

# **ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**DESARROLLO DE UN SISTEMA PARA LA ESCUELA DE  
BIODANZA SRT EN QUITO-PUEMBO**

**DESARROLLO DE UN *BACKEND***

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO  
SUPERIOR EN DESARROLLO DE SOFTWARE**

**JORGE DANIEL ORTIZ GALLEGOS**

**DIRECTOR: IVONNE FERNANDA MALDONADO SOLIZ**

**Quito, marzo 2023**

## CERTIFICACIONES

Yo, Jorge Daniel Ortiz Gallegos declaro la originalidad de este presente trabajo de integración curricular redactado ante mi autoría; que no ha sido presentado previamente a ningún grado o entidad profesional; y, que ha sido agregado las referencias bibliográficas que se presenta en este documento.



---

**Jorge Ortiz**

jorge.ortiz@epn.edu.ec

**ortizjorge319@gmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por Jorge Daniel Ortiz Gallegos, bajo mi supervisión.



---

**Ing. Ivonne Maldonado, MSc.**

**DIRECTOR**

**ivonne.maldonado@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.

Jorge Daniel Ortiz Gallegos

## **DEDICATORIA**

Este tema de proyecto de Integración Curricular está dedicado para el elemento más importante de mi vida "Mi Familia". Dedico este proyecto a mi Madre porque no hay mejor bendición que conozco que es haberme dado la vida. A mi Padre por haberme formado en las mejores enseñanzas de tener disciplina y carácter ante las situaciones difíciles.

A mi hermana por darme la valentía de nunca darme por vencido, y si todo fracasara, siempre dar la mejor sonrisa y volver a intentarlo. Espero que este proyecto sea el resultado de todos los sacrificios que dieron por mi bienestar y felicidad, dando como consecuencia la mejor satisfacción para ustedes y que un día sientan satisfacción y complacencia de tener a un hijo que valora el esfuerzo de la familia.

**Jorge Daniel Ortiz Gallegos**

## **AGRADECIMIENTO**

Agradezco ante todo a la Escuela Politécnica Nacional por darme la oportunidad de construir mi futuro con las mejores enseñanzas que esta institución puede ofrecer. A mis profesores por haberme dado sus conocimientos con el mejor carisma y dedicación hacia la profesión. Especialmente a los ingenieros; Ing. Ivonne Maldonado e Ing. Byron Loarte por haberme compartido sus conocimientos con todo el amor, la paciencia y voluntad. Gracias a ustedes he sentido un ambiente de felicidad y tranquilidad.

Espero que este proyecto de Integración Curricular sea del agrado de ustedes; Ingenieros e Institución, que me abrieron sus puertas para formarme como todo un profesional listo para surgir y elevarme ante el éxito. Es un gran honor haber pertenecido a esta noble Institución y anhelo cumplir con los buenos deseos que tengan hacia a mí.

**Jorge Daniel Ortiz Gallegos**

# ÍNDICE DE CONTENIDO

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO .....	V
RESUMEN.....	VII
ABSTRACT .....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general .....	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco teórico.....	3
2 METODOLOGÍA.....	5
2.1 Metodología de Desarrollo.....	5
Roles .....	6
Artefactos .....	6
2.2 Diseño de la arquitectura .....	8
Patrón Arquitectónico.....	8
2.3 Herramientas de desarrollo.....	9
Librerías .....	10
3 RESULTADOS .....	11
3.1 Sprint 0. Configuración del espacio de desarrollo .....	11
Recopilación y definición de requerimientos .....	11
Elaboración del Modelo de Datos .....	13
Roles de usuarios.....	13
3.2 Sprint 1. Desarrollo de endpoints de funcionalidades básicas de Usuarios Administrador y Cliente .....	14
Implementar endpoints para el inicio de sesión, cierre de sesión y recuperación de contraseña.....	14
Implementar endpoints para visualizar y editar el perfil.....	16
Implementar endpoints para el registro del usuario.....	17
3.3 Sprint 2. Desarrollo de endpoints para Usuario Administrador .....	18
Implementar endpoints para la gestión de contenido de la página principal.....	18
Implementar endpoints para la actualización de información de cada módulo emocional .....	20
Implementar endpoints para la gestión de la lista de reproducción de cada módulo emocional .....	22

3.4	Sprint 3. Desarrollo de endpoints para funcionalidades de interacción de Usuarios Administrador y Cliente .....	24
	Implementar endpoints para comentarios y sugerencias .....	24
	Implementar endpoints para eventos promocionales de la Escuela de Biodanza .....	25
	Implementar endpoints para reservación de eventos .....	27
3.5	Sprint 4. Desarrollo de endpoints funcionalidades extras.....	29
	Implementar endpoints para la gestión de publicidades .....	29
	Implementar endpoints para habilitar y deshabilitar usuarios clientes .....	31
3.6	Sprint 5. Pruebas y Despliegue.....	33
	Pruebas de rendimientos .....	33
	Pruebas de carga .....	34
	Despliegue del backend.....	35
4	CONCLUSIONES .....	36
5	RECOMENDACIONES .....	37
6	REFERENCIAS BIBLIOGRÁFICAS.....	38
7	ANEXOS.....	42
	ANEXO I.....	1
	ANEXO II.....	2
	ANEXO III.....	25
	ANEXO IV.....	26
	ANEXO V.....	27

## RESUMEN

Las aplicaciones web y móviles son parte de la vida cotidiana de la población, están presentes ante el consumo de contenido audiovisual y/o informativo por medio de sus equipos tecnológicos.

Por otro lado, Biodanza es un proceso conocido por la integración de temas de uso de movimientos corporales por medio de la sensación de la música, que produce una terapia psicológica y física, que se con un grupo de participantes.

El presente documento, muestra el desarrollo del componente *backend*; en la implementación de una API RESTful que provee *endpoints* de funcionalidades acorde a dos roles establecidos: administrador y cliente. El rol administrador ejerce las funciones de múltiples actividades en la actualización y/o creación de nuevos contenidos informativos o archivos por medio del uso de una aplicación web. Mientras que el rol cliente por medio de una aplicación móvil visualiza los contenidos de multimedia, eventos o publicaciones a través de las actividades realizadas del administrador, así como el poder reservar una clase en particular.

Este documento tiene la siguiente estructura de desarrollo: la primera sección donde de detallar los antecedentes, objetivos, alcance y marco teórico. la segunda sección que explica la metodología de desarrollo ágil *Scrum* junto con su guía de implementación, además se muestra la arquitectura y herramientas para este proyecto. La tercera sección detalla los resultados de implementación de la metodología por *sprints*. Y finalmente las secciones cuatro y cinco presentan conclusiones y recomendaciones que se ha presentado en el desarrollo de este proyecto.

**PALABRAS CLAVE:** *Backend*, API Restful, web, Scrum, Biodanza

## ABSTRACT

Web and mobile applications are part of the daily life of the population, they are present before the consumption of audiovisual and/or informative content through their technological equipment.

In this context, Biodanza is a process of integration and organic renewal, which uses, music, dance, body movements, is carried out through group encounter situations, and its constant practice produces therapeutic effects.

The present work shows the development of the *Backend* component; in the implementation of a RESTful API that provides functionality *endpoints* according to two established roles: administrator and client. The administrator role exercises the functions of multiple activities in updating and/or creating new information content or files using a web application. While the client role through a mobile application displays multimedia content, events or publications through the activities carried out by the administrator, as well as being able to book a particular class.

This document has the following development structure: the first section where the background, objectives, scope, and theoretical framework are detailed. The second section that explains the Scrum Agile Development Methodology together with its implementation guide, also shows the architecture and tools for this project. The third section details the results of the implementation of the methodology by Sprints. And finally, sections four and five present conclusions and recommendations that have been presented in the development of this project.

**KEYWORDS:** backend, API Restful, web, Scrum, Biodanza.

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Según el INEC, en 2019, se registró un total de 17.8% de índice de suicidios de la población ecuatoriana de entre 12 a 17 años, incluyendo motivos de soledad, problemas familiares y problemas psicológicos. Resultando así que Ecuador este dentro de los 10 primeros países con mayor índice de suicidio a nivel mundial [1].

En tiempos de confinamiento, por el COVID-19, los sucesos de tendencias suicidas en los jóvenes aumentaron por situaciones de aislamiento y problemas de depresión. Y a la fecha se ha registrado sucesos de problemas psicológicos por motivo del alto índice de pobreza y falta de empleo, afectando notablemente la salud mental [2].

Estos datos son alarmantes, puesto que indican que la ciudadanía ecuatoriana sufre de severos problemas en la conducta psicológica, ante sucesos de tragedias profesionales, familiares y/o sociales. La Escuela de Biodanza STR en Quito-Puembo es especialista en biodanza, un sistema de transformación personal que pretende que las personas se conecten consigo, con los demás y con el entorno, de manera que puedan ir recuperando la alegría. Esta técnica se enfoca en movimientos acompañados de música que integran y armonizan el cuerpo, la mente y las emociones, de ahí el hecho de que la Escuela de Biodanza STR en Quito - Puembo tenga como principal objetivo ayudar a las personas a sobrellevar problemas psicológicos evitando en lo posible la idea del suicidio, esto debido a que los expertos ven a la biodanza como una opción terapéutica para mejorar ciertos problemas psicológicos, de salud y bienestar, ayudando a controlar de forma voluntaria los estados de estrés y ansiedad [3] [4] .

Por otro lado, en la actualidad el *software* constituye una herramienta poderosa para los negocios, que por medio de páginas web o aplicativos móviles llegan a un mayor número de clientes, de ahí que el presente Trabajo de Integración Curricular tenga como objetivo el desarrollo del componente del *backend* para la página web y la aplicación móvil de la Escuela de Biodanza STR en Quito - Puembo.

La construcción del *backend* se ha realizado con la ayuda de Laravel, *framework* que ha permitido el desarrollo de una API RESTful con múltiples *endpoints*, MySQL para el almacenamiento de la información y Cloudinary para el almacenamiento de archivos multimedia.

El *backend* cuenta con diferentes *endpoints* que son consumidos por el usuario administrador quien gestiona los contenidos de la parte web, información almacenada

acorde al requerimiento del usuario, quien a su vez puede visualizar e interactuar con los temas direccionados por medio de la parte móvil.

## 1.1 Objetivo general

Desarrollar el *backend* de un sistema para la escuela de Biodanza SRT en Quito-Puembo.

## 1.2 Objetivos específicos

1. Levantar los requerimientos para el *backend*.
2. Diseñar el modelo de Base de Datos y arquitectura para el *backend*.
3. Codificar los múltiples servicios(*endpoints*) de acuerdo con los requerimientos.
4. Comprobar la funcionalidad del *backend*.
5. Desplegar el *backend*.

## 1.3 Alcance

Todo lo que se puede observar en el internet no tendría mayor sentido si no estuviera presente el desarrollo *backend*, conocido como el desarrollo del lado del servidor es lo que hace que todo “funcione” en el interior de una aplicación web o móvil y en esencia es la parte invisible para el usuario, pero sin esta el usuario no podría realizar casi nada. El desarrollo del *backend* implica la lógica y la integración del lado del servidor, la creación de API y el trabajo con componentes del sistema [5].

En el presente Proyecto de Integración Curricular se ha desarrollado un *backend* que es consumido por la parte web y por la parte móvil de la aplicación para la Escuela de Biodanza STR en Quito – Puembo a la que se ha denominado “AlphaO”.

En lo que se refiere a la parte web se ha desarrollado varios *endpoints* para que el usuario con perfil “administrador” gestione todo lo referente a la información y servicios de la escuela, mientras que en lo referente a la parte móvil se ha desarrollado varios *endpoints* para que el usuario con perfil “cliente” visualice dicha información e interactúe de acuerdo con la lógica del negocio.

A continuación, se lista los diferentes *endpoints* implementados de acuerdo con el perfil de usuario.

### Administrador – Parte Web

- *Endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña.

- *Endpoints* para modificar el perfil de usuario.
- *Endpoints* para modificar la información de la página principal.
- *Endpoints* para modificar los 5 módulos de Biodanza.
- *Endpoints* para gestionar la lista de reproducción por cada módulo de Biodanza.
- *Endpoints* para el control de reservaciones.
- *Endpoints* para la administración de usuarios clientes.
- *Endpoints* para modificar publicidades de futuras reuniones de Biodanza.
- *Endpoints* para gestionar comentarios y/o sugerencias.

#### **Cliente – Parte Móvil**

- *Endpoints* para el registro por medio de un formulario de inscripción.
- *Splash* de imágenes acerca de la aplicación de Biodanza.
- *Endpoints* para iniciar sesión, cerrar sesión y recuperar contraseña.
- *Endpoints* para la actualización del perfil de usuario.
- *Endpoints* para visualizar el contenido de la página principal.
- *Endpoints* para visualizar la información general de los 6 módulos de la Biodanza.
- *Endpoints* para visualizar y reproducir el contenido multimedia.
- *Endpoints* para el envío de comentarios y sugerencias.
- *Endpoints* para la visualización de publicidades.
- *Endpoints* para realizar reservaciones.

### **1.4 Marco teórico**

Las aplicaciones web poseen una gran cantidad de información almacenada en una base de datos, en su mayoría el usuario accede al contenido por medio de una autenticación de credenciales.

En la actualidad las aplicaciones web y móviles utilizan APIs RESTful, que son interfaces de desarrollo que se basan en el protocolo HTTP y que devuelven como respuesta archivos en formato JSON o XML. Estas interfaces ayudan en la obtención de información de una

base de datos o en el enlace comunicativo de máquina a máquina, maquina-cliente o servidor [6].

JSON (*JavaScript Object Notation*) es un formato que permite el trámite de datos que son usados para el consumo de información en aplicaciones web y móviles. Contiene datos estructurados formados de dos parámetros: clave y valor. Es utilizado para la transferencia de datos por medio de peticiones o llamados *request* [7].

En el desarrollo de *software* intervienen principalmente dos componentes, el *Frontend* y el *backend*, ambos logran lo que se ve y cómo funciona una página web o aplicación móvil.

Se conoce como *backend* a la capa de acceso restringido para el usuario y que es usado para obtener información de un *software*. Es la lógica que contiene la aplicación web o móvil de cómo se maneja los datos y funcionalidades. El *backend* se comunica con el servidor para el trámite de solicitudes devolviendo respuestas que la aplicación debe ejercer para un funcionamiento satisfactorio, estableciendo una comunicación segura y haciendo que su desempeño no confunda la experiencia de usuario. El *backend* está presente en la lógica de las aplicaciones web y móvil de “AlphaO” [8].

La API se ha desarrollado con la ayuda de Laravel, un *framework* de PHP que permite el diseño de aplicaciones web y APIS RESTful, contiene una extensa documentación sobre cómo desarrollar nuevos métodos y brinda herramientas solidas de inyección de dependencias, pruebas unitarias, colas y eventos de análisis en tiempo real [9].

El desarrollo de nuevas aplicaciones web y móviles necesitan múltiples herramientas que ayuden en cuento al almacenamiento de archivos en un contenedor para su consumo exitoso. Cloudinary es un Storage de almacenamiento de múltiples archivos con un tiempo de guardado ilimitado para desarrolladores que quiere agregar en sus aplicaciones contenidos audiovisuales o archivos comunes a cualquier aplicación, de esta manera ayuda con el almacenamiento general de topo tipo de extensión, es gratuito y accesible para el usuario con los siguientes métodos [10]:

- *Cloudinary Upload*, el cual permite enviar un contenido de archivo con extensión jpg, jpeg, png y svg y devolver una URL de visualización [11].
- *Cloudinary UploadVideo*, el cual permite enviar un archivo de multimedia a la base de datos y devuelve una url de visualización [11].
- *Cloudinary UploadFile*, el cual permite el almacenamiento de archivos especiales y en conjunto con las anteriores extensiones [11].

## 2 METODOLOGÍA

Hacer uso de una metodología dentro de la solución de cualquier problema resulta un recurso importante puesto que al seguir su guía se logra alcanzar los objetivos planteados de manera rápida, eficiente y organizada [12].

El estudio de casos se caracteriza por necesitar un proceso de indagación y análisis de uno o varios casos, de manera que se logre entender las situaciones o circunstancias en las cuales se plantea el problema a ser resuelto. Esto solo se alcanza con la obtención de información por medio de investigación del campo en donde se está llevando el estudio [13].

Por lo general, el estudio de casos se centra en un tema en particular de la vida real como es el caso del presente Proyecto de Integración Curricular, donde en base a un problema existente se ha implementado una solución para la Escuela de Biodanza STR en Quito – Puenbo por medio del desarrollo de un *backend* que es consumido tanto por un sistema web como por una aplicación móvil.

### 2.1 Metodología de Desarrollo

Las metodologías de desarrollo de *software* son un conjunto de técnicas y métodos organizativos que ayudan en el diseño de búsqueda de soluciones al crear aplicaciones. Su principal objetivo es organizar el equipo de trabajo para que puedan desarrollar funciones en el sistema o proyecto de la mejor manera posible [14].

Por su lado, las metodologías ágiles busca proporcionar pequeñas piezas de avance del *software* que funcionen correctamente en un lapso corto para aumentar la confiabilidad del cliente. Estas metodologías utilizan enfoques de una buena calidad de trabajo con el equipo de desarrollo y una buena relación con el representante de la aplicación de manera presencial durante el ciclo de vida de desarrollo del proyecto de *software* [15].

*Scrum* es una metodología ágil que transforma proyectos con una dificultad compleja de un entorno dinámico en pequeños fragmentos que sean demasiado flexibles de desarrollarlo. Está basado en la presentación de avances del proyecto final hacia el cliente y es una metodología que sirve para mejorar el rendimiento colaborativo de los equipos de desarrollo de *software* [16]. Por ello el presente Proyecto de Integración Curricular ha hecho uso de esta metodología ágil, permitiendo distribuir las actividades para todo el equipo de desarrolladores, cumpliendo con los tiempos establecidos y logrando los objetivos planteados.

## Roles

Los roles dentro de *Scrum* cumplen un papel central debido a las funciones y responsabilidades dentro de su participación indispensable para alcanzar el éxito de un proyecto, ya que definen el compromiso dentro del mismo. En el presente Proyecto de Integración Curricular se tienen los 3 roles centrales: *Product Owner*, *Scrum Master* y *Development Team* [16].

### **Product Owner**

Es el representante de los clientes. Es quien explica las necesidades, requerimientos y características del producto. El producto debe cumplir con sus exigencias para poder considerarlo como un proyecto satisfactorio. La **TABLA I** evidencia la persona que cumple con este rol [16].

### **Scrum Master**

Es el líder, su función es estar en contacto con todo el equipo de manera que esté al tanto de todo el avance del proyecto, realiza el seguimiento de proyecto encargándose de administrar y controlar los avances del mismo por medio de múltiples reuniones. La **TABLA I** evidencia la persona que cumple con este rol [16].

### **Development Team**

Es el grupo de programadores con las habilidades necesarias para cumplir cada una de las tareas proporcionadas por el *Product Owner*, también son los encargados de presentar entregables funcionales al cliente por cada *Sprint* que se haya planificado [16]. La **TABLA I** evidencia la persona que cumple con este rol.

**TABLA I:** Asignación de Roles.

<b>ROLES</b>	<b>NOMBRES</b>
<i>Product Owner</i>	María Susana Vela Witt.
<i>Scrum Master</i>	Ing. Ivonne Maldonado, MSc.
<i>Development Team</i>	Jorge Ortiz

## Artefactos

Los artefactos son elementos que el equipo *Scrum* deben seguir para el buen desarrollo del producto, las acciones a realizar y el cómo ir creando el sistema esta transparentado

por estos documentos [17]. El componente *backend* sigue los siguientes artefactos de manera que se garantiza el desarrollo y funcionamiento del proyecto establecido.

