

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SISTEMA WEB PARA LA ADMINISTRACIÓN DE EVENTOS ORGANIZADOS POR INCONCERTO

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

Jhonathan Xavier Pizarra Chiriboga

jhonnathan.pizarra@epn.edu.ec

DIRECTOR: ING. Edwin Gonzalo Salvador Pesantes, MSC.

edwin.salvador@epn.edu.ec

CODIRECTOR: PhD. Tania Elizabeth Calle Jiménez, MSC.

tania.calle@epn.edu.ec

Quito, septiembre 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr. Pizarra Chiriboga Jhonnathan Xavier como requerimiento parcial a la obtención del título de TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS, bajo nuestra supervisión:



Ing. Edwin Salvador, MSc.

DIRECTOR DEL PROYECTO

Dra. Tania Calle, MSc.

CODIRECTORA DEL PROYECTO

DECLARACIÓN

Yo Jhonnathan Pizarra Xavier Chiriboga con CI: 1725358525 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

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

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

DEDICATORIA

A Dios, quien ha sido mi motivación, mi fortaleza en las asperezas y vicisitudes presentadas a lo largo de la vida académica.

A mi familia, los cuales son una fuente de inspiración por su duro trabajo y sacrificio para cumplir los objetivos propuestos.

JHONNATHAN XAVIER PIZARRA CHIRIBOGA

AGRADECIMIENTO

Quiero agradecer primero a Dios, por brindarme la oportunidad estudiar en la Escuela Politécnica Nacional, por haberme guiado siempre por el buen camino, para obrar en bien del prójimo, y también, por darme salud y entendimiento para poder culminar este proyecto.

A mi madre y abuelita, Zulema y Dolores por su infinito amor y apoyo. Gracias a ellas he logrado cumplir este objetivo de mi vida, además de convertirme en una persona de bien, y el profesional que soy.

A mis hermanos, Nelson, Marco y Omar quienes me apoyaron, cuidaron y creyeron en mi durante mis estudios. Gracias por sus valiosos consejos.

A mis compañeros y amigos de prepo de la “ESFOT 39” y todas las personas que estuvieron desde el comienzo de mi carrera. Gracias por brindarme su amistad, su respeto y compañía, sin ustedes nada hubiera sido igual.

A mis docentes, por haberme transmitido sus conocimientos y formar mi vida profesional. De manera especial agradezco a los ingenieros Luis Pillajo, Richard Rivera, Ivonne Maldonado, Luis Ponce, Silvia Armendaris, Mauricio Rojas, Byron Loarte, Luis Almeida, Gabriela Cevallos, Mónica Vinueza, y Juan Pablo Zaldumbide, a todos, gracias por su apoyo y sus enseñanzas.

Finalmente, a mis ingenieros Edwin Salvador y Tania Calle, por la paciencia y apertura en la elaboración del presente proyecto quienes con sus valiosos consejos, enseñanzas y correcciones demostraron tener profesionalismo y vocación de ser docentes.

JHONNATHAN XAVIER PIZARRA CHIRIBOGA

ÍNDICE DE CONTENIDOS

1	Introducción	1
1.1	Objetivo general	1
1.2	Objetivos específicos.....	2
1.3	Alcance.....	2
2	Metodología.....	4
2.1	Metodología de Desarrollo.....	4
2.1.1	Roles	4
2.1.2	Artefactos	5
2.2	Diseño de interfaces	7
2.2.1	Herramienta utilizada para el diseño.....	7
2.3	Diseño de la arquitectura.....	8
2.3.1	Patrón arquitectónico.....	8
2.4	Herramientas de desarrollo.....	9
3	Resultados y Discusión.....	11
3.1	Sprint 0. Configuración del ambiente de desarrollo.....	11
3.2	Sprint 1. Inicio de sesión del usuario administrador, usuarios, eventos, conciertos, ensayos, artistas, calendarios y notificaciones.	18
3.3	Sprint 2. Transportes, hospedajes, lugares de concierto, tareas.....	28
3.4	Sprint 3. Recursos	31
3.5	Sprint 4. Cuadros alimenticios, lugares de alimentación	32
3.6	Sprint 5. Iniciar sesión del usuario operario	34
3.7	Sprint 6. Pruebas y despliegue del Sistema Web	34
4	Conclusiones y Recomendaciones	39
4.1	Conclusiones.....	39
4.2	Recomendaciones.....	40
5	Referencias Bibliográficas	41

6	ANEXOS.....	i
6.1	Manual Técnico	i
6.2	Manual de Usuario.....	i
6.3	Manual de Instalación.....	i

ÍNDICE DE FIGURAS

Fig. 1: Interfaz del inicio de sesión del Sistema Web.....	8
Fig. 2: Arquitectura del Sistema Web	9
Fig. 3: Base de datos	12
Fig. 4: Ejecución del proyecto en Laravel.....	13
Fig. 5: Estructura del proyecto.....	14
Fig. 6: Migraciones creadas	15
Fig. 7: Columnas de la tabla users	15
Fig. 8: Rutas de entidad concierto	16
Fig. 9: Perfiles, usuarios y módulos del Sistema Web	17
Fig. 10: Página de inicio del Sistema Web	19
Fig. 11: Inicio de sesión para los usuarios registrados en el Sistema Web.....	20
Fig. 12: Formulario de solicitud de correo.....	20
Fig. 13: Bandeja de entrada del usuario.....	21
Fig. 14: Formulario de restablecimiento de contraseña	21
Fig. 15: Listado de usuarios	22
Fig. 16: Listado de eventos	23
Fig. 17: Listado de conciertos.....	24
Fig. 18: Listado de ensayos.....	25
Fig. 19: Listado de artistas	26
Fig. 20: Listado de calendarios.....	27
Fig. 21: Notificación del próximo evento.....	27
Fig. 22: Listado de transportes	28
Fig. 23: Listado de hospedajes.....	29
Fig. 24: Listado de lugares de concierto.....	30
Fig. 25: Listado de tareas.....	31
Fig. 26: Lista de recursos	32
Fig. 27: Lista de cuadros alimenticios.....	33
Fig. 28: Lista de lugares alimenticios.....	34
Fig. 29: Configuración del grupo de hilos en JMeter.....	35
Fig. 30: Resultados de la prueba #1 con 100 peticiones simultaneas.....	36
Fig. 31: Despliegue del Sistema Web.....	38

ÍNDICE DE TABLAS

TABLA I: Equipo de trabajo y asignación de roles	5
TABLA II: Historia de Usuario 1 – Inicio de sesión del usuario Administrador	6
TABLA III: Herramientas utilizadas en el desarrollo del Sistema Web.....	9
TABLA IV: Librerías utilizadas en el desarrollo del Sistema Web.....	10
TABLA V: Versiones de navegadores.....	37
TABLA VI: Prueba de funcionalidad - Inicio de sesión del administrador	37

RESUMEN

El presente trabajo surge de la necesidad de una gestión de eventos organizados por la fundación Inconcerto; por esta razón, se ha implementado un Sistema Web para llevar un control de artistas invitados, hospedajes, conciertos, ensayos, notificaciones de próximos eventos a realizarse, entre otros.

El desarrollo de este Sistema Web para el control de eventos organizados por Inconcerto permite gestionar adecuadamente los próximos eventos a realizarse apoyando a la difusión de la música clásica, reconocimiento y actividades de la institución, así como a personas con discapacidad, personas privadas de libertad, tercera edad, niños y niñas, habitantes de barrios periféricos, pacientes hospitalarios, escuelas de bajos recursos y al público en general que disfrute conciertos de música clásica.

Este trabajo se ha realizado bajo un entorno de desarrollo con Laravel, el cual es un *Framework* de PHP, React, que es una librería de JavaScript para el desarrollo de interfaces de usuario y el Sistema Gestor de Base de Datos (SGBD) MySQL, debido a su rendimiento y escalabilidad. Durante el desarrollo del Sistema Web se ha utilizado la metodología ágil *Scrum*, aportando un sin número de beneficios para el proyecto integrador, como: interacción directa con el usuario, adecuada recopilación de requerimientos, definición de roles, flexibilidad, entre otros. Permitiendo de ese modo que el desarrollo se realice en los tiempos establecidos. Posterior a ello, se presenta los resultados que se han obtenido en las tareas realizadas por cada *Sprint* y el despliegue a producción del Sistema Web. Finalmente, en la última sección se mencionan las conclusiones y recomendaciones que se han obtenido a lo largo del desarrollo del proyecto.

PALABRAS CLAVE: *Scrum*, Sistema Web, Gestión de eventos, Laravel, MySQL.

ABSTRACT

This work arises from the need to manage events organized by the Inconcerto foundation; For this reason, a Web System has been implemented to carry out a control of invited artists, accommodations, concerts, rehearsals, notifications of upcoming events to be held, among others.

The development of this Web System for the control of events organized by Inconcerto manage adequately allows the next events to be held, supporting the dissemination of classical music, recognition and activities of the institution, as well as people with disabilities, people deprived of, third age, boys and girls, residents of suburbs, hospital patients, low-income schools and the general public who enjoy classical music concerts.

This work has been carried out under a development environment with Laravel, which is a PHP Framework, React, which is a JavaScript library for the development of user interfaces and the MySQL Database Management System (DBMS), for its performance and scalability. During the development of the Web System, the agile Scrum methodology has been used, which has provided a series of benefits for the integrating project, such as: adequate requirements gathering, flexibility, well-defined roles, direct interaction with the user, among others. This has allowed the development to be carried out with quality and on time. Subsequently, the results obtained in the tasks performed by each Sprint and the deployment to production of the Web System on the server provided by the foundation are presented. Finally, the last section mentions the conclusions and recommendations that have been obtained throughout the development of this project.

KEYWORDS: *Scrum*, Web System, event management plan, Laravel, MySQL

1 INTRODUCCIÓN

El presente informe se centra en el desarrollo de un sistema web para la gestión de eventos organizados por la fundación Inconcerto, aplicando la metodología *Scrum*. Logrando de esta manera la automatización de los procesos llevados manualmente, mejorando la organización y experiencia de gestión para la fundación.

Inconcerto es un colectivo interdisciplinario de artistas, sociólogos, comunicadores y gestores culturales que crean espacios y canales para difundir y revalorizar la música clásica en esta época; a través de proyectos que resignifican los protocolos que han situado a ésta, como un arte de y para una determinada clase social e intelectual [1].

A partir de la experiencia de músicos, gestores e investigadores, se ha identificado la concentración de la música clásica en un sector minoritario y un número limitado de espacios, en su mayoría en las principales urbes del país, lo cual ha impedido su constitución como una práctica cultural de los distintos grupos de la sociedad, en particular de poblaciones en condiciones de doble vulnerabilidad.

Adicionalmente, la larga trayectoria que ha tenido desde el año 2012 con actividades artísticas para los distintos públicos, en particular con personas que no han tenido contacto con este tipo de arte, han comprobado el potencial de la música de provocar, despertar y reactivar afectos y reacciones únicas en las audiencias.

Actualmente, el colectivo no cuenta con un sitio web en el que se pueda publicar actualizaciones sobre los eventos que se organizan a lo largo del año y la logística de todos los eventos organizados son llevados mediante un documento de Excel, lo que, en algunas ocasiones, genera conflictos para la correcta organización de los eventos y la comunicación entre los involucrados.

Por lo citado anteriormente, se ha desarrollado un sistema web que permite llevar un control de los eventos, artistas invitados, hospedajes, conciertos, notificaciones sobre los próximos eventos a realizarse, entre otros. Mejorando la organización y experiencia de gestión para la fundación.

1.1 Objetivo general

Desarrollar un sistema web para la administración de eventos organizados por la fundación Inconcerto.

1.2 Objetivos específicos

- Levantar los requerimientos del sistema web para la administración de eventos.
- Diseñar la arquitectura bajo la metodología Scrum para desarrollar el sistema web.
- Implementar el sistema web para la administración de la fundación Inconcerto.
- Probar la funcionalidad, usabilidad y rendimiento.
- Implantar el sistema en el servidor proporcionado por la fundación.

1.3 Alcance

Internet cada día avanza a pasos agigantados y con él crecen las oportunidades de negocio, estudio, comunicación, y muchas otras cosas basadas en posicionar y ofrecer a los internautas información, productos y servicios que puedan encontrar a un clic de distancia [2]. Cuando una organización cuenta con un sitio web incrementa su reputación, a la vez que los usuarios elevan su nivel de confianza en el producto o servicio que ofrece [3]. La importancia de un sitio web para una organización radica en que esto supondrá en la credibilidad de esta, además, si es una web dinámica, representa que está al día y al tanto de lo que millones de usuarios necesitan, y existe un buen número de organizaciones que no cuentan con un sitio web [4] .

Por tal motivo, se ha procedido a la implementación de un sistema web capaz de brindar y cubrir las necesidades del colectivo, permitiendo obtener y organizar información de manera detallada, eficiente y personalizada sobre los eventos, conciertos, recursos, entre otros. Con el objetivo de ofrecerles una herramienta de solución tecnológica.

El sistema web propuesto garantiza al colectivo la integridad, consistencia y seguridad de los datos, por medio de la autenticación de credenciales, a su vez, permite la gestión de diferentes ámbitos que componen un evento, como son:

- Gestión de artistas invitados: Los artistas son nacionales e internacionales por lo que se lleva el control de compras de pasajes, tiempos de llegada y salida, observaciones de alimentación especiales, transporte, etc.
- Gestión de hospedajes: Se coordina donde se hospedan los artistas invitados cada años o evento teniendo en cuenta los horarios de entrada y salida de los artistas con el fin de optimizar los gastos.
- Gestión de auspicios: La fundación cuenta con varios auspicios, entre los cuales están hostales y hoteles, alimentación, transporte, lugares para conciertos y auspicios directamente económicos para cubrir los gastos de los eventos. Se

lleva un control detallado de los auspicios con los que se cuentan cada año para poder utilizar los recursos de la mejor manera.

