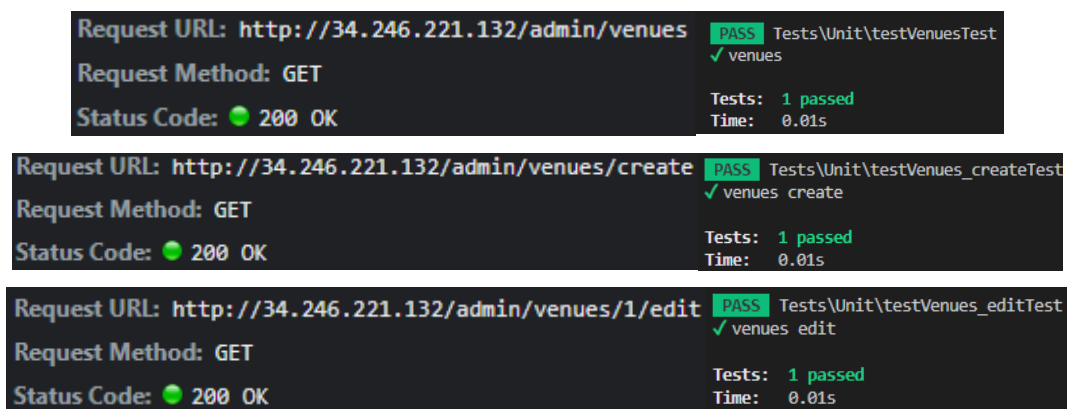
```
public function testVenuesTest()
{
    $response = $this->get('/admin/venues');
    $response->assertStatus(200);
}

public function testVenues_createTest()
{
    $response = $this->get('/admin/venues/create');
    $response->assertStatus(200);
}

public function testVenues_editTest()
{
    $response = $this->get('/admin/venues/1/edit');
    $response->assertStatus(200);
}
```

Fig 88. Pruebas unitarias #12. Acceso a ubicaciones (edición y creación)

La **Fig 89** muestra los resultados que indican que las pruebas automatizadas *testVenuesTest()*, *testVenues_createTest()* y *testVenues_editTest()* de los archivos *testVenuesTest.php*, *testVenues_createTest.php* y *testVenues_editTest.php* ubicados en la carpeta *Tests\Unit* han pasado exitosamente. La prueba verificó si las páginas de */admin/venues*, */admin/venues/create* y */admin/venues/1/edit* se cargaron correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a las URL <http://34.246.221.132/admin/venues>, <http://34.246.221.132/admin/venues/create> y <http://34.246.221.132/admin/venues/1/edit>. El método utilizado para realizar las solicitudes es GET. Las respuestas (*Response*) recibidas del servidor fueron un código de estado HTTP 200 OK, lo que significa que las solicitudes se realizaron correctamente y las páginas correspondientes a */admin/venues*, */admin/venues/create* y */admin/venues/1/edit* se cargaron sin problemas.



```
Request URL: http://34.246.221.132/admin/venues PASS Tests\Unit\testVenuesTest
Request Method: GET ✓ venues
Status Code: 200 OK Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/venues/create PASS Tests\Unit\testVenues_createTest
Request Method: GET ✓ venues create
Status Code: 200 OK Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/venues/1/edit PASS Tests\Unit\testVenues_editTest
Request Method: GET ✓ venues edit
Status Code: 200 OK Tests: 1 passed
Time: 0.01s
```

Fig 89. Resultado pruebas unitarias #12

Estos códigos definen unas funciones públicas llamadas *testFaqsTest()*, *testFaqs_createTest()* y *testFaqs_editTest()*, que se utilizan para realizar unan pruebas automatizadas. Las pruebas verifican si las rutas */admin/faqs*, */admin/faqs/create* y */admin/faqs/1/edit* devuelven una respuesta HTTP 200, que indican que las páginas se cargaron correctamente. Para lograr esto, la función utiliza el método *get()* del objeto *\$this* (que hace referencia a una instancia de la clase *TestCase*), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método *assertStatus()* para verificar si las respuestas recibidas tienen un estado HTTP 200, que indica que las solicitudes se realizaron correctamente. Muestra en la **Fig 90**.

```

public function testFaqsTest()
{
    $response = $this->get('/admin/faqs');
    $response->assertStatus(200);
}

public function testFaqs_createTest()
{
    $response = $this->get('/admin/faqs/create');
    $response->assertStatus(200);
}

public function testFaqs_editTest()
{
    $response = $this->get('/admin/faqs/1/edit');
    $response->assertStatus(200);
}

```

Fig 90. Pruebas unitarias #13. Acceso a preguntas frecuentes (edición y creación)

La **Fig 91** muestra los resultados que indican que las pruebas automatizadas *testFaqsTest()*, *testFaqs_createTest()* y *testFaqs_editTest()* de los archivos *testFaqsTest.php*, *testFaqs_createTest.php* y *testFaqs_editTest.php* ubicados en la carpeta *Tests\Unit* han pasado exitosamente. La prueba verificó si las páginas de */admin/faqs*, */admin/faqs/create* y */admin/faqs/1/edit* se cargaron correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a las URL <http://34.246.221.132/admin/faqs>, <http://34.246.221.132/admin/faqs/create> y <http://34.246.221.132/admin/faqs/1/edit>. El método utilizado para realizar las solicitudes es GET. Las respuestas (*Response*) recibidas del servidor fueron un código de estado HTTP 200 OK, lo que significa que las solicitudes se realizaron correctamente y las páginas correspondientes a */admin/faqs*, */admin/faqs/create* y */admin/faqs/1/edit* se cargaron sin problemas.