### Recopilación de requerimientos

Los requerimientos de *software* son las necesidades que debe cumplir el proyecto, aquellas funcionalidades estas esperadas, el levantamiento de estas necesidades son claves para alcanzar los objetivos planteados [18]. Para conseguir las necesidades del presente Proyecto de Integración Curricular han sido necesarias reuniones con el *Product Owner*, lo que ha permitido la generación de la lista de requerimientos misma que se encuentra en **ANEXO II** del presente documento.

### Historias de Usuario

Es una explicación simple e informal donde se detalla la funcionalidad del sistema acorde a la visión del usuario final. Su propósito es articular las funcionalidades del proyecto, basándose en las necesidades del usuario final [19]. En la **TABLA II** se presenta un ejemplo de las Historias de Usuario utilizadas en el desarrollo del presente proyecto, las restantes se encuentran detalladas en el **ANEXO II** del documento mencionado.

**TABLA II:** Historia de usuario Nro.5.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU005	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Gestionar contenido multimedia.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 2	
<b>Responsable (es):</b> Jorge Ortiz	
<p><b>Descripción:</b> El <i>componente backend</i> por medio del rol de administrador podrá gestionar el contenido de la lista de reproducción de los temas de Biodanza:</p> <ul style="list-style-type: none"> <li>• Crear contenido multimedia.</li> <li>• Visualizar su contenido multimedia.</li> <li>• Editar la descripción del contenido multimedia.</li> </ul>	
<b>Observación:</b> El usuario administrado una vez dentro de la aplicación AlphaO puede agregar nuevos elementos musicales o audiovisuales de cada Temática de Biodanza.	

### ***Product Backlog***

Es el listado de todas las tareas asignadas que tienen como objetivo ser finalizadas acorde al desarrollo del proyecto. Debe estar visible para el equipo y tener una visión panorámica sobre lo que se quiere realizar [20]. En el **ANEXO II** del presente documento se encuentra a detalle el *Product Backlog* del presente proyecto.

### ***Sprint Backlog***

Es una lista con mayor detalle sobre cada una de las tareas que se deben realizar, estas son colocadas y clasificadas por medio de *Sprints* [19]. Para desarrollar cada una de las tareas estas cuentan con un tiempo determinado para su cumplimiento. En el **ANEXO II** del documento mencionado se encuentra a detalle el *Sprint Backlog* del presente proyecto.

## **2.2 Diseño de la arquitectura**

Se define como la arquitectura de *software* a la planificación de modelos, patrones y abstracciones teóricos en el desarrollo de un componente que presenta un nivel de complejidad y la necesidad de implementación que requiere para su funcionalidad [22]. Por lo anteriormente mencionado se presenta la arquitectura de software con las librerías de importación necesarias que necesita el componente *backend* para su aplicación.

### **Patrón Arquitectónico**

El patrón arquitectónico definido para el desarrollo del componente *backend* es Modelo-Vista-Controlador (MVC).

- **Modelo:** define la cantidad de datos que contiene el sistema o aplicación. Si existe alguna actualización del modelo, se notifica a la vista para su respectivo despliegue de información de datos [23].
- **Vista:** muestra la información de los datos que posee el sistema o aplicación [23].
- **Controlador:** contiene la lógica que controla y actualiza al modelo y vista por medio de las respuestas de entrada por parte del usuario [23].

La **Fig. 1** presenta el patrón arquitectónico que se ha usado para el desarrollo del *componente backend*, este presenta una serie de ventajas tales como: estructura organizada, escalable y de fácil mantenimiento.

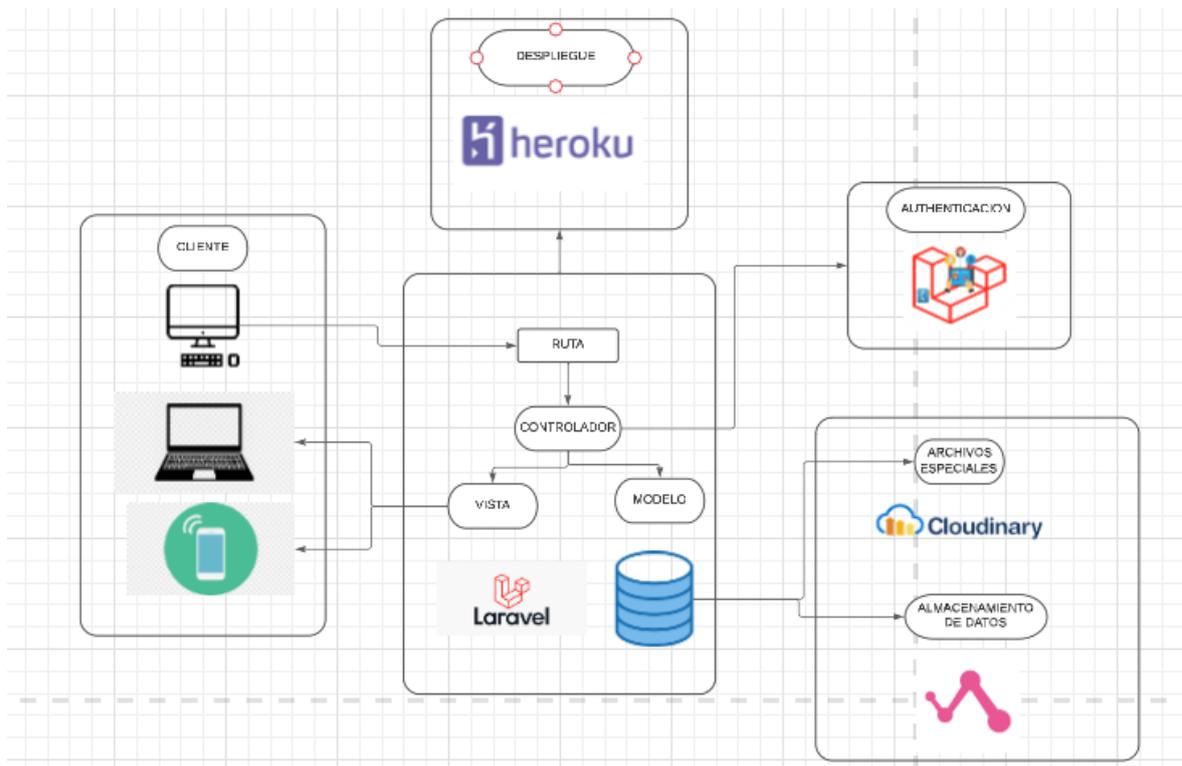


Fig. 1: Modelo Arquitectónico – *backend*.

## 2.3 Herramientas de desarrollo

Una vez establecido el patrón arquitectónico del *backend*, a continuación, la **TABLA III** muestra las herramientas de desarrollo del componente *backend*, junto con un detalle y justificación de su uso ya que han facilitado, simplificado y aclarado el desempeño del trabajo establecido; permitiendo una solución simple al problema.

**TABLA III:** Herramientas para el desarrollo del *backend*.

HERRAMIENTA	JUSTIFICACIÓN
<b><i>Xampp</i></b>	Es un servidor web local multiplataforma que se enfoca en la gestión de paquetes de bases de datos y el despliegue de desarrollos web sin necesidad de internet para poder ser ejecutadas [24]. Se ha utilizado para realizar las primeras pruebas del <i>backend</i> únicamente en el desarrollo local.
<b><i>Laravel Authentication</i></b>	Ha permitido implementar de manera automática la capa de autenticación de usuarios [25].

<b><i>AlwaysData</i></b>	Es un alojamiento gratis de una capacidad de 100MB donde se encuentra almacenada la base de datos de presente Trabajo de Integración Curricular [26].
<b><i>Heroku</i></b>	Es una plataforma de servicio que ayuda en el despliegue de aplicaciones web de una forma sencilla donde el objetivo es desarrollar únicamente la codificación [27]. Ha permitido que el despliegue del <i>backend</i> se lo realice de manera simple y sin mayor inconveniente.
<b><i>Laravel</i></b>	Es un <i>Framework</i> por parte del servidor con el uso de herramientas de migraciones de datos y creación de diferentes componentes entre controladores, modelos y bases de datos en base al Modelo MVC [28]. Ha permitido la creación del proyecto de manera sencilla, además de permitir la importación de librerías de manera fácil, permitiendo que el desarrollo sea organizado y mantenible.

### Librerías

La **TABLA IV** muestra las librerías que se han implementado para el desarrollo del *backend* y la incorporación con las herramientas listadas anteriormente.

**TABLA IV:** Librerías para el desarrollo del *backend*.

<b>LIBRERÍA</b>	<b>DESCRIPCIÓN</b>
<b><i>Cloudinary</i></b>	Cloudinary es un servicio en la nube que almacena los archivos en secciones y para recuperarlas se accede mediante la recuperación de URLs [29]. Ha permitido almacenar archivos mp3, mp4 y fotografías para el aplicativo.
<b><i>Storage Laravel</i></b>	Es un contenedor de almacenamiento de vía local que ayuda a resguardar y comprobar el flujo de comunicación entre las peticiones de guardado y actualización de archivos de datos [30]. Ha permitido realizar las primeras pruebas de almacenamiento por medio de un desarrollo local.

## 3 RESULTADOS

Esta sección, presenta los resultados obtenidos por medio del desarrollo establecido para el componente *backend*. Toda la información se detalla por medios de *Sprints* para clasificar los resultados que se han determinado en el Sprint Backlog.

### 3.1 *Sprint 0*. Configuración del espacio de desarrollo

El *Sprint 0* contiene los siguientes resultados:

- Recopilación y definición de requerimientos.
- Elaboración del Modelo de Datos
- Roles de usuarios.

#### **Recopilación y definición de requerimientos**

##### **Implementar *endpoints* para el registro de usuarios**

El componente *backend* cuenta con un *endpoint* que permite el registro de un usuario, a este se le asigna el rol de cliente.

##### **Implementar *endpoints* para el iniciar sesión, cerrar sesión y cambiar contraseña.**

El componente *backend* contiene *endpoints* para el proceso de inicio de sesión (autenticación), cierre de sesión y cambio de contraseña; por medio de la creación de un *token Bearer*.

##### **Implementar *endpoints* para recuperación de contraseña**

El *backend* crea varios *endpoints* que le permiten al usuario cliente recuperar contraseña de acceso, por medio de un correo electrónico que cuenta con un enlace que redirige al inicio de sesión.

##### **Implementar *endpoints* para modificar el perfil de usuario**

El componente *backend* tiene varios *endpoints* que permiten la visualización y actualización de datos personales e imagen de perfil del usuario administrador y cliente.

##### **Implementar *endpoints* para la gestión de la página principal**

El componente *backend* permite al administrador realizar modificaciones de contenido en la página principal. Los cambios que el administrador realice, el cliente lo visualiza tanto en la parte web como en la parte móvil.

### Implementar *endpoints* para gestionar módulos emocionales establecidos de Biodanza.

El componente *backend* crea *endpoints* para modificar el contenido descriptivo de los 5 modelos establecidos de los géneros emocionales: Ira, Miedo, Tristeza, Ansiedad y Soledad.

### Implementar *endpoints* para gestionar las listas de reproducciones de musica

El componente *backend* crea *endpoints* para gestionar (modificar, crear, eliminar y reproducir el contenido multimedia) las listas de reproducción que contiene cada módulo.

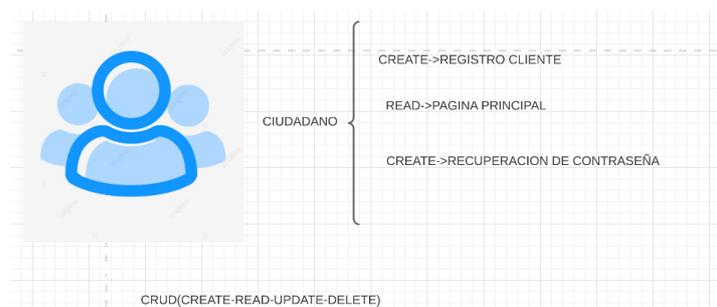
### Implementar *endpoints* para la gestión de usuarios de la aplicación

El componente *backend* crea *endpoints* para que el administrador pueda listar los usuarios clientes de la aplicación móvil y pueda bloquear el acceso si así lo viera necesario. El bloqueo es notificado por medio de un mensaje de correo electrónico con la respectiva observación de la causa del bloqueo.

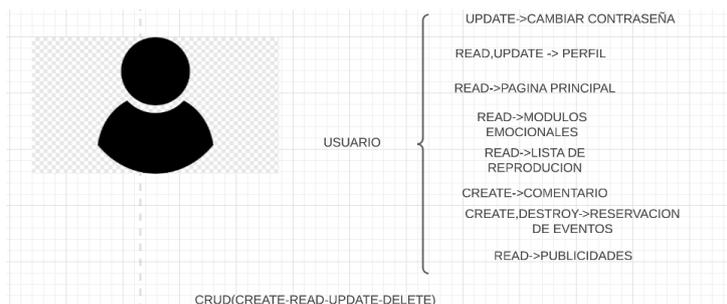
### Implementar *endpoints* para gestionar los espacios de opinión o sugerencias

El componente *backend* crea *endpoints* para que el usuario cliente pueda realizar comentarios, estos comentarios pueden ser gestionados por el usuario administrador.

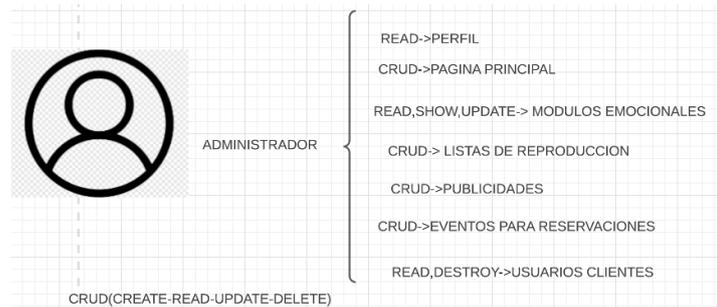
La **Fig. 2**, **Fig. 3** y **Fig. 4** muestran las diferentes actividades que realizan los usuarios que utilizan el *backend* en sus respectivas aplicaciones.



**Fig. 2:** Perfil de usuario invitado.



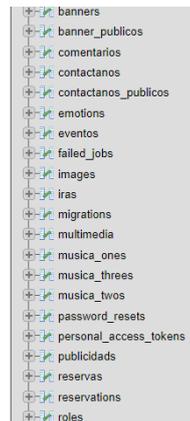
**Fig. 3:** Perfil de usuario cliente.



**Fig. 4:** Perfil de usuario administrador.

### Elaboración del Modelo de Datos

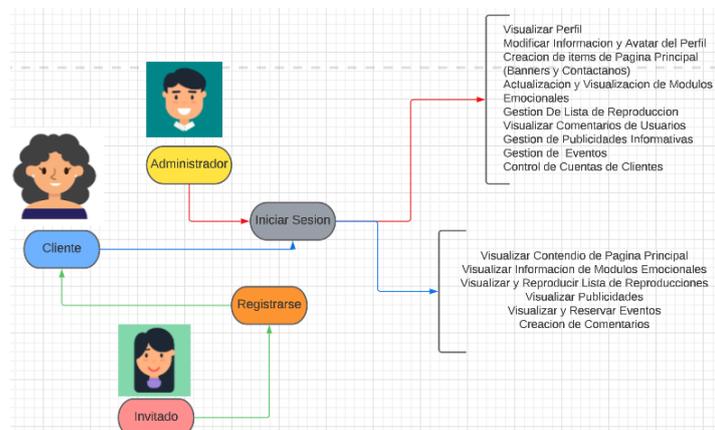
Para el almacenamiento de datos se ha elegido la plataforma de *Alwaysdata*, misma que cuenta con *MySQL* y *PhpMyadmin*. La **Fig. 5** presenta la lista de tablas del modelo de base de datos del componente *backend*. El esquema completo se presenta en el **ANEXO II** de este documento.



**Fig. 5:** Tablas del Modelo de datos en *Alwaysdata*.

### Roles de usuarios

A continuación, la **Fig. 6** muestra los usuarios y las funcionalidades respectivas de acuerdo a la lógica de AlphaO.



**Fig. 6:** Perfiles de usuario.

## 3.2 *Sprint* 1. Desarrollo de *endpoints* de funcionalidades básicas de Usuarios Administrador y Cliente

El *Sprint* 1 contiene los siguientes resultados:

- Implementar *endpoints* para el inicio de sesión, cierre de sesión y recuperación de contraseña.
- Implementar *endpoints* para visualizar y editar perfil.
- Implementar *endpoints* para el registro del usuario.

### Implementar *endpoints* para el inicio de sesión, cierre de sesión y recuperación de contraseña

El *endpoint* para el inicio de sesión solicita correo y contraseña para la autenticación como se muestra en la **Fig. 7**. El *endpoint* para la recuperación de contraseña, solicita el correo del usuario como se muestra en la **Fig. 8** y, como consecuencia se envía un enlace especial al correo del usuario, dicho correo contiene un token para el restablecimiento de la contraseña. El *endpoint* para cerrar sesión permite al usuario a desconectarse del *backend* como se muestra en la **Fig. 9**.

En *Scrum* cada *sprint* tiene un avance demostrable, es decir para poder afirmar que se ha cumplido con las tareas determinadas en cada *sprint* se realizan pruebas unitarias que permiten la verificación del correcto funcionamiento de pequeñas partes del código, con lo que se aumenta la confiabilidad del desarrollo y la calidad en la satisfacción del cumplimiento de las necesidades del cliente por cada iteración. De ahí el hecho de que en cada *sprint* se ha realizado las diferentes pruebas unitarias que han permitido verificar la implementación de cada *endpoint* de manera correcta.

La **Fig. 10** y **Fig. 11** muestran la prueba unitaria (bloque de código y resultado) del inicio de sesión. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el **ANEXO II** de este documento.

```

public function login(Request $request)
{
    $request -> validate([
        'email' => ['required', 'string', 'email'],
        'password' => ['required', 'string'],
    ]);

    $user = User::where('email', $request['email'])->first();

    if (!$user || !$user->state || in_array($user->role->slug, $this->discarded_role_names) ||
        !Hash::check($request['password'], $user->password))
    {
        return $this->sendResponse(message: 'The provided credentials are incorrect.', code: 404);
    }

    if ($user->state!=1)
    {
        return $this->sendResponse(message: 'This account is blocked for any motive.', code: 404);
    }

    if (!$user->tokens->isEmpty())
    {
        $user->tokens()->delete();
    }

    $token = $user->createToken('auth-token')->plainTextToken;

    return $this->sendResponse(message: 'Successful authentication.', result: [

```

Fig. 7: Método para iniciar sesión.

```

public function resendLink(Request $request)
{
    // Validación de los datos de entrada
    $request->validate([
        'email' => ['required', 'email'],
    ]);

    // enviar el link de restablecimiento de contraseña al mail
    // https://laravel.com/docs/9.x/passwords#requesting-the-password-reset-link
    $status = Password::sendResetLink(
        $request->only('email')
    );

    // Se invoca a la función padre
    return $status === Password::RESET_LINK_SENT
        ? $this->sendResponse(__($status))
        : $this->sendResponse(
            message: 'Link reset failure.',
            errors: ['email' => __($status)],
            code: 422
        );
}

```

Fig. 8: Método para recuperar contraseña.

```

public function logout(Request $request)
{
    $request->user()->tokens()->delete();

    return $this->sendResponse(message: 'Logged out.');
```

Fig. 9: Método para cerrar sesión.

```

public function test_inicio_de_sesion()
{
    $test_request = $this->post('/api/alpha/login', [
        'email' => "andres89@example.com",
        'password' => "secret"
    ]);
    $test_request->assertStatus(200);
}

```

Fig. 10: Prueba unitaria de inicio de sesión

```

PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\GeneralTest
✓ inicio de sesion

Tests: 1 passed
Time: 6.23s

PS D:\Tesis\alphaomega>

```

Fig. 11: Resultado de prueba unitaria de inicio de sesión

## Implementar *endpoints* para visualizar y editar el perfil

El método para visualizar la información del usuario detalla los datos personales y la dirección de imagen del avatar (ver Fig. 12).

