

Manual Técnico

Proyecto Teoría de Sistemas 2 2024

BARTER PLACE

#14

Erick Daniel Morales Xicará

201930699

## MANUAL TÉCNICO

El software puede ser utilizado en cualquier sistema operativo ya que es un proyecto web, hecho con lenguajes de programación: Typescript y Php, también se puede ejecutar en la mayoría de computadoras que tengan php 8.3.6++, angular 15.2.4 +, Node 16.19.0+, y para su creación fueron utilizados IDE's como IntelliJ y VsCode.

El software consiste en el manejo que permite poner en práctica los conceptos de gestión de proyectos y teoría de sistemas para implementar un sistema hecho a mla medidas que permita el intercambio de bienes y servicios en una comunidad (Comercio electrónico), teniendo características como ser una alternativa al comercio tradicional utilizando las TICs, a su vez poder manejar varios tipos de roles, manejados de distintas maneras donde puede hacer publicaciones y a su vez poder comprar , hacer voluntarios o vender productos, a su vez, manejo de moneda de aplicación y moneda local.

## Backend:

Se refiere a la parte de un sistema informático que gestiona y procesa la lógica del negocio, la manipulación de datos y la interacción con la base de datos. En el desarrollo de web, el backend se encarga de manejar las operaciones del servidor y la comunicación con el frontend (Interfaz del usuario), por lo cual se utilizara Laravel:

### [Documentación Laravel](#)



Laravel es un framework de desarrollo web en PHP, que se utiliza comúnmente para construir el backend de aplicaciones. Algunas razones por las que Laravel es un popular y considerado bueno para el desarrollo web incluyen:

- Elegancia y Simplicidad
- ORM Eloquent : tiene un sistema de mapeo objeto-relacional llamado Eloquent, ya que utiliza una sintaxis para la base de datos orientada a objetos
- Migraciones a la BD
- Sistema de Plantillas Blade
- Rutas y Controladores: tiene un sistema de enrutamiento simple y poderoso junto con controladores que facilitan la organización del código y la gestión de solicitudes HTTP.

### Instalación:

Para instalar php y laravel deberá ejecutar los siguientes comandos en la terminal:

- `sudo apt install curl php-cli php-mbstring`
- `curl -sS https://getcomposer.org/installer -o composer-setup.php`
- `sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer`

Si se está utilizando un sistema de windows o otro sistema de linux, se puede abocar a la página oficial de Laravel:

### [Página Oficial Laravel](#)

Veremos algunas de las carpetas

## **Http/Controllers**

Esta carpeta obtendrá todos los archivos que son utilizados para poder realizar las consultas con la base de datos:

- AccountBankController
- BankController
  - updateAplicationCurrency:
  - updateAplicationCurrency2
  - updateAplicationCurrency3
  - updateAplicationCurrency4
  - updateRealMoney
  - updateBuyPack
  - returnMoneyIfBlocked
  - transferUsers
- CategoryController
- ChatsController
  - verExistenciaChat
- MessagesController
- PackMoneyController
- PublicationController
  - publicationsP
  - reenviarPublication
  - inactivePublication
  - getPublicationsMain
  - publicationsByCategory
  - comprobarNumeroPublicaciones
  - blockPublication
- PublicationCopyController
  - verificarParticipacionAnt
- PublicationTyprController
- RecordBuyPackController
- RecordRechargeController

- ReportPublicationController
  - changeStatus
  - bloquearPublication
  - infoBloqueoPublication
- RequestSellerController
- TarjetController
- TransferController
- UserController
  - getAllAdmins
  - deleteAdmin
  - updateImg

Los anteriores archivos también contienen los métodos:

- index : para obtener toda la lista
- store : para añadir un nuevo registro
- show : para pedir un registro en específico
- update: actualizar un registro
- destroy : eliminar un registro

## **Models**

Este directorio contiene los modelos de la base de datos, definiendo la llave primaria de la tabla, y el fiable para poder corroborar que vengan los datos necesarios para un nuevo registro en su mayoría.