Request URL: http://34.246.221.132/admin/faqs	PASS Tests\Unit\testFaqsTest ✓ faqs
Request Method: GET	Tests: 1 passed
Status Code: 200 OK	Time: 0.01s

Request URL: http://34.246.221.132/admin/faqs/create	PASS Tests\Unit\testFaqs_createTest ✓ faqs create
Request Method: GET	Tests: 1 passed
Status Code: 200 OK	Time: 0.01s

Request URL: http://34.246.221.132/admin/faqs/1/edit	PASS Tests\Unit\testFaqs_editTest ✓ faqs edit
Request Method: GET	Tests: 1 passed
Status Code: 200 OK	Time: 0.01s

Fig 91. Resultado pruebas unitarias #13

Estos códigos definen unas funciones públicas llamadas *testAmenitiesTest()*, *testAmenities_createTest()* y *testAmenities_editTest()*, que se utilizan para realizar unan pruebas automatizadas. Las pruebas verifican si las rutas */admin/amenities*, */admin/amenities/create* y */admin/amenities/1/edit* devuelven una respuesta HTTP 200, que indican que las páginas se cargaron correctamente. Para lograr esto, la función utiliza el método *get()* del objeto *\$this* (que hace referencia a una instancia de la clase *TestCase*), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método *assertStatus()* para verificar si las respuestas recibidas tienen un estado HTTP 200, que indica que las solicitudes se realizaron correctamente. Muestra en la Fig 92.

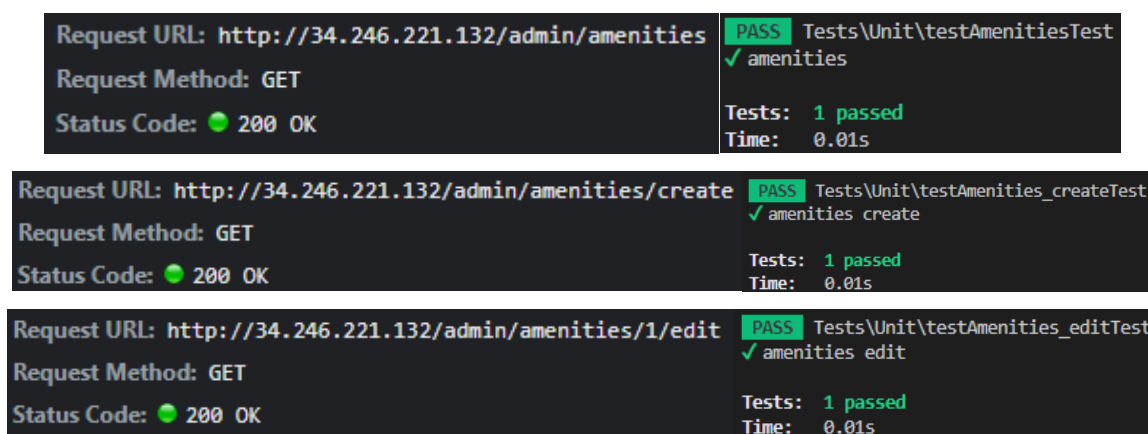
```
public function testAmenitiesTest()
{
    $response = $this->get('/admin/amenities');
    $response->assertStatus(200);
}

public function testAmenities_createTest()
{
    $response = $this->get('/admin/amenities/create');
    $response->assertStatus(200);
}

public function testAmenities_editTest()
{
    $response = $this->get('/admin/amenities/1/edit');
    $response->assertStatus(200);
}
```

Fig 92. Pruebas unitarias #14. Acceso a preguntas beneficios (edición y creación)

La **Fig 93** muestra los resultados que indican que las pruebas automatizadas *testAmenitiesTest()*, *testAmenities_createTest()* y *testAmenities_editTest()* de los archivos *testAmenitiesTest.php*, *testAmenities_createTest.php* y *testAmenities_editTest.php* ubicados en la carpeta *Tests\Unit* han pasado exitosamente. La prueba verificó si las páginas de */admin/amenities*, */admin/amenities/create* y */admin/amenities/1/edit* se cargaron correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a las URL <http://34.246.221.132/admin/amenities>, <http://34.246.221.132/admin/amenities/create> y <http://34.246.221.132/admin/amenities/1/edit>. El método utilizado para realizar las solicitudes es GET. Las respuestas (*Response*) recibidas del servidor fueron un código de estado HTTP 200 OK, lo que significa que las solicitudes se realizaron correctamente y las páginas correspondientes a */admin/amenities*, */admin/amenities/create* y */admin/amenities/1/edit* se cargaron sin problemas



```
Request URL: http://34.246.221.132/admin/amenities
Request Method: GET
Status Code: 200 OK
PASS Tests\Unit\testAmenitiesTest
✓ amenities
Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/amenities/create
Request Method: GET
Status Code: 200 OK
PASS Tests\Unit\testAmenities_createTest
✓ amenities create
Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/amenities/1/edit
Request Method: GET
Status Code: 200 OK
PASS Tests\Unit\testAmenities_editTest
✓ amenities edit
Tests: 1 passed
Time: 0.01s
```