El método para actualizar información de perfil se lo realiza por medio del identificador ID (ver Fig.13).

Tanto el método visualizar como actualizar están implementados tanto para el usuario administrador como para el usuario cliente.

La Fig.14 y Fig.15 muestran los resultados de la prueba unitaria de visualizar perfil. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el ANEXO II de este documento.

```

public function show()
{
    // Se obtiene el usuario autenticado
    // https://laravel.com/docs/9.x/authentication#retrieving-the-authenticated-user
    $user = Auth::user();
    // Se invoca a la función padre
    return $this->sendResponse(message: "User's profile returned successfully", result: [
        'user' => new ProfileResource($user),
        'avatar' => $user->getAvatarPath(),
        'imagen' => ImageHelper::getDiskImageUrl($user->getAvatarPath())
    ]);
}

```

Fig. 12: Método para visualizar perfil de usuario.

```

public function store(Request $request)
{
    // Validar que el usuario sea mayor de edad
    $allowed_date_range = [
        'max' => date('Y-m-d', strtotime('-70 years')),
        'min' => date('Y-m-d', strtotime('-18 years')),
    ];

    // Validación de los datos de entrada
    $request->validate([
        'first_name' => ['required', 'string', 'min:3', 'max:35'],
        'last_name' => ['required', 'string', 'min:3', 'max:35'],
        // https://laravel.com/docs/9.x/validation#rule-unique
        'username' => ['required', 'string', 'min:5', 'max:20', Rule::unique('users')->ignore($request->user)],
        // https://laravel.com/docs/9.x/validation#rule-after
        'birthdate' => ['nullable', 'string', 'date-format:Y-m-d', 'after_or_equal:{$allowed_date_range["min"]}',
            'before_or_equal:{$allowed_date_range["max"]}'],
        'personal_phone' => ['required', 'numeric', 'digits:10'],
        'home_phone' => ['nullable', 'numeric', 'digits:9'],
        'address' => ['required', 'string', 'min:5', 'max:50'],
    ]);

    // Se obtiene el modelo del usuario
    $user = $request->user();
    // Se actualiza el modelo en la BDD
    // https://laravel.com/docs/9.x/queries#update-statements
    $user->update($request->all());
    // Se invoca a la función padre
    return $this->sendResponse("Profile updated successfully");
}

```

Fig.13: Método para editar perfil de usuario.

```

public function test_visualizacion_de_perfil()
{
    $user = User::where('id', 1)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/profile');
    $test_request->assertStatus(200);
}

```

Fig.14: Prueba unitaria de visualizar perfil.

```

PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\GeneralTest
✓ visualizacion de perfil

Tests: 1 passed
Time: 3.18s

```

Fig.15: Resultado de la prueba unitario de visualizar perfil.

### Implementar *endpoints* para el registro del usuario

Se ha implementado un método que permite la creación de una cuenta con el rol cliente para el acceso desde la aplicación móvil. El método cuenta con un formulario de registro (ver Fig. 16).

La Fig. 17 y Fig. 18 muestran los resultados de la prueba unitaria de registrar usuario cliente. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el ANEXO II de este documento.

```

public function store(Request $request)
{
    $request->validate([
        'first_name' => ['required', 'string', 'min:3', 'max:35'],
        'last_name' => ['required', 'string', 'min:3', 'max:35'],
        'username' => ['required', 'string', 'min:5', 'max:20', 'unique:users'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'personal_phone' => ['required', 'numeric', 'digits:10'],
        'home_phone' => ['required', 'numeric', 'digits:9'],
        'address' => ['required', 'string', 'min:5', 'max:50'],
        'password' => ['required', 'string', 'min:5', 'max:20']
    ]);
    $request->request->add([
        'password' => Hash::make($request['password'])
    ]);

    $role = Role::where('slug', $this->role_slug)->first();

    $user = new User($request->all());

    $role->users()->save($user);

    return $this->sendResponse(message: 'User stored successfully');
}

```

Fig. 16: Método para el registro de usuarios

```

public function test_registro_cliente()
{
    $test_request = $this->post('/api/alpha/register', [
        "username" => "Hector89",
        "first_name" => "Hector",
        "last_name" => "Perez",
        "personal_phone" => "0992006598",
        "address" => "453 Chillogallo Oval Apt. 7",

        "email" => "hector89@gmail.com",

        "birthdate" => "1999-08-15",
        "home_phone" => "029637977",

        "password" => "Hola1234*",
    ]);
    $test_request->assertStatus(200);
}

```

Fig. 17: Prueba unitaria de registro de usuario cliente

```

PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\GeneralTest
✓ registro cliente

Tests: 1 passed
Time: 4.54s

```

Fig. 18: Resultado de prueba unitaria de registro invitado a cliente.

### 3.3 *Sprint 2. Desarrollo de endpoints para Usuario Administrador*

El *Sprint 2* contiene los siguientes resultados:

- Implementar *endpoints* para la gestión de contenido de la página principal.
- Implementar *endpoints* para la actualización de información de cada módulo emocional.
- Implementar *endpoints* para la gestión de la lista de reproducción de cada módulo emocional.

#### Implementar *endpoints* para la gestión de contenido de la página principal

Los métodos que permiten la gestión del contenido en la página principal de AlphaO son:

- **Método visualizar:** muestra el banner que se encuentran en la página principal (ver Fig. 19). Por otro lado, también permite visualizar la información de la directiva de la Escuela de Biodanza junto con los contactos (ver Fig. 20)
- **Método crear:** permite la creación del contenido principal de la página (Banner) ingresando un nombre y el archivo. Así también se cuenta con el método crear de

“contáctanos” para ello se requiere el ingreso de: nombre, apellido, cargo, email y número de celular. Para ambos casos existen las validaciones necesarias de acuerdo a la lógica del negocio.

- **Método Actualizar:** permite la actualización del contenido informativo de contactos con los ítems de nombre, apellido, email, puesto y número telefónico.
- **Método Eliminar:** permite eliminar el contenido de la página principal (Banner) ingresando su ID. Así también se cuenta este mismo método para eliminar contenido de contactos.

Cabe mencionar que todos los métodos se encuentran documentados en *notion* y pueden visualizarse en el *link* que se presenta en el **ANEXO III**.

La **Fig. 21** y **Fig. 22** muestran los resultados de la prueba unitaria de creación de contactos. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el **ANEXO II** de este documento.

```
public function index()
{
    //
    $banner=Banner::all();
    return $this->sendResponse(message: 'Lista de banners desplegados satisfactoriamente', result: [
        'banners' => BannerResource::collection($banner),
    ]);
}
```

**Fig. 19:** Método para visualizar publicidades informativas.

```
public function index()
{
    //
    $contacto=Contactanos::all();
    return $this->sendResponse(message: 'Contactos list generated successfully', result: [
        'contactanos' => ContactanosResource::collection($contacto),
    ]);
}
```

**Fig. 20:** Método para visualizar contactos.

```
public function test_crear_contactanos()
{
    $user = User::factory()->make(['role_id' => 2]);
    $contactos = [
        "nombre" => "Luis",
        "apellido" => "Espinoza",
        "correo" => "luis@gmail.com",
        "puesto" => "Director",
        "contactanos" => "0995102824",
    ];
    $test_request = $this->actingAs($user)->post('/api/alpha/contactos/create', $contactos);
    $test_request->assertStatus(200);
}
```

**Fig. 21:** Prueba unitaria de creación de contactos.

```
PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\GeneralTest
✓ crear contactanos

Tests: 1 passed
Time: 2.99s
```

Fig. 22: Resultado de pruebas unitaria de contactos.

## Implementar *endpoints* para la actualización de información de cada módulo emocional

Este método permite actualizar información de texto (tema y descripción) de un módulo emocional establecido (ver Fig. 23). El método implementa la creación de los 5 módulos emocionales establecidos, almacenándolos en la Base de Datos MySQL (Alwaysdata) y archivando su contenido en Cloudinary (ver Fig.24). Este método se utiliza para la visualización de cada módulo emocional específico (ver Fig. 25).

La Fig. 26 y Fig. 27 muestran los resultados de la prueba unitaria de actualizar emociones. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el ANEXO II de este documento.

```
public function update(Request $request,Ira $emotion)
{
    //
    $request->validate([
        'Tema' => ['required', 'string', 'min:3', 'max:45'],
        'descripcion' => ['required', 'string', 'min:3', 'max:600'],
    ]);

    $ira= $request -> validate([
        'video' => ['required','file','mimes:mp4','max:6000000'],
    ]);
    $file = $ira['video'];
    $uploadedFileUrl = Cloudinary::uploadVideo($file->getRealPath(),['folder'=>'emotions']);
    $url = $uploadedFileUrl->getSecurePath();

    $emotion->update([
        "Tema"=>$request->Tema,
        "descripcion"=>$request->descripcion,
        "video"=>$url
    ]);
    return $this->sendResponse('Emotion update succesfully',200);
}
```

Fig. 23: Método para actualizar información de emociones.

```

public function store(Request $request)
{
    //

    $request->validate([
        'Tema' => ['required', 'string', 'min:3', 'max:45'],
        'descripcion' => ['required', 'string', 'min:3', 'max:600'],
    ]);

    $ira= $request ->validate([
        'video' => ['file'],
    ]);
    $file = $ira['video'];

    $uploadedFileUrl = Cloudinary::uploadVideo($file->getRealPath(),['folder'=>'emotions','resource_
    'public_id' => 'myfolder/mysubfolder/dog_closeup',
    'chunk_size' => 6000000,
    'eager' => [
        ['width' => 300, 'height' => 300, 'crop' => 'pad'],
        ['width' => 160, 'height' => 100, 'crop' => 'crop', 'gravity' => 'south']],
    'eager_async' => true, ]);
    $url = $uploadedFileUrl->getSecurePath();
    dd($url);
    $file = $ira['video'];
    $url=(new UploadApi())->upload($file,['folder'=>'emotions','resource_type'=>'video','chunk_size'

    Ira::create(
        [
            "Tema"=>$request->Tema,
            "descripcion"=>$request->descripcion,

```

Fig.24: Método para crear emociones.

```

public function show(Ira $ira)
{
    //
    return $this->sendResponse(message: 'Emotion details', result: [
        'iras' => new IraResource($ira)
    ]);
}

```

Fig. 25: Método para seleccionar una emoción específica.

```

public function test_actualizar_modulo()
{
    $user = User::factory()->make(['role_id' => 1]);
    $modulo = [
        "Tema" => "Ira Biodanza",
        "descripcion" => "Prueba de Modulo",
    ];
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/ira/%u/update',1), $modulo);
    $test_request->assertStatus(200);
}

```

Fig. 26: Prueba unitaria de actualizar emoción.

```

PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\GeneralTest
✓ actualizar modulo

Tests: 1 passed
Time: 3.53s

```

Fig. 27: Resultado de prueba unitaria de actualizar emoción.

## Implementar *endpoints* para la gestión de la lista de reproducción de cada módulo emocional

Para gestionar los contenidos de la lista de reproducción, se ha implementado los siguientes métodos de desarrollo.

- Método visualizar cada contenido audiovisual (ver **Fig. 28**).
- Método actualizar cada contenido audiovisual específico (ver **Fig. 29**).
- Método crear canciones para las listas de reproducción (ver **Fig. 30**).
- Método eliminar canción de la lista de reproducción (ver **Fig. 31**).

La **Fig. 32** y **Fig. 33** muestran los resultados de la prueba unitaria de actualizar canciones de las listas de reproducción. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el **ANEXO II** de este documento.

```
public function index()
{
    //
    $banner=MusicFour::all();
    return $this->sendResponse(message: 'Music Four list generated successfully', result: [
        'musics' => MusicaFourResource::collection($banner),
    ]);
}
```

**Fig. 28:** Método para visualizar canciones.

```
public function update(Request $request, MusicFour $musicfour)
{
    //
    $request->validate([
        'tema' => ['required', 'string', 'min:3', 'max:45'],
        'genero' => ['required', 'string', 'min:3', 'max:200'],
        'descripcion' => ['required', 'string', 'min:3', 'max:500'],
        'duracion' => ['required', 'numeric', 'digits:10'],
    ]);

    $list1= $request -> validate([
        'audio' => ['required', 'file', 'mimes:mp3', 'max:100000000000'],
    ]);
    $list2=$request->validate([
        'imagen'=>['required', 'image', 'mimes:jpg,png,jpeg', 'max:10000'],
    ]);
    $file = $list1['audio'];
    $file2=$list2['imagen'];
    $uploadedFileUrl = Cloudinary::uploadFile($file->getRealPath(),['folder'=>'listFour']);
    $url = $uploadedFileUrl->getSecurePath();
    $uploadedFileUrl1=Cloudinary::upload($file2->getRealPath(),['folder'=>'AudiosFour']);
    $url1=$uploadedFileUrl1->getSecurePath();
    $musicfour->update([
        'tema'=>$request->tema,
        'genero'=>$request->genero,
        'descripcion'=>$request->descripcion,
        'duracion'=>$request->duracion,
        'imagen'=>$url1,
        'audio'=>$url,
    ]);
    return $this->sendResponse("MusicFour update succesfully", 200);
}
```

**Fig. 29:** Método para subir canciones.

```

public function store(Request $request)
{
    //
    $request->validate([
        'tema' => ['required', 'string', 'min:3', 'max:45'],
        'genero' => ['required', 'string', 'min:3', 'max:200'],
        'descripcion' => ['required', 'string', 'min:3', 'max:500'],
        'duracion' => ['required', 'numeric', 'digits:10'],
    ]);

    $list1= $request -> validate([
        'audio' => ['required','file','mimes:mp3','max:100000000000'],
    ]);
    $list2=$request->validate([
        'imagen'=>['required','image','mimes:jpg,png,jpeg', 'max:10000'],
    ]);
    $file = $list1['audio'];
    $file2=$list2['imagen'];
    $uploadedFileUrl = Cloudinary::uploadFile($file->getRealPath(),['folder'=>'listFour']);
    $url = $uploadedFileUrl->getSecurePath();
    $uploadedFileUrl1=Cloudinary::upload($file2->getRealPath(),['folder'=>'AudiosFour']);
    $url1=$uploadedFileUrl1->getSecurePath();
    MusicFour::create(
        [
            "tema"=>$request->tema,
            "genero"=>$request->genero,
            "descripcion"=>$request->descripcion,
            "duracion"=>$request->duracion,
            "imagen"=>$url1,
            "audio"=>$url,
        ]
    );
}

```

Fig. 30: Método para editar una canción.

```

public function destroy(MusicFour $musicfour)
{
    //
    $musicfour->delete();
    return $this->sendResponse("MusicOne delete succesfully", 200);
}

```

Fig. 31: Método para eliminar una canción.

```

public function test_actualizar_lista_de_reproduccion()
{
    $user = User::factory()->make(['role_id' => 1]);
    $modulo = [
        "tema" => "Ira Biodanza- Testeo",
        "genero" => "Ralph Cornejo",
        "descripcion" => "Testeo de Lista de Reproduccion",
        "duracion" => "300",
    ];
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/musicOne/%u/update',1), $modulo);
    $test_request->assertStatus(200);
}

```

Fig. 32: Prueba unitaria de actualizar lista de reproducción.

```

PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

 PASS Tests\Unit\GeneralTest
 ✓ actualizar lista de reproduccion

Tests: 1 passed
Time: 2.89s

```

Fig. 33: Resultado de prueba unitaria de lista de reproducción.

### 3.4 *Sprint 3*. Desarrollo de *endpoints* para funcionalidades de interacción de Usuarios Administrador y Cliente

El *Sprint 3* contiene los siguientes resultados:

- Implementar *endpoints* para comentarios y sugerencias.
- Implementar *endpoints* para eventos promocionales de la Escuela de Biodanza.
- Implementar *endpoints* para reservación de eventos.

#### Implementar *endpoints* para comentarios y sugerencias

Se han implementado métodos que permiten visualizar (todos los comentarios realizados por los usuarios clientes), crear (comentarios y una calificación acerca de la optimo en el uso de la AlphaO. Ver **Fig. 34**) y eliminar comentarios.

Cabe mencionar que todos los métodos se encuentran documentados en *notion* y pueden visualizarse en el *link* que se presenta en el **ANEXO III**.

La **Fig. 35** y **Fig. 36** muestran los resultados de la prueba unitaria de crear comentarios. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el **ANEXO II** de este documento.

```
public function store(Request $request)
{
    //
    $comments= $request -> validate([
        'comentario' => ['required', 'string', 'min:3', 'max:455'],
        'calificacion' => ['required', 'numeric'],
    ]);
    $user=Auth::user();
    $comments= new Comentarios($request->all());
    $user->comentario()->save($comments);

    return $this->sendResponse('Comment created succesfully',204);
}
```

**Fig. 34:** Método para crear un comentario con su calificación.

```
public function test_crear_comentario()
{
    $user = User::where('id', 9)->first();
    $comentario = [
        "comentario" => "Comentario Testeo",
        "calificacion" => "5",
    ];
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/comments/comment-create'), $comentario)
    $test_request->assertStatus(200);
}
```

**Fig. 35:** Prueba unitaria de crear comentarios o sugerencias.

```
PASS Tests\Unit\GeneralTest
✓ crear comentario

Tests: 2 warnings, 1 passed
Time: 2.36s
```

Fig. 36: Resultado de prueba unitaria de crear comentarios o sugerencias.

## Implementar *endpoints* para eventos promocionales de la Escuela de Biodanza

La gestión de eventos es realizada por parte del usuario administrador. Para ello se tienen: Método visualizar lista de eventos generados por el usuario administrador (ver Fig. 37), método crear eventos por medio de un título, descripción, evento, contacto, cupos e imagen (ver Fig. 38), método visualizar evento seleccionado con los respectivos clientes que se registraran por medio de la aplicación móvil (ver Fig. 39), método actualizar eventos (ver Fig. 40) y, método eliminar eventos (ver Fig.41).

La Fig.42 y Fig.43 muestran los resultados de la prueba unitaria de actualizar eventos. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el ANEXO II de este documento.

```
public function index()
{
    //
    $evento=Evento::all();
    return $this->sendResponse(message: 'Event list generated successfully', result: [
        'eventos' => EventoResource::collection($evento),
    ]);
}
```

Fig. 37: Método para visualizar los eventos generados por el administrador.

```
public function store(Request $request)
{
    //
    $mytime=Carbon::now();
    $eventod=new Evento();
    $request->validate([
        'titulo' => ['required', 'string', 'min:3', 'max:45'],
        'descripcion' => ['required', 'string', 'min:3', 'max:600'],
        'evento' => ['required', 'date'],
        'contacto' => ['required', 'numeric', 'digits:10'],
        'cupos' => ['required', 'numeric', 'digits:10'],
    ]);
    $evento= $request -> validate([
        'imagen' => ['required', 'image', 'mimes:jpg,png,jpeg', 'max:10000'],
    ]);
    $file = $evento['imagen'];
    $uploadedFileUrl = Cloudinary::upload($file->getRealPath(),['folder'=>'evento']);
    $url = $uploadedFileUrl->getSecurePath();

    if($eventod->evento<$mytime){
        return $this->sendResponse(message: 'You event must is in present');
    }

    Evento::create([
        'titulo'=>$request->titulo,
        'imagen'=>$url,
        'descripcion'=>$request->descripcion,
        'evento'=>$request->evento,
        'contacto'=>$request->contacto,
        'cupos'=>$request->cupos
    ]);
}
```

Fig. 38: Método para crear el evento generado por el administrador.

```

public function show(Evento $evento)
{
    //
    $reserva=Reserva::where('eventos_id',$evento->id)->get();
    return $this->sendResponse(message: 'Reserva-Event details', result: [
        'eventos' => new EventoResource($evento),
        'reservaciones'=> VercomentarioResource::collection($reserva)
    ]);
}

```

**Fig. 39:** Método para visualizar el evento específico con sus reservaciones.

```

public function update(Request $request, Evento $eventoup)
{
    //
    $request->validate([
        'titulo' => ['required', 'string', 'min:3', 'max:45'],
        'descripcion' => ['required', 'string', 'min:3', 'max:600'],
        'evento' => ['required', 'date'],
        'contacto' => ['required', 'numeric', 'digits:10'],
        'cupos' => ['required', 'numeric', 'digits:10'],
    ]);
    $evento= $request -> validate([
        'imagen' => ['required', 'image', 'mimes:jpg,png,jpeg', 'max:10000'],
    ]);
    $file = $evento['imagen'];
    $uploadedFileUrl = Cloudinary::upload($file->getRealPath(),['folder'=>'evento']);
    $url = $uploadedFileUrl->getSecurePath();

    $eventoup->update([
        "titulo"=>$request->titulo,
        "imagen"=>$url,
        "descripcion"=>$request->descripcion,
        "evento"=>$request->evento,
        "contacto"=>$request->contacto,
        "cupos"=>$request->cupos
    ]);
    return $this->sendResponse('Event update succesfully',204);
}

```

**Fig. 40:** Método para actualizar un evento específico por el administrador.

```

public function destroy(Evento $evento)
{
    //
    $evento->delete();
    return $this->sendResponse("Event delete succesfully", 200);
}

```

**Fig.41:** Método para eliminar un evento específico por el administrador.

```

public function test_actualizar_evento()
{
    $user = User::factory()->make(['role_id' => 1]);
    $evento = [
        "titulo" => "Publicidad Testeo",
        "descripcion" => "Testeo de Publicidad",
        "evento" => "2023-02-07 22:50",
        "contacto" => "0995102824",
        "cupos" => "12",
    ];
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/events/eventupdate/%u',1), $evento);
    $test_request->assertStatus(200);
}

```

**Fig.42:** Prueba unitaria de actualizar un evento.

```
PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\GeneralTest
✓ actualizar evento

Tests: 1 passed
Time: 2.99s
```

Fig.43: Resultado de prueba unitaria de actualizar evento.

