# Objetivo del proyecto:

Dahsboard para visualizacion de cotizaciones y operaciones de criptomonedas

Diseño:

El diseño debe ser elegante con colores en gamas azules y negros, que sea responsive.

Divide la pantalla en dos secciones, donde la seccion cotizaciones debe estar en la parte superior con margen de entre 2 a 3 centimentros de la partee superior. Debajo tendra las cajas de las diferentes criptomonedas obtenidas desde el excel y sus cotizaciones respectivas. A partir de la mitad de la pantalla para abajo que se muestren el titulo de operaciones y debajo la tabla con las operaciones obtenidas desde el Excel.

El codigo debe contemplar el numero de criptomonedas a mostar y ajustar el numero de cajas y sus respectivas posiciones para que quede centrado de la forma que esta en el diseño. O sea, si hay 5 criptos entonces tendremos 5 cajas distruidas horizontalemente, donde los margenes izquierdo y derecho deben estar alineados con la caja del titulo "cotizaciones".

Las funciones deben tenr todas manejo de errores que sea elegante y un log detallado en caso de errors.

Estructura del Proyecto:

app.py: Archivo principal de Flask.

templates/index.html: Archivo HTML para la interfaz.

operaciones.xlsx: Archivo Excel con los datos de las operaciones.

Backend flask:

Importar las librerías necesarias.

(ejemplo: from flask import Flask, render\_template, jsonify

import requests

import pandas as pd

import logging)

Configurar la aplicación Flask y el registro de logs

Crear un diccionario de mapeo de nombres de criptomonedas

Crear Función para obtener precios de criptomonedas

Crea Función para obtener símbolos de criptomonedas desde Excel

Crea Función para obtener operaciones desde Excel

Configurar las rutas en Flask

Frontend:

Crea el archivo HTML (index.html) en base al diseño que te solicito, tengo un template si lo necesitas.

Templates:

Tengo el archivo python y el HTML como base pero no funcionan del todo bien, puedo subirlo si lo necesitas, o si quieres hacerlo en REACT u otra tecnologia entonces has la sugerencia.

Tambien puedo subirte un ppt con el diseño que realice.

1. ¿Podrías especificar qué información quieres que se muestre en las cajas de las criptomonedas? ¿Solo el precio actual o también otros datos como el volumen, el cambio porcentual, etc.?

Precio actual y tambien cambio porcentual en la ultima semana

1. ¿Qué frecuencia de actualización en tiempo real esperas para las cotizaciones de las criptomonedas?

Que se actualice cada 30 minutos con la opción de tener un botón para actualizar manualmente.

1. ¿Prefieres algún servicio específico para obtener los datos en tiempo real (por ejemplo, CoinGecko, Binance API, etc.)?

La que sea gratis y además incluye una lógica para informar si la API deja de responder porque se acaba la cantidad de consultas.

1. ¿Qué detalles específicos deben incluir las operaciones obtenidas desde el Excel (por ejemplo, fecha de la operación, tipo de operación, cantidad, precio, etc.)?

Lo mas importante es el porcentaje de ganacia el cual debe ser comparada con la cotización actual. Me gustaria además que si defino un 10% de ganancia, al calcular la ganacia actual de la operación si es negativa entonces que este en color rojo toda la columna, si es positivo que este en color verde.

1. ¿Tienes alguna preferencia en cuanto a la librería de gráficos o componentes de interfaz de usuario para React (por ejemplo, Material-UI, Chart.js, etc.)?

La que permita mayor flexibilidad y sea mas elegante

1. ¿Cómo prefieres manejar los posibles errores en la obtención de datos, tanto desde las APIs de criptomonedas como desde el archivo Excel?

Podrias incluir una caja en la parte inferior de la pantalla donde se indique los errores al obtener cotizaciones u operaciones. Que este traducido al castellano y si puedes también indica la solución para este error o problema.

1. ¿El diseño debe incluir algún tipo de interacción adicional, como filtros o búsqueda en la tabla de operaciones?

Si, que permita filtrar.

1. ¿Tienes alguna preferencia en cuanto a la tecnología para el backend si se decide cambiar de Flask a otro framework?

De momento estoy utilizando FLASK en mi ordenador, pero si tienes alguna sugerencia indicala. Considera que siempre incluire operaciones en el Excel y desde ahí se deben tomar los datos. Ahora si tienes alguna alternativa de fácil implementación que sea mejor que esto, ofrecela.

1. ¿Hay algún detalle adicional que consideres importante incluir en el dashboard?

De momento ninguna.