Fig 93. Resultado pruebas unitarias #14

Estos códigos definen unas funciones públicas llamadas *testPricesTest()*, *testPrices_createTest()* y *testPrices_editTest()*, que se utilizan para realizar unan pruebas automatizadas. Las pruebas verifican si las rutas */admin/prices*, */admin/prices/create* y */admin/prices/1/edit* devuelven una respuesta HTTP 200, que indican que las páginas se cargaron correctamente. Para lograr esto, la función utiliza el método *get()* del objeto *\$this* (que hace referencia a una instancia de la clase *TestCase*), que realiza una solicitud GET a la ruta especificada. Luego, utiliza el método *assertStatus()* para verificar si las respuestas recibidas tienen un estado HTTP 200, que indica que las solicitudes se realizaron correctamente. Muestra en la **Fig 94**.

```

public function testPricesTest()
{
    $response = $this->get('/admin/prices');
    $response->assertStatus(200);
}

public function testPrices_createTest()
{
    $response = $this->get('/admin/prices/create');
    $response->assertStatus(200);
}

public function testPrices_editTest()
{
    $response = $this->get('/admin/prices/1/edit');
    $response->assertStatus(200);
}

```

Fig 94. Pruebas unitarias #15. Acceso a precios (edición y creación)

La **Fig 95** muestra los resultados que indican que las pruebas automatizadas *testPricesTest()*, *testPrices_createTest()* y *testPrices_editTest()* de los archivos *testPricesTest.php*, *testPrices_createTest.php* y *testPrices_editTest.php* ubicados en la carpeta *Tests\Unit* han pasado exitosamente. La prueba verificó si las páginas de */admin/prices*, */admin/prices/create* y */admin/prices/1/edit* se cargaron correctamente. La solicitud (*Request*) es una petición HTTP GET enviada a las URL <http://34.246.221.132/admin/prices>, <http://34.246.221.132/admin/prices/create> y <http://34.246.221.132/admin/prices/1/edit>. El método utilizado para realizar las solicitudes es GET. Las respuestas (*Response*) recibidas del servidor fueron un código de estado HTTP 200 OK, lo que significa que las solicitudes se realizaron correctamente y las páginas correspondientes a */admin/prices*, */admin/prices/create* y */admin/prices/1/edit* se cargaron sin problemas.


```
Request URL: http://34.246.221.132/admin/prices PASS Tests\Unit\testPricesTest
Request Method: GET ✓ prices
Status Code: 200 OK Tests: 1 passed
Time: 0.04s

Request URL: http://34.246.221.132/admin/prices/create PASS Tests\Unit\testPrices_createTest
Request Method: GET ✓ prices create
Status Code: 200 OK Tests: 1 passed
Time: 0.01s

Request URL: http://34.246.221.132/admin/prices/1/edit PASS Tests\Unit\testPrices_editTest
Request Method: GET ✓ prices edit
Status Code: 200 OK Tests: 1 passed
Time: 0.01s
```

Fig 95. Resultado pruebas unitarias #15

Pruebas de compatibilidad

Las pruebas de compatibilidad son una técnica de evaluación que se utiliza para verificar si el sistema es compatible con diferentes plataformas, navegadores, dispositivos y sistemas operativos. Para la siguiente sección se verifica como se adapta el sistema en diferentes navegadores con la finalidad de encontrar alguna inconsistencia como a partir de la Fig 96 hasta la Fig 98.



Fig 96. Página principal con información general vista desde el navegador Brave



Fig 97. Página principal con información general vista desde el navegador Chrome



Fig 98. Página principal con información general vista desde el navegador Edge

Prueba de compatibilidad en móviles, las pruebas de compatibilidad del sistema *web* en un dispositivo móvil son una parte importante del proceso de estas pruebas ya que ayudarán a garantizar que el sistema es accesible y fácil de usar en diferentes dispositivos y sistemas operativos móviles. El dispositivo de prueba en cuestión es un Redmi Note 8 Pro, se evaluó como se ve y se comporta en el móvil mencionado, se verificó que se adapta correctamente al tamaño de la pantalla y su resolución. Adicional se hizo la prueba del funcionamiento del sistema desde un dispositivo móvil a partir de la **Fig 99** hasta la **Fig 101**.



Fig 99. Intro (móvil)

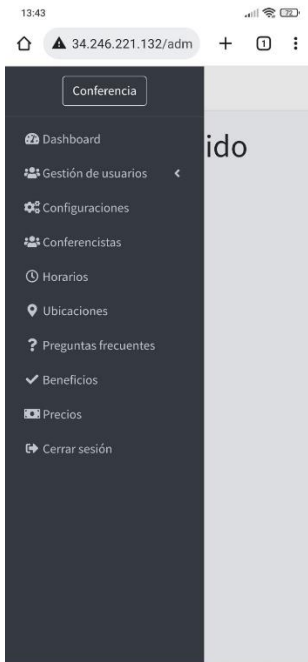


Fig 100. Panel administrador (móvil)



Fig 101. Comprar Tickets (móvil)

Pruebas de usabilidad

Las pruebas de usabilidad son un conjunto de técnicas que se utilizan para evaluar la facilidad de uso y la experiencia de usuario de un sistema o aplicación. Entre estas técnicas, se encuentra la encuesta, que es una herramienta útil para recopilar información de los usuarios y obtener una retroalimentación sobre el sistema *web*. A continuación, se presentarán las preguntas y los resultados obtenidos junto con las respectivas conclusiones. A continuación, el enlace a la encuesta:

https://docs.google.com/forms/d/e/1FAIpQLSe2AQE97VgwCNbJkXO_7yXM6KCwjWAzmFcX6Mg4ZOfRJYtE-Q/viewform

Pregunta #1

Esta pregunta busca recopilar información sobre la facilidad o dificultad que tuvo el usuario al registrarse en un sistema *web*. Tal como se muestra en la **Fig 102**.