- Gestión de espacios para conciertos: La fundación está constantemente en búsqueda de espacios poco usuales para ofrecer conciertos. Se lleva un control de estos espacios para facilitar la organización de cada evento.
- Adicionalmente, la fundación cuenta con medios adecuados para la difusión de sus actividades y un medio de comunicación directo con las personas interesadas en sus eventos.

2 METODOLOGÍA

Las metodologías ágiles son procesos de la ingeniería de software que permiten responder adecuadamente a las condiciones propuestas por el cliente, permitiendo inmediatez y flexibilidad [5]. Una comunicación efectiva con el cliente y miembros del equipo de trabajo ayudan en este tipo de proyectos donde los requerimientos son cambiantes.

Las subsecciones siguientes describen la implementación de la metodología *Scrum* en el desarrollo de este proyecto integrador. Posterior a ello, se procede con las siguientes etapas que son la elaboración de Historias de Usuario, *Product Backlog* y *Sprints*.

2.1 Metodología de Desarrollo

La metodología utilizada en el proyecto es *Scrum*, la cual se basa en el trabajo por iteraciones o *Sprints*. *Scrum* es una metodología ágil que tiene por objetivo planificar y controlar proyectos con una gran medida a cambios de última hora, en donde la incertidumbre se eleva [6]. Garantizando en todo momento la calidad de los sistemas propuestos y el cumplimiento de los requerimientos establecidos.

2.1.1 Roles

Scrum al ser una metodología ágil basado en roles asegura el control de todos los procesos durante el desarrollo del software. Además, las personas a quienes se les asigna un rol deben comprometerse y cumplir con las necesidades del proyecto propuesto [7]. A continuación, se especifican los roles del proyecto integrador:

Product Owner

Es la persona encargada de revisar los resultados y mantener el control de todos los entregables por parte del equipo de desarrollo [7]. Este rol es representado por la productora del programa Inconcerto, quien proporciona todos los requerimientos e información necesaria en los procesos de gestión de los eventos, logrando con esta información determinar perfiles, módulos y funcionalidades para el sistema web.

Scrum Master

Es el líder del proyecto y su responsabilidad es la de guiar y cumplir las reglas de la metodología ágil *Scrum* [7]. Este rol es representado por el director del proyecto integrador, quien posee los conocimientos necesarios y técnicos para ayudar al equipo

de desarrollo a solventar dudas e inconvenientes y lograr el cumplimiento del objetivo planteado.

Development Team

El presente rol está representado por el autor y desarrollador del proyecto, quien dispone de todos los conocimientos y habilidades necesarias para transformar las tareas del *Sprint Backlog* en pequeños entregables funcionales que aporten valor a los usuarios finales [7] quien junto al *Scrum Master* toman las decisiones adecuadas para conseguir los objetivos propuestos durante el desarrollo de cada *Sprint* del proyecto.

Para cumplir con el desarrollo del proyecto integrador los roles se han dividido como se describe en la **TABLA I**.

TABLA I: Equipo de trabajo y asignación de roles

NOMBRE	ROL
Srta. Isabel Paredes	<i>Product Owner</i>
Ing. Edwin Salvador, MSc.	<i>Scrum Master</i>
Jhonnathan Pizarra	<i>Development Team</i>

2.1.2 Artefactos

Son elementos utilizados por *Scrum*, diseñados para garantizar el registro y la transparencia de la información fundamental del proceso de *Scrum* [8]. Permitiendo de esta manera organizar el desarrollo por etapas y que los roles mencionados anteriormente puedan realizar las tareas definidas en cada *Sprint*. A continuación, se detallan los artefactos que se han utilizado:

Recopilación de Requerimientos

Para dar inicio a la primera etapa para el desarrollo del software, se define con claridad la lista de requerimientos del cliente permitiendo desarrollar sistemas óptimos y apegados a la necesidad del negocio [9]. Por tal motivo, para cumplir con esta etapa se han mantenido una serie de reuniones con el *Product Owner* e interesados en el proyecto, permitiendo obtener la lista inicial de requerimientos general para el desarrollo del sistema web, la cual se encuentra detallada dentro del Manual Técnico en el apartado recopilación de requerimientos (pág. 2 - 4).

Historias de Usuario

Consiste en una descripción general de una función de software descrita desde el punto de vista del usuario final. Tiene como propósito estipular cómo esa función aporta valor al negocio del cliente [10]. Estas se basan en reuniones con los miembros del proyecto entre el cliente y el equipo de desarrollo. A continuación, se detalla los elementos que componen una Historia de Usuario:

- **Identificador (ID):** Es un identificador único alfanumérico.
- **Usuario:** Información del perfil del usuario.
- **Nombre Historia:** Nombre para definir la Historia de Usuario.
- **Prioridad en negocio:** Califica la prioridad del negocio en una escala de alta, medio y bajo.
- **Riesgo de desarrollo:** Califica el grado de riesgo en el desarrollo del *Sprint*.
- **Iteración:** Calificativo numérico de un *Sprint*.
- **Responsable:** Miembro del equipo de desarrollo quien es el encargado de dicha funcionalidad.
- **Descripción:** Información obtenida de la Historia de Usuario.
- **Observación:** Información breve que se debe de tomar antes de elaborar dicha funcionalidad.

La **TABLA II** detalla una de las 55 Historias de Usuario que se han obtenido para el desarrollo del sistema web. Por otra parte, se detallan las demás historias de usuario dentro del Manual Técnico en el apartado Historias de Usuario (pág. 5 - 32).

TABLA II: Historia de Usuario 1 – Inicio de sesión del usuario

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Administrador/Operario
Nombre Historia: Iniciar sesión usuario Administrador/Operario	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alta
Iteración Asignada: 1	
Responsable (es): Jhonathan Pizarra	
Descripción: Para el acceso a las funcionalidades, el sistema web posee autenticación de perfiles uno de ellos es del administrador quien para identificarse es necesario el correo institucional y contraseña las cuales fueron proporcionadas por el desarrollador.	

Observación:

El correo debe ser el institucional del colectivo Inconcerto y la clave es proporcionada por el desarrollador del sistema web.

Product Backlog

Es una lista ordenada de todos los requerimientos que sean necesarios para la elaboración del proyecto [11]. En el *Product Backlog* se lista de forma ordenada los requerimientos funcionales del Sistema Web, permitiendo ordenar estos requerimientos funcionales según la prioridad que se tiene en el negocio y el nivel de complejidad que puede tener durante el desarrollo. Esta lista se encuentra detallada dentro del Manual Técnico en el apartado *Product Backlog* (pág. 33 - 34).

Sprint Backlog

El *Sprint Backlog* es una lista de tareas identificadas por el equipo de *Scrum* que se completa durante el *Sprint*. Durante la reunión de planificación del *Sprint*, el equipo selecciona algunas historias de usuario e identifica las tareas necesarias para completarlas [12].

En base a la información obtenida anteriormente y siguiendo la metodología, se procede con el desarrollo de 6 *Sprints*: Configuración del ambiente de desarrollo, Inicio de sesión del usuario, eventos, alimentación, tareas, logística, personal. Finalmente, pruebas de rendimiento y el despliegue del sistema web a producción. Esta lista de actividades por cada *Sprint* se encuentra detallado dentro del Manual Técnico en el apartado *Sprint Backlog* (pág. 35 - 40).

2.2 Diseño de interfaces

Una vez determinado los *Sprints* para el desarrollo del proyecto, a continuación, se detalla la herramienta que se ha utilizado para el diseño de las interfaces del sistema web.

2.2.1 Herramienta utilizada para el diseño

Para el diseño de las interfaces y cada una de las funcionalidades del sistema web se han realizado prototipos (*mockups*) utilizando para ello la herramienta Balsamiq mockups, permitiendo la creación de los prototipos del sistema web de forma rápida y sencilla [13]. Tomando en cuenta la experiencia de usuario (UX) como la consistencia, accesibilidad, usabilidad, disponibilidad y simplicidad.

La **Fig. 1** ilustra el prototipo de la interfaz del inicio de sesión del sistema web, mientras que el detalle de los 23 prototipos restantes se encuentra dentro del Manual Técnico en el apartado Diseño de Interfaces (pág. 41 – 52).

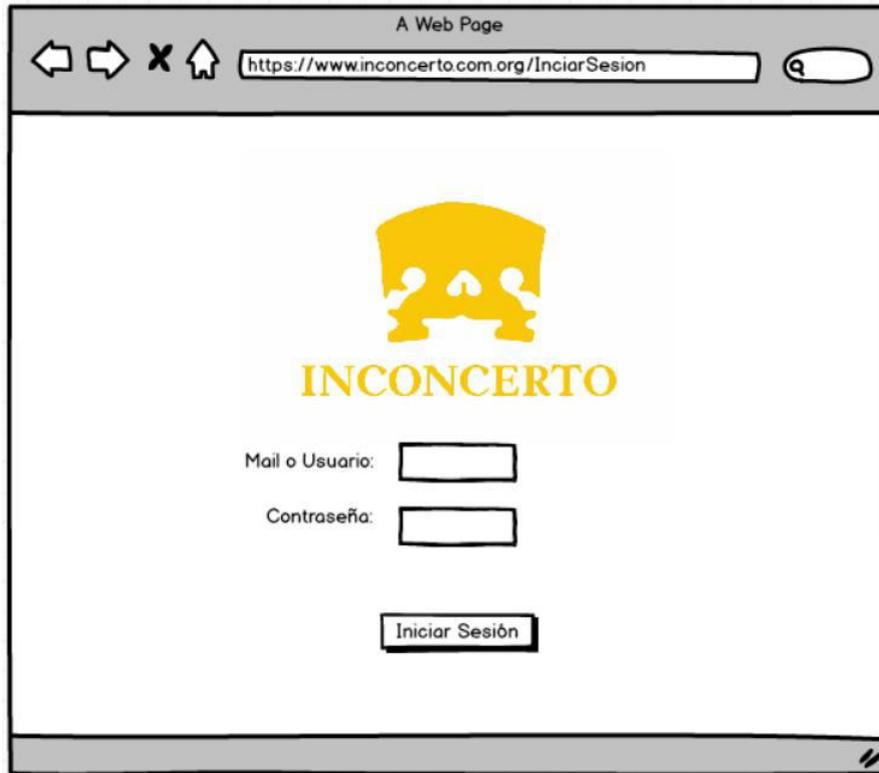


Fig. 1: Interfaz del inicio de sesión del Sistema Web

2.3 Diseño de la arquitectura

Después de haber establecido los *Sprints* para este proyecto conjuntamente con el diseño de las interfaces para el sistema web. A continuación, se describe el patrón de arquitectura que se ha utilizado para el desarrollo.

2.3.1 Patrón arquitectónico

El sistema web está basado sobre el patrón de arquitectura Modelo, Vista y Controlador (MVC). Consiste en un modelo maduro que ha demostrado su validez en todo tipo de aplicaciones, sobre multitud de lenguajes y plataformas de desarrollo [14]. Permitiendo la clasificación de los componentes de una aplicación en tres grupos (o capas) primordiales: el modelo, la vista, y el controlador, y describe cómo se relacionan entre ellos para mantener una estructura organizada, limpia y acoplada entre las distintas capas [15].

Una vez analizado las ventajas que se tiene al usar un patrón de arquitectura de 3 capas. La **Fig. 2** ilustra el detalle de la implementación de la arquitectura junto a las herramientas para el desarrollo del sistema web.

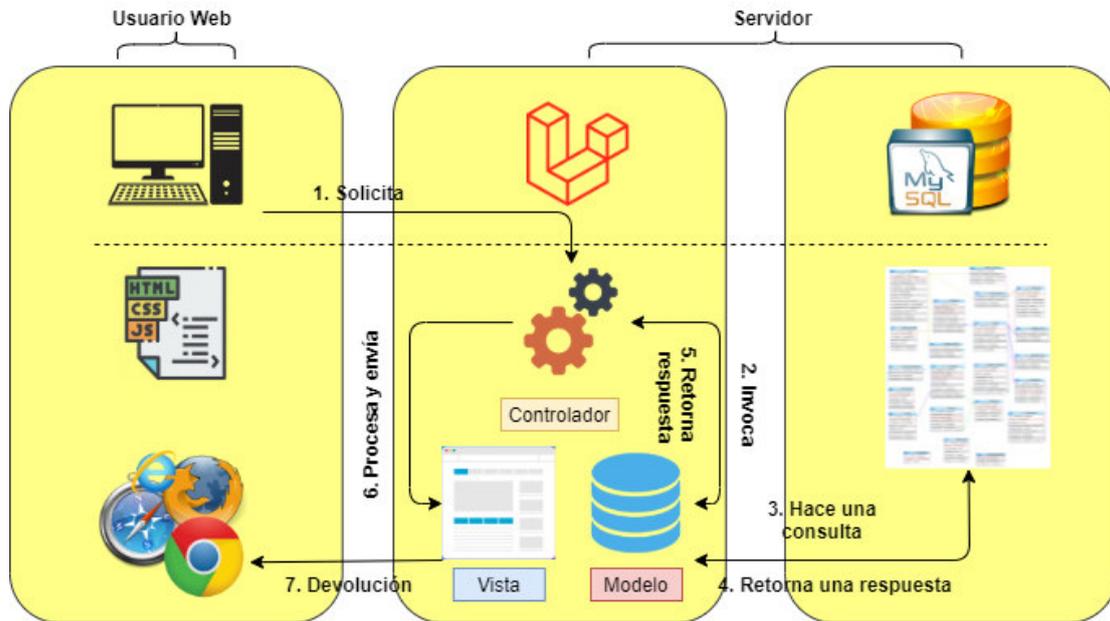


Fig. 2: Arquitectura del Sistema Web

2.4 Herramientas de desarrollo

Una vez establecido el patrón arquitectónico y los requerimientos funcionales. A continuación, en las **TABLA III** y **TABLA IV** se detallan las herramientas y librerías utilizadas para desarrollo del sistema web.

