**Manual Técnico**

**Nombre del Proyecto:**

**Elegans**

**Desarrollado por:**

**Jhonny Alexander Gonzalez Torres**

**Fecha de Publicación:**

(23/10/2024)

Contenido

[Introducción 3](#_bookmark0)

[Descripción General del Proyecto 4](#_bookmark1)

[Arquitectura del Sistema 4](#_bookmark2)

[Configuración del Entorno de Desarrollo 4](#_bookmark3)

[Estructura del Proyecto 6](#_bookmark4)

[Detalles Técnicos del Desarrollo 8](#_bookmark5)

[Seguridad 9](#_bookmark6)

[Mantenimiento y Soporte 9](#_bookmark7)

[Contacto y Escalación de Soporte Técnico 9](#_bookmark8)

[Diagrama de actividades para 10](#_bookmark9)

[Diagrama de caso de uso para 14](#_bookmark10)

[Diagrama de modelo entidad relación 15](#_bookmark11)

[Componentes 15](#_bookmark12)

[Esquema de la base de datos 16](#_bookmark13)

[Diagrama de clase 17](#_bookmark14)

## Introducción

### Propósito del Documento

El objetivo de este documento es proporcionar una guía completa sobre el desarrollo, mantenimiento y operación de **Elegans**, una plataforma web diseñada para la Gestión de solicitudes de Música y posteriormente La reproducción de esas solicitudes de Música. Este manual está destinado principalmente a desarrolladores y personal de soporte técnico.

### Descripción del Proyecto

**Elegans** es una plataforma diseñada para amantes y entusiastas de la Música. Nuestro objetivo es proporcionar un espacio donde los usuarios puedan divertirse y solicitar una amplia variedad de música y diversos géneros para todos los oyentes.

## Descripción General del Proyecto

**Elegans** utiliza una combinación de tecnologías modernas para ofrecer una experiencia de usuario rica y eficiente. Los principales lenguajes de programación utilizados son HTML, PHP, CSS, Tailwind CSS y JavaScript vanilla, y se emplean frameworks como PHPMailer para la gestión de correos electrónicos y Bootstrap para el diseño y la responsividad y consumo de api como lo son **Cloudflare Turnstile y Spotify**.

* **Objetivo Principal:** Proporcionar una buena gestión de solicitudes de música y una experiencia endeble a los usuarios sobre la gestión que hacemos de las solicitudes de música.
* **Público Objetivo:** Personas de todas las edades y orígenes interesados en experimentar una vibra nueva de sonidos propuestos por las demás personas.
* **Características Clave:** Amigable con los usuarios, una buena gestión de las solicitudes de música para un buen ambiente sonoro entre los usuarios.

## Arquitectura del Sistema

### Diagrama de Arquitectura

Un diagrama visual que muestra los componentes principales del sistema y sus interacciones, incluyendo servidores web, bases de datos, y servicios externos.

### Componentes Principales

* **Frontend:** Desarrollado con HTML, CSS y JavaScript vanilla, utilizando Bootstrap para la responsividad.
* **Backend:** Implementado en PHP, manejando la lógica de negocio y la interacción con la base de datos.
* **Base de Datos:** MySQL, utilizada para almacenar la información de los cursos, usuarios y comentarios.
* **Servicios Externos:** PHPMailer para la gestión de correos electrónicos, **Cloudflare Turnstile** para la integración de **CAPTCHA** y también **API Spotify para la lógica de búsqueda de canciones y la reproducción de sí mismas**

## Configuración del Entorno de Desarrollo

### Requisitos de Hardware y Software

* **Hardware:** Requisitos mínimos de CPU, RAM y almacenamiento.
* **Software:** Sistemas operativos soportados, servidores web (como Apache o MySQL), versiones de PHP, MySQL, y otras dependencias.