¿Te resultó fácil registrarte en el sistema web?

20 respuestas

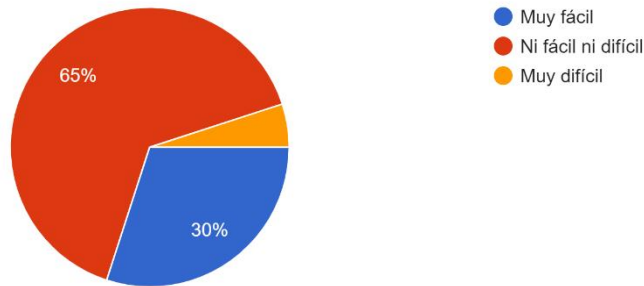


Fig 102. Pregunta de usabilidad #1

De acuerdo con las respuestas dadas, podemos concluir que el 40% de los usuarios encontró la gestión de los conferencistas en el panel de administración del sistema *web* ni fácil ni difícil, lo que sugiere que el proceso de gestión es relativamente sencillo para ellos. Además, el 35% de los usuarios encontró muy fácil gestionar los conferencistas, lo que indica que para algunos usuarios el proceso fue bastante sencillo. Sin embargo, el 25% de los usuarios encontró muy difícil gestionar los conferencistas, lo que sugiere que algunos usuarios tuvieron dificultades significativas en este aspecto.

Pregunta #2

Esta pregunta tiene como objetivo obtener información sobre la facilidad o dificultad que tuvo el usuario al navegar por el sistema *web* en cuestión. Tal como se muestra en la **Fig 103**.

¿Te resultó fácil navegar por el sistema web?

19 respuestas

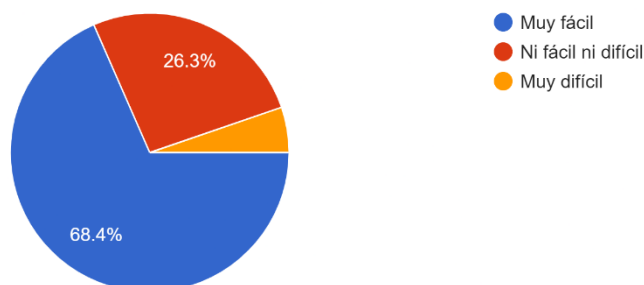


Fig 103. Pregunta de usabilidad #2

De acuerdo con las respuestas dadas, podemos concluir que la mayoría de los usuarios (68.4%) encontró fácil navegar por el sistema *web*. Además, el 26.3% de los usuarios no encontró el proceso de navegación ni fácil ni difícil, lo que sugiere que, en general, el sistema *web* es bastante accesible para ellos. Solo el 5.3% de los usuarios encontró muy difícil la navegación en el sistema *web*, lo que indica que solo una minoría tuvo dificultades significativas.

Pregunta #3

Esta pregunta tiene como objetivo evaluar si el usuario pudo encontrar fácilmente la información que estaba buscando en el sistema *web* en cuestión. Tal como se muestra en la **Fig 104**.

¿Te resultó fácil encontrar la información que necesitabas en el sistema web?
20 respuestas

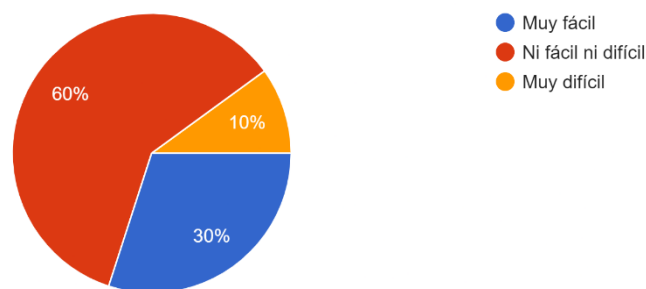


Fig 104. Pregunta de usabilidad #3

De acuerdo con las respuestas dadas, podemos concluir que el 60% de los usuarios no encontró ni fácil ni difícil encontrar la información que necesitaban en el sistema *web*, lo que sugiere que el proceso de búsqueda de información fue relativamente sencillo para ellos. Además, el 30% de los usuarios encontró muy fácil encontrar la información que necesitaban, lo que indica que para algunos usuarios el proceso de búsqueda fue bastante sencillo. Solo el 10% de los usuarios encontró muy difícil encontrar la información que necesitaba, lo que sugiere que solo una minoría tuvo dificultades significativas para encontrar lo que estaban buscando.

Pregunta #4

Esta pregunta tiene como objetivo evaluar la opinión del usuario sobre el diseño del sistema *web* en términos de su atractivo visual y facilidad de comprensión. Tal como se muestra en la **Fig 105**.

¿Crees que el diseño del sistema web es atractivo y fácil de entender?
20 respuestas

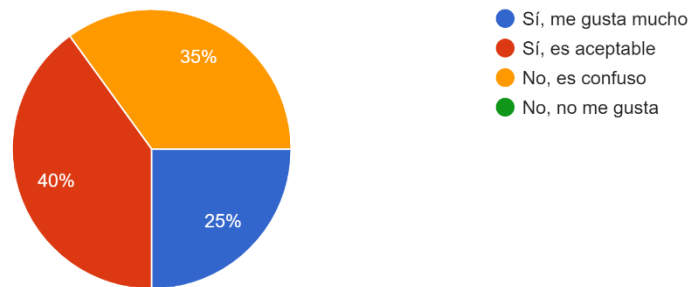


Fig 105. Pregunta de usabilidad #4

De acuerdo con las respuestas dadas, podemos concluir que una mayoría significativa de usuarios (65%) encuentra el diseño del sistema *web* atractivo y fácil de entender, ya sea que lo consideren "aceptable" (40%) o "me gusta mucho" (25%). Sin embargo, un tercio de los usuarios (35%) no encuentra el diseño del sistema *web* fácil de entender, lo que sugiere que hay áreas de mejora para hacer que el diseño del sistema *web* sea más claro y accesible para los usuarios. Afortunadamente, ningún usuario indicó que no le gusta el diseño del sistema *web*, lo que indica que, en general, el diseño parece ser aceptable para los usuarios.

