

## Resumen

En esta etapa los equipos deben ejecutar y modificar la aplicación web con el Framework Flask del proyecto con la información cargada y calculada de las etapas 2.1 y 2.2.

## Sobre Flask

Flask es un marco de desarrollo web ligero y flexible para Python. Fue creado para facilitar la creación de aplicaciones web de manera rápida y sencilla, centrándose en la simplicidad y la extensibilidad. Flask no impone una estructura rígida para la aplicación, lo que significa que tienes la libertad de elegir las herramientas y bibliotecas que desees utilizar.

**Microframework:** Flask es considerado un microframework porque proporciona solo las funciones básicas necesarias para crear aplicaciones web. Aunque es ligero, es lo suficientemente flexible como para escalar y adaptarse a aplicaciones más grandes.

Algunas de sus características son:

1. **Extensibilidad:** Flask utiliza un sistema de extensiones que permite agregar funcionalidades específicas de manera modular. Puedes elegir las extensiones que necesitas para tu aplicación y dejar fuera las que no necesitas.
2. **Rutas y vistas:** Flask utiliza un sistema de rutas para asociar funciones (vistas) con URL específicas. Esto facilita la creación de endpoints de API y la gestión de las solicitudes del usuario.
3. **Plantillas:** Flask admite el uso de plantillas, lo que facilita la generación dinámica de contenido HTML. Jinja2 es el motor de plantillas predeterminado en Flask.
4. **Integración con otras tecnologías:** Flask se integra fácilmente con otras bibliotecas y herramientas de Python, como SQLAlchemy para la manipulación de bases de datos, WTForms para la validación de formularios y más.

## Cómo ejecutar la aplicación Flask

Para ejecutar la aplicación Flask, los equipos deben utilizar el siguiente directorio que pueden descargar en el siguiente enlace:

[https://drive.google.com/drive/folders/1D3XDn0FrmompP6VVPyjiM4GrxOh2EdEL?usp=drive\\_link](https://drive.google.com/drive/folders/1D3XDn0FrmompP6VVPyjiM4GrxOh2EdEL?usp=drive_link)

Este directorio consta de lo siguiente estructura:

```
mi_aplicacion/  
|-- static/  
|   |-- style.css  
|-- templates/  
|   |-- detalle-fondo.html  
|   |-- index.html  
|   |-- layout.html  
|   |-- portafolios.html  
|   |-- rentabilidades.html  
|-- uploads/  
|   |-- detalle_fondo.csv  
|   |-- portafolio_internacional.csv  
|   |-- portafolio_nacional.csv  
|   |-- rentabilidades_acumuladas.csv  
|   |-- series.csv  
|-- Flask_app.py  
|-- README.md  
|-- requirements.txt
```

El directorio **"static/"** contiene el archivo **"style.css"**, el cual consta de un CSS **"Hojas de Estilo en Cascada"**. CSS es un lenguaje de estilo utilizado para describir la presentación de un documento. En el se definen las fuentes, colores, márgenes etc de la web. Esta directamente relacionado con el estilo gráfico de la web.

En la directorio **"templates/"** está compuesta por todos los archivos **"html"** de la página web. HTML es el lenguaje estándar utilizado para crear y diseñar páginas web. Es un estándar que describe la estructura básica de una página web mediante el uso de elementos y etiquetas.

Los directorios **"static/"** y **"templates/"** están asociados **a la vista** en el modelo MVC.

En el directorio **"uploads/"** es donde se cargan los datos que se muestran en la página web. En particular estos archivos están vacíos. **Los equipos deben completar la información según lo calculado y cargado en la etapa 2.1 y 2.2 de la siguiente forma:**

- **"detalle\_fondo.csv", "portafolio\_nacional.csv", "portafolio\_internacional.csv" y "series.csv":** aquí debe cargar la tablas cargadas correspondientes de la parte 2.2 del proyecto, en particular debe cargar la tabla como un csv separado por **";"** (si lo hace con otra separación no lo tomará).

- “rentabilidades\_acumuladas.csv”: en esta parte los equipos deben cargar solo la tabla de rentabilidades acumuladas de la parte 2.1 del proyecto, en particular debe cargar la tabla como un csv separado por “;” al igual que el anterior.

De los archivos cargados anteriormente. ES RELEVANTE QUE LAS COLUMNAS ASOCIADAS AL RUN SE LLAMEN “Run Fondo” Y LAS ASOCIADAS A LA FECHA DE LA INFORMACIÓN SE LLAME “fecha”. Si no las tiene con este nombre debe modificarlas en los archivos csv. A continuación un ejemplo:

```
fecha;Run Fondo;erie;1 mes;3 meses;6 meses;1 año
2021-05-03;10002;A;3%;12%;11%;1%
2021-05-01;10000;B;5%;7%;10%;3%
2021-05-02;10002;APV;5%;7%;10%;2%
```

Se podría considerar que el directorio “uploads” está asociado al “Modelo” en MVC

El archivo “Flask\_app.py” es a un archivo Python que sirve como punto de entrada para una aplicación Flask. Este archivo es el que tiene la configuración y lógica de la aplicación. Este archivo está asociado a el “Controlador” en el modelo MVC. Es este archivo el que deben finalmente ejecutar.

El archivo “README.md” aparecen todas las instrucciones de la creación del ambiente virtual para ejecutar la aplicación y para instalar las librerías requeridas por el proyecto.

Finalmente, el archivo “requirements.txt” contiene el nombre de las librerías que serán instaladas en el ambiente virtual.

