# Memoria de EDA

Escogí estudiar los efectos de los indicadores económicos en los mercados bursátiles de Estados Unidos por que el mundo de las inversiones es mi pasión y podré utilizar este análisis como herramienta para identificar posibles tendencias de mercado y oportunidades de compra o venta.

## Extracción de Datos

### Extracción de Datos Económicos

El primer reto que me encontré fue la extracción de datos económicos. Resultó bastante complicado ya que hay muchas páginas web y entidades que ofrecen dichos datos de manera poco fiable. Sin embargo, encontré una la página web oficial del banco central estadounidense ([FRED](https://fred.stlouisfed.org/)); que ofrece una API pública y gratuita para la extracción de una infinidad de datos económicos y de mercado.

Al aprender a utilizar su API a través de los códigos de los indicadores económicos y de la librería de Python que ofrecen, desarrollé una función que se encarga de extraer los datos de los indicadores deseados de manera dinámica.

### Extracción de Datos de Mercado

El segundo reto fue la extracción de los datos de mercados bursátiles de USA. Encontré una librería de Python que extrae los datos de una acción, índice bursátil o un fondo de inversión individual a través de la API de [Yahoo Finanzas](https://finance.yahoo.com/). Desarrollé una función para extraer múltiples índices de manera dinámica.

Ambas extracciones guardan los datos en la carpeta ‘datos/raw’ del proyecto para su posterior limpieza y procesado.

## Limpieza de Datos

Para limpiar los datos, primero se extraen de la carpeta ‘raw’ y se definen las variables del filtro temporal (start y end) con un rango dinámico establecido por defecto desde hoy hasta hace 23 años.

### Limpieza de Datos Económicos

Para preparar y limpiar los datos económicos se le aplica la máscara que filtra los datos extraídos según el filtro temporal preestablecido.

Los diferentes datos económicos están dispuestos en diferentes franjas temporales (diario, semanal, mensual, trimestral o anual). Ya que la tabla global de datos económicos de ‘raw’ dispone todos los indicadores en la misma serie temporal, está repleta de celdas vacías (Nan).

Para abordar ese problema, se ‘limpió’ la tabla global rellenando las celdas vacías con los datos anteriores y (en el caso de que los vacíos aparecieran al inicio de la serie temporal) se rellenaron los datos vacíos con el primer dato encontrado.

Este método solo nos sirve para la visualización de los datos históricos, no se ha utilizado para realizar el análisis de los datos comparándolos con los mercados bursátiles.

Para obtener los datos que se utilizarán en el análisis, se hizo lo siguiente:

1. Se genera un diccionario que contiene los tipos de indicadores (keys) y una lista con cada uno de los indicadores (values)
2. Se utiliza una función que:
   1. Lee la tabla global de ‘raw’
   2. Elimina los datos vacíos
   3. Guarda los datos limpios en la carpeta ‘data/processed’

### Limpieza de Datos de Mercado

Para preparar y limpiar los datos de mercado se le aplica una máscara que filtra los datos extraídos según el filtro temporal preestablecido.

Posteriormente, a partir de la extracción inicial, se generan dos tablas de datos calculadas que se utilizarán en el análisis:

1. Se calcula el rendimiento diario de los mercados (la variación diaria en el valor de los mercados)
2. Se calcula el rendimiento acumulado de los mercados (la acumulación de las variaciones diarias en el tiempo)
3. [En desarrollo] Se replican los pasos anteriores para las diferentes franjas temporales (semanal, mensual, trimestral, anual) para comparar los datos en la misma franja que los datos económicos.

Finalmente, se guardan los datos procesados en la carpeta ‘data/processed’

## Análisis Univariante

Para el análisis univariante, se creó una función que calcula varias medidas de tendencia central, variabilidad y forma, y ​​crea un histograma y un diagrama de caja usando seaborn. También crea una tabla HTML de las medidas para cada indicador y la muestra junto con los gráficos. Finalmente, la función devuelve el marco de datos "df\_measures" con las medidas calculadas para cada indicador.

## Análisis Multivariante

Para el análisis Multivariante, se creó una función que hace lo siguiente:

Para cada indicador y su correspondiente DataFrame en 'indicators':

* Calcula el cambio porcentual del indicador para obtener las variaciones temporales.
* Guarda ese DataFrame como un archivo csv.
* Fusiona los datos de variaciones del indicador y los datos de rendimiento del mercado.
* Calcula los valores de correlación mínimos y máximos de los datos de rendimiento del mercado.
* Traza un mapa de calor de la matriz de correlación de los datos combinados.
* Traza un diagrama de caja de los datos combinados.
* Traza un diagrama de pares de los datos combinados.

## Análisis Bivariante

Para el análisis Bivariante, se creó una función llamada "analisis\_bivariante" que genera gráficos y tablas de análisis bivariante para datos bursátiles e indicadores económicos. La función toma varios argumentos que incluyen: la lista de nombres de índices de mercado a analizar, la lista de nombres de indicadores a analizar, fechas de inicio y finalización del análisis, los datos con rendimientos de mercado y rendimientos acumulados y el diccionario de datos de indicadores. Esta función devuelve los datos combinados de mercado e indicadores. La función entra en un bucle complejo que recorre los datos de cada índice de mercado para combinarlos y generar cuatro gráficos que muestran la distribución de los datos del mercado y del indicador, un gráfico de caja de los dos conjuntos de datos, un gráfico de regresión y un gráfico de líneas que muestra los rendimientos históricos combinados del mercado y del indicador. La función también calcula y muestra una tabla de correlación para cada combinación de mercado e indicador.

## Análisis de Regresión

[En desarrollo]

## Conclusiones

[En desarrollo]