MANUAL TECNICO DEL SISTEMA

MERCARQ

Contenido

Lista de Figuras

Lista De Tablas

Presentación

Este manual contiene toda la información necesaria para utilizar, instalar y mantener el software de gestión de planos arquitectónicos Mercarq. El sistema está diseñado para facilitar la administración y distribución de planos de arquitectura, permitiendo obtener un control preciso sobre cada plano disponible para la venta. Además, proporciona una visión clara del flujo y comportamiento de las cotizaciones de clientes, así como de las transacciones realizadas y otros procesos relacionados.

La gestión eficaz del catálogo de planos arquitectónicos es esencial para mantener las operaciones comerciales y optimizar los procesos de venta y distribución. Ehrhardt y Brigham (2007) afirman que la gestión de inventarios digitales tiene tres objetivos principales:

- Garantizar la disponibilidad continua de planos para los clientes
- Mantener un catálogo organizado y actualizado para facilitar la búsqueda y selección
- Optimizar los procesos de venta para maximizar la rentabilidad del negocio

Manejar grandes catálogos de planos arquitectónicos puede resultar complicado y costoso, especialmente con técnicas tradicionales como la "gestión manual de archivos", que requiere recursos humanos significativos y tiempo considerable, siendo además propensa a errores de clasificación y pérdida de información. Esta situación es común en empresas arquitectónicas y de diseño que manejan múltiples proyectos y variaciones de planos.

Desde hace tiempo, la venta segura de planos arquitectónicos ha enfrentado importantes dificultades en la gestión del catálogo digital, procesamiento de pagos seguros, protección de la propiedad intelectual y entrega eficiente de archivos, debido a métodos de gestión ineficaces y plataformas inadecuadas para este tipo de productos especializados.

En respuesta a esta necesidad, este manual detalla el diseño y funcionamiento de la plataforma web Mercarq, que permite a los usuarios registrarse, explorar el catálogo de planos arquitectónicos y realizar pagos seguros por medio de WhatsApp, con escalabilidad futura para implementar pasarelas de pago integradas. La plataforma conecta arquitectos y diseñadores con clientes potenciales, mostrando planos de manera atractiva para promocionarlos efectivamente. El sistema administra de forma eficiente toda la información sobre los planos comercializados, incluyendo precios, especificaciones técnicas y estado de disponibilidad. Además, el manual incluye una guía detallada para desarrolladores que deseen conocer, utilizar o realizar mejoras en el software, proporcionando una comprensión profunda de la estructura y arquitectura de desarrollo del aplicativo.

Resumen

La revolución digital ha transformado radicalmente la comercialización de productos arquitectónicos, exigiendo soluciones innovadoras que trasciendan los métodos tradicionales de gestión. En este contexto emerge Mercarq, una plataforma web vanguardista que redefine los paradigmas establecidos en la venta de planos arquitectónicos, integrando tecnología de punta con una experiencia de usuario excepcional que satisface las demandas del mercado contemporáneo.

El ecosistema Mercarq constituye una arquitectura digital robusta que amalgama funcionalidades de registro intuitivo, catálogo dinámico y sistema de pagos híbrido a través de WhatsApp, estableciendo un puente tecnológico entre arquitectos visionarios y clientes exigentes. Esta plataforma trasciende las limitaciones convencionales del comercio electrónico especializado, ofreciendo una gestión integral que optimiza cada eslabón de la cadena de valor desde la catalogación hasta la entrega final.

La propuesta de valor se fundamenta en tres pilares estratégicos: la democratización del acceso a planos arquitectónicos de calidad premium, la protección integral de la propiedad intelectual mediante sistemas de seguridad avanzados, y la creación de un Marketplace especializado que fomenta la conexión directa entre creadores y consumidores. Esta sinergia genera un entorno comercial dinámico donde la innovación arquitectónica encuentra su canal de expresión más efectivo.

Este manual representa la culminación de un proceso de software meticulosamente diseñado, proporcionando una guía comprehensiva que abarca desde la perspectiva del usuario final hasta los aspectos más técnicos del desarrollo. La documentación establece un estándar de excelencia que facilita la adopción, mantenimiento y evolución continua de la plataforma, asegurando su sostenibilidad y crecimiento en el competitivo mercado de soluciones arquitectónicas digitales.

Objetivo

Proporcionar una guía integral y estructurada que facilite la comprensión, implementación y utilización óptima de la plataforma web Mercarq, habilitando a usuarios, administradores y desarrolladores para aprovechar al máximo las capacidades del sistema de gestión y comercialización de planos arquitectónicos, garantizando una experiencia fluida, segura y eficiente en todos los niveles de interacción con la plataforma.

Finalidad Del Manual

Este manual tiene como finalidad proporcionar la documentación técnica completa del software Mercarq, una plataforma web para la gestión y comercialización de planos arquitectónicos. El documento está dirigido específicamente a desarrolladores, programadores y personal técnico que requieran comprender la arquitectura del sistema, realizar mantenimiento, implementar nuevas funcionalidades o modificar el código existente. La finalidad es facilitar el trabajo de desarrollo mediante la documentación detallada de la estructura del código, base de datos, APIs, procedimientos de instalación y configuración, así como las mejores prácticas para el mantenimiento y escalabilidad del software.

Introducción

El presente manual técnico tiene como objetivo principal brindar una guía completa y detallada sobre el software Mercarq desde una perspectiva técnica e informática. Su contenido está diseñado para dotar al personal encargado de la administración, edición y configuración del aplicativo con los conocimientos y herramientas necesarias para realizar estas tareas de manera apropiada y eficiente.