Si siguen los pasos se ejecutará una aplicación en su navegador en la ruta “<http://127.0.0.1:5000/>” como se muestra a continuación:



**App Fondos Mutuos**

**Descubre los Fondos Mutuos en Chile**

**Inversión Inteligente para Crecimiento Financiero**

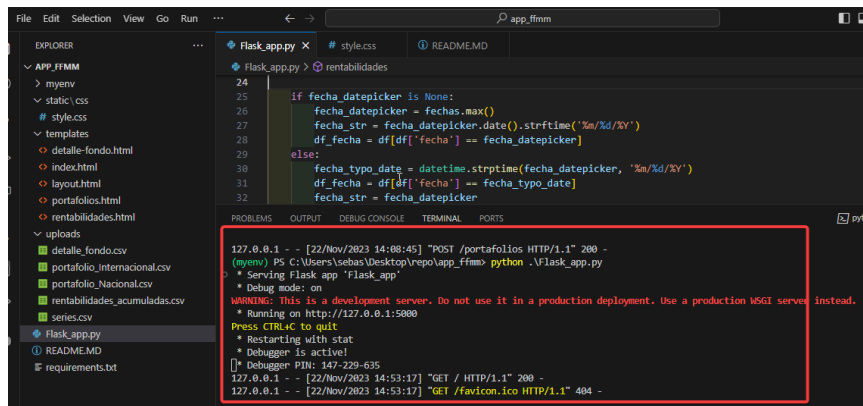
**¿Qué son los Fondos Mutuos?**

Los Fondos Mutuos en Chile representan una opción de inversión colaborativa, permitiendo a los inversionistas unir sus recursos para acceder a carteras diversificadas de activos financieros. El método de inversión es gestionado por Administradoras Generales de Fondos (AGF) de la Superintendencia de Valores y Seguros (SVS).

**Diversificación para Mitigar Riesgos**

Los Fondos Mutuos agrupan inversiones en una variedad de instrumentos financieros: bonos y otros valores. Esta diversificación reduce el riesgo asociado con la inversión.

La terminal se verá como sigue:



```

24
25 if fecha_datepicker is None:
26     fecha_datepicker = fechas.max()
27     fecha_str = fecha_datepicker.date().strftime('%m/%d/%Y')
28     df_fecha = df[df['fecha'] == fecha_datepicker]
29 else:
30     fecha_tipo_date = datetime.strptime(fecha_datepicker, '%m/%d/%Y')
31     df_fecha = df[df['fecha'] == fecha_tipo_date]
32     fecha_str = fecha_datepicker
  
```

```

127.0.0.1 - - [22/Nov/2023 14:08:45] "POST /portafolios HTTP/1.1" 200 -
(myenv) PS C:\Users\sebas\Desktop\repo\app_ffmm> python .\Flask_app.py
* Serving Flask app 'Flask_app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 147-229-635
127.0.0.1 - - [22/Nov/2023 14:53:17] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/Nov/2023 14:53:17] "GET /favicon.ico HTTP/1.1" 404 -
  
```

Cuando ejecute la aplicació, en la barra lateral podrá navegar en 3 partes:

- “Sobre FFMM”: aparecerá una página con información de fondos mutuos (imagen anterior).
- “Rentabilidades FFMM”: aquí aparecerán las rentabilidades para un día particular que elija:

## Rentabilidad FFMM

Seleccionar Fecha:

fecha	Run Fondo	serie	1 mes	3 meses	6 meses	1 año
2021-05-03	<a href="#">10002</a>	A	3%	12%	11%	1%

Además, podrá hacer click en el run del fondo y ver el detalle del fondo y las series que contiene.

- “Portafolio FFMM”: aquí podrá seleccionar un día y revisar la cartera nacional o internacional de un fondo seleccionado:

Fondos Mutuos
  
  
Rentabilidades FFMM
  
  
Portafolio FFMM

### Portafolio FFMM

Seleccionar Fecha:

Seleccionar cartera:

Seleccionar fondo:

fecha	Run Fondo	nombre_fondo	nombre_corto	rut_administradora	nombre_administrado
2023-03-31	<a href="#">9765</a>	FONDO MUTUO BANCOESTADO MI FUTURO ACCESIBLE	MI FUTURO ACCESIBLE	96836390	BANCOESTADO S.A. ADMINISTRADORA GENERAL DE FONDOS

**ES MUY IMPORTANTE SEÑALAR QUE SI AL SELECCIONAR UNA FECHA EN PARTICULAR Y NO LE APARECE INFORMACIÓN ES PORQUE NO CARGÓ LOS DATOS CORRECTAMENTE EN EL DIRECTORIO “uploads/”**

### **¿Qué deben presentar los equipos?**

Además de realizar la carga correctamente en el directorio “/Uploads” y ejecutar la aplicación Flask los equipos deben crear un repositorio en “GIT” con Github con al menos 5 commits con modificaciones “estéticas” (esto quiere decir en los htmls o CSS) y enviar el link del repositorio el día anterior a la presentación. El día de la presentación deben mostrar al profesor la aplicación ejecutándose con el repositorio con los 5 commits.

Algunas consideraciones:

- No debe realizar “push” del ambiente virtual. Para eso debe crear el .gitignore.
- Bonus: Si realiza modificación lógica en el archivo Flask\_app.py para agregar una nueva pestaña en la barra lateral que muestre una página interesante tendrá puntos adicionales.
- El día anterior a la presentación debe enviar el “esquema de la base de datos” corregido.
- La cantidad de columnas que tienen los archivos csv (que provienen de sus tablas) las tomará igual Flask. Solo importa que las columnas asociadas al run se llamen “Run Fondo” y las asociadas a la fecha de la información se llamen “fecha” las demás columnas las puede mantener como aparecen en su base de datos.