TABLA III: Herramientas utilizadas en el desarrollo del Sistema Web

HERRAMIENTA	JUSTIFICACIÓN
XAMPP	Consiste en un sistema de gestión de base de datos MySQL y de servidor Apache, así como de intérpretes de lenguajes PHP y Perl [16]. Es una herramienta idónea para convertir el computador en un servidor Apache y MySQL, perfecto para el desarrollo del Sistema Web [16].
Composer	Es un gestor de dependencias en proyectos, que permite gestionar los paquetes de software en los que se basa un proyecto de PHP [17].

JWT	Se define como un mecanismo para propagar de forma segura la información de un usuario, a través de un método de autenticación en servicios API. En dónde el cliente envía sus credenciales de acceso, y luego, la API le retorna un token que se envía en todas las peticiones que el usuario haga, para que ésta compruebe que tiene acceso a las acciones que se quieran realizar [18].
Laravel	La implementación de este <i>Framework</i> de tipo MVC permite codificar todo el Sistema Web, haciendo énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad [19].
Material-UI	La implementación de Material-UI permitió la creación del sistema web de forma más rápidas, hermosa y accesible [20]. Además de tener un diseño adaptable a los diferentes dispositivos.
JavaScript	Es un lenguaje de secuencias de comandos que permite implementar funciones complejas en las páginas web y crear contenido dinámico en el proyecto [21].
HTML5	La integración de HTML permite en el desarrollo del proyecto distribuir de forma ordenada todo el contenido del sistema web en base a etiquetas preestablecidas [22].
CSS	La integración de CSS permite en el desarrollo del proyecto establecer una capa de personalización sobre el sistema web para que sea visualmente agradable para el usuario [23].
MySQL	La integración de MySQL permite trabajar con bases de datos relacionales, almacenar información y administrar los datos de nuestro proyecto a través de las relaciones de las tablas [24].

TABLA IV: Librerías utilizadas en el desarrollo del Sistema Web

LIBRERÍA	DESCRIPCIÓN
"react-hook-form": " ^7.2.1",	Librería para usar formularios [25].
"yup": " ^0.32.9"	Librería para validar formularios [26].

"@material-ui/icons": "^4.11.2",	Librería para usar íconos [27].
----------------------------------	---------------------------------

3 RESULTADOS Y DISCUSIÓN

En esta sección, se detallan los resultados obtenidos en cada una de las tareas de los 6 *Sprints* de desarrollo.

3.1 Sprint 0. Configuración del ambiente de desarrollo

Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 0* detalla las actividades concernientes a la configuración del ambiente de desarrollo contemplan las siguientes tareas:

- Creación de la Base de Datos.
- Creación del proyecto con Laravel.
- Estructura del proyecto.
- Creación de migraciones, modelos, rutas y controladores.
- Usuarios.
- Requerimientos específicos para el Sistema Web.

3.1.1 Creación de la Base de Datos

Para la creación de la Base de Datos se ha utilizado MySQL, el cual almacena toda la información de usuarios, eventos, recursos, artistas, tareas, notificaciones de los próximos eventos, entre otros.

La **Fig. 3** ilustra la Base de Datos creada en MySQL, esta Base de Datos contiene 22 tablas relacionadas entre sí por medio de las claves primarias¹ (PK) y foráneas² (FK) siguiendo los parámetros establecidos por el *Framework* para una fácil integración y comunicación. Finalmente, el diseño completo se encuentra dentro del Manual Técnico en el apartado Diseño de la Base de Datos (pág. 52).

¹ **Clave Primaria:** Es una llave primaria o identificación de una tabla en la Base de Datos relacional.

² **Clave foránea:** Es una llave foránea empleada para las tablas secundarias que necesitan conectarse a la principal en la Base de Datos relacional.

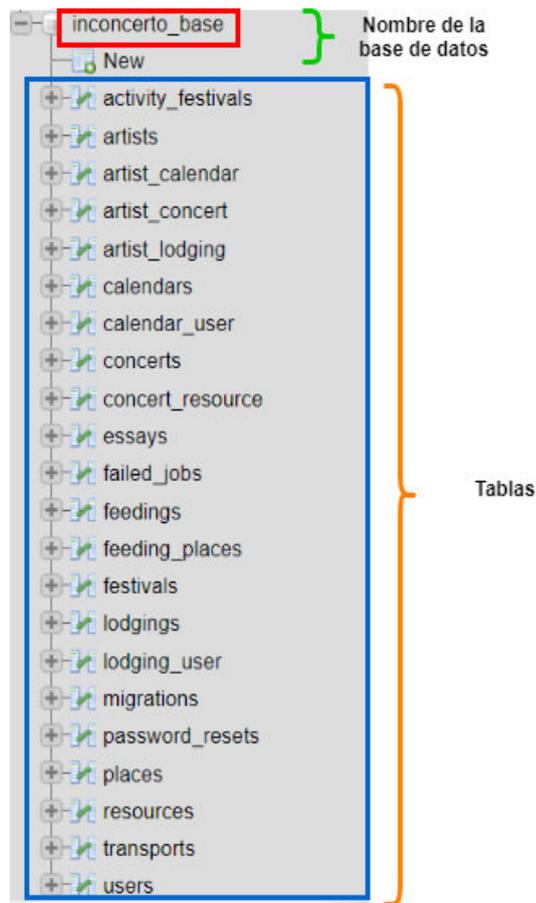


Fig. 3: Base de datos

3.1.2 Creación del proyecto con Laravel

Se procede a la creación de un proyecto en Laravel, usando la línea de comandos y el sistema gestor de dependencias *Composer*³, el cual permite gestionar las instalaciones de todos los paquetes y recursos necesarios para la creación de un proyecto en Laravel. A continuación, se presenta los comandos que se han utilizado para la creación del proyecto. Mientras que la **Fig. 4**, ilustra la ejecución del proyecto de forma satisfactoria.

```
composer create-project laravel/laravel PROYECTO_Inconcerto-Back-End
```

```
php artisan serve
```

³ **Composer:** es un administrador de dependencias para la instalación, modificación y eliminación de paquetes [36].



Fig. 4: Ejecución del proyecto en Laravel

3.1.3 Estructura del proyecto

Para la codificación y estructura de directorios y archivos del sistema web, se ha utilizado *PhpStorm*⁴, tomando en cuenta el patrón arquitectónico de tres capas (MVC) como ilustra la **Fig. 5**. Por último, se detalla el contenido de los directorios y archivos más relevantes:

- **Directorio app:** contiene los controladores, modelos, middleware, providers, reglas de validación entre otros.
- **Directorio config:** contiene archivos de rutas, cors, entre otras configuraciones del proyecto.
- **Directorio database:** contiene las tablas y los seeders creadas por las migraciones.
- **Directorio public:** contiene los archivos y directorios públicos para el usuario final.
- **Directorio resources:** contiene los archivos de la personalización de la respuesta del API.
- **Directorio routes:** contiene el archivo para el direccionamiento de las rutas del proyecto.
- **Archivo. env:** contiene la configuración del proyecto.

⁴ **PhpStorm:** editor de texto para HTML, JavaScript y PHP, con análisis de código y funciones automatizadas [35].

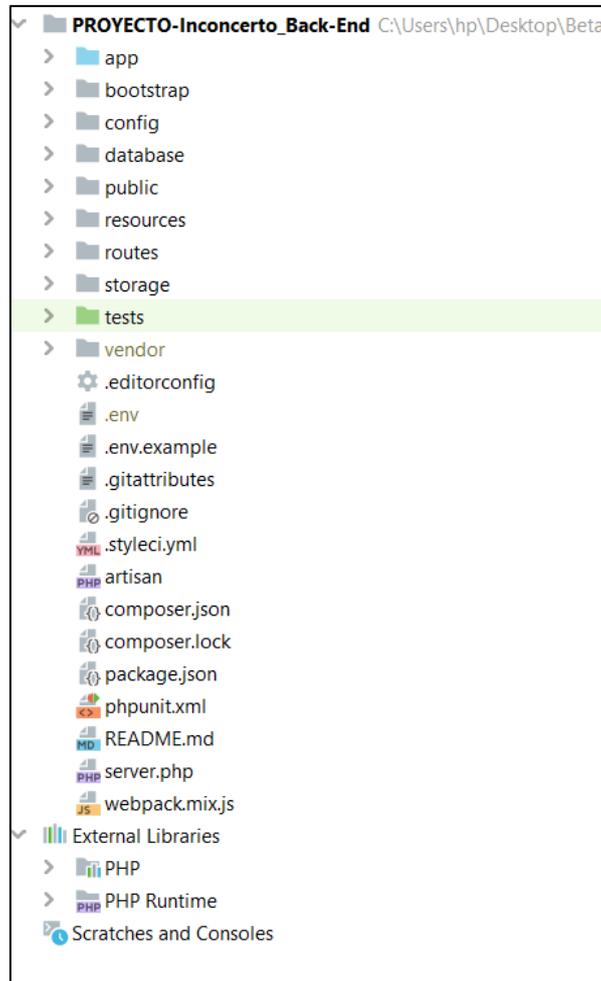


Fig. 5: Estructura del proyecto

3.1.4 Creación de migraciones, modelos, rutas y controladores

Migraciones

En Laravel, las migraciones permiten crear tablas, modificarlas, eliminarlas y establecer relaciones mediante la consola de comandos [28]. Es por ello que, para obtener las 22 tablas de la Base de Datos se ha utilizado las migraciones de Laravel, las cuales, permiten tener un control más estructurado en los cambios que se realicen en la Base de Datos. A continuación, se presenta el comando que se ha utilizado para la creación de la migración mientras que la **Fig. 6**, ilustra la creación de las migraciones localizadas dentro del directorio *database* subcarpeta *migrations*

```
php artisan make:migration create_users_table
```

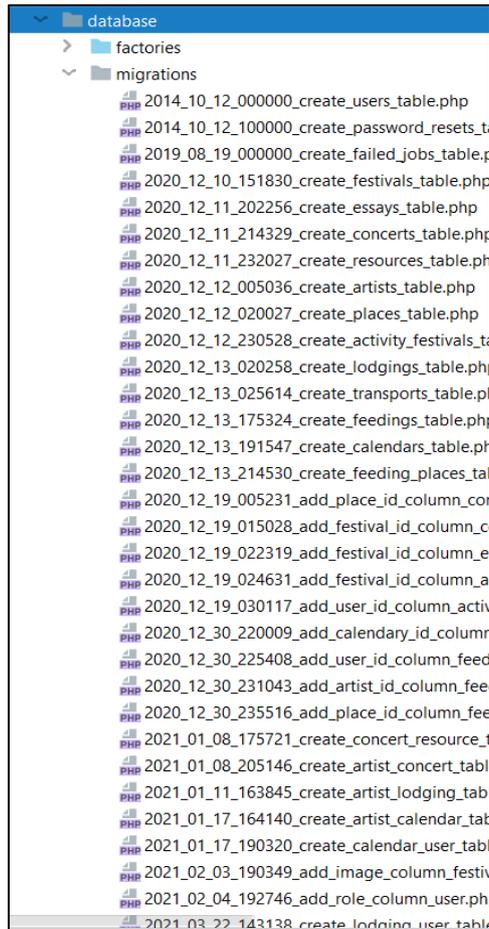


Fig. 6: Migraciones creadas

Posterior a la creación de las migraciones, se establece por cada tabla el nombre de la columna, tipo de dato, clave primaria, claves foráneas, timestamps, como ilustra la **Fig. 7.**

```

public function up()
{
    Schema::create( table: 'users', function (Blueprint $table) {
        $table->id();
        $table->string( column: 'name');
        $table->string( column: 'email')->unique();
        $table->timestamp( column: 'email_verified_at')->nullable();
        $table->string( column: 'password');
        $table->rememberToken();
        $table->timestamps();
    });
}

```

Fig. 7: Columnas de la tabla users

Por último, el siguiente comando sirve para ejecutar las migraciones automáticamente y con ello crear las tablas en la Base de Datos, como se ilustra en la **Fig. 3.**

```
php artisan migrate
```

Modelos

Los modelos son uno de los componentes principales de las aplicaciones desarrolladas bajo el patrón MVC, tienen la responsabilidad de mantener lo que se llama la lógica de negocio, es decir, un conjunto de reglas que deben cumplirse para acceder a los datos, modificarlos, etc.[29]. Para crear un modelo, se debe respetar su convención, el nombre del modelo debe estar en singular y en idioma inglés.

```
php artisan make:model User
```

Controladores

Permiten agrupar la lógica peticiones HTTP relacionadas, aquí residen las acciones solicitadas por el cliente que son referenciadas desde las rutas, de esta forma permiten organizar mejor el código [30]. A continuación, se presenta el comando utilizado para la creación de los controladores en el desarrollo del Sistema Web.

```
php artisan make:controller userController
```

Rutas

Las rutas definen la dirección URL y el método por el cual se puede ingresar a dicha ruta, se encargan de manejar el flujo de solicitudes y respuestas, desde y hacia el cliente (GET, POST, etc.) [31], como se ilustra en la **Fig. 8** un ejemplo de las rutas de la entidad concierto.

```
//CONCIERTO
//GET
Route::get(uri: '/concerts', [ConcertController::class, 'index']);
//GET by ID
Route::get(uri: '/concerts/{concert}', [ConcertController::class, 'show']);
//POST
Route::post(uri: '/concerts', [ConcertController::class, 'store']);
//PUT
Route::put(uri: '/concerts/{concert}', [ConcertController::class, 'update']);
//DELETE
Route::delete(uri: '/concerts/{concert}', [ConcertController::class, 'delete']);
```

Fig. 8: Rutas de entidad concierto

3.1.5 Usuarios

En la **Fig. 9** ilustra los usuarios que pueden interactuar con el sistema web, cada uno de ellos tiene un rol y permisos asignados a los diferentes módulos una vez que se realice la autenticación en el sistema web.