El documento se encuentra estructurado en las siguientes secciones:

- **Aspectos Teóricos:** En esta sección, se abordarán los conceptos fundamentales, definiciones y explicaciones teóricas relacionadas con los diferentes componentes de la plataforma web de planos arquitectónicos. Esto permitirá al lector adquirir una comprensión sólida del funcionamiento del sistema de gestión de catálogo digital y las herramientas involucradas.
- **Aspectos Técnicos Del Desarrollo Del Sistema:** Se buscará profundizar en los aspectos técnicos del desarrollo de la plataforma, incluyendo detalles sobre el almacenamiento de datos de planos arquitectónicos, la estructura del código fuente, las tecnologías web utilizadas, la integración con WhatsApp para pagos y las mejores prácticas para garantizar un uso adecuado y óptimo del aplicativo.
- **Requerimientos Del Software:** En esta sección, se detallarán los requisitos de hardware, software y configuración necesarios para el correcto funcionamiento de Mercarq. Esto permitirá a los usuarios preparar adecuadamente su entorno de trabajo y asegurar una experiencia fluida durante la implementación y el uso del sistema de gestión de planos.

A lo largo de este manual, se encontrarán instrucciones paso a paso, ejemplos prácticos y recomendaciones que facilitarán la comprensión y el dominio de Mercarq desde una perspectiva técnica.

1. Aspectos Técnicos

Mercarq es una plataforma web diseñada para facilitar la gestión eficiente de planos arquitectónicos en el mercado digital. Este sistema permite un control preciso sobre el catálogo de planos disponibles para la comercialización, brindando una visión clara del flujo y comportamiento de las transacciones, cotizaciones y procesos de venta realizados en la plataforma.

Este manual está destinado exclusivamente al personal autorizado para administrar, editar o configurar la plataforma Mercarq. Se recomienda no compartir ni divulgar la información contenida en este documento, ya que podría comprometer la integridad y seguridad de los datos almacenados en la base de datos del sistema.

1.1. Herramientas Utilizadas Para El Desarrollo

En esta sección, se detallan las herramientas informáticas empleadas para el desarrollo de la plataforma Mercarq, incluyendo lenguajes de programación, frameworks, bases de datos y entornos de desarrollo integrados (IDEs).

1.1.1. Visual Studio Code

Es un editor de código fuente ligero y potente desarrollado por Microsoft. Fue el entorno de desarrollo principal utilizado por el equipo para la edición, depuración y desarrollo del código fuente de la plataforma. Es una herramienta multiplataforma, gratuita y de código abierto que ofrece soporte nativo para JavaScript, TypeScript, Node.js y una amplia gama de extensiones que facilitan el desarrollo web. Su interfaz intuitiva, sistema de autocompletado inteligente y capacidades de debugging integradas lo convierten en una opción ideal para proyectos web modernos.

1.1.2. GitHub

Plataforma líder mundial de alojamiento de repositorios de código fuente y control de versiones basada en Git. Se utilizó un repositorio en GitHub para el proyecto Mercarq, lo que permitió al equipo de desarrollo colaborar de manera segura y coordinada. GitHub facilita el seguimiento de cambios en el código, la gestión de ramas de desarrollo, la resolución de conflictos y la implementación de flujos de trabajo colaborativos. Además, proporciona herramientas esenciales como la gestión de issues, pull requests, wikis de documentación, sistemas de integración continua y despliegue automatizado, garantizando un desarrollo organizado y profesional.

1.1.3. Laravel

Framework de desarrollo web de código abierto para PHP, reconocido por su sintaxis elegante, expresiva y fácil comprensión. Laravel fue seleccionado como la base principal para construir el backend y la lógica de negocio de la plataforma Mercarq, aprovechando su robusta arquitectura Modelo-Vista-Controlador (MVC). Este framework ofrece características avanzadas como el sistema de enrutamiento RESTful, migraciones de base de datos, ORM Eloquent para manejo de

datos, sistema de autenticación y autorización integrado, motor de plantillas Blade, y un ecosistema completo de paquetes y librerías que aceleran significativamente el desarrollo de aplicaciones web seguras y escalables.

1.1.4. PhpMyAdmin

Herramienta de código abierto escrita en PHP que proporciona una interfaz web intuitiva para administrar servidores MySQL. En el desarrollo de Mercarq, se utilizó PhpMyAdmin en conjunto con Laravel para realizar operaciones de consulta, creación, modificación y eliminación de datos en la base de datos MySQL del aplicativo de manera sencilla, facilitando la gestión del catálogo de planos y las transacciones.

1.1.5. MySQL

Sistema de gestión de bases de datos relacionales (RDBMS) de código abierto y ampliamente utilizado en el desarrollo web. MySQL se utilizó como el sistema de base de datos principal para almacenar y gestionar los datos del aplicativo Mercarq, aprovechando su integración nativa con Laravel y PHP.

1.1.6. Bootstrap 5

Framework de front-end de código abierto y ampliamente popular, que proporciona una colección de herramientas y componentes de interfaz de usuario (UI) basados en HTML, CSS y JavaScript. En Mercarq, se utilizó Bootstrap para construir una interfaz de usuario atractiva y responsive, destacando el uso de colores naranjas en el diseño, lo que permite una experiencia visualmente atractiva y adaptable a diferentes dispositivos, optimizando la promoción de planos arquitectónicos.