- AccountBank
- Bank
- Category
- Chats
- Client
- Messages
- PackMoney
- Publication
- PublicationCopy
- PublicationType

- RecordBuyPack
- RecordRecharge
- ReportPublication
- RequestSeller
- Target
- Transfer
- User

## **Migrations**

Este directorio contiene información sobre las migraciones de las bases de datos, hechas desde laravel, teniendo los atributos de cada tabla y las relaciones de las mismas tablas, con métodos como:

- up
- down

## **Routes/api**

Este directorio contiene la información sobre las rutas de acceso, es decir, las rutas de toda la api, teniendo en cuenta el uso de métodos:

- get
- post
- put
- delete

# FRONTEND:

En el contexto de un software se refiere a la parte de la aplicación o sistema que interactúa directamente con los usuarios. Es la interfaz de usuario y la experiencia que tienen los usuarios al utilizar la aplicación. El frontend se encarga de presentar la información de manera visual, gestionar la interacción del usuario y enviar solicitudes al “Backend”. Por lo anterior la herramienta a utilizar será Angular:



[Documentación Angular](#)

Angular es un framework de desarrollo de aplicaciones web y móviles desarrollado y mantenido por Google. Se utiliza para construir aplicaciones del lado del cliente con funcionalidades avanzadas y complejas. Aquí hay algunas razones por las cuales angular es considerado bueno para el manejo del frontend:

- Arquitectura MVVM
- Datos Bidireccionales
- Inyección de Dependencias
- Componentes Reutilizables
- Tipado Estático con TypeScript
- Routing Integrado
- Amplia Comunidad y Soporte
- Herramientas de Desarrollo

Angular utiliza herramientas como:

- Lenguaje TypeScript: Lenguaje tipado estático a JavaScript, lo que facilita el desarrollo de aplicaciones más seguras y mantenibles.
- Node.js y npm: Node.js es usado para ejecutar el entorno de desarrollo y npm (Node Package Manager) para gestionar las dependencias y bibliotecas necesarias.
- Angular CLI (Command Line Interface)
- RxJS (Reactive Extensions for JavaScript)

## Instalación:

Si se quiere instalar node, npm y Angular se necesita copiar y pegar los siguientes comandos en la terminal (Sistema Operativo basado en Linux)

- `sudo apt install nodejs npm`
- `npm install -g @angular/cli`

Si se utiliza windows o otro sistema de linux, se puede abocar a la página oficial de Angular:

[Página Oficial Angular](#)

### **Models:**

Este directorio contiene los archivos con las clases, que representan los objetos de las tablas en la base de datos, y permite controlar algunas vistas que tendrá el usuario

- AccountBank
- Bank
- Category
- Chats
- Client
- Messages
- PackMoney
- Publication
- PublicationCopy
- PublicationType
- RecordBuyPack
- RecordRecharge
- ReportPublication
- RequestSeller
- Target
- Transfer
- User

### **Service**

Este directorio contiene las carpetas de los servicios utilizados para la implementación de las rutas que acceden a la api, para el manejo de la información

- Admin Service: Este archivo contiene los métodos utilizados para el consumo de la api, y a su vez tiene la variable que permite controlar la pantalla que se le mostrará al administrador
  - getGlobalVariableLimitMinPublication()
  - getPantalla()
  - setPantalla()



- obtenerAdmins()
- eliminarAdmin()
- updateUser()
- updateImage()
- addCategory()
- getCategory()
- findCategory()
- getPackMoney()
- addPackMoney()
- deletePackageMoney()
- getPublicationsUser()
- getReportPublicationUpdate()
- bloquearPublication()
- Guest Service: Este archivo contiene los métodos utilizados para el consumo de la api, y a su vez tiene la variable que permite controlar la pantalla que se le mostrará al usuario visitante
  - getPantalla()
  - setPantalla()
  - saveUser()
  - getUser()
  - loginUser()
  - getBank()
  - getPubliByCategory()
- Test Service: Este solo tiene un método para testear la conexión del frontend con la api
  - getTest()
- User Service: Este archivo contiene métodos utilizados para el consumo de la api, y a su vez tiene la variable que permite controlar la pantalla que se le mostrará al usuario, como la variable para mostrar el bank del mismo
  - getPantalla()
  - setPantalla()
  - setCurrentAplicationMoney()
  - getCurrentAplicationMoney()