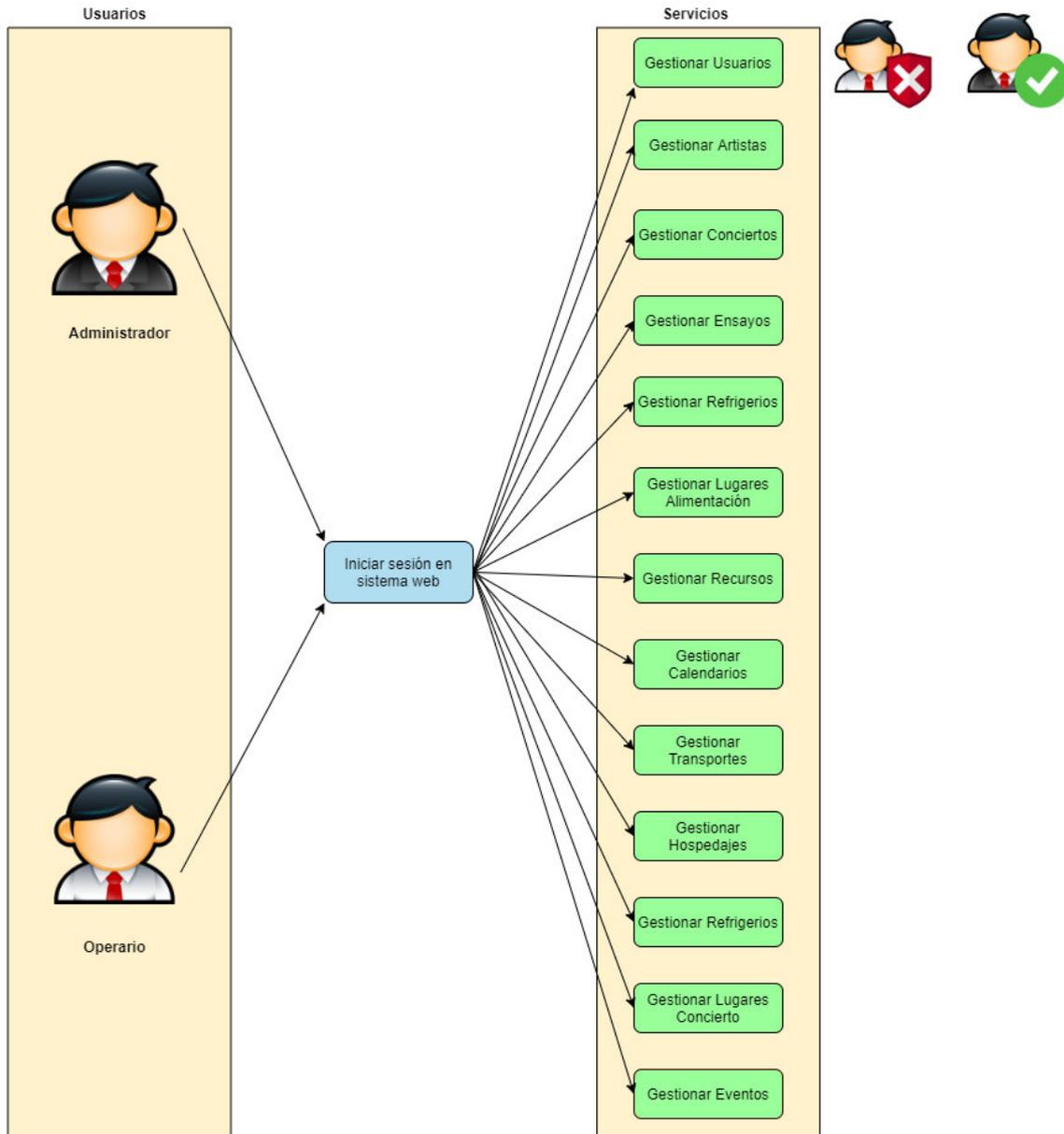


Fig. 9: Perfiles, usuarios y módulos del Sistema Web

3.1.6 Requerimientos específicos del Sistema Web

Autenticación de usuarios

Todos los usuarios tienen la posibilidad de acceder al Sistema Web con sus credenciales que son email y contraseña.

El Sistema Web verifica que el usuario que intenta ingresar esté registrado, lo que implica que, si un usuario no está registrado no podrá gestionar sobre el sistema web.

Gestión de roles

El usuario con perfil administrador es el encargado de gestionar los roles y permisos para cada usuario en el sistema web. También puede gestionar los demás módulos del sistema web.

Gestión de usuarios

El usuario con perfil administrador es quien debe de gestionar los usuarios del sistema web, en este caso si un usuario ingresa por primera vez al sistema web su rol por defecto es operario.

Gestión de eventos, alimentación, tareas, logística y personal

Tanto el usuario con perfil administrador como el usuario con perfil operario son encargados de registrar, modificar y eliminar eventos, alimentación, tareas y logística. Además, el usuario administrador tiene la posibilidad de gestionar usuarios.

Envío de notificaciones de eventos

Tanto el usuario con perfil administrador como el usuario con perfil operario tienen la posibilidad de enviar notificaciones de los nuevos eventos al crear un nuevo festival.

3.2 Sprint 1. Inicio de sesión del usuario administrador, usuarios, eventos, conciertos, ensayos, artistas, calendarios y notificaciones.

Siguiendo el lineamiento del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 1* contempla las siguientes tareas:

- Visualización de la página inicio del Sistema Web.
- Inicio de sesión del usuario administrador.
- Listado, registro, actualización y eliminación de usuarios.
- Listado, registro, actualización y eliminación de eventos.
- Listado, registro, actualización y eliminación de conciertos.
- Listado, registro, actualización y eliminación de ensayos.
- Listado, registro, actualización y eliminación de artistas.
- Listado, registro, actualización y eliminación de calendarios.

- Notificaciones sobre próximos mantenimientos.

3.2.1 Visualización de la página inicio del Sistema Web

La **Fig. 10** ilustra la página de inicio del sistema web la cual contiene el nombre del proyecto, información, un video tutorial y en la parte inferior se encuentra los enlaces a redes sociales. Mientras que, las demás secciones de la página web se detallan en el Manual de Usuario.



Fig. 10: Página de inicio del Sistema Web

3.2.2 Inicio de sesión del usuario administrador

La **Fig. 11** ilustra el formulario y la validación de campos para el ingreso al sistema web. Posterior a ello, el sistema web valida las credenciales del usuario en la Base de Datos, si estas son válidas, lo redirecciona a la página home. Por último, el procedimiento para el inicio de sesión y restablecimiento de contraseñas es el mismo para todos los perfiles.

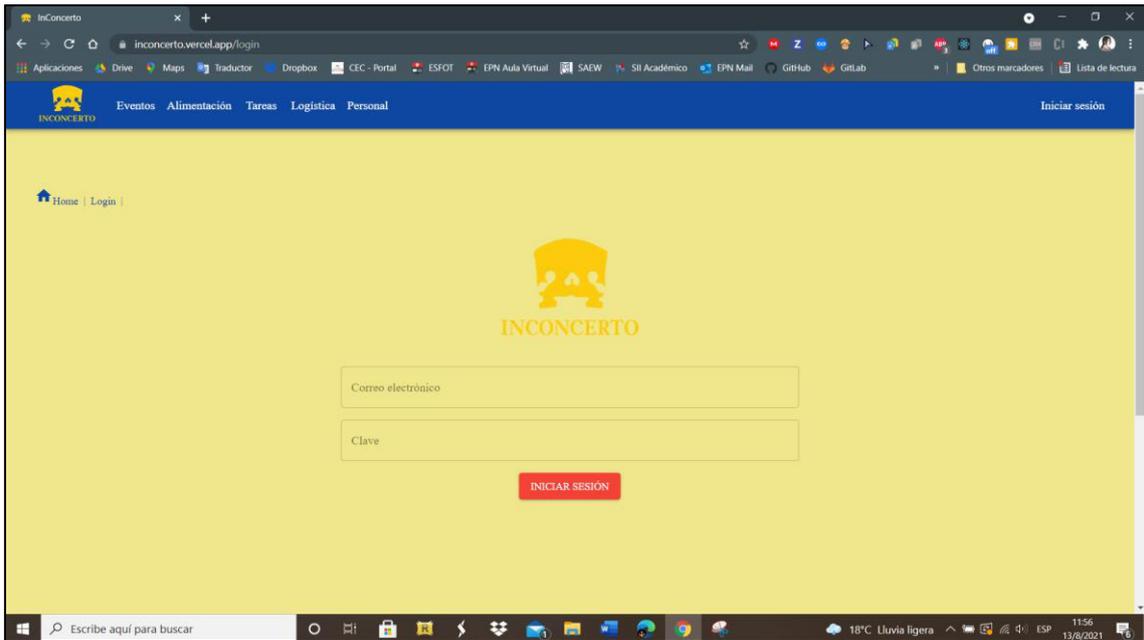


Fig. 11: Inicio de sesión para los usuarios registrados en el Sistema Web

En el caso de que el usuario administrador u operario olvide su contraseña, existe la opción de restablecer contraseña. La **Fig. 12** ilustra el formulario de solicitud de correo a donde se envía un enlace de restablecimiento de contraseña.

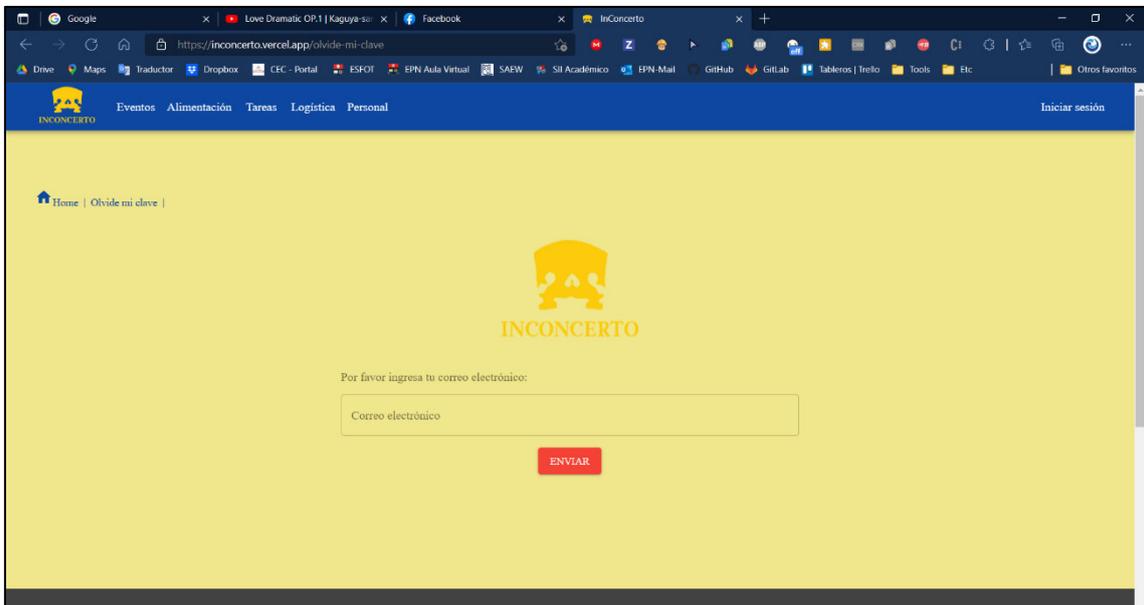


Fig. 12: Formulario de solicitud de correo

Finalmente, la **Fig. 13** ilustra la bandeja de entrada del usuario que consta de un enlace para acceder al formulario de restablecimiento de contraseña; donde cambia sus credenciales de acceso para poder ingresar al sistema web, como lo indica la **Fig. 14**

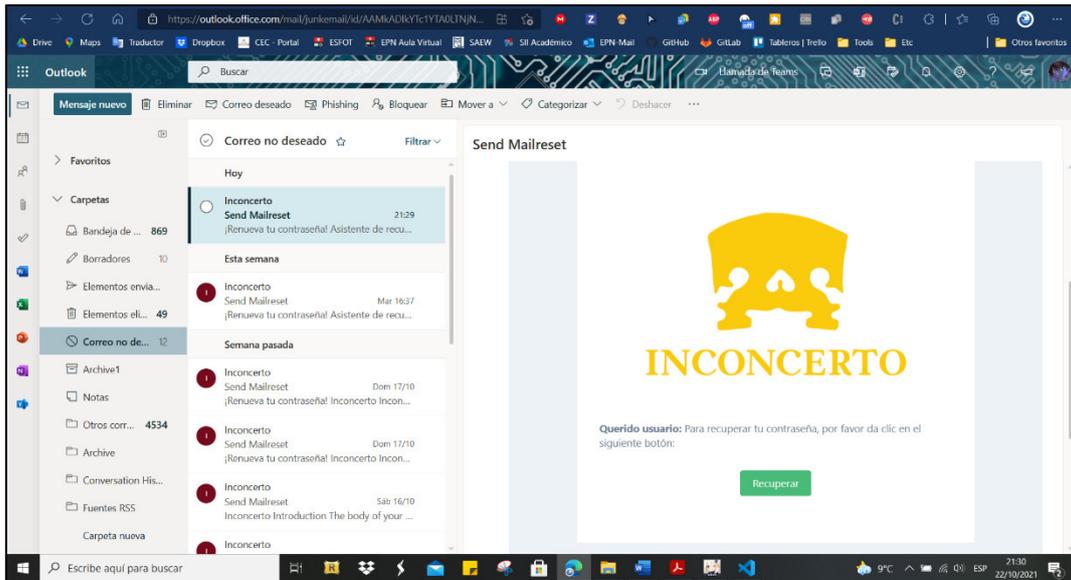


Fig. 13: Bandeja de entrada del usuario

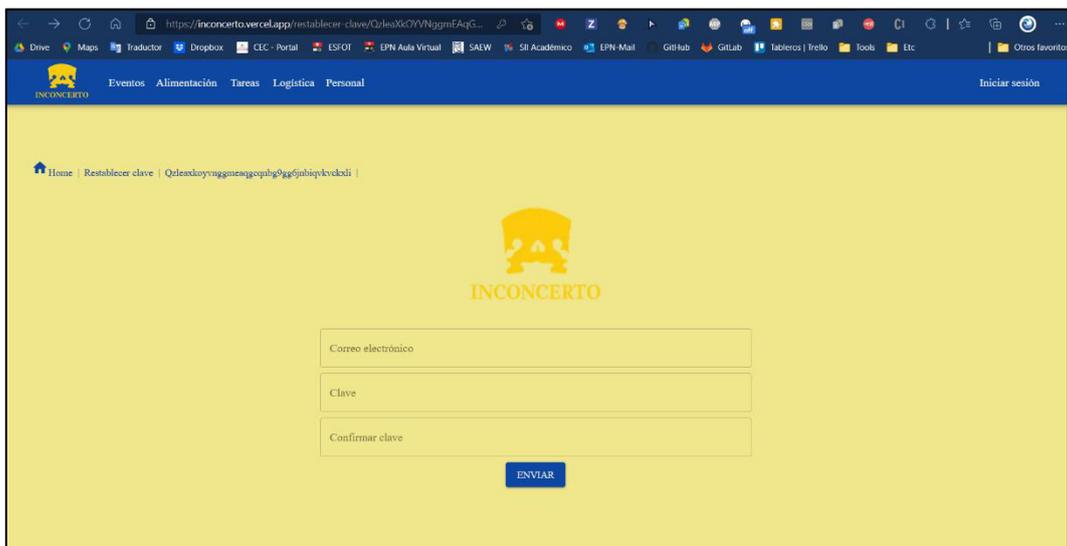


Fig. 14: Formulario de restablecimiento de contraseña

3.2.3 Listado, registro, actualización y eliminación de usuarios

El procedimiento para registrar, modificar y eliminar usuarios del sistema web lo puede realizar el usuario administrador únicamente.

La Fig. 15 ilustra el listado de usuarios registrados en el sistema web, el cual presenta información como: nombre, correo electrónico y opciones que se pueden realizar cuando sobre el usuario.

Por último, los usuarios que son registrados en el sistema web deben de tener un correo organizacional, si un usuario es identificado con un correo diferente es eliminado del sistema web manualmente por el administrador.

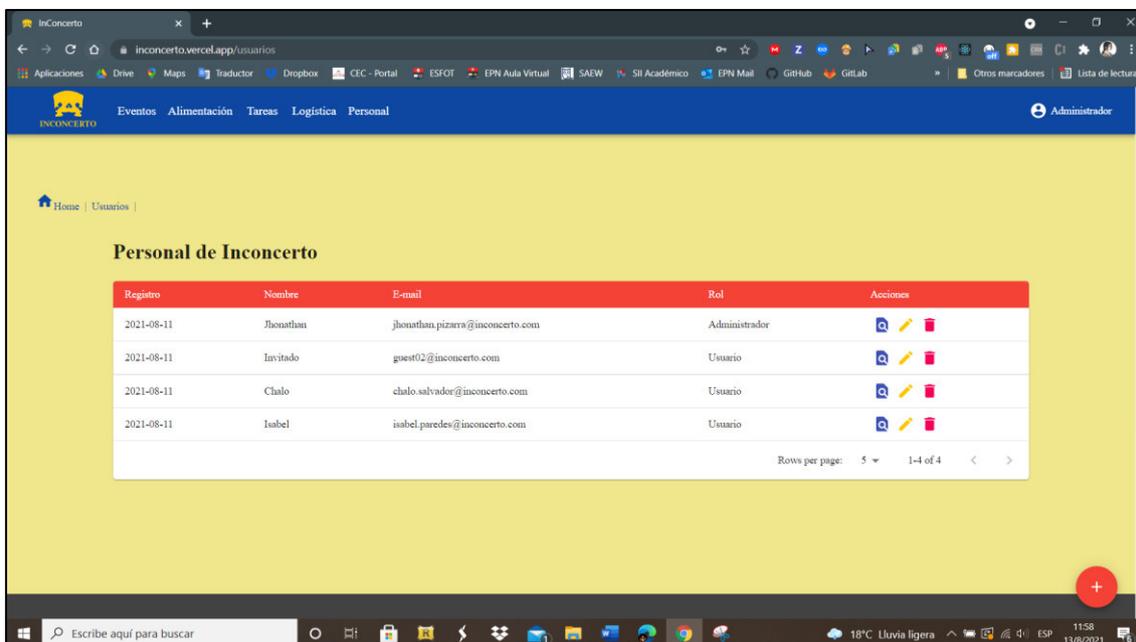


Fig. 15: Listado de usuarios

3.2.4 Listado, registro, actualización y eliminación de eventos

El procedimiento para registrar, modificar y eliminar eventos (llámense también festivales) lo pueden realizar tanto el usuario administrador, como el usuario operario.

La Fig. 16 ilustra el listado de eventos registrados en el sistema web, el cual presenta información como: nombre, descripción, imagen y las opciones que se pueden realizar sobre el evento. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

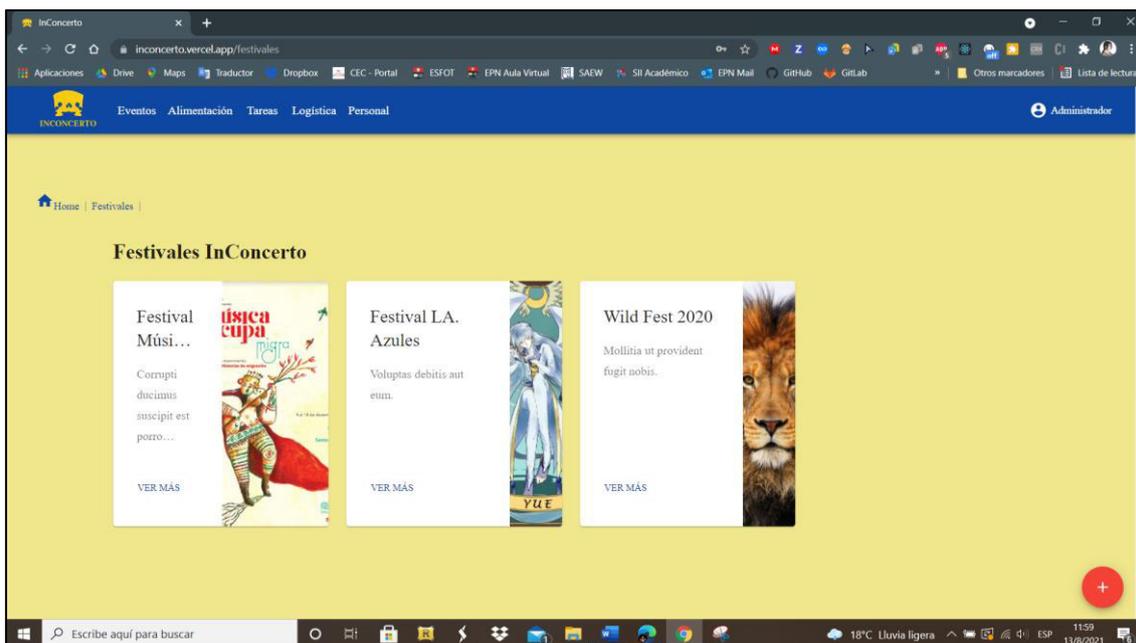


Fig. 16: Listado de eventos

3.2.5 Listado, registro, actualización y eliminación de conciertos

El procedimiento para registrar, modificar y eliminar conciertos lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 17** ilustra el listado de los conciertos registradas en el sistema web, el cual presenta información como: fecha de concierto, hora, nombre, gratuidad, modalidad y las opciones que se pueden realizar sobre cada concierto. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

Concierto	Fecha	Hora	Gratuidad (Si - No)	Inítilo (Si - No)	Lugar	Festival	Acciones
Beatles Concert	1980-12-03	06:11:00	Si	Si	La Caponata	Festival Música Ocupa	[Iconos de acciones]
Live Aid	1994-08-22	11:07:00	Si	Si	La Caponata	Festival Música Ocupa	[Iconos de acciones]
Janis Joplin Concert	2014-03-23	04:28:00	Si	Si	Orfmatto	Wild Fest 2020	[Iconos de acciones]

Rows per page: 5 | 1-3 of 3

Fig. 17: Listado de conciertos

3.2.6 Listado, registro, actualización y eliminación de ensayos

El procedimiento para registrar, modificar y eliminar ensayos lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 18** ilustra el listado de los ensayos registrados en el sistema web, el cual presenta información como: fecha del ensayo, nombre, lugar, festival para el que se ensaya y las opciones que se pueden realizar sobre el ensayo. Mientras que en el Manual de Usuario se detalla el procedimiento, interfaces y validaciones pertinentes para este módulo.

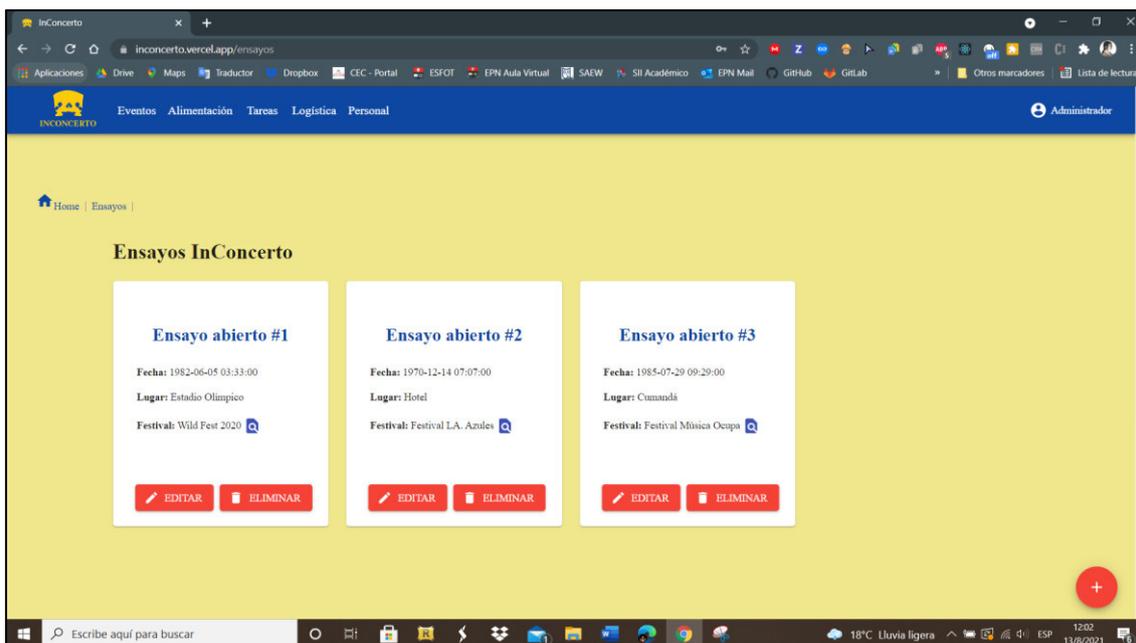


Fig. 18: Listado de ensayos

3.2.7 Listado, registro, actualización y eliminación de artistas

El procedimiento para registrar, modificar y eliminar artistas lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 19** ilustra el listado de artistas, el cual presenta información como: cédula, nombre artístico, nombre del artista, apellido, nacionalidad, mail, teléfono, pasaje, instrumentos que toca, teléfono de emergencia, mail de emergencia, grupo alimenticio, observación y las acciones que se pueden realizar sobre el artista. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

Identificación	Nombre artístico	Nombre	Apellido	Nacionalidad	Email	Teléfono	Pasaje	Instrumentos	Teléfono emergencia	Email emerg
93938	The Beatles	Paul	MacCartney	British	paulmacartney@hotmail.com	0875488854	Si	Viviane Purdy	0999873989	ethelyn25@l
8032	The Beatles	Ringo	Star	EEUU	ringo.star@hotmail.com	0985784485	Si	Lelia Sawaya	3885248255	shayna74@g
10619	The Beatles	George	Harrison	Mexico	george.harrison@hotmail.com	0968552585	Si	Chasity Bohan DDS	0987578884	joynay66@h
1725885548	The Beatles	John	Lennon	EEUU	jhon.lennon@hotmail.com	0986521145	Si	Voz, Guitarra, Piano	0856255548	yokoono@h
1852225845	Janis Joplin	Janis	Joplin	EEUU	janis.joplin@gmail.com	0987554155	Si	Voz, Guitarra	0985852654	jimi.hendrix

Fig. 19: Listado de artistas

3.2.8 Listado, registro, actualización y eliminación de calendarios

El procedimiento para registrar, modificar y eliminar calendarios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 20** ilustra el listado calendarios, el cual presenta información como: llegada del artista, salida del artista, país del que viene, número de vuelo, y las acciones que se pueden realizar sobre el calendario. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