1.1.7. GitBash

Aplicación de línea de comandos que proporciona una emulación de la Shell de Bash en sistemas operativos Windows, lo que permite utilizar herramientas de línea de comandos de Unix. En el desarrollo de Mercarq, GitBash se utilizó en conjunto con Visual Studio Code y GitHub para realizar operaciones de control de versiones, como la creación de commits, el envío de cambios al repositorio remoto y la fusión de ramas de código.

1.1.8. Librerías Utilizadas en Laravel

El desarrollo de Mercarq incorporó diversas librerías de Laravel y paquetes adicionales para optimizar su funcionalidad. Las siguientes son las principales dependencias utilizadas:

Producción:

- **PHP:** Versión mínima requerida de PHP para ejecutar el proyecto.
- **laravel/framework:** - El núcleo del framework Laravel, que proporciona la estructura MVC y funcionalidades esenciales.

- **laravel/tinker:** - Herramienta de consola interactiva para depurar y probar código en Laravel.

Desarrollo y Pruebas:

- **fakerphp/faker:** - Generador de datos ficticios para pruebas y seeding de bases de datos.
- **laravel/breeze**- Paquete para autenticación y scaffolding de interfaces de usuario simplificadas.
- **laravel/pail:** - Herramienta de registro (logging) avanzada para monitoreo en tiempo real.
- **laravel/pint:** - Herramienta de formateo de código para mantener consistencia en el estilo del código.
- **laravel/sail**- Entorno de desarrollo con contenedores Docker para simplificar la configuración local.
- **mockery/mockery**- Biblioteca para crear mocks y pruebas unitarias.
- **nunomaduro/collision**- Mejora las excepciones con una interfaz más legible para depuración.
- **pestphp/pest:** - Framework de pruebas ligero y moderno.
- **pestphp/pest-plugin-laravel:** - Extensión de Pest para integrarse con Laravel.

Estas librerías se gestionaron mediante Composer, asegurando una instalación y actualización eficientes de las dependencias del proyecto.

2.3. Diccionario de Datos

Diccionario de datos modelo solicitudes

Nombre de la Columna	Tipo de Dato	Descripción
id	bigint UNSIGNED NOT NULL	Identificador único de la solicitud
blueprint_id	bigint UNSIGNED DEFAULT NULL	Identificador del plano relacionado
tipo_solicitud	varchar(255) NOT NULL	Tipo de solicitud (e.g., whatsapp, descarga_gratuita)
nombre_solicitante	varchar(255) NOT NULL	Nombre del solicitante
email_solicitante	varchar(255) DEFAULT NULL	Correo del solicitante
telefono_solicitante	varchar(255) DEFAULT NULL	Teléfono del solicitante
mensaje	text	Mensaje o detalles de la solicitud
ip_address	varchar(255) DEFAULT NULL	Dirección IP del solicitante
created_at	timestamp NULL DEFAULT NULL	Fecha y hora de creación
updated_at	timestamp NULL DEFAULT NULL	Fecha y hora de última actualización

Fuente: Por los autores

Diccionario de datos modelo purchases

Nombre de la Columna	Tipo de Dato	Descripción
`id`	bigint UNSIGNED NOT NULL	Identificador único de la compra
`user_id`	bigint UNSIGNED NOT NULL	Identificador del usuario comprador
`blueprint_id`	bigint UNSIGNED NOT NULL	Identificador del plano comprado
`amount`	decimal(10,2) NOT NULL	Monto de la compra
`payment_method`	varchar(255) DEFAULT NULL	Método de pago (e.g., WhatsApp)
`payment_reference`	varchar(255) DEFAULT NULL	Referencia del pago
`status`	varchar(255) NOT NULL DEFAULT 'completed'	Estado de la compra (por defecto 'completed')
`created_at`	timestamp NULL DEFAULT NULL	Fecha y hora de creación
`updated_at`	timestamp NULL DEFAULT NULL	Fecha y hora de última actualización

Fuente: Por los autores

Diccionario de datos modelo blueprints

Nombre de la Columna	Tipo de Dato	Descripción
id	bigint UNSIGNED NOT NULL	Identificador único del plano
user_id	bigint UNSIGNED NOT NULL	Identificador del usuario creador
title	varchar(255) NOT NULL	Título del plano
description	text	Descripción del plano
file_path	varchar(255) NOT NULL	Ruta del archivo del plano
price	int NOT NULL DEFAULT '0'	Precio del plano (en pesos)
whatsapp_number	varchar(20) NOT NULL	Número de WhatsApp para pagos
file_size	bigint DEFAULT NULL	Tamaño del archivo en bytes
is_public	tinyint(1) NOT NULL DEFAULT '0'	Indica si el plano es público (0/1)
created_at	timestamp NULL DEFAULT NULL	Fecha y hora de creación
updated_at	timestamp NULL DEFAULT NULL	Fecha y hora de última actualización

Fuente: Por los autores

Diccionario de datos modelo users

Nombre de la Columna	Tipo de Dato	Descripción
id	bigint UNSIGNED NOT NULL	Identificador único del usuario
name	varchar(255) NOT NULL	Nombre del usuario
email	varchar(255) NOT NULL	Correo electrónico del usuario (único)
avatar_path	varchar(255) DEFAULT NULL	Ruta del avatar del usuario
email_verified_at	timestamp NULL DEFAULT NULL	Fecha y hora de verificación del email
password	varchar(255) NOT NULL	Contraseña encriptada del usuario
role	varchar(255) NOT NULL DEFAULT 'cliente'	Rol del usuario (por defecto 'cliente')
remember_token	varchar(100) DEFAULT NULL	Token para recordar sesión
created_at	timestamp NULL DEFAULT NULL	Fecha y hora de creación
updated_at	timestamp NULL DEFAULT NULL	Fecha y hora de última actualización

Fuente: Por los autores

3. Aspecto Técnico Del Desarrollo Del Sistema

En la siguiente sección se procede a realizar una descripción detallada sobre los aspectos técnicos del aplicativo relacionado con la instalación de las herramientas necesarias para realizar modificaciones requeridas de manera ordenada

3.1. Modificación Local

Si el desarrollador desea realizar modificaciones del software de manera local, tendrá que realizar la instalación de componentes adicionales, para empezar, se debe de instalar php versión 8 o superiores (para poder trabajar con laragon o Docker), el cual se consigue de manera gratuita en la página https://www.php.net/downloads.php, por consiguiente, se deberá de seguir los siguientes pasos

Figura 1. Página web de descarga de PHP

Fuente: (S/f). Php.net.

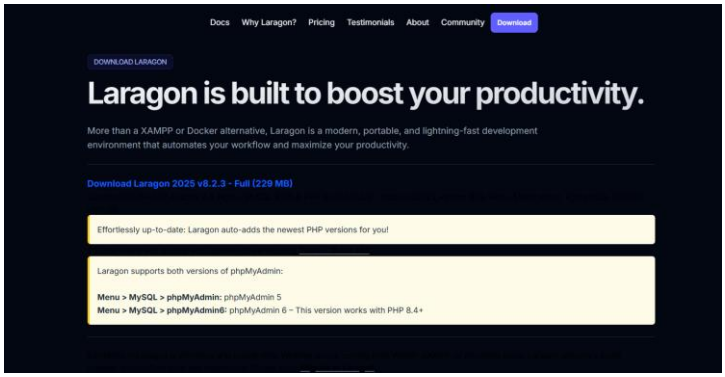
Al descargar PHP podemos proceder con su debida instalación una vez que hemos instalado y configurado el PHP en las variables de entorno de nuestro equipo, podemos proceder a instalar el editor de texto, el cual para el desarrollo de este proyecto fue realizado con visual Studio Code, nos dirigimos a la pagina de descarga https://code.visualstudio.com/download y descargamos la última versión en la pagina oficial, esta descarga es con licencia gratuita

Figura 2 Página de descarga de Visual Studio Code

Fuente: Visual Studio Code. (s/f).

Se continúa descargando el aplicativo de laragon el cual se realiza desde la pagina oficial, laragon es una herramienta de desarrollo específico para Windows el cual brinda todo lo necesario para iniciar el desarrollo en el menor tiempo posible

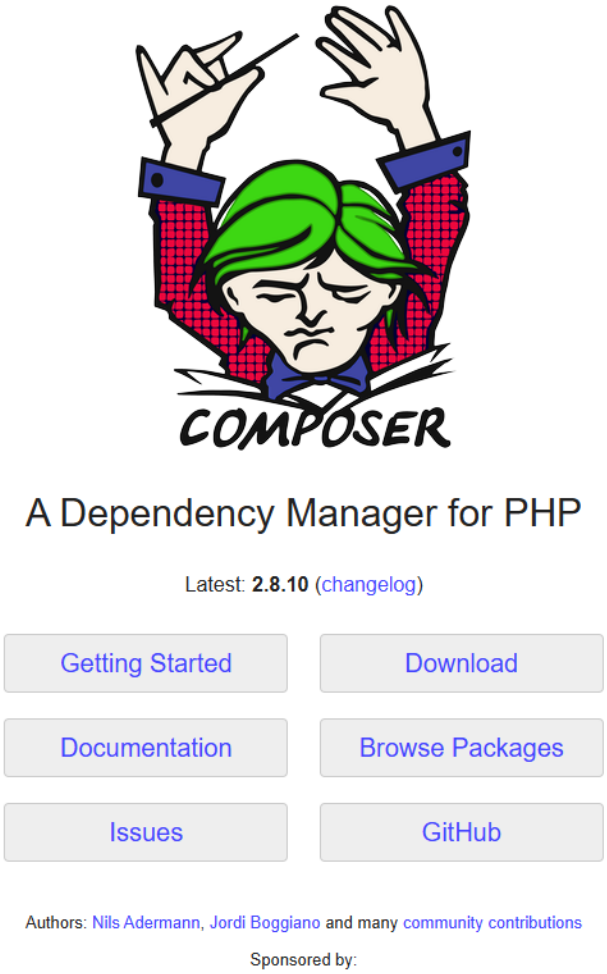
Figura 3 Página de descarga de Laragon



Fuente: LeoKhoa. (s/f). Laragon.