### Implementar *endpoints* para reservación de eventos

Para generar reservas acordes a los eventos establecidos se ha desarrollado los siguientes métodos: Método visualizar los eventos registrados por el usuario administrador (ver Fig. 44), método registrar una reservación acorde al evento seleccionado (ver Fig. 45), método visualizar reservaciones realizadas por el cliente (ver Fig. 46), método visualizar el evento seleccionado (ver Fig. 47) y, método cancelar reservación por parte del cliente (ver Fig. 48).

La Fig. 49 y Fig. 50 muestran los resultados de la prueba unitaria de crear reservación. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el ANEXO II de este documento.

```
public function index()
{
    //
    $evento=Evento::all();
    return $this->sendResponse(message: 'Event list generated successfully', result: [
        'eventos' => EventoResource::collection($evento),
    ]);
}
```

Fig. 44: Método para visualizar los eventos disponibles para su reservación.

```
public function indexuser()
{
    //
    $user=Auth::user();
    $reservas=Reserva::where('user_id',$user->id)->get();
    if(!$reservas->first()){
        return $this->sendResponse(message: 'The client not have a reservation');
    }

    return $this->sendResponse(message: 'Reserv list generated successfully', result: [
        'reservas' => ReservaResource::collection($reservas),
    ]);
}
```

Fig. 45: Método para visualizar las reservaciones del usuario cliente.

```

public function store(Request $request ,Evento $evento)
{
    //
    if($evento->cupos==0){
        return $this->sendResponse(message: 'No exist reservs for this event');
    }

    $user=Auth::user();
    $reservacion=new Reserva();
    $num=$evento->cupos;
    $evento->cupos=$evento->cupos-1;
    $reservacion->numero=$num;
    $evento->save();
    $reservacion->eventos_id=$evento->id;
    $user->reserva()->save($reservacion);
    return $this->sendResponse(message: 'Event list generated successfully ');
}

```

**Fig. 46:** Método para generar una reservación por el usuario cliente.

```

public function show(Evento $evento)
{
    //
    return $this->sendResponse(message: 'Reserva-Event details', result: [
        'eventos' => new EventoResource($evento),
    ]);
}

```

**Fig. 47:** Método para visualizar el evento para su reservación.

```

public function destroy(Reserva $reserva)
{
    $evento=Evento::where('id',$reserva->eventos_id)->first();
    if($evento->cupos==0){
        return $this->sendResponse(message: 'No exist reservs for this event');
    }

    $user=Auth::user();
    if(!$user->reserva->first()){
        return $this->sendResponse(message: 'This user not have reservs');
    }

    if($reserva->user_id!=$user->id){
        return $this->sendResponse(message: 'This reservs not have you');
    }

    $evento->cupos=$evento->cupos+1;
    $evento->save();
    $reserva->delete();
    return $this->sendResponse("Reserv delete succesfully", 200);
}

```

**Fig. 48:** Método para cancelar la reservación solicitada.

```

public function test_crear_reserva()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/reservas/reserva-create/%u',1));
    $test_request->assertStatus(200);
}

```

**Fig. 49:** Prueba unitaria de crear reservas.

```
PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\AdministradorTest
✓ crear contactanos
✓ crear banner
✓ crear emoción
✓ crear lista reproduccion
✓ crear publicidad
✓ crear comentario
✓ crear evento
✓ crear reserva
```

Fig. 50: Resultado de la prueba unitaria de crear reservas.

### 3.5 *Sprint 4. Desarrollo de endpoints funcionalidades extras*

El *Sprint 4* contiene los siguientes resultados:

- Implementar *endpoints* para la gestión de publicidades.
- Implementar *endpoints* para habilitar y deshabilitar usuarios clientes.

#### Implementar *endpoints* para la gestión de publicidades

La gestión de publicaciones cuenta con: un método crear que permite la creación de publicidades por medio de un título, descripción, evento e imagen (ver Fig. 51), un método visualizar que muestra todas las publicidades creadas (ver Fig. 52), un método seleccionar publicidad en específico por parte del cliente (ver Fig. 53), un método actualizar que permite modificar la información de la publicidad seleccionada (ver Fig. 54) y, un método eliminar publicidad seleccionada (ver Fig. 55).

La Fig. 56 y Fig. 57 muestran los resultados de la prueba unitaria de actualizar publicidades. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el ANEXO II de este documento.

```
public function index()
{
    //
    $publicidad=Publicidad::all();
    return $this->sendResponse(message: 'Publish list generated successfully', result: [
        'publ' => PublicidadResource::collection($publicidad),
    ]);
}
```

Fig. 51: Método para visualizar las publicidades.

```

public function store(Request $request)
{
    //
    $request->validate([
        'titulo' => ['required', 'string', 'min:3', 'max:45'],
        'descripcion' => ['required', 'string', 'min:3', 'max:600'],
        'evento' => ['required', 'date'],
    ]);

    $publicidad= $request -> validate([
        'imagen' => ['required', 'image', 'mimes:jpg,png,jpeg', 'max:10000'],
    ]);
    $file = $publicidad['imagen'];
    $uploadedFileUrl = Cloudinary::upload($file->getRealPath(),['folder'=>'publicidad']);
    $url = $uploadedFileUrl->getSecurePath();

    Publicidad::create([
        "titulo"=>$request->titulo,
        "imagen"=>$url,
        "descripcion"=>$request->descripcion,
        "evento"=>$request->evento
    ]);
    return $this->sendResponse('Publicidad created succesfully',204);
}

```

Fig. 52: Método para crear la publicidad.

```

public function show(Publicidad $publicidad)
{
    //
    return $this->sendResponse(message: 'Publish details', result: [
        'publ' => new PublicidadResource($publicidad)
    ]);
}

```

Fig. 53: Método para seleccionar la publicidad.

```

public function update(Request $request, Publicidad $publicidad)
{
    //
    $request->validate([
        'titulo' => ['required', 'string', 'min:3', 'max:45'],
        'descripcion' => ['required', 'string', 'min:3', 'max:600'],
        'evento' => ['required', 'date'],
    ]);

    $publi= $request -> validate([
        'imagen' => ['required', 'image', 'mimes:jpg,png,jpeg', 'max:10000'],
    ]);
    $file = $publi['imagen'];
    $uploadedFileUrl = Cloudinary::upload($file->getRealPath(),['folder'=>'publicidad']);
    $url = $uploadedFileUrl->getSecurePath();

    $publicidad->update([
        "titulo"=>$request->titulo,
        "imagen"=>$url,
        "descripcion"=>$request->descripcion,
        "evento"=>$request->evento
    ]);
    return $this->sendResponse('Publicidad update succesfully',204);
}

```

Fig. 54: Método para actualizar la publicidad seleccionada.

```

public function destroy(Publicidad $publicidad)
{
    //
    $publicidad->delete();
    return $this->sendResponse("Publish delete succesfully", 200);
}

```

Fig. 55: Método para eliminar una publicidad seleccionada.

```

public function test_actualizar_publicidad()
{
    $user = User::factory()->make(['role_id' => 1]);
    $publicidad = [
        "titulo" => "Publicidad Testeo",
        "descripcion" => "Testeo de Publicidad",
        "evento" => "2023-02-07 22:50",
    ];
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/publicidad/%u/update',1), $publicidad);
    $test_request->assertStatus(200);
}

```

Fig. 56: Prueba unitaria de actualizar publicidades.

```

PS D:\Tesis\alphaomega> php artisan test
Warning: TTY mode is not supported on Windows platform.

PASS Tests\Unit\GeneralTest
✓ actualizar publicidad

Tests: 1 passed
Time: 2.88s

```

Fig. 57: Resultado de prueba unitaria de actualizar publicidades.

### Implementar *endpoints* para habilitar y deshabilitar usuarios clientes

Para habilitar y deshabilitar las cuentas de los usuarios clientes se ha implementado: un método para visualizar todos los usuarios clientes (ver Fig. 58), otro método que permite habilitar y deshabilitar las cuentas de los usuarios clientes (ver Fig. 59) y un método de envío de notificaciones por correo electrónico sobre la habilitación (ver Fig. 60) y deshabilitación(ver Fig. 61) de cuentas.

La Fig. 62 y Fig. 63 muestran los resultados de la prueba unitaria de deshabilitar cuenta de usuario cliente. Las restantes pruebas unitarias de los *endpoints* de esta tarea se encuentran en el ANEXO II de este documento.

```

public function index()
{
    //
    $role = Role::where('slug', 'client')->first();
    // Obtener los usuarios en base a la relación
    $users = $role->users;
    // Invoca el controlador padre para la respuesta json
    return $this->sendResponse(message: 'User list generated successfully', result: [
        'users' => UserResource::collection($users),
    ]);
}

```

Fig. 58: Método para visualizar los usuarios de rol cliente.

```

public function destroy(User $user ,Request $request)
{
    // Obtiene el estado del usuario
    $user_state = $user->state;
    // Crear un mensaje en base al estado del usuario
    $message = $user_state ? 'inactivated' : 'activated';
    // Cambiar el estado
    $user->state = !$user_state;
    // Guardar en la BDD
    $user->save();
    // Validamos si existen solicitudes para este tecnico
    if ($user->state == 0) {
        // Validacion de datos de entrada
        $request->validate([
            'observacion' => ['required', 'string', 'min:5', 'max:500']
        ]);
        $userad = Auth::user();
        // Llamamos la notificacion

        $this->DesactivateUser($user,$request->observacion, $userad->email, $userad->personal_phone)
    }
}

```

**Fig. 59:** Método para habilitar y deshabilitar al usuario cliente.

```

// Funcion para notificar el tecnico que ha sido
private function DesactivateUser(User $user,string $observacion, string $email_admin, int $number_ad
{
    $user->notify(
        new ClienteUso(
            user_name: $user->getFullName(),
            role_name:$user->role->name,
            observacion:$observacion,
            email_admin: $email_admin,
            number_admin: $number_admin
        )
    );
}

```

**Fig. 60:** Método para enviar la notificación de correo de deshabilitación.

```

private function ActivateUser(User $user,string $observacion, string $email_admin, int $number_admin
{
    $user->notify(
        new ClienteUsoActivate(
            user_name: $user->getFullName(),
            role_name:$user->role->name,
            observacion:$observacion,
            email_admin: $email_admin,
            number_admin: $number_admin
        )
    );
}

```

**Fig. 61:** Método de notificación de correo para habilitación de cuentas.

```

public function test_crear_inactividad_actividad()
{
    $user = User::where('id', 1)->first();
    $usuario = [
        "observacion" => "Activado o Inactivado Testeo",
    ];
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/clientes-admin/%u/destroy',6), $usuario);
    $test_request->assertStatus(200);
}

```

**Fig. 62:** Prueba unitaria de habilitación y deshabituación de clientes.

```
PASS Tests\Unit\GeneralTest
✓ crear inactividad actividad

Tests: 2 warnings, 1 passed
Time: 3.48s
```

Fig. 63: Resultado de prueba unitaria de habilitación y deshabilitación de clientes.

### 3.6 Sprint 5. Pruebas y Despliegue

El *Sprint 5* contiene los siguientes resultados:

- Pruebas de rendimiento
- Pruebas de carga
- Despliegue del *backend*

#### Pruebas de rendimientos

Las pruebas de rendimiento determinan si el software cumple con los requisitos de velocidad, accesibilidad, escalabilidad y estabilidad mediante diferentes acciones de trabajo que se ha de someter una aplicación al cliente [31]. La **Fig. 64** muestra el resultado obtenido por medio de *Page Speed*, una herramienta de Google que ayuda a determinar la velocidad de carga y el rendimiento de un sitio además de proveer sugerencias y recomendaciones para saber cómo y dónde optimizar el sitio. En el **ANEXO II** se puede visualizar los resultados del índice de rendimiento de la parte web.

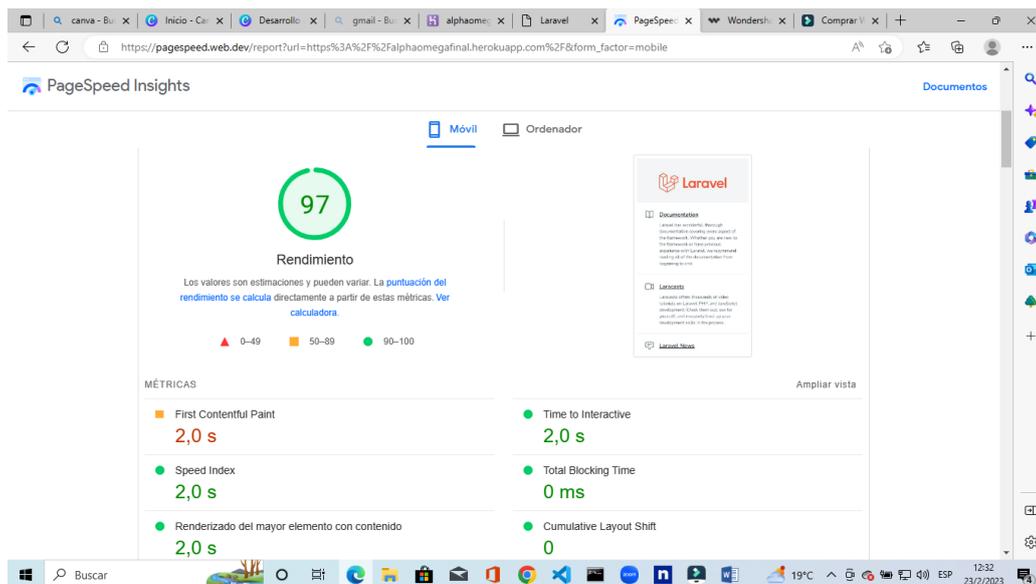
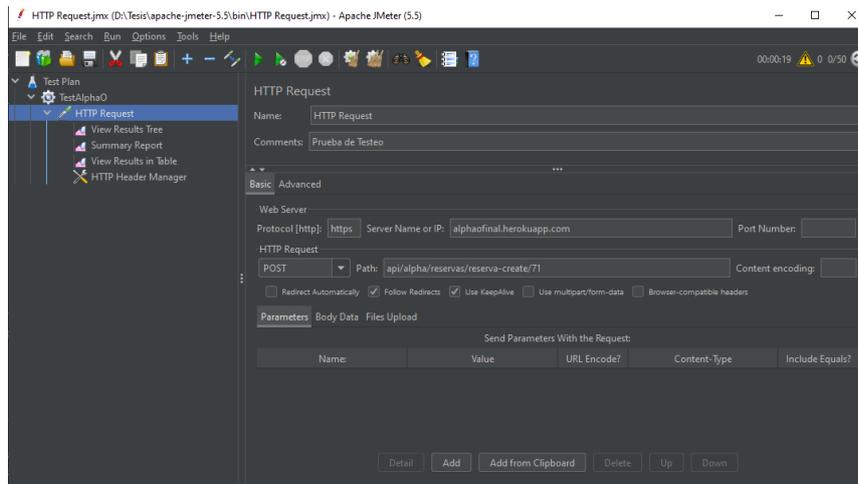


Fig. 64: Prueba de rendimiento de PageSpeed – Prueba parte móvil

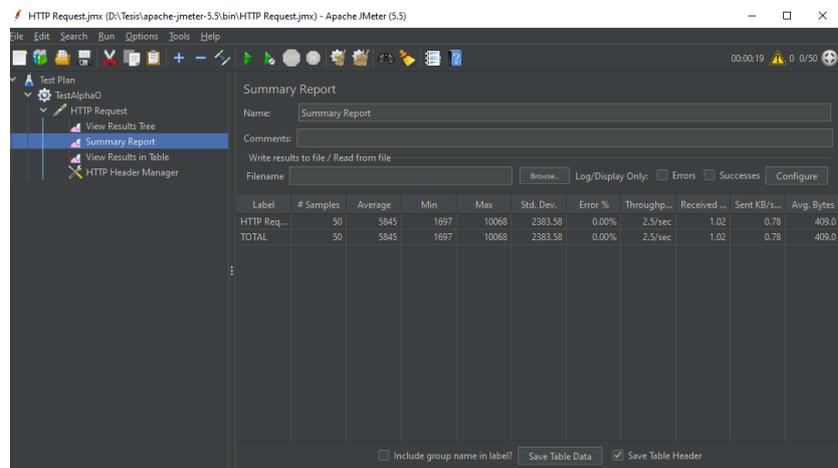
## Pruebas de carga

La prueba de carga es una técnica para determinar el índice de rendimiento y el número de solicitudes que puede soportar una API o aplicativo web. Su objetivo es determinar si un sistema puede permanecer estable con determinado número de peticiones [32].

La **Fig. 65** y **Fig. 66** muestran un componente de carga con el *endpoint* “crear una reserva” donde se visualiza el número de solicitudes POST que un grupo de 50 usuarios. Las pruebas restantes se encuentran en el **ANEXO II** de este documento.



**Fig. 65:** Prueba de estrés de JMeter con el *endpoint* de reservas.



**Fig. 66:** Resultado de carga que presenta el *endpoint* reserva.

Los resultados que se han obtenido de las pruebas de carga realizadas dan un resultado positivo acorde a la lógica del negocio de la Escuela de Biodanza puesto que 50 peticiones son respondidas de manera satisfactoria en un tiempo de alrededor de 10 segundos.