Tabla Operacions

Quiero que la tabla operaciones tengo solamente las siguientes columnas:

* Criptomoneda
* Precio de Compra
* Cantidad
* Dinero Invertido
* Moneda
* % Ganacia Objetivo
* Precio Objetivo
* % Ganancia Actual
* Ganancia Neta Actual

Quiero que estas columnas tengan la siguiente función, ajusta el codigo de la app para que cumpla con lo solicitado:

* Criptomoneda: Codigo de la criptomoneda
* Precio de Compra: Precio obtenido desde el Excel de la columna “Precio de Compra”
* Cantidad: Precio obtenido desde el Excel de la columna “Cantidad”
* Dinero Invertido: Precio obtenido desde el Excel de la columna “Dinero Invertido”
* Moneda: Precio obtenido desde el Excel de la columna “Moneda”
* Porcentaje Ganancia Objetivo: Precio obtenido desde el Excel de la columna “Porcentaje Objetivo de Ganancia”
* Precio Objetivo: Calcula el precio objetivo de la cripto en base al precio de compra y el porcentaje de ganancia objetivo. Esto es para conocer el precio en el cual debería vender esas criptomonedas.
* % Ganancia Actual: Calcula el porcentaje de ganancia usando el precio de compra y el precio actual que obtienes de Coingeko. Si la ganancia es mayor a cero entonces que este en color verde y si es menor a cero puedes expresarla en negativa y que este de color rojo.
* Ganancia Neta Actual: calcula la ganancia neta en la moneda fiat correspondiente. Este valor debería reflejar la ganacia que obtendría si vendiese la criptomoneda en el acto.

# Mejoras

1. Alertas personalizadas:
   * Implementa un sistema de alertas que permita al usuario establecer sus propios criterios (por ejemplo, cuando el precio alcance cierto nivel o cuando el RSI esté en cierto rango).
   * Usa notificaciones push o correos electrónicos para alertar incluso cuando no estés viendo el dashboard.
2. Análisis de tendencias:
   * Añade gráficos de velas (candlestick charts) para cada criptomoneda que estés siguiendo.
   * Implementa indicadores técnicos adicionales como Medias Móviles, MACD, o Bandas de Bollinger.
3. Integración con exchanges:
   * Conecta tu dashboard con las APIs de los exchanges que usas para obtener datos en tiempo real de tus operaciones abiertas.
4. Análisis de sentimiento:
   * Incorpora un feed de noticias relevantes y análisis de sentimiento del mercado.

Para hacer tu proyecto más flexible y accesible, considera estas opciones:

1. Migración a una aplicación web completa:
   * Usa un stack MERN (MongoDB, Express, React, Node.js) para crear una aplicación web completa.
   * Esto te permitiría acceder a tu dashboard desde cualquier dispositivo con conexión a internet.
2. Implementación de una API backend:
   * Crea una API propia que maneje la lógica de negocio, cálculos y almacenamiento de datos.
   * Esto separaría la lógica del frontend, haciendo tu aplicación más escalable.
3. Uso de servicios en la nube:
   * Considera usar servicios como AWS, Google Cloud o Azure para hostear tu aplicación.
   * Esto te permitiría implementar funciones serverless para tareas como el cálculo de indicadores o el envío de alertas.
4. Implementación de una Progressive Web App (PWA):
   * Convierte tu aplicación React en una PWA para que pueda ser instalada en dispositivos móviles y funcionar offline.
5. Base de datos en tiempo real:
   * Usa una base de datos en tiempo real como Firebase para sincronizar datos instantáneamente entre diferentes dispositivos.
6. Containerización con Docker:
   * Usa Docker para empaquetar tu aplicación y sus dependencias, facilitando su despliegue en diferentes entornos.
7. Implementación de WebSockets:
   * Usa WebSockets para recibir actualizaciones en tiempo real de precios y otros datos relevantes.
8. Autenticación y autorización:
   * Implementa un sistema de login para que puedas acceder a tu dashboard de forma segura desde cualquier lugar.

DISEÑO

El diseño debe ser elegante con colores en gamas azules y negros.

Titulo: DASHBOARD

Divide la pantalla en tres secciones:

* Cotizaciones: debe estar en la parte superior izquierda. Contendra 3 cajas para representar la cotizaciones indicadas.
* Indicadores: debe estar en la parte superior derecha y contendrá indicadores que desarrollaremos, como ser un medidor gauge.
* Operaciones: De la mitad de la pantalla para abajo incluye una tabla con las operaciones que obtendrás desde el Excel.

**Modularización de componentes**

1. **Bloques de contenido**: se crearán bloques de texto, imágenes, botones y otros elementos que se puedan combinar y reorganizar según sea necesario.
2. **Layouts**: se definirán layouts flexibles que permitan ajustar la posición y tamaño de los bloques de contenido en función del tamaño de la pantalla.
3. **Estilos y temas**: se crearán estilos y temas predefinidos que permitan personalizar la apariencia de la pantalla sin afectar su estructura y diseño general.

**Alineación y distribución**

1. **Grid system**: se utilizará un sistema de grid para definir las áreas de la pantalla y asegurarse de que los componentes se alineen correctamente.
2. **Responsive design**: se diseñará la pantalla para que se adapte a diferentes tamaños y orientaciones de pantalla, garantizando una experiencia óptima en dispositivos móviles y de escritorio.
3. **Margenes y padding**: se utilizarán margenes y padding para crear espacio entre los componentes y evitar colisiones.

**Edición y mantenimiento**

1. **Drag-and-drop**: se permitirá arrastrar y soltar los bloques de contenido para reorganizar la pantalla y editar su contenido sin afectar el diseño general.
2. **Configuración de estilos y temas**: se permitirá personalizar los estilos y temas sin afectar la estructura y diseño individual de los componentes.
3. **Versiones y historial**: se mantendrá un historial de versiones y cambios para poder revertir a versiones anteriores si es necesario.