Después de tener PHP, Visual Studio y laragon instalados en nuestro equipo nos hace falta realizar una descarga, la cual es composer, composer es un gestor de paquetes para PHP el cual posibilita la administración, descargas instalaciones y dependencias

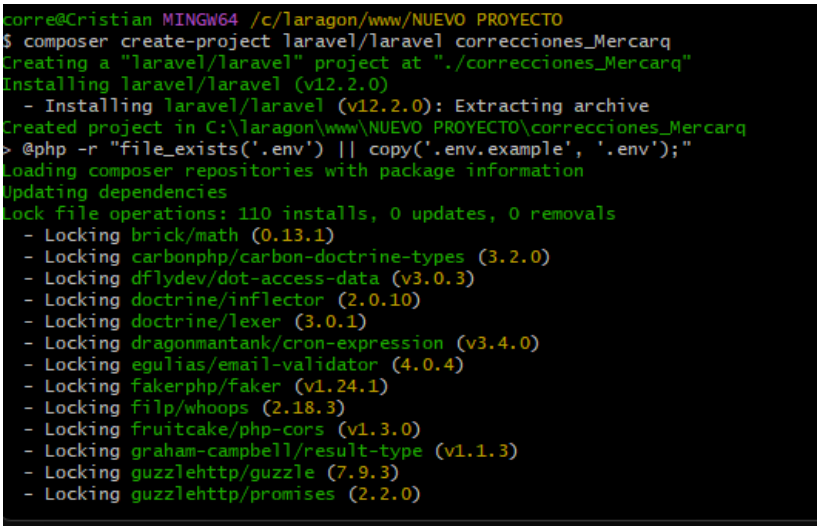
Figura 4 Página de descarga de Composer



Fuente: Composer. (s/f).

Teniendo todo lo anterior podemos utilizar git bash para descargar el repositorio del proyecto o hacer un proyecto desde cero, Ejecutando el comando composer créate-Project laravel/laravel (Nombre Proyecto)

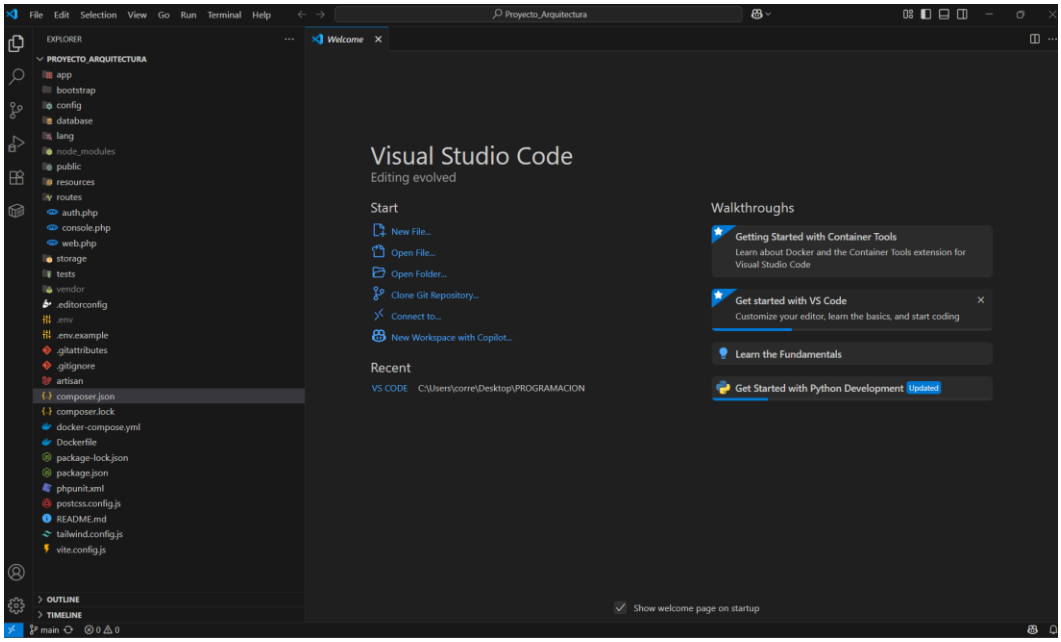
Figura 5 Clonacion repositório Github



Fuente: Por los autores

Una vez realizada la clonación del repositorio de GitHub en nuestra carpeta se debe de proceder con la ejecución del IDE visual studio code en donde a la vez se deberá de seleccionar la carpeta donde hemos clonado el repositorio

