

Brainstorm

- Servicio de aviso de noticias a Day traders para evitar ciertas franjas horarias de alta volatilidad y riesgo de deslizamiento de posiciones.
- La idea es crear un servicio automatizado de envío de emails a clientes para que puedan verificar las noticias de alto y mediano impacto que tendrán lugar ese día y los pares de divisa sobre la que aplica la noticia.
- Los datos se extraen durante la sesión Asiática previa a la apertura de la bolsa de Frankfurt y Londres.
- La página sobre la que aplicamos el Crawler para extraer estos datos es:
- <https://es.investing.com/economic-calendar/>
- Me centro en el mercado de Forex por ser el mercado con mayor volumen de transacciones diaria y debido a que las noticias que le afectan suelen ser noticias programadas de distintos organismos. Noticias tales como las IPC, decisiones de tipos de interés, inventario de crudo, o las Non-Farm Payroll.
- Todas estas noticias tienen fecha y hora y es buena practica por parte de los day traders, alejarse de los gráficos y de toda operativa durante el transcurso de esta noticia, para evitar picos de volatilidad.

Diseño del DAaaS

Definición la estrategia del DAaaS

Servicio de email diario automatizado con las noticias diarias durante las sesiones de operativa.

El servicio contara con una arquitectura basada en un Dataproc para el procesamiento de datos y un Crawler para la extracción de los datos.

Arquitectura DAaaS

- Google Cloud Functions + Email diario a los usuarios con envío del .csv.
- Job procesamiento + almacenamiento en Google Storage del .csv
- Google Cloud Dataproc(Hadoop) + Initialization actions(Job procesamiento)+ Scheduled deletion (1hora)
- Google Cloud Scheduler + Google Cloud Functions (crea cluster en Dataproc)
- CSV en Google Storage con datos diarios.
- Google Cloud Scheduler + Google Cloud Functions(lanza mi Crawler)
- Crawler de <https://es.investing.com/economic-calendar/> que almacena el .csv resultante en mi bucket de Google Storage.

DAaaS Operating Model Design and Rollout

- Una vez al día a las 5:00 am se lanza automáticamente con Google Cloud Scheduler la Google Cloud Functions que hace Crawler sobre mi pagina y almacena los datos en Google Storage.
- Una vez al día a las 5:30 se lanza automáticamente con Google Cloud Scheduler la Google Cloud Functions que crea un Cluster en dataproc programado para borrarse en 1 hora.
- Durante esa hora, mi Clúster con sus Initialization actions ejecuta un Job que hace que mi clúster procese los datos para que mi csv resultante solo contenga noticias de mediano y alto impacto y lo almacene en Google Storage de nuevo.
- Habiendo ya almacenado mi csv resultante en Google Storage, la última Cloud Function lo detecta y envía un email automáticamente a mis clientes con los datos del .csv.

Desarrollo de la plataforma DAaaS. (ligera descripción del desarrollo)

```
import scrapy

class BignewsSpider(scrapy.Spider):
    name = "bignews"
    allowed_domains = ["es.investing.com"]
    start_urls = ["https://es.investing.com/economic-calendar/"]

    def parse(self, response):
        # Find all items with class 'js-event-item'
        events = response.css('.js-event-item')

        for event in events:

            contry = event.css('span[title]::attr(title)').get() # Extract the
country affected by the news
            hour = event.css('td.js-time::text').get() # Extract the time of the
news (Madrid - Spain)
            level = event.css('td.noWrap::attr(data-img_key)').get() # Extract the
level of relevance
            volatility = event.css('td.noWrap::attr(title)').get() # Extract
commentary on volatility
            news = event.css('td.event a::text').get() # Extract the headline of
the news
            link = event.css('td.event a::attr(href)').get() # Extract the link
from the news
            if contry:
                yield {
                    'contry': contry.strip(),
                    'hour': hour.strip(),
                    'level': level.strip(),
                    'volatility': volatility.strip(),
                    'news': news.strip(),
                    'link': link.strip(),
                }
```

Link a Diagrama:

