Brainstorm

- Sistema de recomendación de vuelos.
- Servicios web de recomendación de planes de viajes.
- Servicio web para recomendar destino de viajes con baja demanda.
- Salud, Detección de anomalías examen de cáncer de mama.
- Predicción de demanda en cervecería.
- Servicio de notificaciones push para predecir picos de demanda en granja de aguacates.

Diseño del DAaaS

Definición la estrategia del DAaaS

Definir el catálogo de servicios que proporcionará la plataforma DAaaS, que incluye incorporación de datos, limpieza de datos, transformación de datos, datapedias, bibliotecas de herramientas analíticas y otros.

Idea Seleccionada:

- **Servicio de notificaciones push para predecir picos de demanda en granja de aguacates.**

La estrategia de DAaas que se utilizara consiste en recopilar y analizar datos en tiempo real de sensores en la granja, información histórica de producción de cosechas pasadas almacenados en sistemas propios donde se tiene todo el control de la operación (volúmenes de producción, costos, ventas, etc.) y datos externos de tiendas en línea así como de eventos especiales como por ejemplo el Super Bowl, Carnavales Nacionales, Ligas de Fútbol, etc. donde se identifique el comportamiento de la demanda del producto históricamente.

El objetivo principal con esta recopilación y análisis de datos consiste en hacer un pronóstico en las fluctuaciones de la demanda de aguacates, permitiendo a los agricultores optimizar su producción, inventario y estrategias de precios, maximizando los ingresos y reduciendo el desperdicio, de esta manera la granja de aguacates estará adaptándose a las dinámicas del mercado.

Finalmente, la notificación alertara a los usuarios de la probabilidad de un futuro pico en la demanda de aguacates.

Arquitectura DAaaS

Definir la selección de componentes, la definición de procesos de ingeniería y el diseño de interfaces de usuario. Diseño y ejecución de Proofs-of-Concept (PoC) para demostrar la viabilidad del enfoque DAaaS.

Componentes

1) Adquisición de datos de sensores.

- **Servicio de Pub/Sub de Google Cloud** como puntos de recepción de mensajes bajo protocolo de comunicación liviano MQTT para la lectura de los datos de los sensores.

- **Servicio Google Cloud Dataflow** para procesamiento de streams de datos recibidos de los sensores, limpieza de los datos e ingesta de estos en Google Cloud Bigtable y Dataproc.

- **Servicio Google Cloud Bigtable**, almacenamiento de los datos procesados provenientes de los sensores en una base de datos noSQL. Utilizamos este tipo de base de datos puesto que la integridad referencial no es un requisito obligatorio para el almacenamiento de los datos.

2) Adquisición de datos mediante carga manual de sistemas de información propios.

- **Servicio Google Cloud DataFlow rol ETL**, se utiliza este componente como punto de integración entre los sistemas de información propios (On-Premise) y la nube de Google, mediante este componente se clasificará la información para posteriormente enviarla a Google Cloud Dataproc.

3) Ejecución Manual de Crawler

   El sistema además de leer datos de sensores en campo y de alimentarse de sistemas de información propios de la granja con información histórica del funcionamiento de esta, ejecutara un crawler a diferentes sitios de interés para extraer información de los precios de aguacates en almacenes de cadena (en este caso se adjunta Crawler básico a mercadona.es).

- **Servicio de google Cloud Compute Engine**, se dispondrá de una máquina virtual para ejecutar el crawler que estará dirigido a páginas web de almacenes de cadena y tiendas de pequeñas y mediana superficie con el objetivo de obtener los datos de los aguacates semana a semana, así como estimar un stock.

4) Almacenamiento de documentos Google Cloud Storage

   Posterior a la lectura de los sensores, de sistemas de información propios y el crawler a páginas web de E-Commerce de supermercados y tiendas, se procede a almacenar el resultado de los procesos mencionados en la adquisición de datos en el servicio de Google Cloud Storage.

- **Servicio de google Cloud Storage**, se almacenarán archivos generados por los procesos de adquisición de datos y adicionalmente un **dataset de Kaggle** con la información de consumo de productos por eventos relevantes en nuestro estudio (Super Bowl, torneos, festividades, etc). Adicionalmente se almacenará el resultado con la predicción/pronostico realizado en Dataproc.