Figura 6 Visualización del Proyecto En VSC

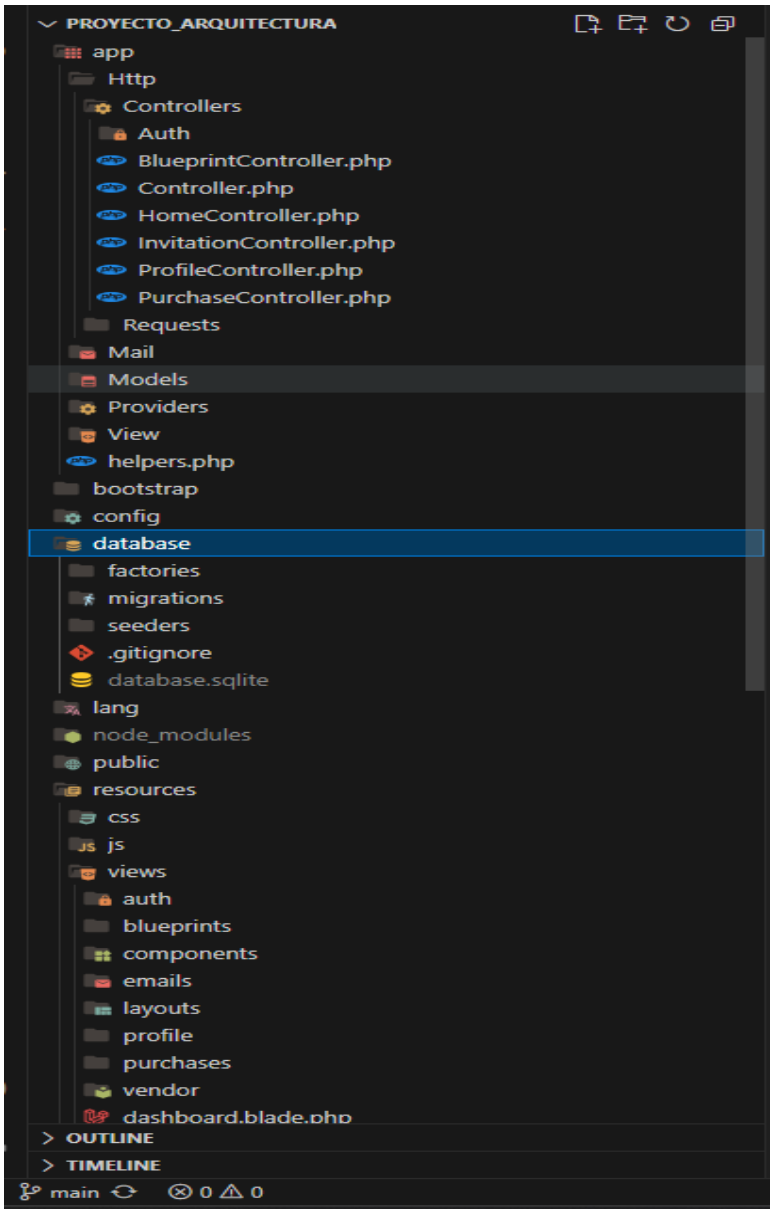


Fuente: Por los autores

Por consiguiente, se nos mostraran una gran variedad de carpetas y programas sin embargo no debe de alterarse, ya que el software solamente utiliza el patrón modelo vista controlador para

este desarrollo eso quiere decir que solo se utilizan las carpetas de controllers model view migrations etc.

Figura 7 Carpetas utilizadas y Patrón MVC



Fuente: Por los autores

El patrón modelo-vista-controlador (MVC) es un patrón de arquitectura de software ampliamente utilizado en el desarrollo de aplicaciones web. Este patrón separa la lógica de la aplicación en tres componentes principales: El modelo, la vista y el controlador. Esta separación de responsabilidades facilita, el desarrollo el mantenimiento y la escalabilidad de las aplicaciones

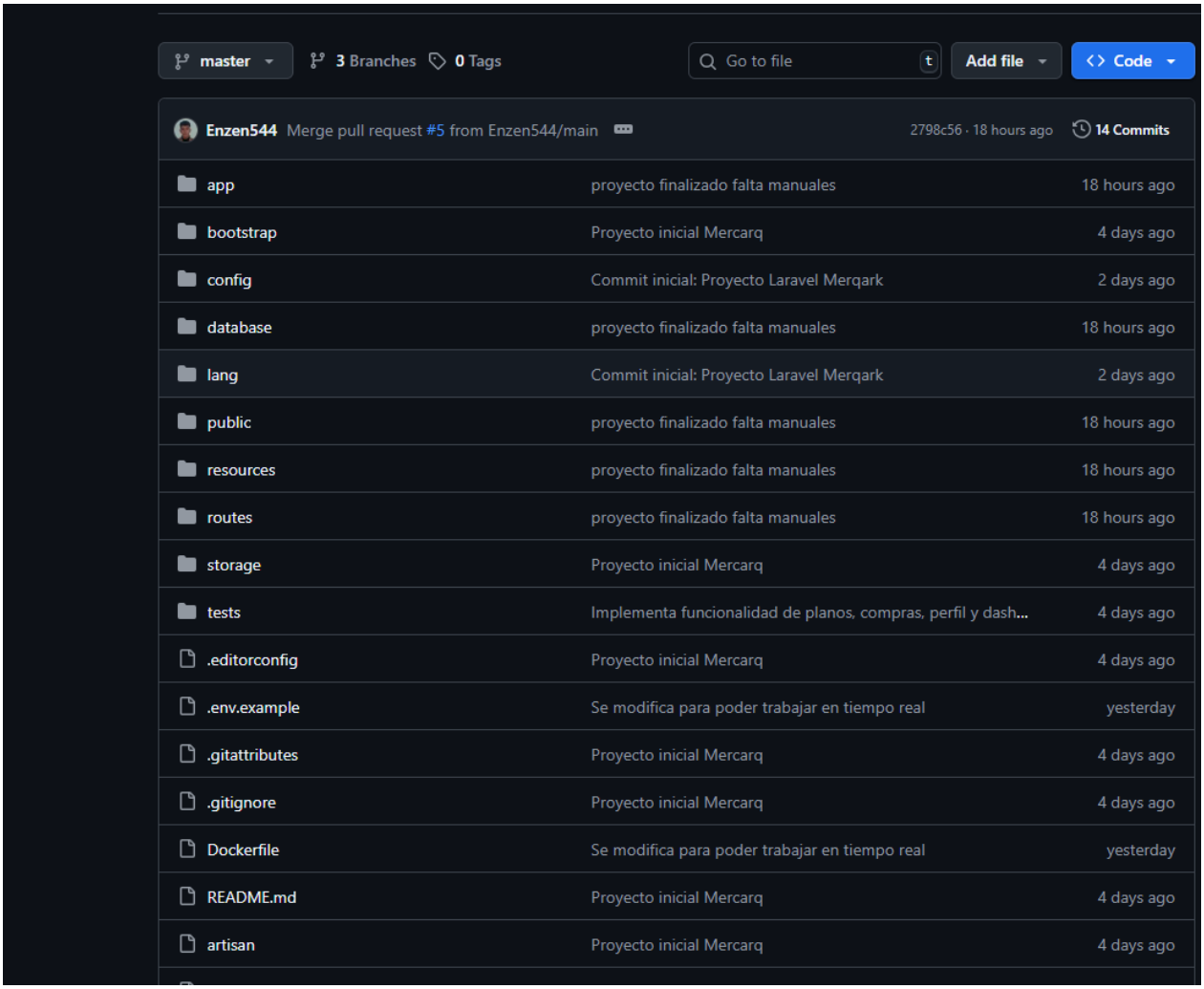
- **Modelo**: El modelo representa los datos de la aplicación y las reglas de negocio asociadas. Es responsable de la gestión de los datos, como la recuperación, la inserción la actualización y la eliminación. En laravel, los modelos se utilizan para interactuar con la base de datos a través del sistema de eloquent ORM (Object-Relational Mapping)
- **Vista**: La vista es responsable de la presentación de los datos al usuario. Se encarga de generar la interfaz de usuario y mostrar la información recibida desde el Modelo. En laravel, las vistas se crean utilizando archivos de plantillas que contienen una combinación de HTML, CSS y código PHP más conocidas como Blade
- **Controlador**: El controlador actúa como intermediario entre el Modelo y la vista Es responsable de manejar las solicitudes del usuario, recuperar los datos necesarios del modelo y pasarlos a la vista para su presentación, En laravel los controladores son clases PHP que contienen métodos que corresponden a diferentes rutas y acciones de la aplicación.