Pregunta #5

esta pregunta tiene como objetivo evaluar la facilidad o dificultad que tuvo el usuario al acceder al panel de administración del sistema *web* en cuestión. Tal como se muestra en la **Fig 106**.

¿Te resultó fácil acceder al panel de administración del sistema web?

20 respuestas

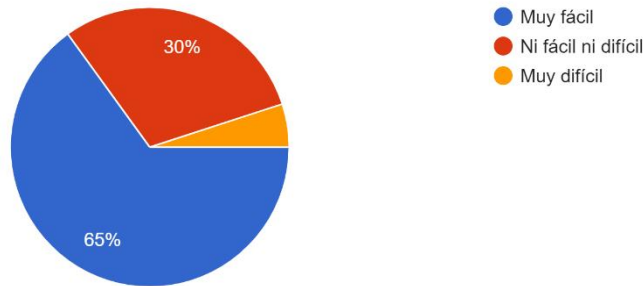


Fig 106. Pregunta de usabilidad #5

De acuerdo con las respuestas dadas, podemos concluir que la mayoría de los usuarios (65%) encontró fácil acceder al panel de administración del sistema *web*. Además, el 30% de los usuarios no encontró el proceso ni fácil ni difícil, lo que sugiere que el acceso al panel de administración es relativamente sencillo para ellos. Solo el 5% de los usuarios encontró muy difícil acceder al panel de administración, lo que indica que solo una minoría tuvo dificultades significativas.

Pregunta #6

Esta pregunta tiene como objetivo evaluar la facilidad o dificultad que tuvo el usuario al gestionar los roles y permisos de los usuarios en el panel de administración del sistema *web* en cuestión. Tal como se muestra en la **Fig 107**.

¿Te resultó fácil gestionar los roles y permisos de los usuarios en el panel de administración del sistema web?

18 respuestas

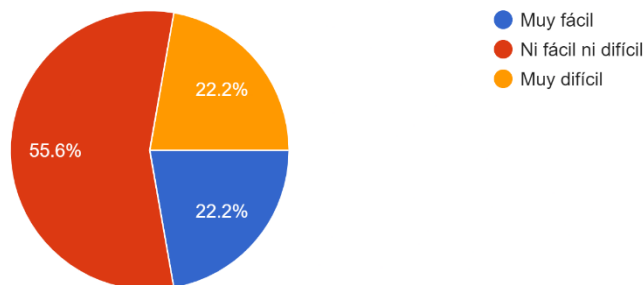


Fig 107. Pregunta de usabilidad #6

De acuerdo con las respuestas dadas, podemos concluir que la mitad de los usuarios (55.6%) encontró la gestión de roles y permisos de los usuarios en el panel de

administración del sistema *web* ni fácil ni difícil, lo que sugiere que el proceso de gestión es relativamente sencillo para ellos. Sin embargo, el 22.2% de los usuarios encontró muy difícil gestionar los roles y permisos de los usuarios, lo que indica que algunos usuarios tuvieron dificultades significativas. Por otro lado, solo el 22.2% de los usuarios encontró muy fácil gestionar los roles y permisos de los usuarios, lo que sugiere que solo una minoría tuvo una experiencia muy positiva en este aspecto.

Pregunta #7

Esta pregunta tiene como objetivo evaluar la facilidad o dificultad que tuvo el usuario al gestionar los conferencistas en el panel de administración del sistema *web* en cuestión. Tal como se muestra en la **Fig 108**.

¿Te resultó fácil gestionar los conferencistas en el panel de administración del sistema web?
20 respuestas

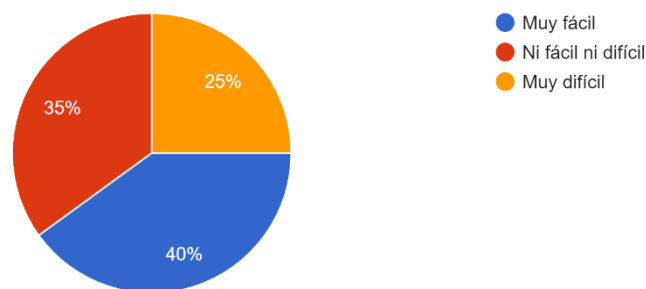


Fig 108. Pregunta de usabilidad #7

De acuerdo con las respuestas dadas, podemos concluir que el 40% de los usuarios encontró la gestión de los conferencistas en el panel de administración del sistema *web* ni fácil ni difícil, lo que sugiere que el proceso de gestión es relativamente sencillo para ellos. Además, el 35% de los usuarios encontró muy fácil gestionar los conferencistas, lo que indica que para algunos usuarios el proceso fue bastante sencillo. Sin embargo, el 25% de los usuarios encontró muy difícil gestionar los conferencistas, lo que sugiere que algunos usuarios tuvieron dificultades significativas en este aspecto.