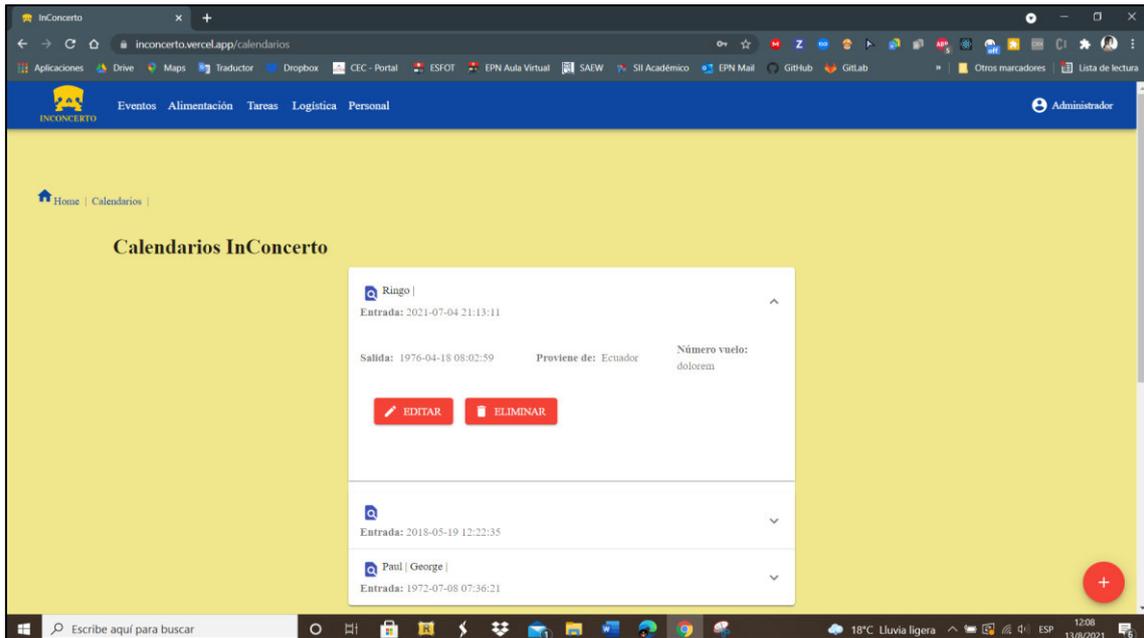


Fig. 20: Listado de calendarios

3.2.9 Notificaciones sobre próximos eventos

El procedimiento para enviar notificaciones lo puede realizar tanto el usuario administrador como el usuario operario.

La Fig. 21 ilustra una notificación recibida automáticamente luego de crear un nuevo festival, donde constan: nombre del nuevo evento, descripción e imagen. Mientras que en el Manual de Usuario se puntualiza el procedimiento, interfaces y validaciones pertinentes para este módulo.

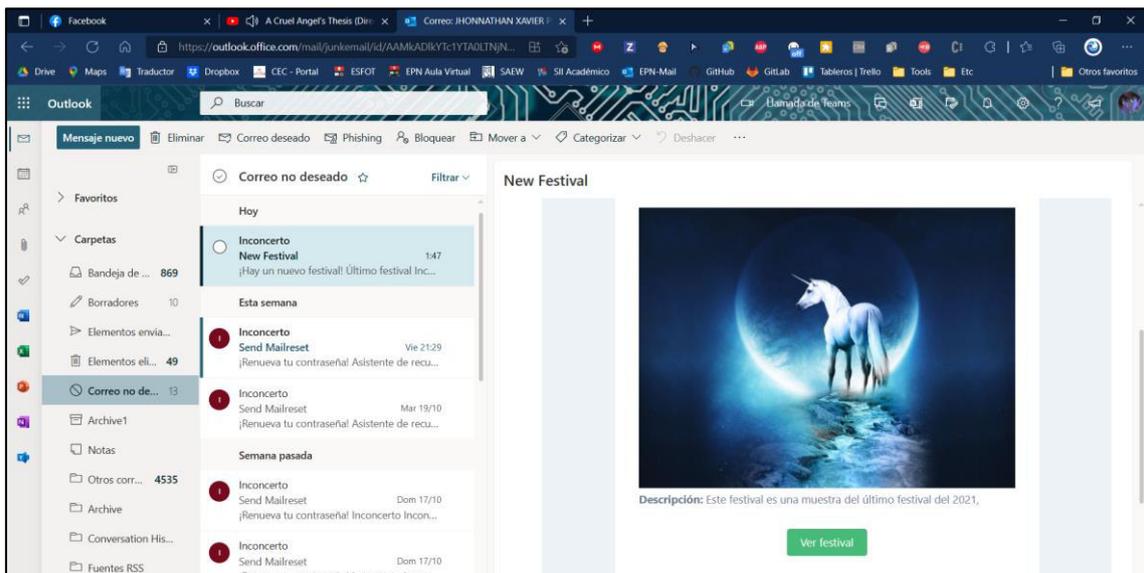


Fig. 21: Notificación del próximo evento

3.3 Sprint 2. Transportes, hospedajes, lugares de concierto, tareas

Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 2* se contemplan las siguientes tareas:

- Listado, registro, actualización y eliminación de transportes.
- Listado, registro, actualización y eliminación de hospedajes.
- Listado, registro, actualización y eliminación de lugares de concierto.
- Listado, registro, actualización y eliminación de tareas.

3.3.1 Listado, registro, actualización y eliminación de transportes

El procedimiento para registrar, modificar y eliminar calendarios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 22** ilustra el listado de transportes, el cual presenta información como: tipo, capacidad, disponibilidad, matrícula, y las acciones que se pueden realizar sobre el transporte. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

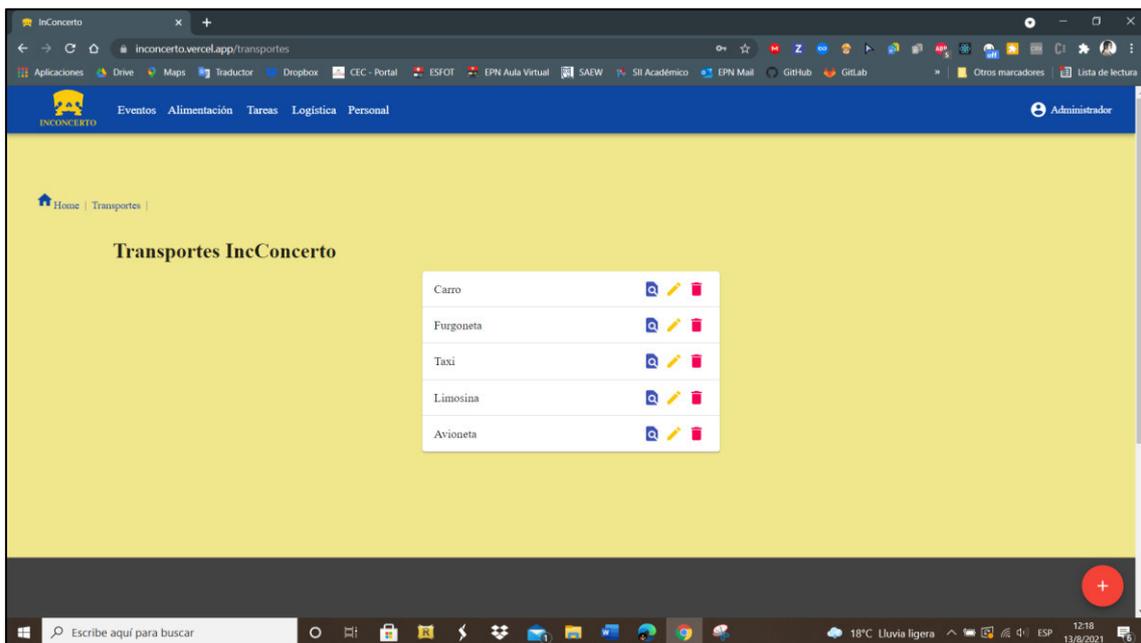


Fig. 22: Listado de transportes

3.3.2 Listado, registro, actualización y eliminación de hospedajes

El procedimiento para registrar, modificar y eliminar hospedajes lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 23** ilustra el listado de hospedajes, el cual presenta información como: tipo, nombre, descripción, observación, entradas, salidas, y las acciones que se pueden realizar sobre el hospedaje. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

Nombre	Características	Descripción	Observación	Fecha de entrada	Fecha de salida	Acciones
Quijus	quidem	saepe	Architecto libero accusamus sed recusandae non voluptas.	2006-05-14 00:00:00	1972-07-25 00:00:00	[Iconos de acciones]
Casa Blanca	temporibus	eum	Animi eveniet dolor cum velit porro ut ut.	1986-08-03 00:00:00	2013-02-11 00:00:00	[Iconos de acciones]
Swissotel	quia	occaecati	Et ex consequatur quas.	2005-12-05 00:00:00	1972-06-11 00:00:00	[Iconos de acciones]

Fig. 23: Listado de hospedajes

3.3.3 Listado, registro, actualización y eliminación de lugares de concierto

El procedimiento para registrar, modificar y eliminar lugares de concierto lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 24** ilustra el listado de lugares de concierto, el cual presenta información como: nombre, dirección, disponibilidad, aforo, y las acciones que se pueden realizar sobre el lugar. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

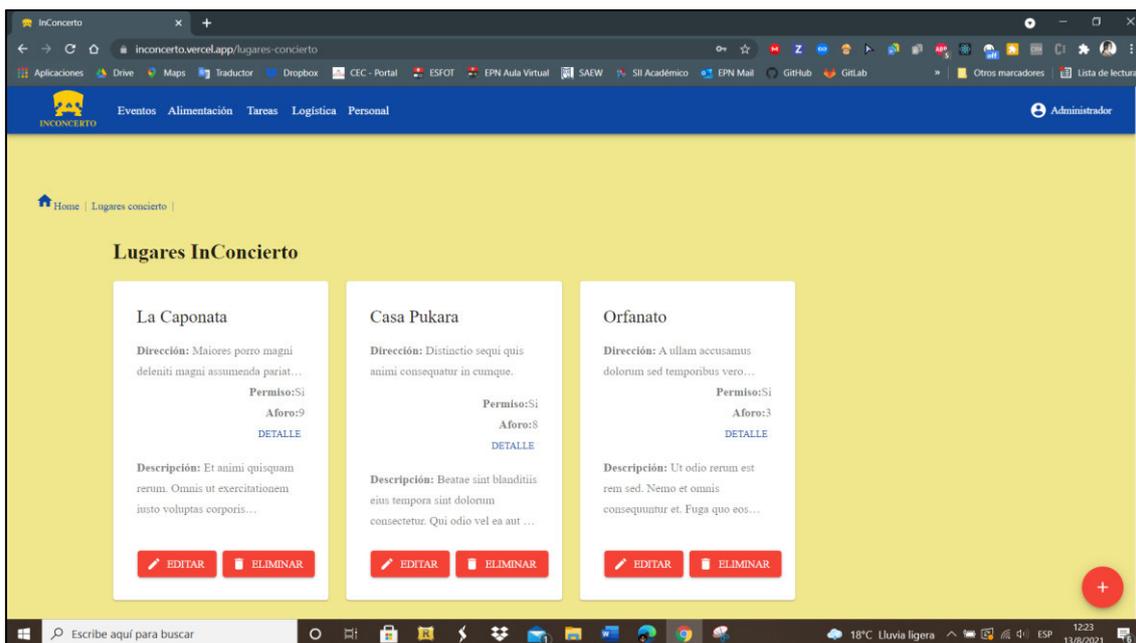


Fig. 24: Listado de lugares de concierto

3.3.4 Listado, registro, actualización y eliminación de tareas

El procedimiento para registrar, modificar y eliminar tareas lo pueden realizar tanto el usuario administrador, como el usuario operario.