El uso del patrón modelo vista controlador en Laravel y PHP nos ofrece múltiples beneficios:

- **Separación de responsabilidades:** Mejora la organización y el mantenimiento del código al dividirlo en lógica, interfaz y flujo.
- **Escalabilidad:** Permite integrar nuevas funciones sin afectar el funcionamiento existente.
- **Colaboración en equipo:** Facilita que varios desarrolladores trabajen en paralelo sin conflictos. **Reutilización de código:** Ahorra tiempo y esfuerzo al permitir reaprovechar vistas y controladores
- **Integración con herramientas de Laravel:** Potencia el patrón con funcionalidades como rutas, middleware y migraciones.

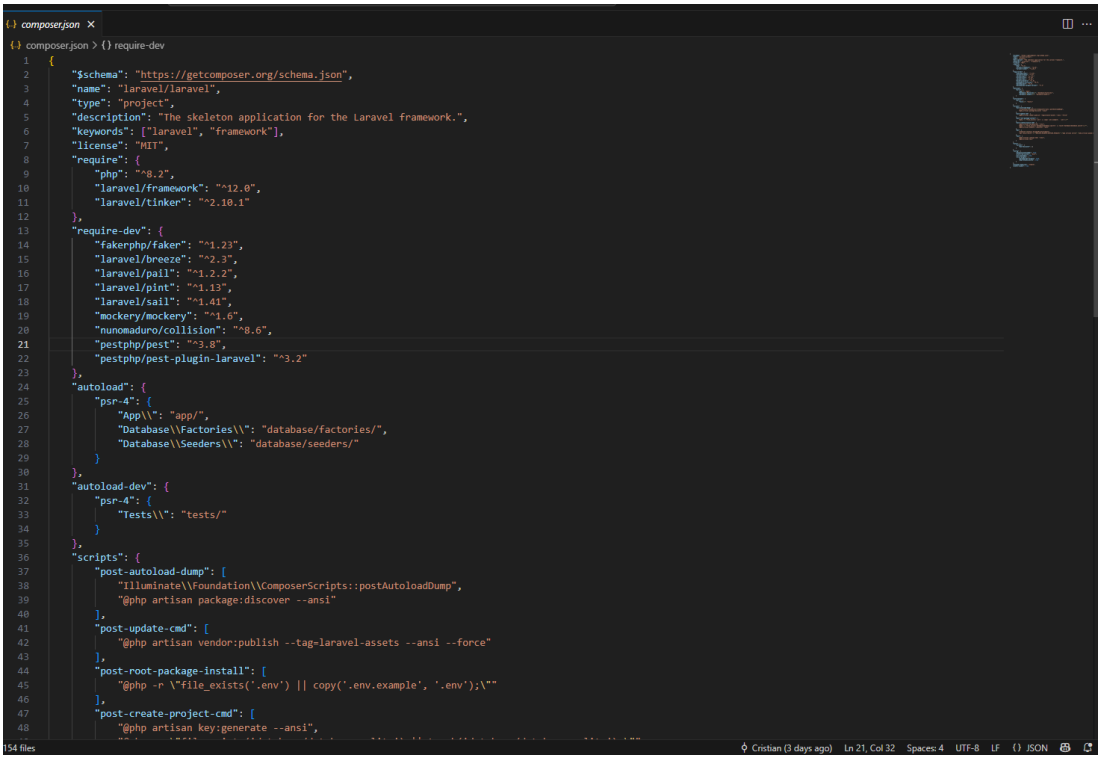
Una vez implementadas las mejoras y modificaciones al código se deberá de subir los cambios al repositorio de GitHub, para ello inicializamos nuestra consola de Bash, y agregamos los comandos típicos para subirlos git init, git add . , git commit – m “descripción de lo que hiciste”, git push origin main o master dependiendo de cual sea tu rama principal si usas main el cambio estará en local y tendrás que desde la plataforma autorizar esta subida haciendo el commit y el merge . Una vez realizados los cambios en la plataforma se nos demostrara quien trabajo que hizo y con qué cambios