### Instalación y Configuración

Instalación

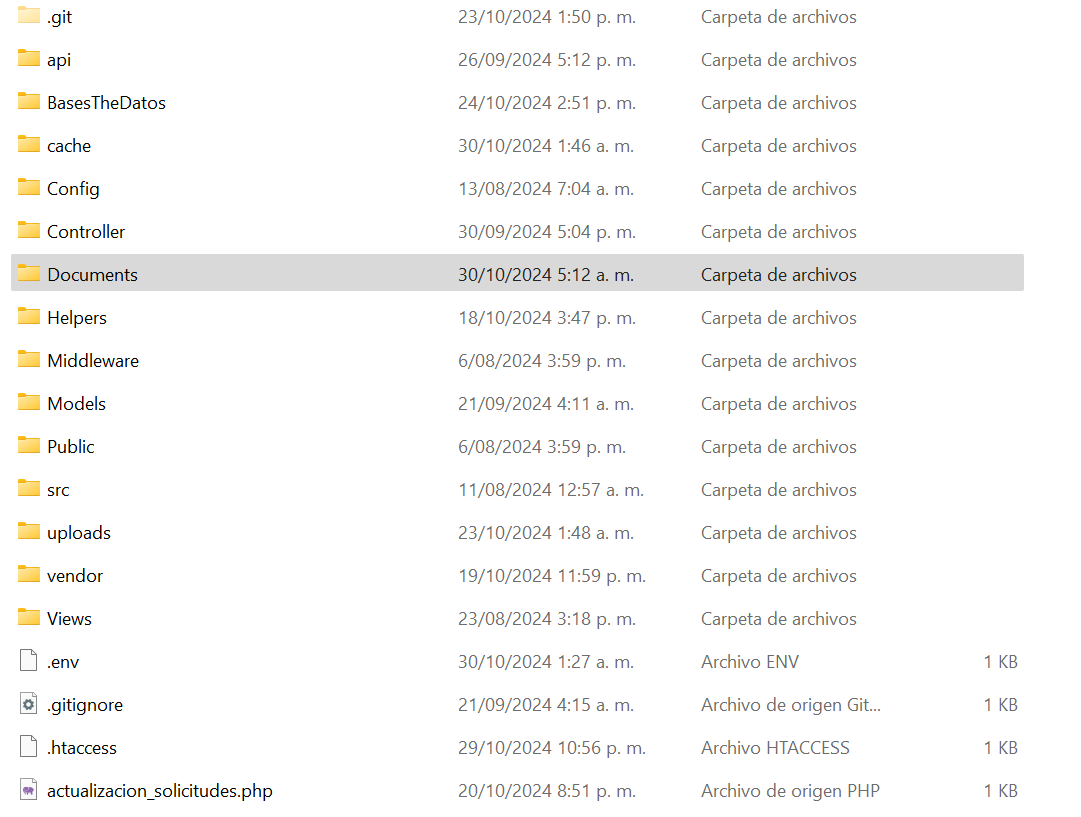
En la instalación se debe proporcionar los siguientes datos en el formulario:

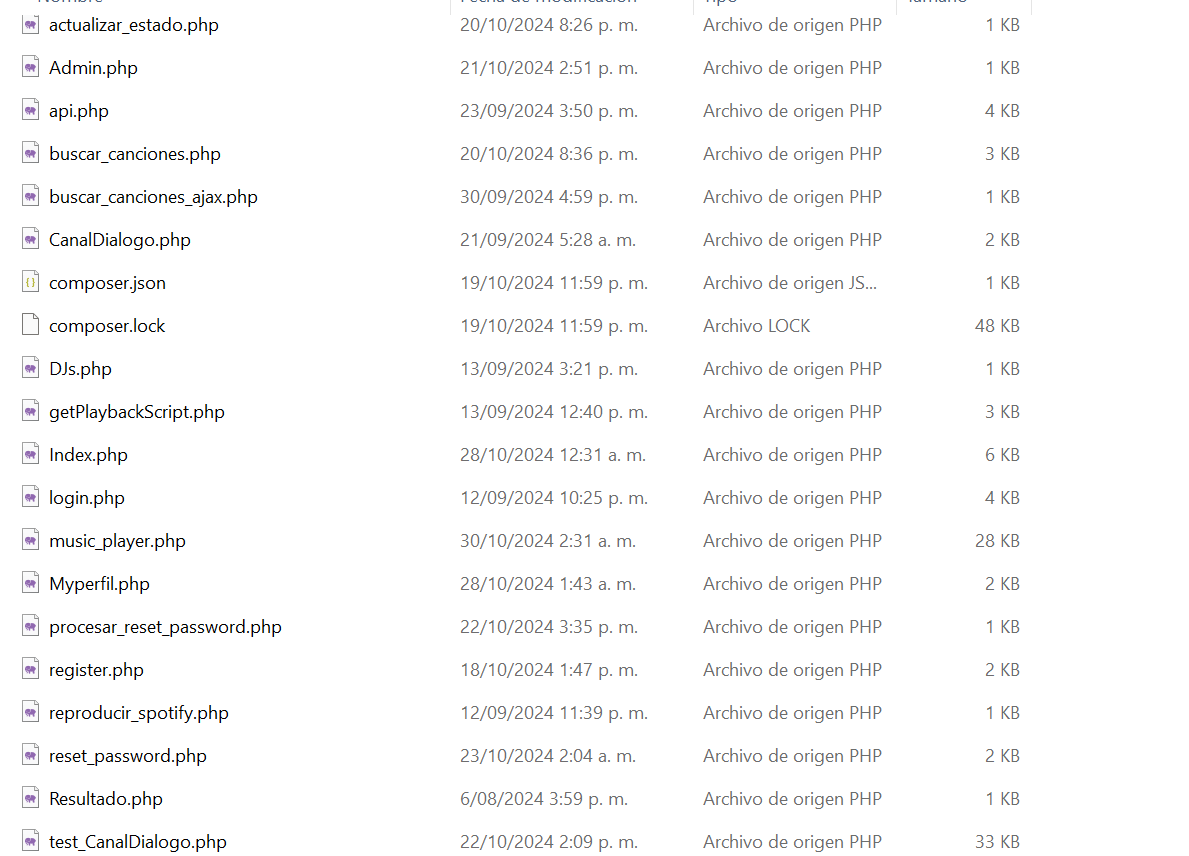
* + Host: localhost
  + Nombre de la base de datos: Elegans
  + Usuario: root
  + Contraseña.: “ “;

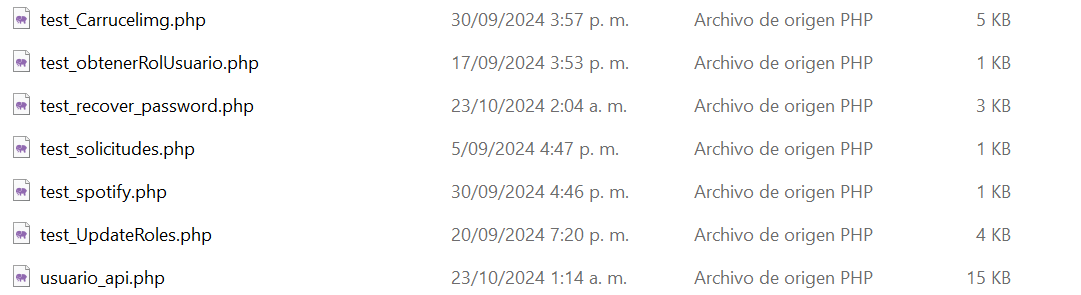


## Estructura del Proyecto

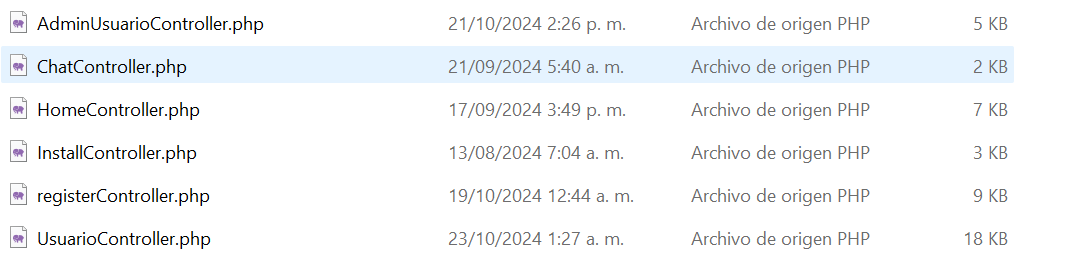
### Organización de Directorios y Archivos

