

Pregunta 1: Desarrollo y Deploy de Servicio de Optimización de Portafolios

Crea un servicio en Python para estimar portafolios óptimos a partir de retornos diarios y publícalo como API REST.

Requisitos del servicio:

- Método HTTP: [POST](#)
- Endpoint: [/optimize-portfolio](#)
- Formato de entrada: [multipart/form-data](#)

Parámetros de entrada:

- [file](#): archivo CSV con los retornos diarios del universo de activos en formato ticker/fecha (mira el ejemplo en este [link](#), el header es el ticker).
- [risk_level](#) (float): nivel máximo de riesgo permitido.
- [max_weight](#) (float): peso máximo permitido por activo.

Respuesta esperada (formato JSON):

```
{ "optimal_portfolio": {  
    "ticker_1": weight_1,  
    "ticker_2": weight_2,  
    ...  
}
```

Ejemplo de solicitud CURL:

```
curl -X POST http://<YOUR_SERVICE_URL>/optimize-portfolio \  
-H "Content-Type: multipart/form-data" \  
-F "file=@returns.csv" \  
-F "risk_level=1.0" \  
-F "max_weight=0.15"
```

Puedes definir libremente:

- La métrica de riesgo (vol, Var, CVar, beta, etc.).
- El criterio de optimalidad.
- El modelo de optimización a utilizar (ej. Markowitz, Black-Litterman, Uryasev, etc.).

Entrega esperada:

- Repositorio en GitHub (o archivo [.zip](#)) con el código fuente. Incluye en el [README.md](#) qué método de optimización usaste y por qué lo elegiste.
- URL del servicio desplegado, accesible públicamente para pruebas.

Pregunta 2: Arquitectura del Servicio para Producción

Propón una arquitectura técnica para disponibilizar el servicio desarrollado en la Pregunta 1 como una solución escalable y accesible para clientes, incluyendo:

- Componentes principales (API, orquestación, despliegue, frontend si aplica, etc.).
- Herramientas y tecnologías sugeridas (por ej.: Docker, Kubernetes, servicios cloud, CI/CD, monitoreo, etc.).
- Justificación breve de tus decisiones técnicas.