5) Análisis con BigQuery

- **Google Gloud BigQuery,** posterior al procesamiento de datos en dataproc se procederá a guardar el resultado en Google Cloud Storage, pero también en BigQuery, la cual será una herramienta destinada para quitar carga al sistema general y realizar tareas de reportes y análisis de negocio.
-

6) Notificación push

- **Google cloud Pub/Sub,** se utilizará como punto final de salida con el proceso de forecasting, este estará conectado a la salida de dataproc.

- **Google Cloud Compute Engine,** se utilizará una segunda máquina virtual para procesar el resultado y formatear el mensaje que se enviara a Google Cloud App Engine y que se transformara en la notificación Push.

- **Google Cloud App Engine,** será el punto final de salida del sistema donde ser enviará la notificación push final a los usuarios encargados de la administración y operación en la granja de aguacates.

Inventario final de servicios en la nuble

- **Google Cloud Pub/Sub Streaming** - Puntos para lectura de sensores.
- **Google Cloud Dataflow** - Transaction streams.
- **Cloud BigTable** - NoSQL Database.
- **ETL Dataflow** - Ingesta e integración de datos de sistemas propios.
- **Dataproc** - Análisis de la información proveniente de los procesos de adquisición de datos.
- **Crawler** a páginas web de supermercados y tiendas en línea - Máquina virtual en Google Compute Engine (Página objetivo mercadona.es).
- **Dataset de Kaggle**, información sobre eventos relevantes en el año (super Bowl, festividades, torneos, etc).

- **Google Cloud Storage** - Almacenamiento de archivos resultantes de los procesos de adquisición de datos (Crawler, sistemas propios, sensores) y el dataset de kaggle.
- **Google Cloud BigQuery -** Análisis de reportes y analiticas.
- **Máquina virtual en Google Compute Engine,** formateo de mensaje push.
- **Google Cloud Pub/Sub,** punto final de salida hacia la notificación push.
- **Google App Engine** para notificaciones Push.

DAaaS Operating Model Design and Rollout

Personalizar los modelos operativos DAaaS para cumplir con los procesos, la estructura organizacional, las reglas y el gobierno de los clientes individuales. Realizar seguimiento de consumo y mecanismos de informe.

- Los datos de los sensores se enviarán en (tiempo real), sin embargo, dependiendo de la relevancia de estos se plantea configurar el envío diario o semanal de un promedio de las lecturas con el objetivo de disminuir el ancho de banda y los recursos de nube utilizados.

- La carga manual de los sistemas propios (Sistemas operativos de la granja y, sistema contable) se integrarán por medio de la ETL con frecuencia (semanal).

- El craweler se ejecutará con una frecuencia (diaria).

- Cuando el procesamiento de datos y la generación del pronóstico a cargo del servicio de Google Cloud Dataproc finaliza se dispararán tres tareas:

  1. Almacenamiento de archivo CSV en Google Cloud Storage, se almacenará con cada ejecución en este caso se realizará con frecuencia (diaria).

  2. Almacenamiento de pronóstico en Google BigQuery, al igual que el archivo CSV del punto 1 se almacenará con frecuencia (diaria).