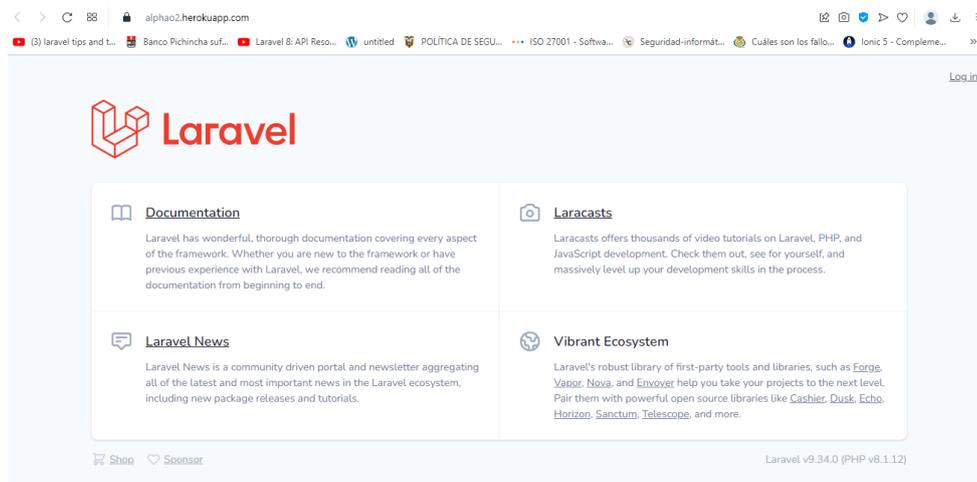
## Despliegue del *backend*

Una vez culminado con la implementación de todos los *endpoints* solicitados y de haber obtenido resultados satisfactorios en las pruebas se realiza el despliegue del *backend* en Heroku, como se muestra en la **Fig. 67**.

Para acceder al componente desarrollado se lo puede realizar por el siguiente URL:

<https://alphaofin.herokuapp.com>

El detalle del despliegue se encuentra en el **ANEXO IV** del presente documento.



**Fig. 67:** *Backend* desplegado en la plataforma de Heroku.

## 4 CONCLUSIONES

En esta sección se presenta las conclusiones que se han registrado en el transcurso para la finalización de este proyecto de Integración Curricular.

- Se ha implementado de manera satisfactoria cada uno de los métodos que permiten cumplir con todos los requerimientos del *backend*.
- El uso de la metodología Scrum, ha ayudado con la implementación del componente *backend* de manera exitosa y en los tiempos establecidos, debido al control en cada *sprint*, reuniones junto con todo el equipo, entrega de avances parciales y pruebas.
- El uso de MVC ha permitido un desarrollo de *backend* de forma ordenada dando la oportunidad de realizar nuevas implementaciones o crear múltiples eventos que el usuario pueda consumir de manera simple.
- El uso de una Base de Datos SQL ha ayudado notablemente no solo con el almacenamiento sino con la organización de múltiples recursos necesarios para el desarrollo de la API Restful, debido a que ha facilitado el acceso a tablas, columnas y datos por medio de la conexión con Laravel por medio del uso simple de métodos descritos en la documentación oficial.
- Las pruebas han permitido garantizar la calidad del *backend* entre su funcionalidad, accesibilidad y comprensión de uso.
- Heroku ha facilitado el despliegue del *backend*, permitiendo un consumo sencillo de los *endpoints* según la necesidad del usuario (parte web o parte móvil).

## 5 RECOMENDACIONES

En este apartado se presentan las recomendaciones que se han obtenido en el transcurso de este trabajo de integración curricular.

- Es de gran importancia visualizar la versión que dispone Laravel, por el motivo que algunos softwares de despliegue solicitan versiones de *frameworks* con un nivel más actualizado de desarrollo.
- Se debe analizar que la versión que se trabaja en Laravel sea compatible con la versión de Composer, caso contrario presenta interrupciones y errores al generar un nuevo proyecto.
- Se recomienda que se cambie el plan de servicio de Cloudinary cuando se presente un aumento de ingreso de usuarios, eventos, emociones, banners y publicidades, por la mayor demanda y capacidad de almacenamiento que soporta la plataforma.
- Se recomienda que, si se desea implementar nuevas funcionalidades se entienda primeramente la lógica de cada método implementado para prevenir errores en la actualización de la aplicación.
- Se recomienda generar la documentación respectiva al implementar nuevas actualizaciones, esto con el fin de entender la lógica y objetivo de implementación.
- Heroku ha sido actualizado provocando que las peticiones se vean comprometidas con un “Error 503 Service Request Interrupted H18”, como alternativa el *backend* tienen su funcionalidad intacta en un servidor local. Por ello se recomienda tener en cuenta errores con prefijo H18 y H20 (que son externos y no manejables por el desarrollador) para contar con otras alternativas de despliegue.
- Se ha informado de las actualizaciones y consecuencias de Heroku *al Product Owner* donde unos de los problemas de su actualización se dan en el envío de archivos mp4. El *Product Owner* ha dado la aceptación de definir videos quemados en la sección de módulos emocionales por el momento. Se recomienda que para futuras versiones se encuentre una mejor solución.
- Los errores encontrados serán corregidos en una nueva versión de los Aplicativos con su respectivo *backend*.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Pimentel, «El suicidio es la principal causa de muerte de adolescentes en Ecuador,» El Universo, 28 Marzo 2021. [En línea]. Available: <https://www.eluniverso.com/noticias/seguridad/el-suicidio-es-la-principal-causa-de-muerte-de-adolescentes-en-ecuador-nota/>. [Último acceso: 12 Febrero 2023].
- [2] C. d. D. d. I. OCDE, «Making Development Happen,» de *Impacto social del COVID-19 en Ecuador*, Quito, Centro de Desarrollo de la OCDE, 2020, p. 23.
- [3] S. Diez, «Qué es la biodanza y cómo bailar para recuperar la alegría de vivir,» Cuerpo Mente, 2 Marzo 2021. [En línea]. Available: [https://www.cuerpomente.com/salud-natural/terapias-naturales/que-es-biodanza-y-como-bailar-para-recuperar-alegria-vivir\\_7899](https://www.cuerpomente.com/salud-natural/terapias-naturales/que-es-biodanza-y-como-bailar-para-recuperar-alegria-vivir_7899). [Último acceso: 28 Febrero 2023].
- [4] Publico, «Todo lo que debes saber de la Biodanza,» Publico, 1 Julio 2021. [En línea]. Available: <https://www.publico.es/psicologia-y-mente/todo-lo-que-debes-saber-de-la-biodanza/#:~:text=Beneficios%20de%20la%20Biodanza&text=Ayuda%20a%20controlar%20de%20forma,estados%20emocionales%20de%20otras%20personas..> [Último acceso: 28 Febrero 2023].
- [5] Startechup, «LOS FUNDAMENTOS DEL DESARROLLO DEL BACKEND,» Startechup, 8 Julio 2022. [En línea]. Available: <https://www.startechup.com/es/blog/back-end-development/#:~:text=El%20desarrollo%20del%20back-end,suelen%20constituir%20el%20back-end..> [Último acceso: 17 Febrero 2023].
- [6] B. ApiMarket, «API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos,» BBVA ApiMarket, 23 Marzo 2016. [En línea]. Available: <https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/>. [Último acceso: 12 Febrero 2023].
- [7] M. W. Docs, «Trabajando con JSON,» MDN Web Docs, 12 Febrero 2023. [En línea]. Available: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>. [Último acceso: 12 Febrero 2023].
- [8] N. Chapaval, «Qué es Frontend y Backend: diferencias y características - Platzi,» Platzi, 12 Febrero 2018. [En línea]. Available: <https://platzi.com/blog/que-es-frontend-y-backend/>. [Último acceso: 12 Febrero 2023].
- [9] Laravel, «Getting Started,» Laravel, 12 Febrero 2023. [En línea]. Available: <https://laravel.com/docs/9.x/installation>. [Último acceso: 12 Febrero 2023].
- [10] AntonioFernandez, «OPTIMIZA LA CARGA DE IMÁGENES DE TU APLICACIÓN WEB, APP O PÁGINA CON CLOUDINARY,» AntonioFernandez.com, 09 Junio 2016. [En línea]. Available: <https://antoniofernandez.com/optimizar-carga-de-imagenes-cloudinary/>. [Último acceso: 12 Febrero 2023].

- [11] Cloundinary, «Cloundinary,» Cloundinary, 12 Febrero 2023. [En línea]. Available: <https://cloundinary.com/documentation>. [Último acceso: 12 Febrero 2023].
- [12] Importancia, «Importancia de la Metodología,» Importancia, 17 Febrero 2023. [En línea]. Available: <https://www.importancia.org/metodologia.php>. [Último acceso: 17 Febrero 2023].
- [13] I. R. Salvador, «Estudio de caso: características, objetivos y metodología,» Psicología y Mente, 8 Marzo 2018. [En línea]. Available: <https://psicologiaymente.com/psicologia/estudio-de-caso>. [Último acceso: 17 Febrero 2023].
- [14] S. Universidades, «Metodologías de desarrollo de software: ¿qué son?,» Santander, 21 Diciembre 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 12 Diciembre 2023].
- [15] R. Hat, «¿Qué es la metodología ágil?,» Red Hat, 19 Julio 2022. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology>. [Último acceso: 12 Febrero 2023].
- [16] J. S. HURTADO, «Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla,» IEBSchool, 3 Diciembre 2021. [En línea]. Available: <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>. [Último acceso: 12 Febrero 2022].
- [17] C. HARRIS, «Artefactos del scrum ágil,» Atlassian, 17 Febrero 2023. [En línea]. Available: <https://www.atlassian.com/es/agile/scrum/artifacts#:~:text=¿Qué%20son%20los%20artefactos%20del,tareas%20realizadas%20durante%20el%20proyecto..> [Último acceso: 17 Febrero 2023].
- [18] Northware, «Requerimientos en el desarrollo de software y aplicaciones,» Northware, 26 Mayo 2022. [En línea]. Available: <https://www.northware.mx/blog/requerimientos-en-el-desarrollo-de-software-y-aplicaciones/#:~:text=requerimientos%20de%20software%20son%20simplemente,las%20necesidades%20de%20los%20usuarios..> [Último acceso: 17 Febrero 2023].
- [19] S. Vergara, «Historias de usuario con ejemplos y plantilla,» Itdo, 10 Agosto 2021. [En línea]. Available: <https://www.itdo.com/blog/que-son-las-historias-de-usuario/>. [Último acceso: 17 Febrero 2023].
- [20] Programacionymas, «Scrum: ¿Qué es el Product Backlog?,» Programacionymas, 17 Febrero 2023. [En línea]. Available: <https://programacionymas.com/blog/scrum-product-backlog>. [Último acceso: 17 Febrero 2023].
- [21] M. Garcia, «¿Qué es el SPRINT BACKLOG?,» Ittude, 17 Julio 2020. [En línea]. Available: <https://ittude.com.ar/b/scrum/que-es-el-sprint-backlog/>. [Último acceso: 17 Febrero 2023].

- [22] KeepCoding, «¿Qué es una Arquitectura de Software?,» KeepCoding, 1 Abril 2022. [En línea]. Available: <https://keepcoding.io/blog/que-es-arquitectura-software/>. [Último acceso: 17 Febrero 2023].
- [23] U. d. Alicante, «Modelo vista controlador (MVC),» Universidad de Alicante, 17 Febrero 2023. [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 17 Febrero 2023].
- [24] M. Garcia, «¿QUE ES XAMPP Y COMO PUEDO USARLO?,» Nettix, 20 Mayo 2020. [En línea]. Available: <https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo/>. [Último acceso: 17 Febrero 2023].
- [25] Anonimo, «7 Tips esenciales de Laravel Auth,» Dev, 3 Agosto 2021. [En línea]. Available: <https://dev.to/themustachecode/7-tips-esenciales-de-laravel-auth-hnj#:~:text=Laravel%20Auth%20es%20el%20kit,usuarios%20en%20nuestras%20aplicaciones%20webs..> [Último acceso: 12 Febrero 2023].
- [26] Yeswelab, «Alwaysdata,» Yeswelab, 17 Febrero 2023. [En línea]. Available: <https://app.yeswelab.com/herramienta/alwaysdata/r/recbRrlozNRWylgSw>. [Último acceso: 17 Febrero 2023].
- [27] R. Celis, «¿Qué es Heroku? Cómo funciona la plataforma y para qué sirve,» Platzi, 14 Febrero 2018. [En línea]. Available: <https://platzi.com/blog/que-es-heroku/>. [Último acceso: 17 Febrero 2023].
- [28] R. A. Vera, «Qué es Laravel: Características y ventajas,» OpenWebinar, 31 Marzo 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/>. [Último acceso: 17 Febrero 2023].
- [29] A. fernandez, «OPTIMIZA LA CARGA DE IMÁGENES DE TU APLICACIÓN WEB, APP O PÁGINA CON CLOUDINARY,» Antonio Fernandez, 9 Junio 2016. [En línea]. Available: <https://antoniofernandez.com/optimizar-carga-de-imagenes-cloudinary/>. [Último acceso: 17 Febrero 2023].
- [30] D. Palacios, «Almacenamiento De Archivos – Documentación de Laravel 6,» Styde, 29 Mayo 2020. [En línea]. Available: <https://styde.net/laravel-6-doc-almacenamiento-de-archivos/>. [Último acceso: 17 Febrero 2023].
- [31] A. Rios, «Pruebas de rendimiento: cuándo y cómo,» Encora, 13 Enero 2021. [En línea]. Available: <https://www.encora.com/es/blog/pruebas-de-rendimiento-cuando-y-como>. [Último acceso: 17 Febrero 2023].
- [32] K6, «Stress testing,» K6, 17 Febrero 2023. [En línea]. Available: [https://k6.io/docs/es/tipos-de-prueba/stress-testing/#:~:text=Las%20pruebas%20de%20estrés%20\(stress,sistema%20bajo%20una%20carga%20pesada..](https://k6.io/docs/es/tipos-de-prueba/stress-testing/#:~:text=Las%20pruebas%20de%20estrés%20(stress,sistema%20bajo%20una%20carga%20pesada..) [Último acceso: 17 Febrero 2023].
- [33] GCFGlobal, «¿Qué son las aplicaciones web?,» GCFGlobal, 12 Febrero 2023. [En línea]. Available: <https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/>. [Último acceso: 12 Febrero 2023].

- [34] Mailify, «¿Qué es un dominio y un subdominio?,» Mailify, 12 Febrero 2023. [En línea]. Available: <https://www.mailify.com/es/help/entregabilidad/nombres-de-dominios/que-es-un-dominio>. [Último acceso: 12 Febrero 2023].
- [35] M. A. Alvare, «Qué es React. Por qué usar React,» Desarrollo Web, 25 Febrero 2019. [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>. [Último acceso: 12 Febrero 2023].
- [36] J. A. Blanes, «¿Qué es React Native?,» Deloitte, 12 Febrero 2023. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html>. [Último acceso: 12 Febrero 2023].
- [37] F. Flores, «Qué es Visual Studio Code y qué ventajas ofrece,» OpenWebinar, 22 Julio 2022. [En línea]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. [Último acceso: 17 Febrero 2023].
- [38] KeepCoding, «¿Qué son las pruebas unitarias de software?,» KeepCoding, 1 Agosto 2022. [En línea]. Available: [https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/#Que\\_son\\_las\\_pruebas\\_unitarias\\_de\\_software](https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/#Que_son_las_pruebas_unitarias_de_software). [Último acceso: 17 Febrero 2023].

## 7 ANEXOS

Como siguiente punto se presenta los siguientes **ANEXOS** que contiene información detallada y relevante de la documentación y se compone de los siguientes puntos.

- **ANEXO I.** Certificado de Originalidad
- **ANEXO II.** Manual Técnico.
- **ANEXO III.** Manual de Usuario.
- **ANEXO IV.** Manual de Instalación.
- **ANEXO V.** Justificaciones

# ANEXO I

A continuación, se presenta el certificado que la directora del proyecto de titulación ha emitido, donde se evidencia el resultado que se ha obtenido por la herramienta anti-plagio Turnitin.



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

## CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 02 de marzo de 2023

De mi consideración:

Yo, IVONNE FERNANDA MALDONADO SOLIZ, en calidad de Director del Trabajo de Integración Curricular titulado DESARROLLO DE UN BACKEND asociado al DESARROLLO DE UN SISTEMA PARA LA ESCUELA DE BIODANZA SRT EN QUITO-PUEMBO elaborado por el estudiante JORGE DANIEL ORTIZ GALLEGOS de la carrera en TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones (sin anexos), 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,



---

Ivonne Maldonado  
Docente Ocasional a Tiempo Completo  
ESFOT

## ANEXO II

### Recopilación de Requerimientos

La **TABLA V** muestra el fichero de requerimientos levantados como parte inicial del proyecto.

**TABLA V:** Recopilación de requerimientos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
<b>Backend</b>	<b>RR001</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Iniciar y cerrar sesión.</li><li>• Recuperar contraseña.</li></ul>
	<b>RR002</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Gestionar perfil personal.</li></ul>
	<b>RR003</b>	Como usuario administrador necesita diferentes generar <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Gestionar información de la página principal.</li></ul>
	<b>RR004</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Actualizar contenido de cada módulo emocional.</li></ul>
	<b>RR005</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Gestionar información de lista de reproducción de cada módulo emocional.</li></ul>
	<b>RR006</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Desarrollar comentarios y/o sugerencias.</li></ul>
	<b>RR007</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Realizar reservas.</li></ul>
	<b>RR008</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Gestionar publicidad.</li></ul>
	<b>RR09</b>	Como usuario administrador necesita generar diferentes <i>endpoints</i> para: <ul style="list-style-type: none"><li>• Habilitar y Deshabilitar usuarios.</li></ul>

## Historias de Usuario

Después de la obtención de la tabla de requerimientos, se dispone con la formulación de las historias de Usuarios donde se detalla todas las especificaciones de desarrollo desde la **TABLA VI** a la **TABLA XV**: Historia de usuario Nro.11.

**TABLA VI:** Historia de usuario Nro.1.

HISTORIA DE USUARIO	
<b>Identificador (ID):</b> HU001	<b>Usuario:</b> Administrador- Cliente
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para inicio de sesión, cierre de sesión y cambio de contraseña.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 3	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> genera varios <i>endpoints</i> para que el perfil administrador pueda: <ul style="list-style-type: none"><li>• Iniciar sesión</li><li>• Cerrar sesión</li><li>• Recuperar contraseña</li></ul>	
<b>Observación:</b> El <i>backend</i> valida la información de entrada que en este caso es usuario y contraseña, caso contrario muestra un mensaje de error. Además, para la recuperación de contraseña el <i>backend</i> envía un correo electrónico para recuperar la contraseña del usuario. El usuario administrador y cliente tendrá que llenar de forma correcta tanto el email como la contraseña caso contrario se mostrará una alerta de datos incorrectos o campos vacíos.	

**TABLA VII:** Historia de usuario Nro.2.

HISTORIA DE USUARIO	
<b>Identificador (ID):</b> HU002	<b>Usuario:</b> Administrador – Cliente
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para visualizar y editar perfil	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 3	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> por medio del rol administrador y cliente podrá realizar gestionar la información del perfil: <ul style="list-style-type: none"><li>• Nombre y Apellido</li><li>• Teléfono Celular</li><li>• Teléfono de Casa</li><li>• Fecha de nacimiento</li><li>• Correo Electrónico</li></ul>	

<ul style="list-style-type: none"> <li>• Avatar</li> <li>• Nombre de Usuario</li> </ul>
<p><b>Observación:</b> Como usuario administrador y cliente el usuario una vez dentro de la aplicación podrá realizar el cambio de información de sus datos personales y será realizado por un formulario.</p>

**TABLA VIII:** Historia de usuario Nro.3.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU003	<b>Usuario:</b> Invitado
<b>Nombre Historia:</b> Implementar un <i>endpoints</i> para el registro del usuario	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Alta
<b>Iteración Asignada:</b> 1	
<b>Responsable (es):</b> Jorge Ortiz	
<p><b>Descripción:</b> El componente <i>backend</i> por medio del rol de invitado podrá acceder al contenido que ofrece la aplicación AlphaO por medio de un formulario de registro con los siguientes datos:</p> <ul style="list-style-type: none"> <li>• E-mail.</li> <li>• Contraseña.</li> <li>• Nombre.</li> <li>• Apellido</li> <li>• Numero celular</li> <li>• Número de teléfono</li> <li>• Usuario</li> <li>• Dirección</li> </ul>	
<p><b>Observación:</b> EL usuario invitado al momento de registrarse y completar el formulario de registro tendrá el acceso de consumir las funcionalidades de la app AlphaO.</p>	

**TABLA IX:** Historia de usuario Nro.4.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU004	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para la gestión de contenido de la página principal	

<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 2	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> por medio del rol administrador podrá modificar información que se presente en la página de inicio de la aplicación.	
<b>Observación:</b> El usuario administrador es el único que podrá gestionar el contenido que se presente en la página principal con nueva información.	