Pregunta #8

Esta pregunta tiene como objetivo evaluar la facilidad o dificultad que tuvo el usuario al gestionar los horarios en el panel de administración del sistema *web* en cuestión. Tal como se muestra en la **Fig 109**.

¿Te resultó fácil gestionar los horarios en el panel de administración del sistema web?

20 respuestas

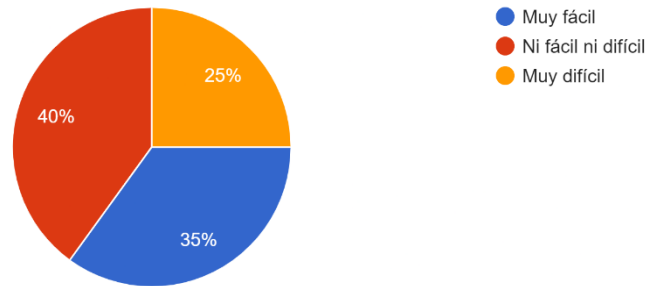


Fig 109. Pregunta de usabilidad #8

De acuerdo con las respuestas dadas, podemos concluir que el 40% de los usuarios encontró la gestión de los horarios en el panel de administración del sistema *web* muy fácil, lo que sugiere que el proceso de gestión es relativamente sencillo para una parte importante de los usuarios. El 35% de los usuarios consideró que la gestión de los horarios ni fue fácil ni difícil, lo que indica que para algunos usuarios el proceso fue algo complicado, pero no imposible. El 25% de los usuarios encontró muy difícil gestionar los horarios en el panel de administración del sistema *web*, lo que sugiere que para una minoría significativa de usuarios el proceso resultó difícil de comprender y ejecutar.

Pregunta #9

Esta pregunta tiene como objetivo evaluar la facilidad o dificultad que tuvo el usuario al gestionar las ubicaciones en el panel de administración del sistema *web* en cuestión. Tal como se muestra en la **Fig 110**.

¿Te resultó fácil gestionar las ubicaciones en el panel de administración del sistema web?

20 respuestas

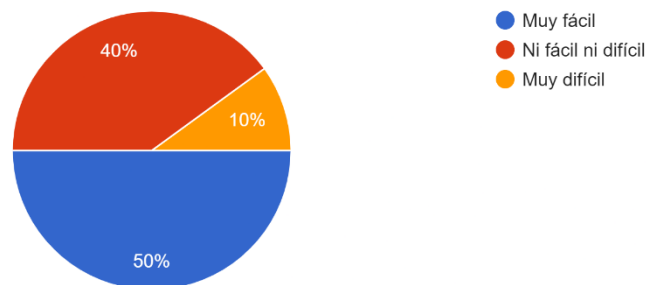


Fig 110. Pregunta de usabilidad #9

De acuerdo con las respuestas dadas, podemos concluir que la mitad de los usuarios (50%) encontró la gestión de las ubicaciones en el panel de administración del sistema *web* muy fácil, lo que sugiere que el proceso de gestión es bastante sencillo para la mayoría de los usuarios. Además, el 40% de los usuarios encontró la gestión de las ubicaciones ni fácil ni difícil, lo que indica que para algunos usuarios el proceso no fue demasiado complicado, pero tampoco resultó completamente sencillo. Solo el 10% de los usuarios encontró muy difícil gestionar las ubicaciones, lo que sugiere que solo una minoría tuvo dificultades significativas en este aspecto.

Pregunta #10

Esta pregunta tiene como objetivo evaluar la facilidad o dificultad que tuvo el usuario al gestionar las preguntas frecuentes en el panel de administración del sistema *web* en cuestión. Tal como se muestra en la **Fig 111**.

¿Te resultó fácil gestionar las preguntas frecuentes en el panel de administración del sistema *web*?
20 respuestas

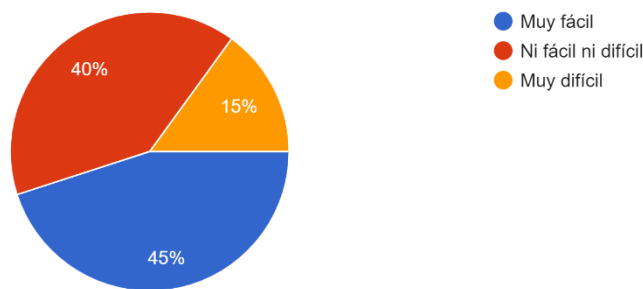


Fig 111. Pregunta de usabilidad #10

De acuerdo con las respuestas dadas, podemos concluir que el 45% de los usuarios encontró la gestión de las preguntas frecuentes en el panel de administración del sistema *web* muy fácil, lo que sugiere que el proceso de gestión es bastante sencillo para una gran parte de los usuarios. Además, el 40% de los usuarios encontró la gestión de las preguntas frecuentes ni fácil ni difícil, lo que indica que para algunos usuarios el proceso no fue demasiado complicado, pero tampoco resultó completamente sencillo. Solo el 15% de los usuarios encontró muy difícil gestionar las preguntas frecuentes, lo que sugiere que solo una minoría tuvo dificultades significativas en este aspecto.

Pregunta #11

Esta pregunta tiene como objetivo evaluar la facilidad o dificultad que tuvo el usuario al gestionar los precios y beneficios en el panel de administración del sistema *web* en cuestión. Tal como se muestra en la **Fig 112**.

