

Documentación del Proyecto: Plataforma de Trading

Autor: Jesús Arranz Navas

1. Explicación de la Aplicación

Esta plataforma de trading permite a los usuarios simular la compra y venta de criptomonedas utilizando un monedero virtual en USDC.

Las principales funcionalidades incluyen:

- Registro e inicio de sesión (con correo y Google).
- Consulta de datos de mercado actualizados usando la API de CoinMarketCap.
- Visualización de gráficas de velas (OHLC) desde CoinGecko.
- Simulación de compra y venta de Bitcoin.
- Depósito virtual de fondos.
- Cálculo automático de la equivalencia de criptomonedas.
- Vista del valor total de la cartera (balance en USDC + valor de BTC).

2. Cumplimiento de la Rúbrica

- ****Uso de bases de datos y operaciones CRUD****: se utilizan modelos para registrar el balance virtual (VirtualBalance) y las criptomonedas adquiridas (CryptoHolding). Las operaciones de compra/venta modifican estos registros.
- ****Trabajo con APIs externas****: CoinMarketCap y CoinGecko son usadas para obtener información actualizada del mercado y gráficas.
- ****Decoradores y login_required****: se protege el acceso a todas las vistas mediante autenticación.
- ****Rutas limpias y funcionales****: como /buy-btc/, /sell-btc/, /coin-details/, /index-2/.
- ****Pruebas automáticas****: se implementan tests que cubren depósitos, compras con fondos suficientes e insuficientes, venta de BTC, y acceso a vistas.
- ****Buena estructura de vistas y separación de responsabilidades****: las vistas están organizadas en funciones específicas por funcionalidad.
- ****Diseño responsive con Bootstrap****: aunque se eliminó la vista /portfolio/, el sistema de balance y operaciones sigue funcionando correctamente.

3. Instrucciones de Ejecución

1. Crear y activar un entorno virtual:

```
python -m venv .venv  
.venv\Scripts\activate (Windows)
```

2. Instalar los requisitos:

```
pip install -r requirements.txt
```

3. Crear la base de datos:

```
python manage.py migrate
```

4. Ejecutar el servidor:

```
python manage.py runserver
```

5. Acceder a la app en <http://127.0.0.1:8000/>

6. Ejecutar los tests:

```
python manage.py test
```