**TABLA X:** Historia de usuario Nro.6.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU006	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para la actualización de información de cada módulo emocional de Biodanza.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Alta
<b>Iteración Asignada:</b> 2	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> por medio del rol administrador puede modificar o visualizar los nuevos elementos de cada módulo de Biodanza: <ul style="list-style-type: none"> <li>• Actualizar el tema del modulo</li> <li>• Visualizar su descripción acerca del módulo.</li> <li>• Reproducir un video tutorial de Biodanza.</li> </ul>	
<b>Observación:</b> El usuario con rol administrador podrá realizar varios métodos de actualización y visualización acerca de la información de cada módulo una vez dentro del inicio de sesión.	

**TABLA XI:** Historia de usuario Nro.7.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU007	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para la gestión de la lista de reproducción de cada módulo emocional.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta

<b>Iteración Asignada:</b> 2
<b>Responsable (es):</b> Jorge Ortiz
<b>Descripción:</b> El componente <i>backend</i> por medio del rol administrador podrá realizar las funcionalidades de un CRUD en los elementos de multimedia de cada reproductor de audio.
<b>Observación:</b> El usuario con rol administrador gestiona todos los elementos de cada lista de reproducción, donde ejecuta los métodos CRUD en cada contenido musical.

**TABLA XII:** Historia de usuario Nro.8.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU008	<b>Usuario:</b> Administrador-Cliente
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para el desarrollo de comentarios o sugerencias.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 3	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> por medio del rol administrador crea un espacio de opinión o sugerencia que nos brindara el usuario cliente acerca de la experiencia con AlphaO.	
<b>Observación:</b> El usuario administrador crea un <i>endpoint</i> que ha de ser utilizado por el usuario cliente para enviar sus opiniones o sugerencias.	

**TABLA XIII:** Historia de usuario Nro.9.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU009	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para el desarrollo de eventos y reservaciones que ofrece la Escuela de Biodanza	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 3	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> por medio del perfil administrador realiza un CRUD de eventos que será visualizado por el usuario cliente, posteriormente el usuario	

cliente realizar una reservación acorde a los elementos agregados por el usuario administrador.

**Observación:** El usuario administrador crea un CRUD de eventos para la visualización del usuario Cliente y un endpoint especial para el desarrollo de una reservación.

**TABLA XIV:** Historia de usuario Nro.10.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU010	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para la gestión de publicidades en la Aplicación de AlphaO.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 3	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> por medio del perfil administrador crea un CRUD de publicidades informativas que ofrece la Escuela de Biodanza.	
<b>Observación:</b> El usuario administrador realiza un CRUD de varias publicidades que será una fuente de información acerca de los temas de Biodanza que visualizar el usuario Cliente.	

**TABLA XV:** Historia de usuario Nro.11.

<b>HISTORIA DE USUARIO</b>	
<b>Identificador (ID):</b> HU011	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Implementar <i>endpoints</i> para la habilitación y deshabilitación de los usuarios clientes.	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Iteración Asignada:</b> 3	
<b>Responsable (es):</b> Jorge Ortiz	
<b>Descripción:</b> El componente <i>backend</i> por medio del perfil administrador crea varios <i>endpoints</i> para la visualización de los usuarios clientes para deshabilitar sus cuentas por motivos de mal uso de la aplicación o incumplimientos de reservaciones realizadas.	
<b>Observación:</b> El usuario administrador realiza varios <i>endpoints</i> que controlara el desempeño de los usuarios clientes entre el uso de comentarios o la asignación de	

reservaciones solicitadas por el cliente.

### **Product Backlog**

La **TABLA XVI** muestra las necesidades del *Product Owner* priorizadas.

**TABLA XVI:** *Product Backlog.*

<b>ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>-<i>BACKEND</i></b>				
<b>ID – HU</b>	<b>HISTORIA DE USUARIO</b>	<b>ITERACIÓN</b>	<b>ESTADO</b>	<b>PRIORIDAD</b>
HU001	Implementar <i>endpoints</i> para el inicio de sesión, cierre de sesión y cambio de contraseña	1	Finalizado	Media
HU002	Implementar <i>endpoints</i> para visualizar y editar el perfil de usuario.	1	Finalizado	Media
HU003	Implementar <i>endpoints</i> para el registro de la aplicación AlphaO con el rol asignado Cliente	1	Finalizado	Media
HU004	Implementar <i>endpoints</i> para la gestión de la página principal.	1	Finalizado	Media
HU005	Implementar <i>endpoints</i> para la actualización de información de cada módulo emocional	2	Finalizado	Media
HU006	Implementar <i>endpoints</i> para la gestión de la lista de reproducción de cada módulo emocional.	3	Finalizado	Media
HU007	Implementar <i>endpoints</i> para el desarrollo de comentarios y sugerencias.	3	Finalizado	Media
HU008	Implementar <i>endpoints</i> para el desarrollo de eventos y reservaciones que ofrece la Escuela de Biodanza.	2	Finalizado	Media
HU009	Implementar <i>endpoints</i> para la gestión de publicidades en la aplicación AlphaO.	2	Finalizado	Media
HU010	Implementar <i>endpoints</i> para la habilitación y deshabilitación de los usuarios clientes.	2	Finalizado	Alta

## ***Sprint Backlog***

La **TABLA XVII** muestra los *Sprints* con cada actividad a cumplir en un tiempo determinado.

**TABLA XVII:** *Sprint Backlog.*

<b>ELABORACIÓN DEL <i>SPRINT BACKLOG</i></b>						
<b>ID-SB</b>	<b>NOMBRE</b>	<b>MÓDULO</b>	<b>ID-HU</b>	<b>HU</b>	<b>TAREAS</b>	<b>TIEMPO ESTIMADO</b>
SB000	Configuración del espacio de desarrollo	-	HU000	Estructura de la Base de Datos de cada entidad.	<ul style="list-style-type: none"> <li>• Recopilación y definición de requerimientos.</li> <li>• Elaboración del Modelo de Datos</li> <li>• Roles de usuarios.</li> </ul>	10H
SB001	Desarrollo de <i>endpoints</i> de funcionalidades básicas de Usuarios Administrador y Cliente	Módulo Login	HU001	Iniciar sesión, cerrar sesión y recuperar contraseña.	<ul style="list-style-type: none"> <li>• Crear un Método para el inicio de sesión</li> <li>• Crear un Método para el cierre de sesión</li> <li>• Crear un Método para el cambio de contraseña</li> </ul>	20H
		Módulo de Gestión de Usuario	HU002	Gestionar su perfil personal	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar perfil de usuario</li> <li>• Implementar un Método para actualizar perfil de usuario</li> </ul>	25 H
		Módulo de Registro un Usuario	HU003	Registro de un Cliente por medio de un Formulario	<ul style="list-style-type: none"> <li>• Implementar un método para el registro de clientes.</li> </ul>	10 H

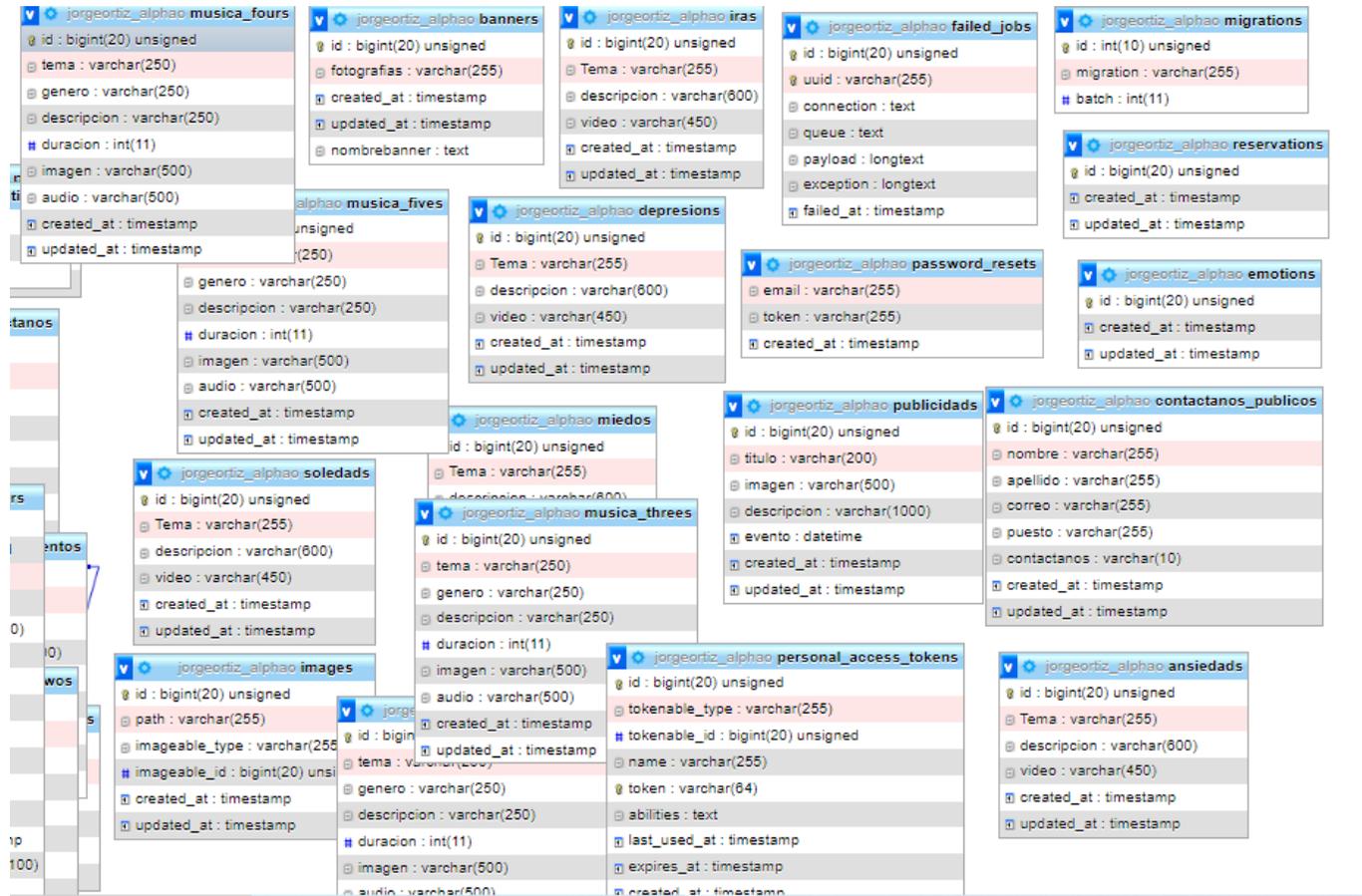
SB002	Desarrollo de <i>endpoints</i> para Usuario Administrador	Módulo de Información principal de la pagina	HU004	Gestionar información de la página principal.	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar publicidades informativas.</li> <li>• Implementar un Método para visualizar contactos de la Escuela de Biodanza.</li> </ul>	25H
		Módulo de Actualizar contenido informativo cada módulo emocional.	HU005	Editar contenido de cada módulo emocional.	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar emociones.</li> <li>• Implementar un Método para actualizar información de emociones.</li> <li>• Implementar un Método para seleccionar una emoción en específico.</li> </ul>	25H
		Módulo de gestión de contenido de lista de reproducción	HU006	Gestionar información de lista de reproducción.	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar canciones.</li> <li>• Implementar un Método para subir canciones.</li> <li>• Implementar un Método para editar una canción.</li> <li>• Implementar un Método para eliminar una canción.</li> </ul>	25H
SB003	Desarrollo de <i>endpoints</i> para funcionalidades	Módulo de comentarios o sugerencias	HU007	Gestionar comentarios y/o sugerencias.	<ul style="list-style-type: none"> <li>• Implementar un Método para crear un comentario con su calificación.</li> </ul>	25H

	de interacción de Usuarios Administrador y Cliente	Módulo de reservas de sesiones de Biodanza	HU008	Gestionar eventos ofertados por la Escuela.	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar los eventos generados por el administrador.</li> <li>• Implementar un Método para crear el evento generado por el administrador.</li> <li>• Implementar un Método para visualizar el evento específico con sus reservaciones.</li> <li>• Implementar un Método para eliminar un evento específico por el administrador.</li> </ul>	25H
		Módulo de Publicidades de Biodanza	HU009	Gestionar reservaciones acorde a los eventos establecidos por la aplicación.	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar los eventos disponibles para su reservación.</li> <li>• Implementar un Método para visualizar las reservaciones del usuario cliente.</li> <li>• Implementar un Método para generar una reservación por el usuario cliente.</li> <li>• Implementar un Método para visualizar el evento para su reservación.</li> <li>• Implementar un Método para cancelar la reservación solicitada.</li> </ul>	25H
SB004	Desarrollo de <i>endpoints</i> funcionalidades extras	Módulo de gestión de publicidades informativas	HU010	Gestionar publicidades informativas acerca del tema de Biodanza.	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar las publicidades.</li> <li>• Implementar un Método para crear la publicidad.</li> <li>• Implementar un Método para seleccionar la publicidad.</li> <li>• Implementar un Método para actualizar la publicidad seleccionada.</li> <li>• Implementar un Método para eliminar una publicidad seleccionada.</li> </ul>	25H

		Habilitación y deshabilitación de los usuarios clientes.	HU011	Gestión de actividad o inactividad de cuentas con el rol Cliente.	<ul style="list-style-type: none"> <li>• Implementar un Método para visualizar los usuarios de rol cliente.</li> <li>• Implementar un Método para habilitar y deshabilitar al usuario cliente.</li> <li>• Implementar un Método para enviar la notificación de correo de deshabilitación.</li> <li>• Implementar un Método de notificación de correo para habilitación de cuentas.</li> </ul>	25H
SB005	Pruebas y Despliegue	----	-----	----	<ul style="list-style-type: none"> <li>• Pruebas de rendimiento</li> <li>• Pruebas de carga</li> <li>• Despliegue de <i>backend</i></li> </ul>	25H

## Modelamiento del esquema de la Base de Datos

En esta sección se comparte la imagen del diseño de la base de datos donde está compuesto la lógica del negocio y que será implementado en el componente *Backend* como se visualiza en la **Fig. 68** y **Fig. 69**.



**Fig. 68:** Modelamiento de Datos Individuales

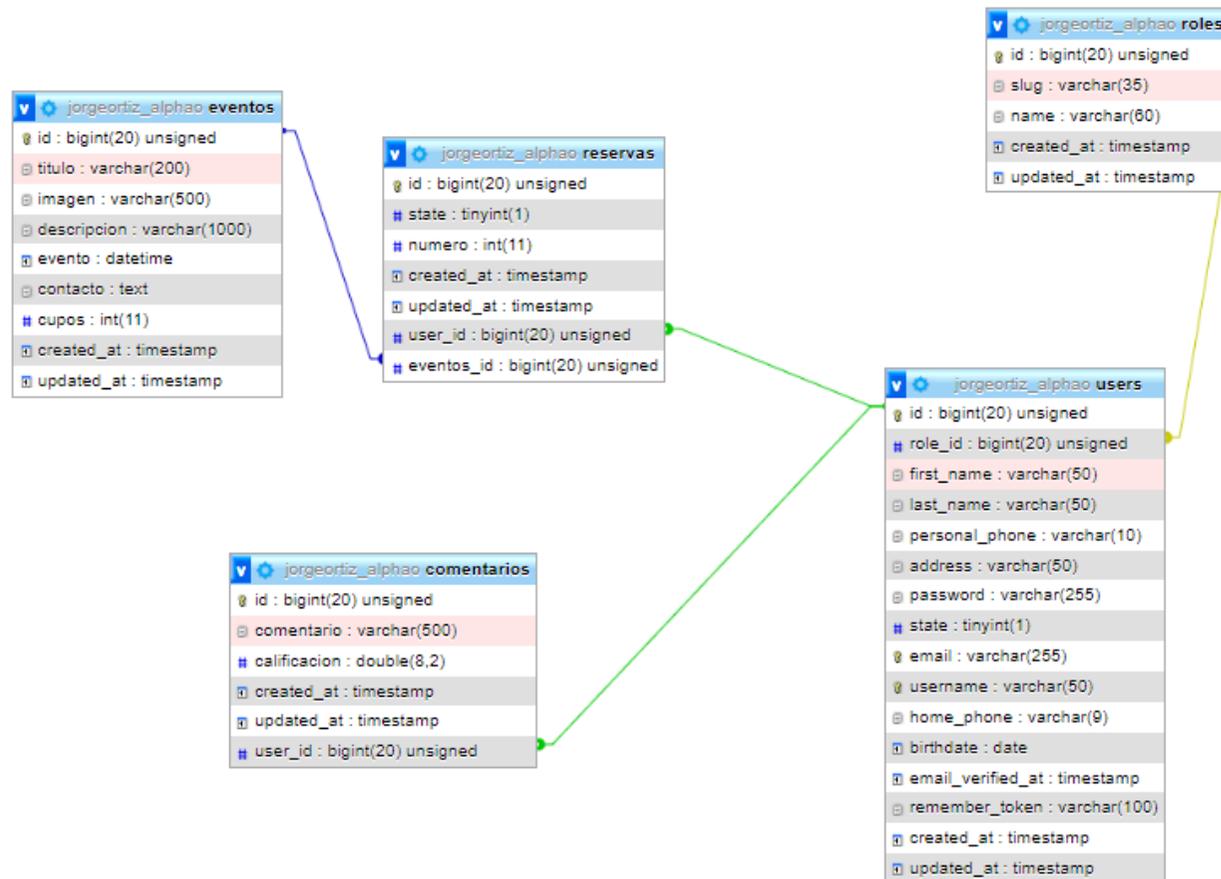
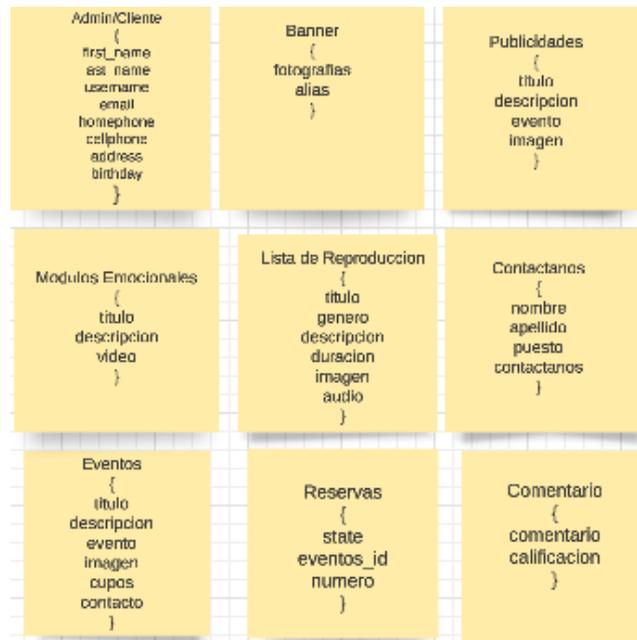


Fig. 69. Modelo de Datos relacional

## Diseño de la Base de datos MySQL y Alwaysdata

La **Fig.70** muestra los 9 elementos que compone la estructura de salida de cada petición, que se presentara en el despliegue de datos con la finalidad de ser claro y entendible el retorno de respuestas de consultas.



**Fig.70: Diseño de la Base de datos relacional.**

## Pruebas

Una vez finalizado el proceso de codificación del componente *backend*, se dispone con el inicio de las Pruebas.