¿Te resultó fácil gestionar los precios y beneficios en el panel de administración del sistema web?
19 respuestas

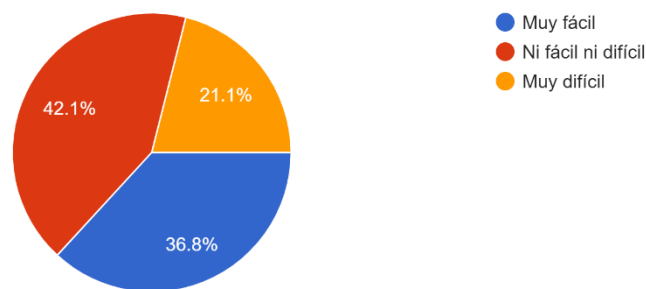


Fig 112. Pregunta de usabilidad #11

De acuerdo con las respuestas dadas, podemos concluir que el 36.8% de los usuarios encontró la gestión de los precios y beneficios en el panel de administración del sistema *web* muy fácil, lo que sugiere que el proceso de gestión es sencillo para una parte de los usuarios. Además, el 42.1% de los usuarios encontró la gestión de los precios y beneficios ni fácil ni difícil, lo que indica que para algunos usuarios el proceso no fue demasiado complicado, pero tampoco resultó completamente sencillo. Por otro lado, el 21.1% de los usuarios encontró muy difícil gestionar los precios y beneficios, lo que sugiere que una minoría tuvo dificultades significativas en este aspecto.

Pregunta #12

esta pregunta tiene como objetivo evaluar la satisfacción general del usuario con el sistema *web* de gestión de conferencias con pagos en línea y su disposición a recomendarlo a otras personas. Tal como se muestra en la **Fig 113**.

¿Recomendarías el sistema web de gestión de conferencias con pagos en línea a tus amigos?

20 respuestas

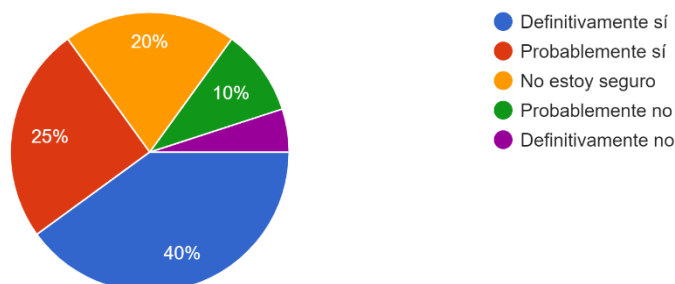


Fig 113. Pregunta de usabilidad #12

Según las respuestas dadas, podemos concluir que el 40% de los usuarios recomendaría definitivamente el sistema *web* de gestión de conferencias con pagos en línea a sus amigos, lo que sugiere que una gran parte de los usuarios se siente satisfecha con la experiencia y la considera útil y efectiva. Además, el 25% de los usuarios probablemente recomendaría el sistema *web* a sus amigos, lo que indica que aún tienen ciertas dudas o preocupaciones, pero en general tienen una opinión positiva. Por otro lado, el 10% de los usuarios probablemente no recomendaría el sistema *web* a sus amigos, y solo el 5% definitivamente no lo recomendaría, lo que sugiere que solo una minoría de los usuarios se siente muy insatisfecha con la experiencia. Sin embargo, el 20% de los usuarios no está seguro de si recomendaría o no el sistema *web*, lo que indica que aún hay ciertas dudas o necesidades no satisfechas que podrían abordarse en futuras actualizaciones del sistema.

Despliegue en servidor privado

Con lo que corresponde al despliegue del sistema en un servidor con dirección IP pública se consideró lo siguiente:

La configuración del servidor: En esta sección, se pueden ver detalles sobre el servidor como el nombre del servidor, el sistema operativo y la versión, la dirección IP, los roles y características de seguridad. También se pueden realizar algunas configuraciones, como cambiar el nombre del servidor, agregar o quitar roles y características, habilitar o deshabilitar características, y cambiar la configuración de energía con el panel de control de *Windows Server*. Tal como se muestra en las **Fig 114** y **Fig 115**.

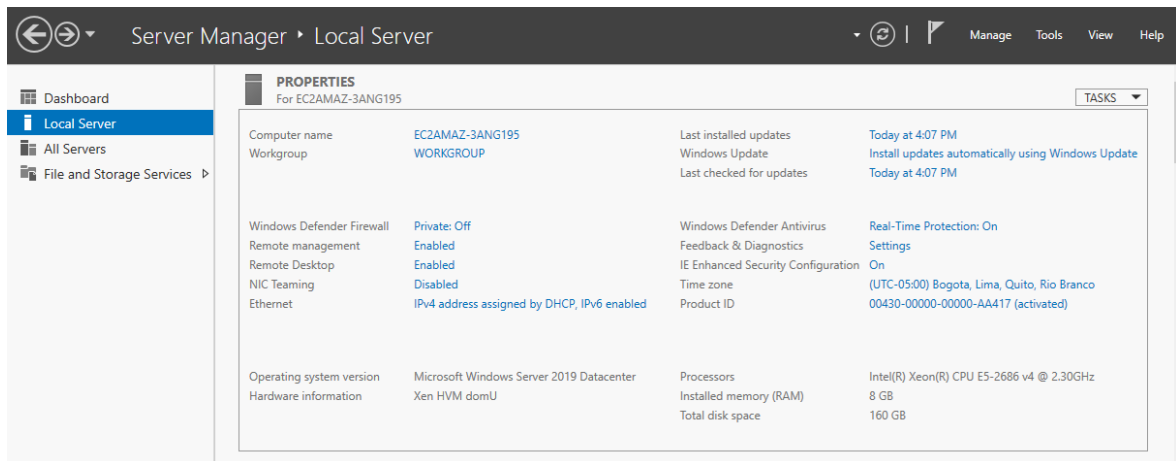


Fig 114. Panel de control de *Windows Server*

```

Hostname: EC2AMAZ-3ANG195
Instance ID: i-064eee77107c82c79
Public IPv4 Address: 34.246.221.132
Private IPv4 Address: 172.26.5.0
IPv6 Address: 2a05:d018:5ee:6100:5a4c:1e88:a359:80a5
Instance Size: t2.large
Availability Zone: eu-west-1a
Architecture: AMD64
Total Memory: 8192 MB
Network Performance: Low to Moderate