Figura 8 Cambios repositorio GitHub



Fuente: Por los autores

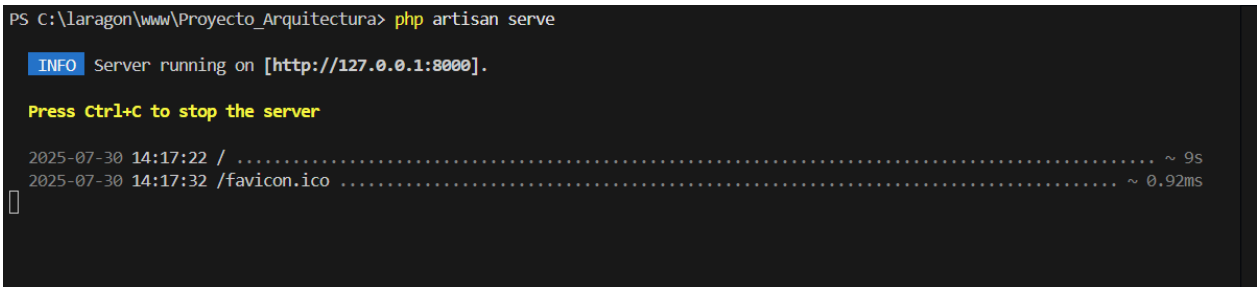
Figura 9 Validación Composer.Json



Fuente: Por los autores

El archivo composer.json en Laravel es como el corazón de nuestro proyecto. Es un archivo que contiene la información sobre todas y cada una de las dependencias del proyecto, es decir, los otros paquetes de código que el proyecto necesita para funcionar correctamente. También contiene información de configuración y se utiliza para gestionar la autocarga de clases

Figura 10 inicialización en servidor local



Fuente: Por los autores

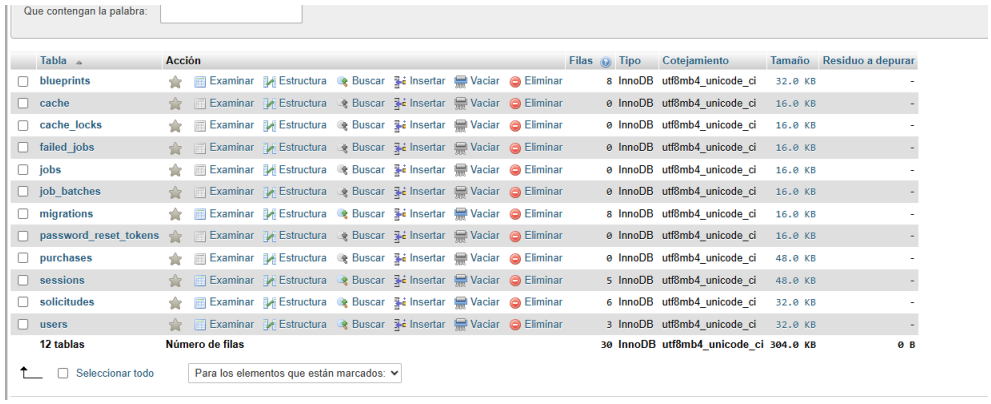
Existen 2 maneras la 1 es utilizando la manera local de nuestro vscode el comando php artisan serve y ejecutar el link que nos aparece, la 2 es abrir laragon y navegar hacia la carpeta donde tenemos alojado el proyecto

Figura 11 Ingreso a la administración de PhpMyAdmin

Fuente: Por los autores

Al ingresar al gestor de bases de datos PhpMyAdmyn se podrá tener la gerencia completa de nuestras tablas en caso de necesitar la inserción creación o modificación de una tabla

Figura 12 Administración en PhpMyAdmin



Fuente: Por los autores

4. Requerimientos Del Software

Requisito	Detalle
Sistema Operativo	Windows 10, Superior o Equivalentes
Procesador	Intel Core i3, superior o Equivalentes
Memoria RAM	8 GB o tamaños superiores
Disco Duro	100 GB o mayor capacidad
Resolución De Pantalla	1280 x 720 Pixeles
Periféricos	Teclado, Ratón Bocinas (Opcional)

Bibliografía

Aprendiendo a usar GitHub — Conociendo GitHub 0.1 documentation. (n.d.). Readthedocs.Io. Retrieved July 30, 2025, from https://conociendogithub.readthedocs.io/en/latest/data/dinamica-de-uso/

Composer. (s/f). Getcomposer.org. Recuperado el 30 de julio de 2025, de https://getcomposer.org/

Download Visual Studio Code. (s/f). Visualstudio.com. Recuperado el 30 de julio de 2025, de https://code.visualstudio.com/download

LeoKhoa. (s/f). *Laragon*. Laragon. Recuperado el 30 de julio de 2025, de https://laragon.org/

¿Qué es Visual Studio Code y cuáles son sus ventajas? (n.d.). Arsys. Retrieved July 30, 2025, from https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas

(S/f). Php.net. Recuperado el 30 de julio de 2025, de https://www.php.net/downloads.php