- setCurrentMoney()
- getCurrentMoney()
- sendRequestSeller()
- getRequestsSeller()
- respondentRequestSeller()
- getPublicationType()
- getAllPublications()
- getAllPublicationsNoUser()
- addPublication()
- getMyPublications()
- getPublicationsPending()
- updateStatusPublication()
- reenviarPublication()
- updatePublication()
- updateAPlicationCUurrency()
- addTarjet()
- getTarjets()
- updateTarjet()
- deleteTarjet()
- getMyAccountBank()
- addAccountBank()
- updateAccountBank()
- deleteAccountBank()
- addRecordRecharge()
- addRecordPackMoney()
- getMyRecordPackMoney()
- comprarProducto()
- revisatVoluntariados()
- venderProdcuto()
- getPublicationsBued()
- reportPublication()
- getInfoPublicBloked()
- getChatUser()

- saveNewChat()
- verficiarChat()
- Transferir()
- getHistoryTranfer

## **Views**

### **Admin Views**

- Category-add: este directorio, contiene el componente para manejar las categorías de la aplicación o de las publicaciones
- create-user: este directorio, contiene el componente para manejar el formulario para el ingreso de un nuevo usuario y su manejo
- home -admin: este directorio, contiene el componente utilizado para el menú principal del administrador
- management-user: este directorio, contiene el directorio para ver a los usuarios administradores del sistema y manejar sus datos
- my-profile: este directorio, contiene el componente para ver el perfil del usuario y poder editarlo si amerita
- pack-money-add: este directorio, contiene el componente utilizado para añadir los paquetes de compras de monedas, y su administración
- report-publish: este directorio , contiene el componente que ve los reportes de las publicaciones para que el administrador pueda rechazar o aceptar dicho reporte
- request-publish: este componente es utilizado para que el administrador pueda ver las solicitudes para publicar, mientras sea menor a la cantidad dicha por la toma de requerimientos
- request-seller: este directorio, contiene el componente que se utiliza para administrar las peticiones de ser vendedor del sistema

### **Guest View**

- login: este directorio contiene el componente para el formulario de inicio de sesión
- main: este directorio contiene el componente que se mostrará cuando se levante la aplicación
- register: este directorio contiene el componente para el formulario de registro de datos

## User Views

- chat-frontend: este directorio contiene el componente que maneja el chat del usuario
- collect-money: este directorio contiene el componente para comprar monedas del sistema, mediante tarjeta o cuentas de banco
- history-transfer: este directorio contiene el componente para poder mostrar el historial de transferencias enviadas y recibidas
- home-user: este directorio contiene el componente necesario para el menú del home del usuario normal
- publish: este directorio contiene el componente para publicar las diferentes publicaciones
- record-buys: este directorio contiene el componente para ver el historial de las publicaciones compradas
- records-pay: este directorio contiene el componente para ver el historial de los pagos realizados
- store: este componente contiene las publicaciones que andan a venta

## DBMS:

Un DBMS es un software que proporciona una interfaz para interactuar con una base de datos. Estos sistemas facilitan la creación, manipulación y administración de bases de datos, permitiendo a los usuarios almacenar, recuperar y gestionar eficientemente la información, y uno de los principales en la categoría de bases de datos relacionales es MySQL.



MySQL es un sistema de gestión de bases de datos relacional de código abierto, que sigue los principios de SQL (Structured Query Language), y por ende este gestor de bases de datos, posee las siguientes características:

- Es una base de datos relacional
- Basada en SQL
- Características ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)
- Soporte de Índices y Optimización de Consultas

## Instalación:

Se necesitara instalar PostgreSQL, deberá ejecutar los siguientes comandos en la terminal

- `sudo apt install mysql-server`
- `sudo systemctl status mysql`
- `sudo systemctl start mysql`

### Documentación MySQL

#### **Ejecución**

Después, para poder usar el proyecto, deberá ejecutar el Backend, y usando el siguiente comando:

- `php artisan serve`

Y para ejecutar el Frontend:

- `ng serve / ng s`

Y acceder a la BD:

- `mysql -u root -p`