```

Fig 115. Características del servidor

Antes de poder desplegar, se debe configurar un servidor *web* en el servidor. Se uso Apache para este caso, para lo cual se configuro lo siguiente:

- La instalación de PHP y las dependencias requeridas: Laravel requiere una versión específica de PHP y varias dependencias adicionales. Clonar el proyecto en el servidor: Para el despliegue, se copió desde el repositorio todos los archivos del proyecto en el servidor.
- Configuración el archivo `.env` para almacenar información de credenciales de la base de datos y claves de seguridad.
- Instalación las dependencias de Composer: Laravel utiliza Composer para manejar sus dependencias.

La presenta una configuración de Apache para el uso de *virtual hosts*. Un *virtual host* es un término utilizado para describir la práctica de alojar más de un sitio *web* en un solo servidor. En la configuración presentada, se definen los *virtual hosts* para el sitio llamado

"conferenciaEPN.com". Cada host se define mediante un contenedor `<VirtualHost>` que especifica la dirección IP y el puerto que el servidor *web* utilizará para servir el sitio *web*. En este caso, el asterisco (*) indica que el servidor escuchará en todas las direcciones IP disponibles en el puerto 80. Dentro de cada contenedor `<VirtualHost>`, se especifican las directivas de Apache necesarias para configurar el sitio *web*. Por ejemplo, `DocumentRoot` especifica la ubicación de los archivos del sitio *web* y `ServerName` especifica el nombre del servidor utilizado para acceder al sitio *web*. Tal como se muestra en la **Fig 116**.

```
35
36  ##<VirtualHost *:80>
37     ##ServerAdmin webmaster@dummy-host2.example.com
38     ##DocumentRoot "C:/xampp/htdocs/dummy-host2.example.com"
39     ##ServerName dummy-host2.example.com
40     ##ErrorLog "logs/dummy-host2.example.com-error.log"
41     ##CustomLog "logs/dummy-host2.example.com-access.log" common
42  ##</VirtualHost>
43  <VirtualHost *:80>
44     DocumentRoot "C:/xampp/htdocs/proyecto/public"
45     ServerName conferenciaEPN.com
46  </VirtualHost>
47
```

Fig 116. Configuración de Apache

En la sección presentada, se especifica que la dirección IP 34.246.221.132 (que es la dirección IP pública) se utilizará para resolver el nombre de host "conferenciaEPN.com". En otras palabras, al acceder al sitio web "conferenciaEPN.com" en el navegador, el sistema operativo lo resolverá como si se estuviera accediendo al servidor local. Tal como se muestra en la **Fig 117**.

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10      x.acme.com              # x client host
#
# localhost name resolution is handled within DNS itself.
#   127.0.0.1      localhost
#   ::1           localhost
34.246.221.132 conferenciaEPN.com
```

Fig 117. Nombre de host

Prueba del despliegue: Finalmente, se probó el despliegue visitando la dirección IP pública del servidor en un navegador web. Se pudo ver el sistema web correctamente en el servidor. Adicionalmente se comprobó que el acceso al sistema web desde solo se hace desde la dirección IP proporcionada. Tal como se muestra en la **Fig 118**.



Fig 118. Prueba del despliegue

ANEXO III

El manual de usuario de este proyecto es un video que muestra la funcionalidad completa del sistema web en acción. A través de una demostración detallada, los usuarios podrán conocer todas las características del sistema y cómo utilizarlas para maximizar su experiencia. El video incluye una visión general de la interfaz de usuario y de todas las opciones disponibles para los usuarios, así como instrucciones paso a paso sobre cómo utilizar las diferentes funcionalidades del sistema, como la gestión de usuarios, conferencias, pagos etc.

En el siguiente enlace se encuentra el manual de usuario mostrando la funcionalidad del sistema web:

https://www.youtube.com/watch?v=PCDvw1ztz_Q

ANEXO IV

El manual de instalación es una guía detallada para instalar y configurar el sistema a través de la clonación del repositorio del proyecto en GitHub. En este anexo, además de las instrucciones para la instalación, también se incluyen las credenciales necesarias para registrarse como administrador en el sistema, lo que permitirá a los usuarios acceder a todas las funcionalidades de este. Además, se proporciona el enlace directo al sistema web, para que los usuarios puedan acceder fácilmente a él.

En el siguiente enlace se encuentra el enlace al repositorio en GitHub en donde se encuentra el código fuente y en el apartado de README los pasos para realizar la instalación:

[GitHub - SebasVa/Trabajo_Integraci-n_Curricular_Sebasti-n_Valencia](#)

Link a la página web:

El sistema web se encuentra subido en un servidor privado por lo que se puede ingresar por medio de esta dirección: <http://34.246.221.132/>

Credenciales de acceso:

Puede iniciar sesión en el panel de administración yendo a /login e iniciar sesión con credenciales

- Usuario: admin@admin.com Clave: password
- Usuario: user@user.com Clave: password