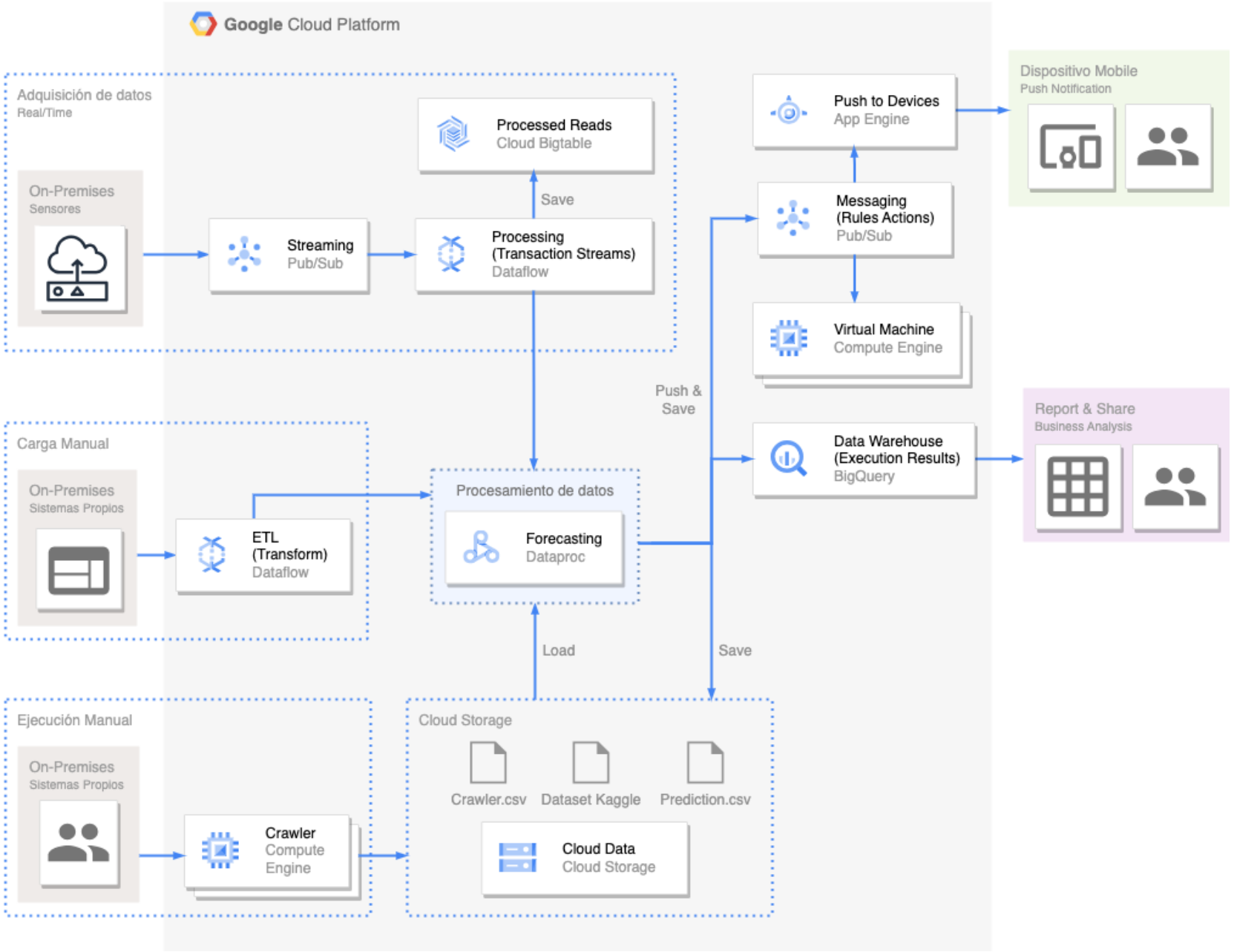
  3. La notificación push, en este caso la máquina virtual en Google Compute Engine se encargará de determinar si hay una anomalía en las lecturas y si se identifica un pico en la demanda de aguacates, de ser así, se disparará la notificación push por medio de Google App Engine por lo que no tiene una frecuencia de envío establecida.

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

Se adjunta un ejercicio básico de crawler escrito en Python a la página de mercadona.es

```
import requests
from bs4 import BeautifulSoup
def main():
    url = 'https://www.mercadona.es/es'
    response = requests.get(url)

    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        products = soup.find_all('div', class_='product')

        avocado_products = [product.find('span', class_='title').text
                            for product in products
                            if 'aguacate' in product.find('span', class_='title').text.lower()]

        if avocado_products:
            print("Productos relacionados con aguacates en Mercadona:")
            for product in avocado_products:
                print("-", product)
        else:
            print("No se encontraron productos relacionados con aguacates en Mercadona.")
    else:
        print("Error al conectar con la página de Mercadona.")

if __name__ == "__main__":
    main()
```

Diagrama:


Google Cloud Platform
Adquisición de datos
Real/Time
On-Premises
Sensores
Streaming
Pub/Sub
Processing
(Transaction Streams)
Dataflow
Save
Processed Reads
Cloud Bigtable
Push to Devices
App Engine
Dispositivo Mobile
Push Notification
Messaging
(Rules Actions)
Pub/Sub
Virtual Machine
Compute Engine
Push &
Save
Carga Manual
On-Premises
Sistemas Propios
ETL
(Transform)
Dataflow
Procesamiento de datos
Forecasting
Dataproc
Data Warehouse
(Execution Results)
BigQuery
Report & Share
Business Analysis
Load
Save
Ejecución Manual
On-Premises
Sistemas Propios
Crawler
Compute
Engine
Cloud Storage
Crawler.csv
Dataset Kaggle
Prediction.csv
Cloud Data
Cloud Storage