### Pruebas Unitarias

A continuación, las figuras presentes entre la **Fig. 71** hasta la **Fig. 87** se visualiza los bloques de código de cada prueba unitaria con su respectivo resultado de aprobación.

```
public function test_crear_contactanos()
{
    $user = User::where('id', 1)->first();
    $contactos = [
        "nombre" => "Rossey",
        "apellido" => "Gallegos",
        "correo" => "rossey@gmail.com",
        "puesto" => "Subdirectora",
        "contactanos" => "0995102824",
    ];
    $test_request = $this->actingAs($user)->post('/api/alpha/contactos/create', $contactos);
    $test_request->assertStatus(200);
}
```

**Fig. 71: Prueba de crear contáctanos.**

```

public function test_crear_banner()
{
    $user = User::where('id', 1)->first();

    $banner = [
        "nombre" => "foto2",
    ];
    $test_request=$this->actingAs($user)->post('/api/alpha/banner/create',$banner);
    $test_request->assertStatus(200);
}

```

**Fig. 72:** Prueba de crear banner.

```

public function test_crear_emocion()
{
    $user = User::where('id', 1)->first();
    $emocion = [
        "Tema" => "EmocionPrueba",
        "descripcion" => "PruebaUnitaria",
    ];
    $test_request = $this->actingAs($user)->post('/api/alpha/ansiedad/create', $emocion);
    $test_request->assertStatus(200);
}

```

**Fig. 73:** Prueba de crear emoción.

```

public function test_crear_listaReproduccion()
{
    $user = User::where('id', 1)->first();
    $musica = [
        "tema" => "ListaPrueba",
        "genero" => "Prueba",
        "descripcion" => "Prueba",
        "duracion" => "250",
    ];
    $test_request = $this->actingAs($user)->post('/api/alpha/musicFive/create', $musica);
    $test_request->assertStatus(200);
}

```

**Fig. 74:** Prueba de crear lista de reproducción.

```

public function test_crear_publicidad()
{
    $user = User::where('id', 1)->first();
    $publicidad = [
        "titulo" => "Unitaria",
        "descripcion" => "Unitaria",
        "evento" => "2023-03-17",
    ];
    $test_request = $this->actingAs($user)->post('/api/alpha/publicidad/create', $publicidad);
    $test_request->assertStatus(200);
}

```

**Fig. 75:** Prueba de crear publicidad.

```

public function test_crear_comentario()
{
    $user = User::where('id', 1)->first();
    $comentario = [
        "comentario" => "Bonito",
        "calificacion" => "4",
    ];
    $test_request = $this->actingAs($user)->post('/api/alpha/comments/comment-create', $comentario);
    $test_request->assertStatus(200);
}

```

**Fig. 76:** Prueba de crear comentarios.

```

public function test_crear_evento()
{
    $user = User::where('id', 1)->first();
    $emocion = [
        "titulo" => "Test",
        "descripcion" => "Unitaria",
        "evento" => "2023-02-14",
        "contacto" => "0958845034",
        "cupos" => "10",
    ];
    $test_request = $this->actingAs($user)->post('/api/alpha/events/event-create', $emocion);
    $test_request->assertStatus(200);
}

```

**Fig. 77:** Prueba de crear eventos.

```

public function test_habilitar_deshabilitar()
{
    $user = User::where('id', 1)->first();
    $usuario = [
        "observacion" => "Activado o Inactivado Testeo",
    ];
    $test_request = $this->actingAs($user)->post(sprintf('/api/alpha/clientes-admin/%u/destroy',10), $usu);
    $test_request->assertStatus(200);
}

```

**Fig. 78:** Prueba de habilitación y deshabilitación de clientes.

```

public function test_ver_clientes()
{
    $user = User::where('id', 1)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/clientes-admin/users');
    $test_request->assertStatus(200);
}

```

**Fig. 79:** Prueba de visualizar usuarios clientes.

```

public function test_ver_contactanos()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/contactos/');
    $test_request->assertStatus(200);
}

```

**Fig. 80:** Prueba de ver contactános.

```

public function test_ver_banner()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/banner/fotos');
    $test_request->assertStatus(200);
}

```

**Fig. 81:** Prueba de visualizar banners.

```

public function test_ver_emocion()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/ansiedad/');
    $test_request->assertStatus(200);
}

```

**Fig. 82:** Prueba de visualizar emoción.

```

public function test_ver_listaReproduccion()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/musicFive/lista');
    $test_request->assertStatus(200);
}

```

**Fig. 83:** Prueba de visualizar lista de reproducción.

```

public function test_ver_publicidad()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/publicidad/publ');
    $test_request->assertStatus(200);
}

```

**Fig. 84:** Prueba de visualizar publicidades.

```

public function test_ver_comentario()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/comments/vercomment');
    $test_request->assertStatus(200);
}

```

**Fig. 85:** Prueba de visualizar comentarios.

```

public function test_ver_evento()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/events/eventlist');
    $test_request->assertStatus(200);
}

```

**Fig. 86:** Prueba de visualizar eventos.

```

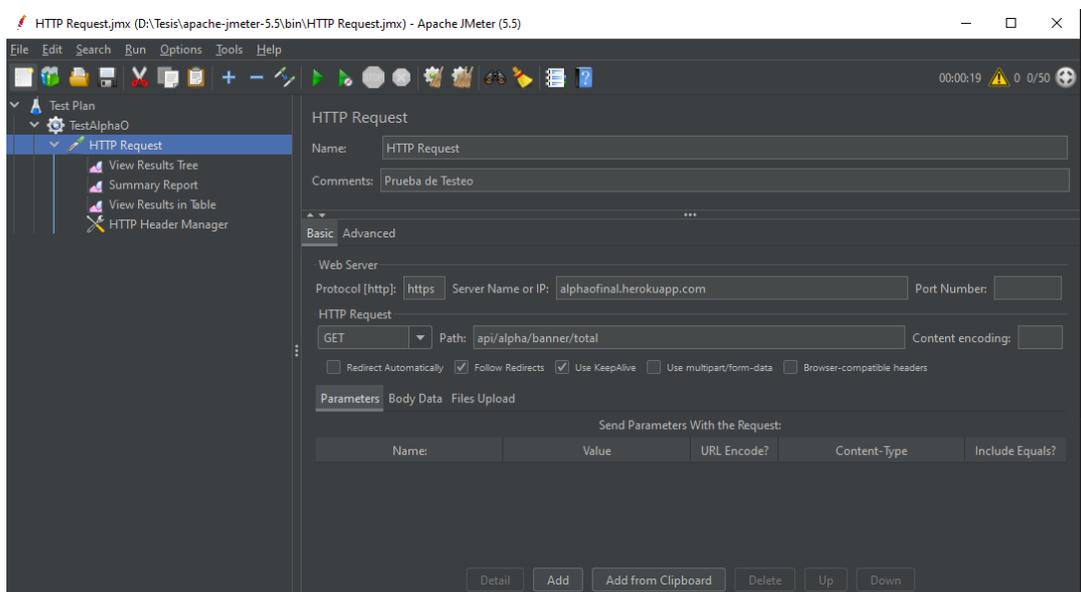
public function test_ver_reserva()
{
    $user = User::where('id', 7)->first();
    $test_request = $this->actingAs($user)->get('/api/alpha/reservas/misreservas');
    $test_request->assertStatus(200);
}

```

**Fig. 87:** Prueba de visualizar reservas.

## Pruebas de carga

A continuación, se muestran las 8 pruebas de carga donde muestra el numero de request que puede solicitar multiples usuarios garantizando la estabilidad que puede mantener la calidad del software del componente *Backend*, desde la **Fig.88** hasta la **Fig. 103**.

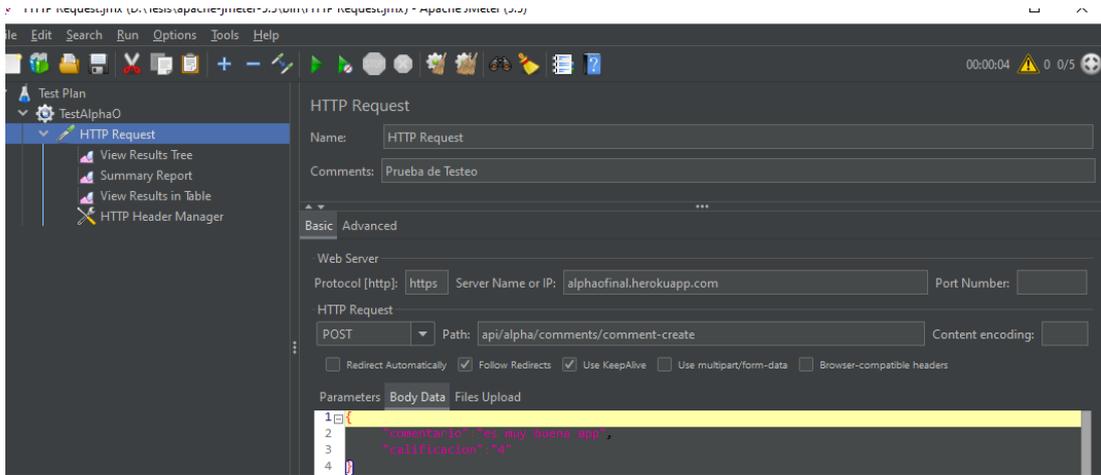


**Fig.88:** Pruebas de carga de visualizar banner.

The screenshot shows the Apache JMeter 5.5 interface displaying the 'Summary Report' for the 'HTTP Request' test. The report includes the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughp...	Received ...	Sent KB/s...	Avg. Bytes
HTTP Req...	50	5574	1900	9636	2303.22	0.00%	2.6/sec	4.26	0.55	1683.0
TOTAL	50	5574	1900	9636	2303.22	0.00%	2.6/sec	4.26	0.55	1683.0

**Fig. 89:** Resultado de las pruebas de carga de visualizar banner.

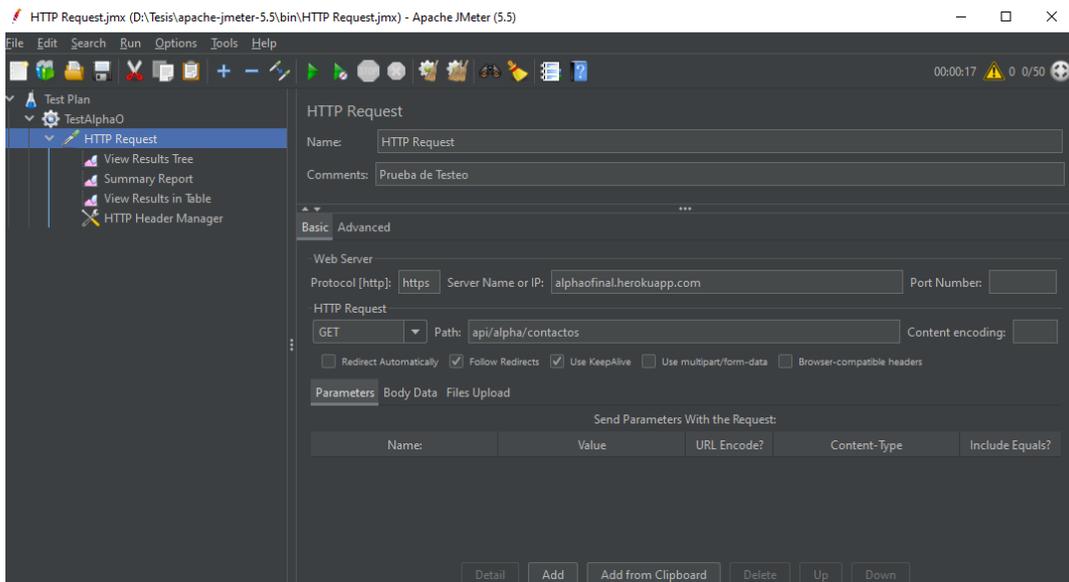


**Fig. 90:** Pruebas de carga de crear comentarios.

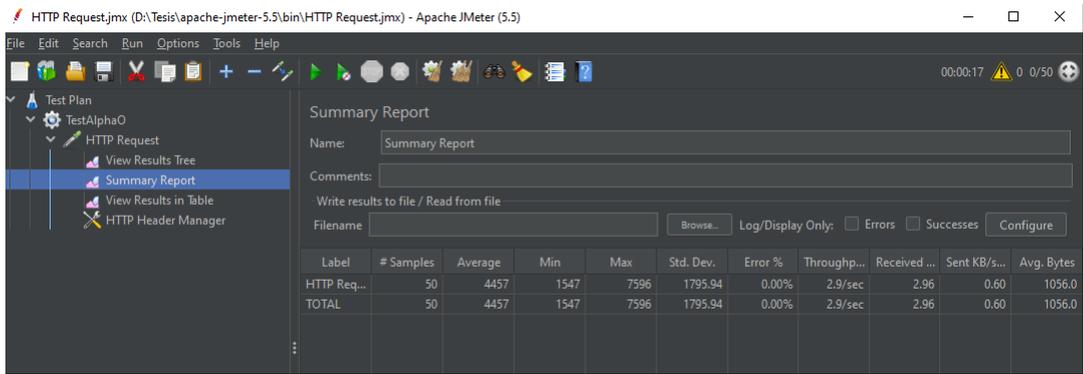
The screenshot shows the 'Summary Report' configuration panel. The 'Name' is 'Summary Report'. Below the configuration, a table displays the test results:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughp...	Received ...	Sent KB/s...	Avg. Bytes
HTTP Req...	5	1978	1638	2240	208.39	0.00%	1.1/sec	0.41	0.44	381.0
TOTAL	5	1978	1638	2240	208.39	0.00%	1.1/sec	0.41	0.44	381.0

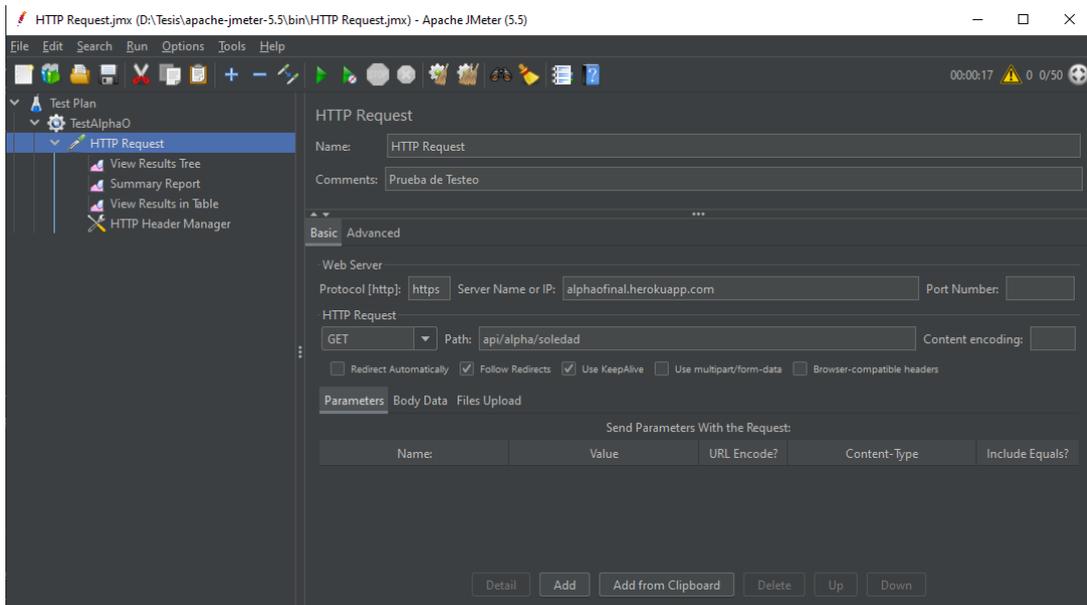
**Fig. 91:** Resultados de la prueba de carga de crear comentarios.



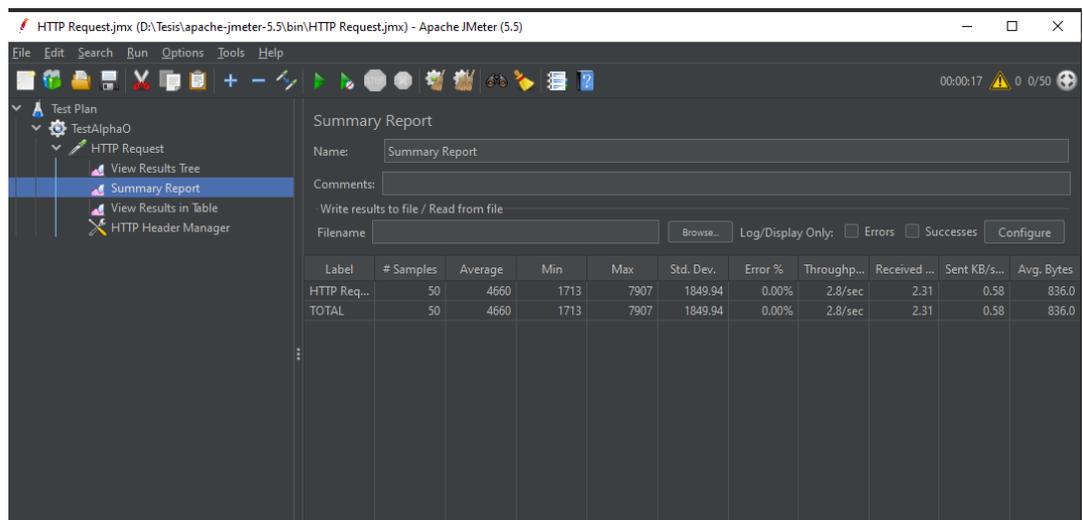
**Fig. 92:** Pruebas de carga de visualizar contactos.



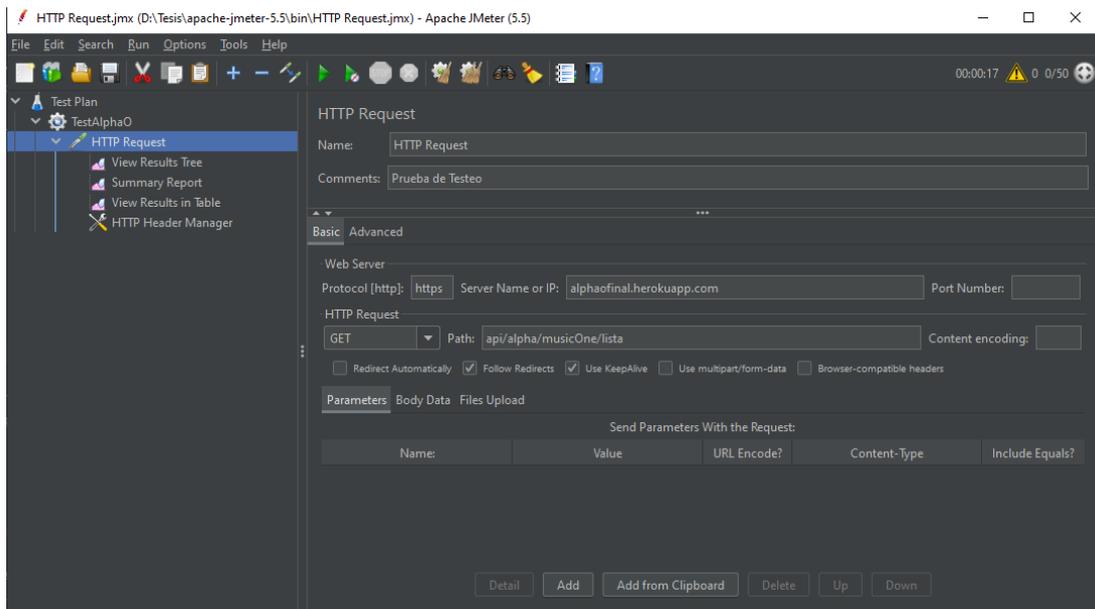
**Fig. 93:** Resultado de pruebas de carga de visualizar contactos.