La Fig. 25 ilustra el listado de tareas, el cual presenta información como: nombre, fecha, descripción, observación, festival al que pertenece, encargado, y las acciones que se pueden realizar sobre la tarea. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

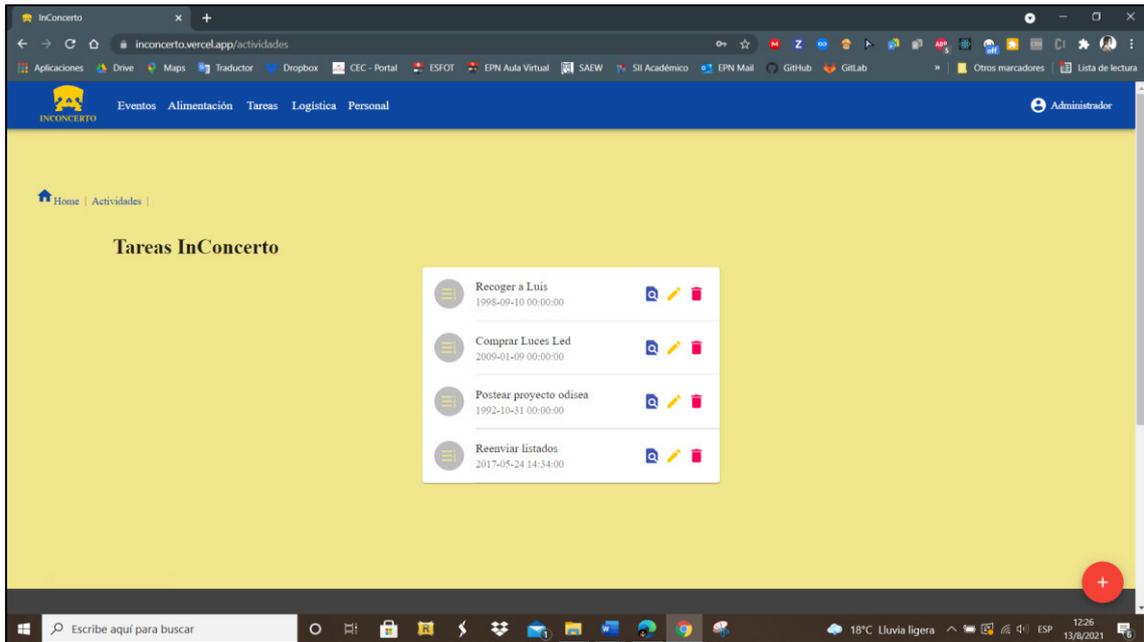


Fig. 25: Listado de tareas

3.4 Sprint 3. Recursos

Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 3* se contemplan las siguientes tareas:

- Listado, registro, actualización y eliminación de recursos

3.4.1 Listado, registro, actualización y eliminación de recursos

El procedimiento para registrar, modificar y eliminar recursos lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 26** ilustra el listado de recursos, el cual presenta información como: nombre, cantidad, descripción, y las acciones que se pueden realizar sobre el recurso. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

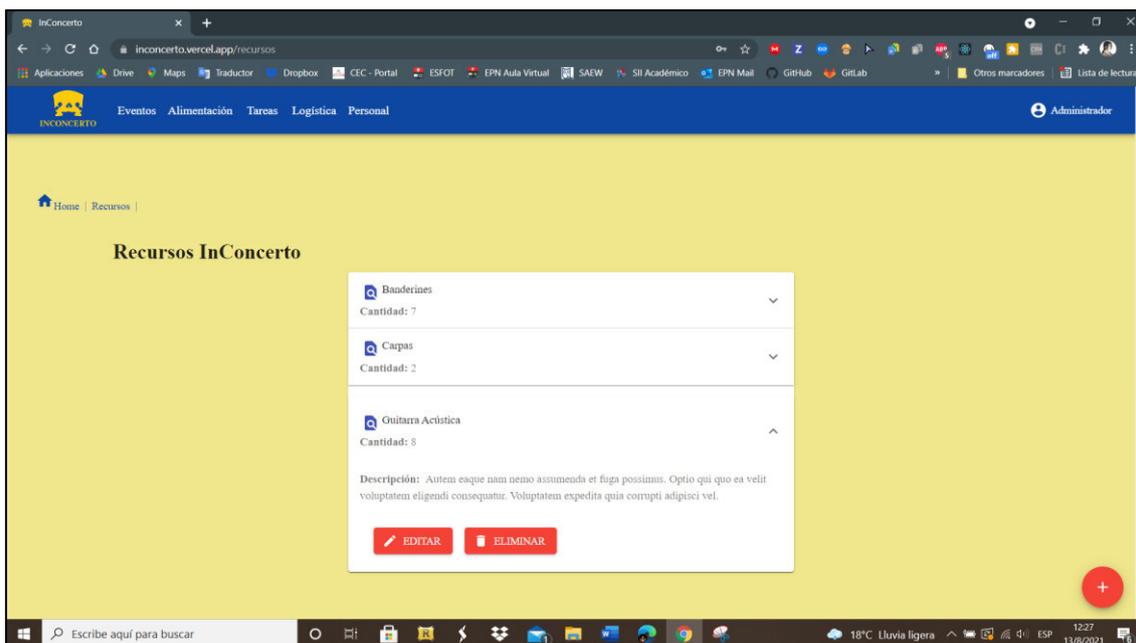


Fig. 26: Lista de recursos

3.5 Sprint 4. Cuadros alimenticios, lugares de alimentación

Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 4* se contemplan las siguientes tareas:

- Listado, registro, actualización y eliminación de cuadros alimenticios.
- Listado, registro, actualización y eliminación de lugares de alimentación.

3.5.1 Listado, registro, actualización y eliminación de cuadros alimenticios

El procedimiento para registrar, modificar y eliminar cuadros alimenticios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 27** ilustra el listado de tareas, el cual presenta información como: nombre, fecha, observación, cantidad, artista al que pertenece, usuario responsable, lugar y las acciones que se pueden realizar sobre el cuadro alimenticio. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

Fecha	Nombre	Cantidad	Observación	Lugar	Artista	Responsable	Acciones
2006-06-26 11:18:00	Onigiris	8	Non quos mínima voluptatem necessitatibus commodi. Voluptatem blanditiis praesentium nam molestiae. Alias nihil exercitationem et atque quis.	Pizza Hut	Paul	Chalo	
2002-01-07 23:12:00	Spaguetti	1	Mínima rerum culpa autem voluptas error. Officiis perferendis et et modi voluptates vel quia. Quas temporibus rerum labore consequatur aut.	Papa Jhon's	Paul	Chalo	
1982-03-31 09:13:00	Pizza	2	A omnis dolores quam unde sit et. Eos ab nihil quo. Sit est praesentium voluptate et debitis qui molestias error. Magni asperiores est aut ut soluta.	Papa Jhon's	George	Chalo	

Rows per page: 5 1-3 of 3

Fig. 27: Lista de cuadros alimenticios

3.5.2 Listado, registro, actualización y eliminación de lugares alimenticios

El procedimiento para registrar, modificar y eliminar lugares alimenticios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 28** ilustra el listado de tareas, el cual presenta información como: nombre, dirección, aforo y las acciones que se pueden realizar sobre los lugares alimenticios. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

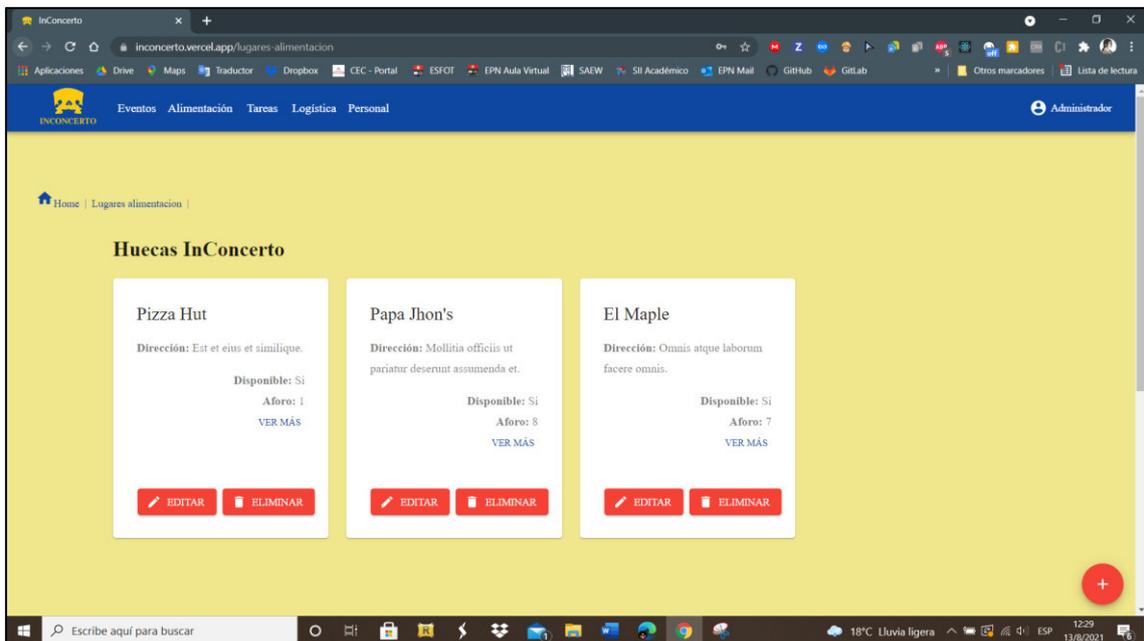


Fig. 28: Lista de lugares alimenticios

3.6 Sprint 5. Iniciar sesión del usuario operario

Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 5* se contemplan las siguientes tareas:

- Inicio de sesión del usuario operario.

3.6.1 Inicio de sesión del usuario operario

La **Fig. 11** ilustra el formulario y la validación de campos para el ingreso al sistema web. Posterior a ello, el sistema valida las credenciales del usuario en la Base de Datos, si estas son válidas, lo redirecciona a la página home. Por último, el procedimiento para el inicio de sesión y restablecimiento de contraseñas es el mismo para todos los perfiles.

3.7 Sprint 6. Pruebas y despliegue del Sistema Web

Siguiendo los lineamientos del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 6* se contemplan las siguientes tareas:

- Pruebas de carga.
- Pruebas de compatibilidad.
- Pruebas de funcionalidad.
- Despliegue del sistema web.

3.7.1 Pruebas de carga

Las pruebas de rendimiento sirven para determinar la rapidez de un sistema mientras realiza una tarea en determinadas condiciones de trabajo [32]. El tipo de pruebas de rendimiento más sencillo, son las pruebas de carga, estas permiten observar el comportamiento de una aplicación bajo una cantidad esperada de peticiones. Esta carga puede interpretarse como el número estimado de usuarios concurrentes al sistema web, que realizan una cantidad determinada de transacciones, durante el tiempo que dura la carga. Esta prueba recopila y presenta los tiempos de respuesta de todas las transacciones realizadas a la aplicación [32].

Existen numerosas herramientas, tanto open source como privativas, para realizar las pruebas de rendimiento: NeoLoad, LoadRunner, LoadUI, WebLOAD, etc [32]. Una de las más populares es JMeter (open source), que permite conocer el comportamiento funcional del sistema web frente a una gran cantidad de peticiones y de esta manera determinar la estabilidad del mismo. La **Fig. 29** ilustra la configuración y la creación de un grupo de hilos de prueba llamado: “Pruebas_Inconcerto”.

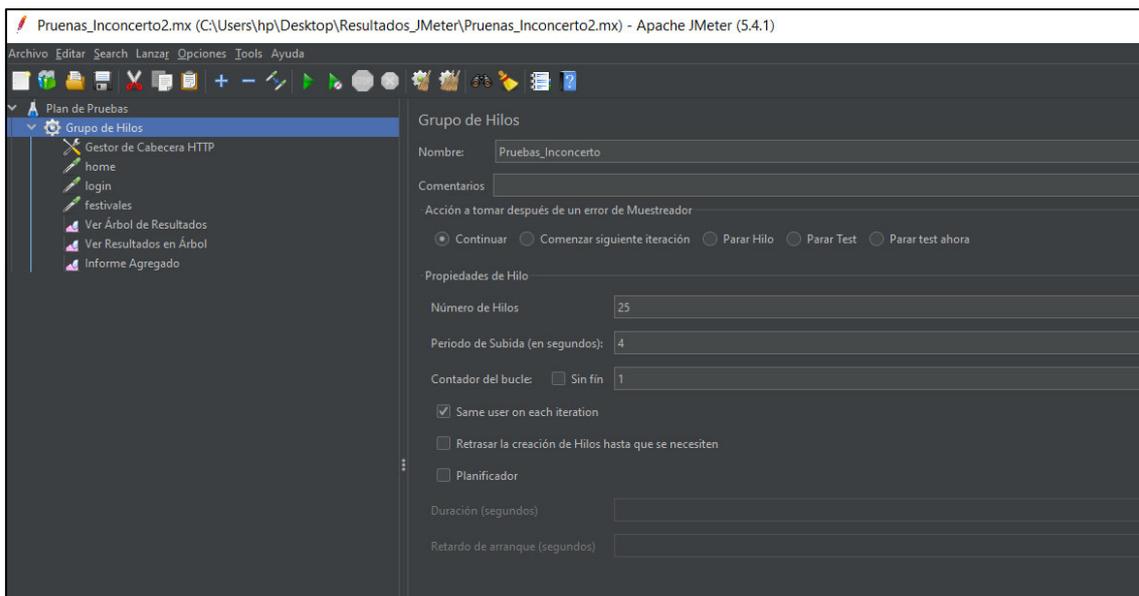


Fig. 29: Configuración del grupo de hilos en JMeter

Donde:

- **Número de Hilos:** es la cantidad de usuarios que van a realizar la petición.
- **Periodo de subida:** es la representación en segundos del intervalo de tiempo que le toma en llegar al máximo hilos.

- **Contador del Bucle:** indica la cantidad de peticiones que se envían por cada hilo.

Posterior a ello se procede a realizar la primera prueba con una carga de 100 peticiones. Mientras que los resultados de las dos pruebas restantes se detallan en el apartado Pruebas de funcionalidad del Manual Técnico (pág. 53).

Primera prueba (100 Peticiones)

A continuación, la **Fig. 30** ilustra el resultado que se ha obtenido de la primera prueba, en donde:

- El porcentaje de error es 0% lo que significa que todas las peticiones se ejecutan exitosamente con un tiempo promedio de 303 milisegundos (0.303 segundos) aproximadamente.
- La ruta Home es la que más tiempo le tomado al servidor en responder las solicitudes con un tiempo de 1048 milisegundos (1.048 segundos) aproximadamente.
- La ruta página de inicio de sesión del sistema, es la más rápida en responder las solicitudes con un tiempo de 127 milisegundos (0.127 segundos) aproximadamente.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error
home	33	628	604	656	697	1048	546	1048	0,00%
login	33	141	143	156	159	162	127	162	0,00%
festivales	33	140	135	158	159	182	128	182	0,00%
Total	99	303	148	619	645	991	127	1048	0,00%

Fig. 30: Resultados de la prueba #1 con 100 peticiones simultaneas

Con los resultados obtenidos de las pruebas de carga se determina que el tiempo de respuesta a las peticiones son adecuadas, obteniendo tiempos de respuestas aceptables para su uso.