****

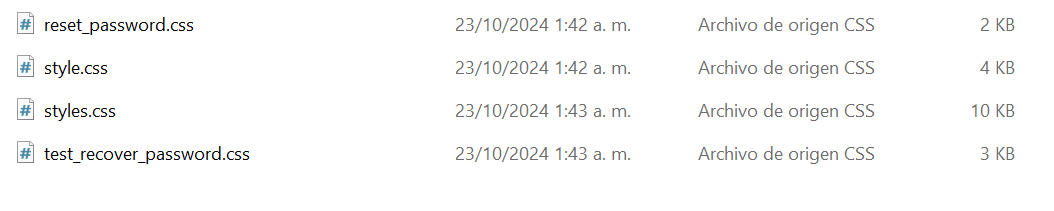
****

****

### Controller

****

### CSS



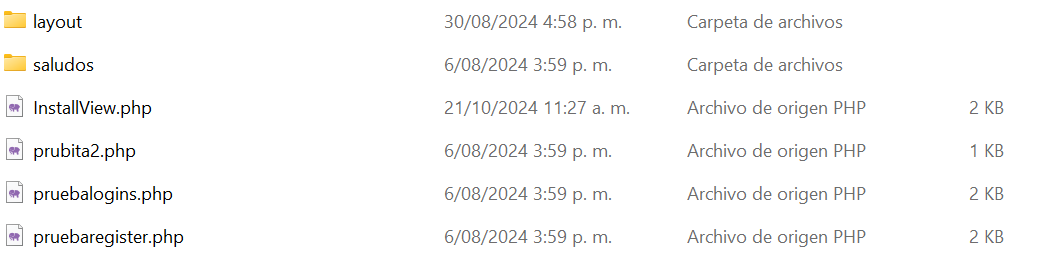
### JS

### 

### Model

### 

### View



### Módulos y Componentes Principales

* **Frontend:** Estructura de las vistas, hojas de estilo y scripts.
* **Backend:** Estructura de los controladores, modelos y rutas.
* **Base de Datos:** Esquema de la base de datos y relaciones entre tablas.

## Detalles Técnicos del Desarrollo

### Tecnologías y Frameworks Utilizados y Consumo de API

* **HTML/CSS/JavaScript vanilla:** Para el desarrollo del frontend.
* **PHP:** Para la lógica del backend.
* **MySQL:** Para la gestión de la base de datos.
* **Bootstrap:** Para el diseño responsivo.
* **PHPMailer:** Para la gestión de correos electrónicos.
* **Cloudflare Turnstile:** Para la verificación inteligente de CAPTCHA.
* **Spotify:** Para la consulta y reproducción de música.
* **Git:** Para el vercionamiento del Proyecto.

### Estructura del Código y Estándares de Codificación

Descripción de las convenciones y mejores prácticas de codificación utilizadas en el proyecto.

## Seguridad

### Medidas de Seguridad Implementadas

* **Autenticación y Autorización:** Métodos utilizados para proteger el acceso a la plataforma.
* **Prevención de Vulnerabilidades Comunes:** Protección contra ataques XSS y SQL inyección, etc.

### Procedimientos de Copias de Seguridad y Recuperación

La gestión adecuada de copias de seguridad y los procedimientos de recuperación son fundamentales para garantizar la integridad y disponibilidad continua del proyecto Elegans. A continuación, se describen los pasos y las mejores prácticas para realizar copias de seguridad y recuperar el repositorio en GitHub nombraba “MVC\_Proyecto-Elegans” Elegans[repositorio](https://github.com/DevMobyEnd/MVC_Proyecto-Elegans) .