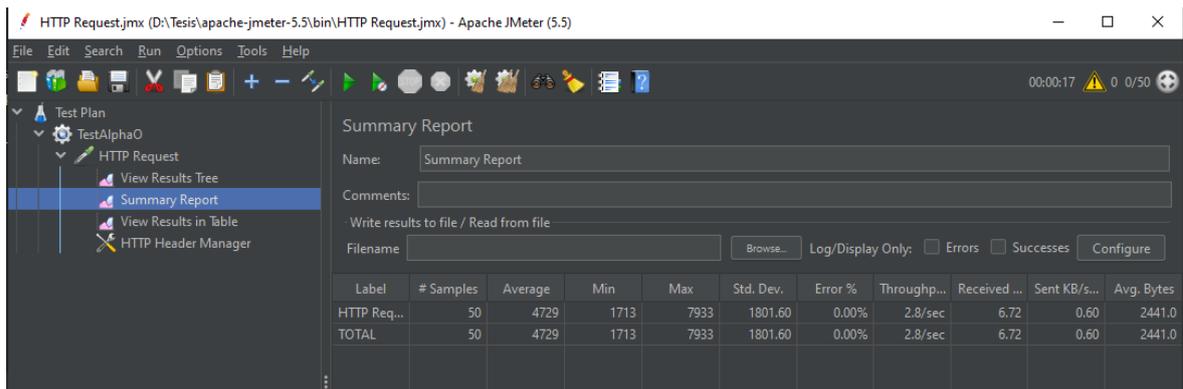
**Fig. 94:** Prueba de carga de visualizar modelo emocional



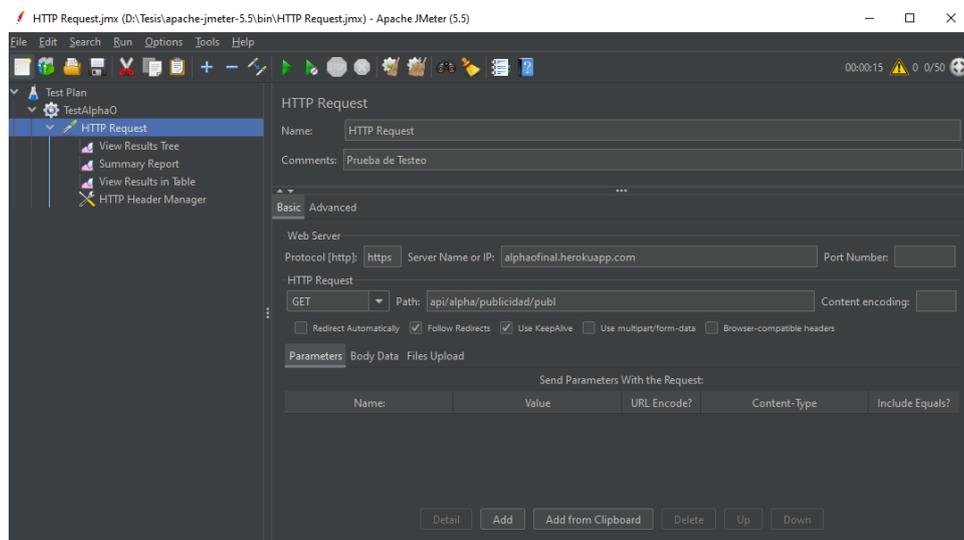
**Fig.95:** Resultado de pruebas de carga de visualizar un modelo emocional.



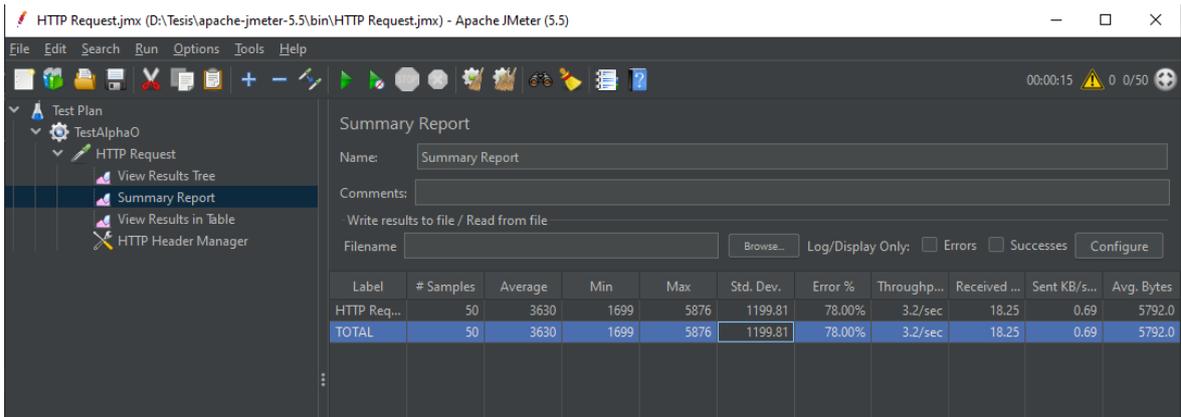
**Fig.96:** Resultado de visualizar lista de reproducción.



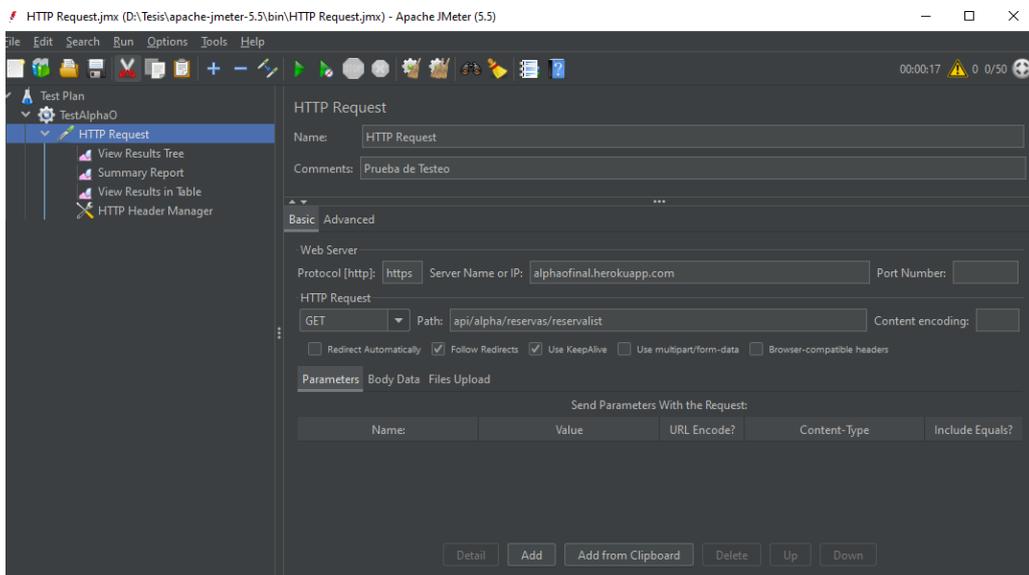
**Fig. 97:** Resultado de pruebas de carga de visualizar lista de reproducción.



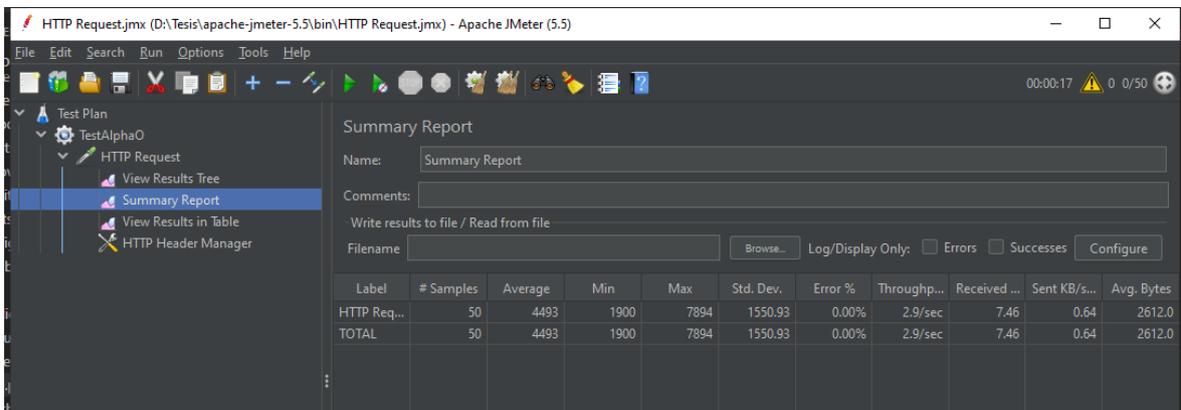
**Fig. 98:** Prueba de carga de visualizar publicidades.



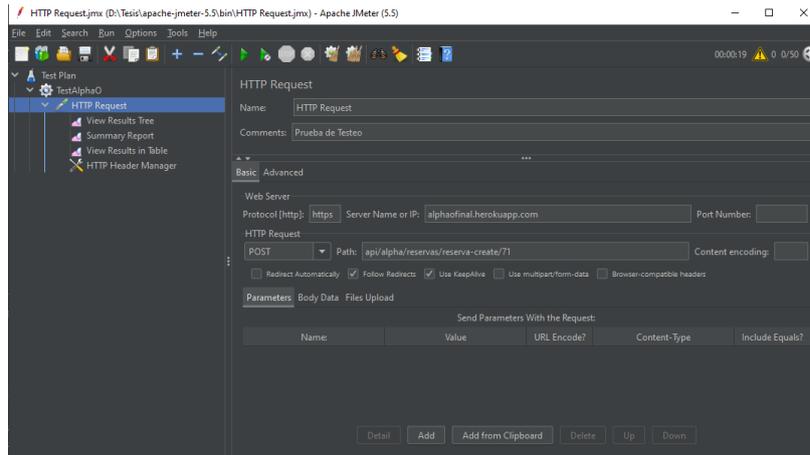
**Fig. 99:** Resultado de pruebas de carga de visualizar contactos.



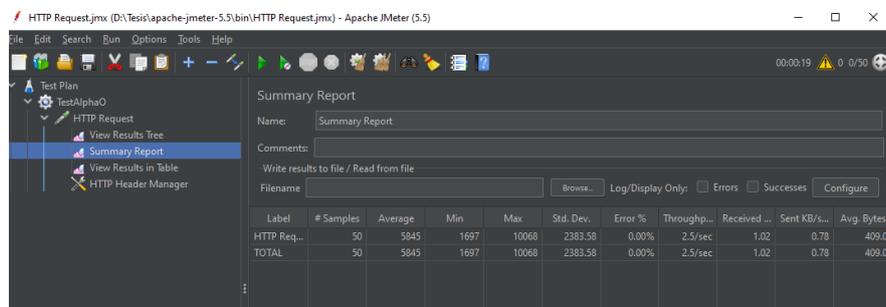
**Fig. 100:** Prueba de carga de visualización de eventos para reservar.



**Fig. 101:** Resultado de pruebas de carga de visualizar eventos para reservar.



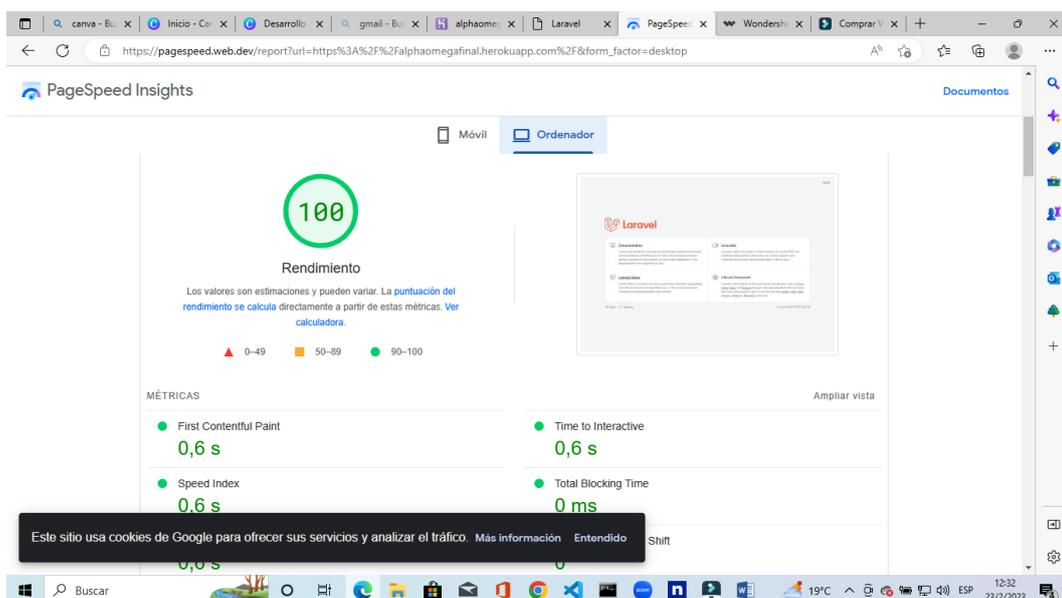
**Fig. 102:** Prueba de carga para realizar una reservación.



**Fig. 103:** Resultado de pruebas de carga de realizar una reservación.

## Pruebas de Rendimiento

Se visualiza en la **Fig. 104** una prueba de rendimiento ante el uso de accesibilidad y funcionalidad de la parte web.



**Fig. 104:** Prueba de rendimiento de PageSpeed - Prueba parte web

## ANEXO III

El Manual de Usuario se encuentra en las siguientes direcciones donde podrá visualizar la documentación y evidencia del correcto funcionamiento de la API *backend* y un video explicativo acerca del proyecto.

Link de Documentación Notion:

<https://smiling-kick-73c.notion.site/Documentaci-n-de-AlphaOmega-Backend-d53b4bc7f6464ee5af2e240db47a6afb>

Link de Video:

[https://youtu.be/db\\_v0Ww2StY](https://youtu.be/db_v0Ww2StY)

Link de Requerimientos:

[Autorización\[1\].docx](#)

## ANEXO IV

A continuación, se presentan el acceso al componente *backend mediante una URL*, así como el acceso al código fuente en el repositorio de GitHub.

### **Acceso para el *Backend***

Para tener acceso al funcionamiento de la API se puede obtener mediante un clic al siguiente enlace.

<https://alphaofin.herokuapp.com/>

### **Repositorio del código fuente del *backend***

EL código generado en la plataforma Github se puede encontrar en el siguiente vinculo donde está almacenado todos los archivos que conforman el componente *backend* de AlphaO:

<https://github.com/LuisOrtiz12/AlphaOFinal.git>

## ANEXO V

### Justificaciones acerca de las actualizaciones de Heroku y sus consecuencias a la Aplicación AlphaO.

En las Fig. 105, Fig. 106, Fig. 107 se presenta las justificaciones de actualizaciones de Heroku que afecta al desempeño de AlphaO con los códigos H18, H20 del error 503 dando a conocer que el error es externo y no por parte del desarrollador.

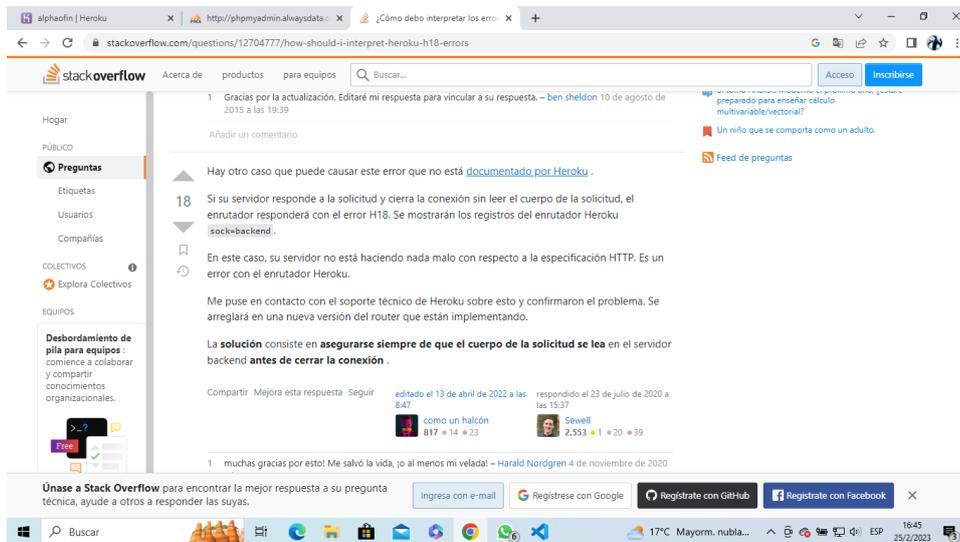


Fig. 105: Definición acerca del error H18 503 de Heroku.

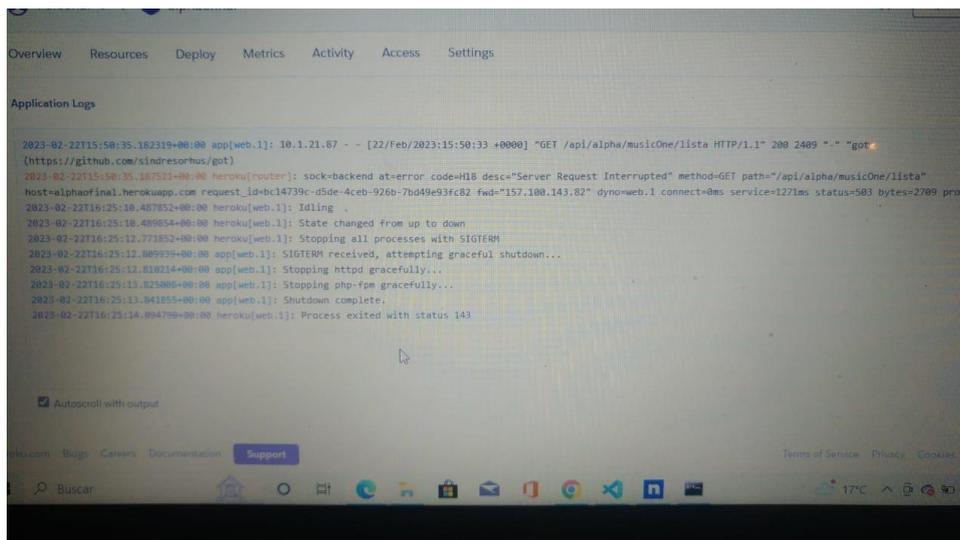


Fig. 106: Evidencia de logs que describe el error de Heroku a AlphaO.



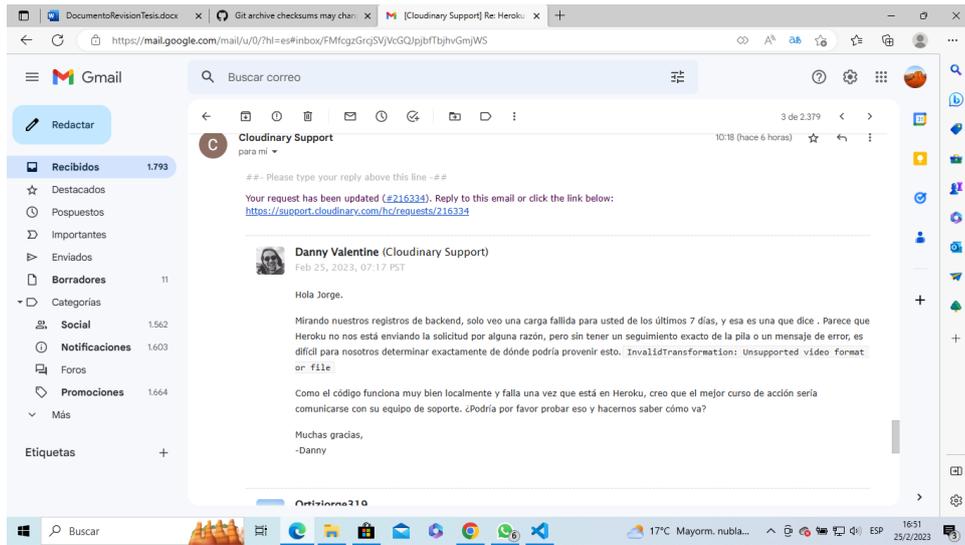


Fig. 109: Respuesta 2 acerca del error a Cloudinary Support