3.7.2 Pruebas de compatibilidad

Las pruebas de compatibilidad son las pruebas que se realizan en un software o aplicación determinada y que comprueban que el desarrollo es compatible con todos los

navegadores de internet y todos los sistemas operativos del mercado [33]. Permitiendo verificar el correcto comportamiento de las interfaces y con ello cumplir con los requerimientos del usuario.

Esta prueba se realiza en tres de los navegadores web más utilizados por los usuarios en la actualidad como se presenta en la **TABLA V**. Mientras que los resultados de las dos pruebas restantes se detallan en el apartado de Pruebas de compatibilidad dentro del Manual Técnico (pág. 54 – 56).

TABLA V: Versiones de navegadores

NOMBRE	VERSIÓN	OBSERVACIÓN
Microsoft Edge	92.0.902.73	Totalmente funcional
Google Chrome	92.0.4515.159	Totalmente funcional
Mozilla Firefox	91.0.1	Totalmente funcional

3.7.3 Pruebas de funcionalidad

Las pruebas funcionales se realizan para comprobar la usabilidad, funcionalidad y otras características críticas para el negocio. Las pruebas funcionales tienen por objetivo garantizar que las funcionalidades del software se comporten según lo esperado sin ningún problema [34]. A continuación, la **TABLA VI** detalla un ejemplo de la prueba de funcionalidad que se ha realizado. Mientras los resultados de las 55 pruebas restantes se detallan en el apartado Pruebas de funcionalidad en el Manual Técnico (pág. 56 – 91).

TABLA VI: Prueba de funcionalidad - Inicio de sesión del administrador

PRUEBA DE FUNCIONALIDAD	
Identificador (ID): PF001	Identificador historia de usuario: HU001
Nombre prueba de funcionalidad: Inicio de sesión del Administrador/Operario	
Descripción: El usuario administrador debe ingresar al sistema web.	
Pasos de ejecución: Ingresar el correo y contraseña (proporcionado por el autor del mismo). Presionar el botón "Ingresar".	

<p>Resultado deseado:</p> <p>El usuario administrador, inicia sesión con su respectivo perfil.</p>
<p>Evaluación de la prueba:</p> <p>Resultado exitoso.</p> <p>El Sistema Web permite el ingreso del usuario administrador.</p> <p>Aprobación del cliente 100%.</p>

3.7.4 Despliegue del Sistema Web

En la **Fig. 31** se ilustra el despliegue del sistema web. El proceso completo de configuración y despliegue se detallan en el Manual de Instalación (pág. 2).



Fig. 31: Despliegue del Sistema Web

Finalmente, la URL para acceder al sistema web es:

<https://inconcerto.vercel.app>

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Al momento de desarrollar un proyecto siempre es importante tener una idea clara de lo que se pretende realizar, marcando objetivos claros, herramientas necesarias, metodologías y técnicas que promuevan las buenas prácticas y organización. De esa manera se concluye que una adecuada recopilación de los requerimientos satisfacer las expectativas y necesidades tanto de los clientes como de los desarrolladores.
- La metodología ágil *Scrum*, brinda reglas, artefactos y funciones fáciles de entender, permitiendo eliminar las ambigüedades durante el desarrollo. Además, la demarcación de funciones y tiempos aseguran que haya transparencia y colaboración durante todo el ciclo de desarrollo.
- La arquitectura MVC implementada en el sistema web hace de este un sistema web escalable y fácil de mantener, permitiendo implementar nuevos módulos sin afectar a los otros componentes de nuestra aplicación ni hacer configuraciones adicionales.
- La utilización del Sistema Gestor de Base de Datos (SGBD) MySQL, permite almacenar los datos del colectivo y al ser un SGBD relacional ayuda al rendimiento, escalabilidad e integración con las diferentes herramientas utilizadas para el desarrollo del proyecto.
- La utilización del *Framework* Laravel ayuda a mantener una estética de código ordenado y al ser un *Framework* de tipo MVC permite que el desarrollo del Sistema Web funcione como una herramienta tecnológica, para cumplir los objetivos propuestos.
- Para comprobar el correcto funcionamiento del sistema web se han realizado pruebas de carga, compatibilidad y funcionalidad, las cuales permiten identificar problemas, medir su eficiencia y comprobar su eficacia.

4.2 Recomendaciones

- Una vez puesto en producción el sistema web se recomienda que los administradores establezcan las frecuencias de mantenimiento con el objetivo de tener un sistema web actualizado, altamente disponible y siempre dispuesto a atender los requerimientos del cliente.
- En el caso de que se requieran mejoras en el sistema web se recomienda realizar un previo análisis de compatibilidad de las librerías utilizadas y de ser necesario, actualizarlas ya que puede ocasionar posibles bugs en el sistema web.
- Se recomienda realizar *backups* de la Base de Datos cuando se requiera hacer algún cambio y que los mismos se envíen al líder del proyecto para su almacenamiento y custodia.
- Se recomienda que el administrador y encargados del laboratorio establezcan políticas de seguridad para el uso del sistema web.
- Se recomienda trazar un plan de contingencia en caso de imprevistos que no permitan realizar algún objetivo trazado durante el desarrollo del proyecto.

5 Referencias Bibliográficas

- [1] S. Gangotena y R. Becerra, «MusicaOcupa,» Carla Moncayo, 06 2017. [En línea]. Available: <https://www.musicaocupa.com/quienessomos>. [Último acceso: 15 01 2020].
- [2] M. Soler, «gradiweb.com,» 2020. [En línea]. Available: <https://www.gradiweb.com/que-es-un-sitio-web-y-su-importancia-en-mi-negocio/>. [Último acceso: 12 01 2020].
- [3] B. P. D. S.A, «impulsopopular.com,» 12 08 2015. [En línea]. Available: <https://www.impulsopopular.com/marketing/la-importancia-de-contar-con-una-pagina-web-de-tu-empresa/>. [Último acceso: 12 01 2020].
- [4] I. S.L.U., «piensoenweb.com,» INFORMATIZAME SL, 2017. [En línea]. Available: <https://piensoenweb.com/entradas-blog/la-importancia-de-una-pagina-web.html>. [Último acceso: 18 01 2020].
- [5] C. Z. Mariño, «Los tres roles de la metodología SCRUM,» [En línea]. Available: <https://xn--zoraidaceballosdemario-4ec.info/scrum/scrum-roles-y-responsabilidades-del-scrum-team/>. [Último acceso: 16 Junio 2021].
- [6] P. C. Alejandro Pérez, «METODOLOGÍA SCRUM,» Sinnaps, 2019. [En línea]. Available: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>. [Último acceso: 07 02 2020].
- [7] Schwaber, k., & Sutherland, J., «The Scrum guide,» noviembre 2017. [En línea]. Available: Ken Schwaber and Jeff Sutherland, 2017.
- [8] V. S.A, «Artefactos Scrum ¿Qué son y para qué sirven?,» Grupo IBM; IBM España S.A., 2021. [En línea]. Available: <https://www.viewnext.com/artefactos-scrum/>. [Último acceso: 17 06 2021].
- [9] proyectosagiles.org, «Lista de objetivos / requisitos priorizada (Product Backlog),» [En línea]. Available: Lista de objetivos / requisitos priorizada (Product Backlog). [Último acceso: 14 octubre 2020].
- [10] M. REHKOPF, «Historias de usuario con ejemplos y plantilla,» Atlassian, 2021. [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 17 06 2021].
- [11] K. Kaizenia, «¿Qué son los artefactos Scrum?,» Grupo Oyarde S.A, 2021. [En línea]. Available: <https://kzi.mx/que-son-los-artefactos-de-scrum/>.
- [12] M. Cohn, «Sprint Backlog,» Mountain Goat Software, 2021. [En línea]. Available: <https://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/sprint-backlog>. [Último acceso: 17 06 2021].
- [13] P. SANTAMARÍA, «Balsamiq mockup, una muy buen herramienta para esbozar tus futuras apps,» Applesfera, 25 11 2013. [En línea]. Available: <https://www.applesfera.com/aplicaciones-os-x-1/balsamiq-mockup-una-muy-buen-herramienta-para-esbozar-tus-futuras-apps>. [Último acceso: 17 06 2021].

- [14] U. D. ALICANTE, «Servicio de Informática ASP.NET MVC 3 Framework,» Universidad de Alicante, [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 18 06 2021].
- [15] J. M. Aguilar, «¿Qué es el patrón MVC en programación y por qué es útil?,» Campus MVP, 15 10 2019. [En línea]. Available: <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>. [Último acceso: 18 06 2021].
- [16] J. G. Gimeno, «Xampp. Herramienta indispensable para desarrollo web,» Código Binario, 12 06 2017. [En línea]. Available: <https://www.codigo-binario.es/xampp-herramienta-para-dev-web/>. [Último acceso: 18 06 2021].
- [17] M. Alvarez, «Composer, gestor de dependencias para PHP,» desarrolloweb.com, 24 01 2020. [En línea]. Available: <https://desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html>. [Último acceso: 18 06 2021].
- [18] J. L. G. Folgado, «Autenticación con JWT en Laravel,» NIGMACODE, 2020. [En línea]. Available: <https://www.nigmacode.com/laravel/jwt-en-laravel/>. [Último acceso: 18 06 2021].
- [19] C. S.L., «Laravel,» desarrolloweb.com, [En línea]. Available: <https://desarrolloweb.com/home/laravel>. [Último acceso: 18 06 2021].
- [20] S. Silbermann, «Material-UI,» DoIT International, 2021. [En línea]. Available: <https://material-ui.com/es/>. [Último acceso: 18 06 2021].
- [21] M. contributors, «¿Qué es JavaScript?,» MDN Web Docs, [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript. [Último acceso: 18 06 2021].
- [22] OpenWebinars, «Qué es HTML5,» 20 enero 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-html5/>.
- [23] A. Robledano, «Qué es CSS y para qué sirve,» OpenWebinars, 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-css/>. [Último acceso: 18 06 2021].
- [24] NeoAttack, «MySQL,» 2021. [En línea]. Available: <https://neoattack.com/neowiki/mysql/>. [Último acceso: 18 06 2021].
- [25] E. Aristizábal, «Formularios de usuarios dinámicos con React Hooks,» Kushki, 16 11 2020. [En línea]. Available: <https://www.kushki.com/en/blog/como-creamos-nuestros-formularios-de-usuario-en-kushki>. [Último acceso: 18 06 2021].
- [26] A. Tinwala, «Formularios en React: React Hook Forms con Material UI y YUP,» DEV Community, 06 09 2020. [En línea]. Available: <https://dev.to/ammartinwala52/forms-in-react-react-hook-forms-with-material-ui-and-yup-4gfb#sub2>. [Último acceso: 18 06 2021].
- [27] ICHI.PRO, «Guía completa para usar iconos de diseño de materiales en React,» ICHI.PRO, 2020. [En línea]. Available: <https://ichi.pro/es/guia-completa-para-usar-iconos-de-diseno-de-materiales-en-react-233576377954331>. [Último acceso: 23 06 2021].
- [28] M. Canales Zapata, «INTRODUCCIÓN A LAS MIGRACIONES CON LARAVEL FRAMEWORK,» ANEXSOFT, 2021. [En línea]. Available: <https://anexsoft.com/introduccion-a-las-migraciones-con-laravel-framework>. [Último acceso: 22 06 2021].

- [29] C. R. Ruso, «Introducción a modelos en Laravel,» desarrolloweb.com, 14 09 2015. [En línea]. Available: <https://desarrolloweb.com/articulos/introduccion-modelos-laravel.html>. [Último acceso: 22 06 2021].
- [30] D. Palacios, «Controladores en Laravel,» Styde, 27 10 2017. [En línea]. Available: <https://styde.net/controladores-en-laravel/>. [Último acceso: 22 06 2021].
- [31] LaravelTIP.com, «¿Qué son los Routes en Laravel?,» 01 02 2019. [En línea]. Available: <https://www.laraveltip.com/que-son-los-routes-en-laravel/>. [Último acceso: 22 06 2021].
- [32] J. F. Sánchez, «Pruebas de rendimiento con JMeter.,» 02 07 2018. [En línea]. Available: <https://www.sdos.es/blog/pruebas-de-rendimiento-con-jmeter-ejemplos-basicos>. [Último acceso: 23 06 2021].
- [33] F. Acero, «PRUEBAS DE COMPATIBILIDAD Y USABILIDAD,» Prezi, 01 11 2013. [En línea]. Available: <https://prezi.com/e164alfch6wx/pruebas-de-compatibilidad-y-usabilidad/>. [Último acceso: 23 06 2021].
- [34] G. Lee, «Tipos de pruebas de software: diferencias y ejemplos,» loadview, 16 10 2020. [En línea]. Available: <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>. [Último acceso: 23 06 2021].
- [35] Darkcritz, «PhpStorm, un excelente IDE para PHP multiplataforma,» LinuxAdictos, [En línea]. Available: <https://www.linuxadictos.com/phpstorm-un-excelente-ide-para-php-multiplataforma.html>. [Último acceso: 22 06 2021].
- [36] Yair, «Qué es Composer y cómo usarlo,» Styde, 23 12 2019. [En línea]. Available: <https://styde.net/que-es-composer-y-como-usarlo/>. [Último acceso: 23 06 2021].

6 ANEXOS

6.1 Manual Técnico

- Índice del Manual Técnico
- Recopilación de Requerimientos
- Historias de Usuario
- *Product Backlog*
- *Sprint Backlog*
- Diseño de Interfaces
- Diseño de la Base de Datos
- Procedimientos almacenados
- Pruebas de carga
- Pruebas de compatibilidad
- Pruebas de funcionamiento

6.2 Manual de Usuario

- <https://www.youtube.com/watch?v=EQHqZdkye-o>

6.3 Manual de Instalación

- Despliegue del sistema web