# Mantenimiento y Soporte

## Contacto y Escalación de Soporte Técnico

Para soporte técnico, puedes contactar con el equipo a través del correo electrónico: [elegansdevmobyend@gmail.com](elegansdevmobyend@gmail.com%20) Sigue los procedimientos de escalación descritos en este documento para resolver problemas críticos de manera eficiente.

## Diagrama de actividades para

* Eliminar cuenta .



* Solicitud de Música.



* Iniciar sección.



## Diagrama de caso de uso para

****

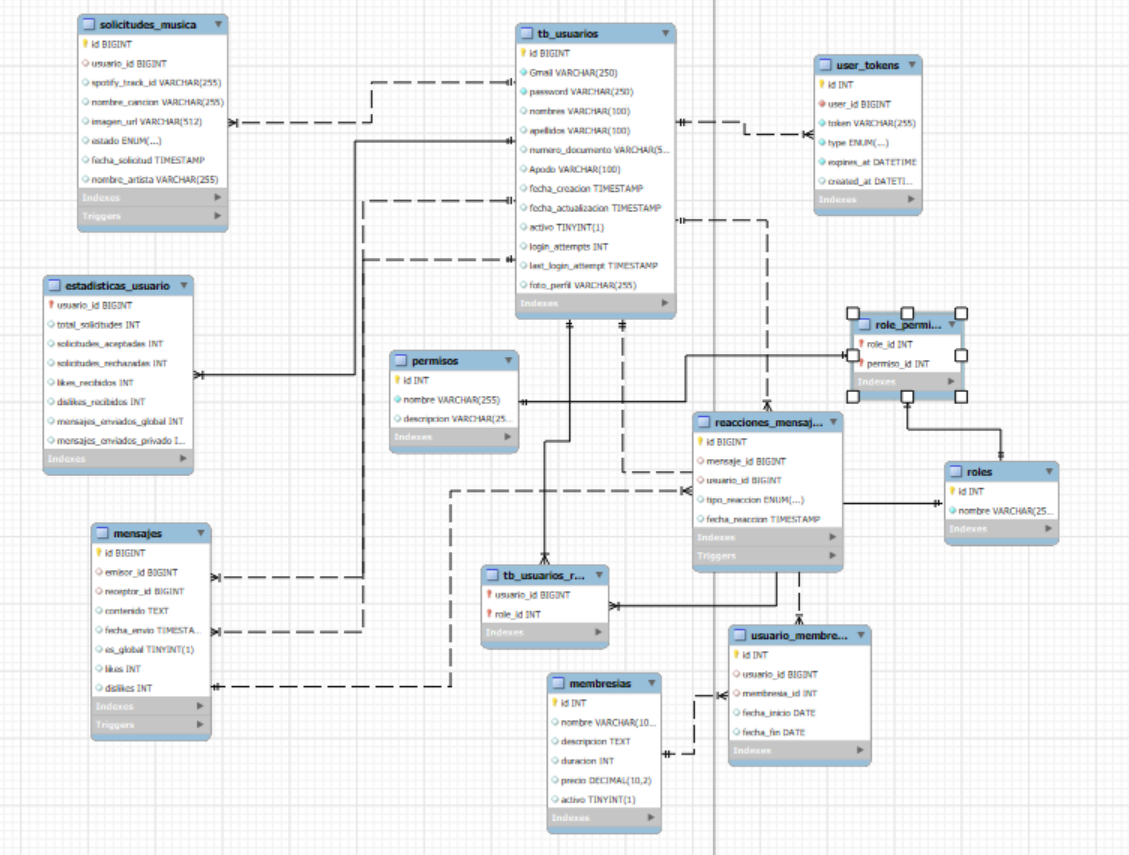
## Diagrama de modelo entidad relación

****

## Componentes

****

## Esquema de la base de datos

****

## Diagrama de clase

****