Practica Big Data Arquitecture

Carlos Javier Ávila Bermejo

Diseño del DAaaS

Definición la estrategia del DAaaS

Recogida de los datos producidos durante la jornada laboral en una compañía para ofrecer los siguientes servicios:

- Visualización del rendimiento de los empleados a través de una Web App con visualización en d3.js.
- Disponer un Data Warehouse para el área de RRHH que integre toda la información relevante de los empleados.
- Enviar mails automáticos con buenas prácticas para los empleados según su rendimiento.

Data Lake:

- Data corporativa:
    - Registros de actividad de dispositivos corporativos (Computadoras):
        - Registros de interacciones con el dispositivo (clicks y teclado).
        - Creación, edición y eliminación de documentos/directorios (Metadata).
    - Registros de actividad del dominio de las herramientas corporativas (Metadatos Suite Microsoft 365):
        - Interacciones de los empleados vía chat y mail.
        - Documentos y directorios compartidos.
        - Creación, edición, eliminación y compartición de documentos en Onedrive.
    - Datos corporativos sobre empleados:
        - Información sobre el puesto, categoría profesional, rango salarial y antigüedad (entre otros posibles).
- Data no corporativa:
    - BBDD sobre puesto, categoría profesional, rango salarial y perfiles profesionales del mercado (Instituto Nacional de Estadística).

Arquitectura DAaaS

La aplicación se soporta principalmente en una solución cloud mediante Google Cloud.

Partes de Software Cloud:

- BBDD SQL para los datos (Google Cloud SQL).
- Dos Web App (Google Cloud Compute) para dar acceso a los datos:
    - Web App Visualization d3.js.
    - Web App HR Data Warehouse.
- Un Web Servicies con una cuenta de correo para el envío automático de mails de recomendación (Levantamiento semanal).
- Clúster de Hadoop (Google Cloud Dataproc – levantamiento diario en horario no laboral). Para la recogida de la información se utilizan las siguientes APIs según su origen y se guardan en Bucket:
    - APIs de BBDD corporativas.
    - INE: Tempus3
    - Suite Office 365: API Azure AD
- KAFKA con dos Virtual Machine para la escucha y la publicación de los eventos recibidos de la aplicación de los registros de los dispositivos corporativos.
- Análisis de interacciones y nodos a través de los metadatos de eventos colaborativos (mails, chats, documentos compartidos,…) mediante la teoría de grafos con Python mediante la biblioteca NetworkX y d3.js.

Partes de Software Local:

- App que registra y envía a la VM Publisher de KAFKA los eventos del usuario al apagar el ordenador, tipo estructura de .json:

```
{
“Timestamp”: “Value”
“IDUser”: “Value”
“Events”: {
    “Clicks”: “NumberClicks”
    “Key_Board”: “Numberkeys”
    “CreateFile”: “NumberCreateFiles”
    “DeleteFile”: “NumberDeleteFiles”
    “EditFile”: “NumberEditFiles”
    “CreateDirectory”: “NumberCreateDirectories”
    “DeleteDirectory”: “NumberDeleteDirectories”
    “EditDirectory”: “NumberEditDirectories”
    “SecondsPowerOn”: “Value”
    “SecondsHibernate”: “Value”
    }
}
```

DAaaS Operating Model Design and Rollout

El proceso de ingesta de data se inicia con diferentes orígenes de datos que alimentan el Bucket del Google Storage:

A. BBDD Corporativas con datos de los empleados que mediante APIs específicas de las diferentes plataformas (Actualización Diaria).
B. BBDD No Corporativa que informa sobre sueldo de mercado de los diferentes perfiles, la salarios y la rotación media. Se informa mensualmente a partir de la API del INE Tempus3 y proporciona un archivo .json.
C. Metadata de Suite Office 365 para tener metadatos de comunicación interna (Emisor, receptores, canal, Timestamp). Actualización diaria a partir del API AZURE AD y se recibe archivo .json.
D. Eventos de Suite Office 365 metadatos del trabajo en la nube de los empleados (Creación/Edición/Eliminación/Compartición de archivos y directorios). Actualización diaria a partir del API AZURE AD se recibe archivo .json.
E. Eventos del usuario en local de la Suite de Office 365 (Creación/Edición/Eliminación/Compartición de archivos y directorios, clicks y keys). Actualización diaria a partir de comunicación antes de apagar el equipo mediante KAFKA.

Todos estos datos se vuelcan a al Bucket de Google Storage mientras el cluster de procesamiento está apagado.

A una hora predefinida por configuración (horario nocturno fuera de jornada laboral para evitar la indisponibilidad de los datos durante el volcado) se procede al procesamiento de los datos del último día, al tratamiento y al volcado a la BBDD Cloud SQL (Google Cloud SQL) mediante un clúster de Dataproc.

Al arrancar el clúster, lanza el Cloud Scheduler que arranca la Cloud Function New Files que revisa y compara los metadatos de los archivos de Bucket con los metadatos del día anterior para inicializar el procesamiento de los datos si son diferentes o apagar el clúster en caso de que no haya cambios (Evitar costes en días no laborales).

Si la información es nueva se arranca el procesamiento de datos y se crea cuatro dataset distintos para sincronizar con la BBDD Cloud SQL:

1. User_Network: Dataset con la información relativa a interacciones con las siguientes columnas:
   “UserIDEmisor”: String
   “UserIDReceptores”: String separado por “,”.
   “Canal”: String {Mail / Chat}
   “Timestamp”: INT (UNIX)

2. Data_Jobmarket: Dataset con un origen de datos externos procedentes del INE con los siguientes campos:
   “Sector”: String
   “Perfil Profesional”: String

“Salario Medio”: Money
“Salario Min”: Money
“Salario Max”: Money
“Días de antigüedad promedio”: INT
“Tiempstamp”: INT (UNIX)

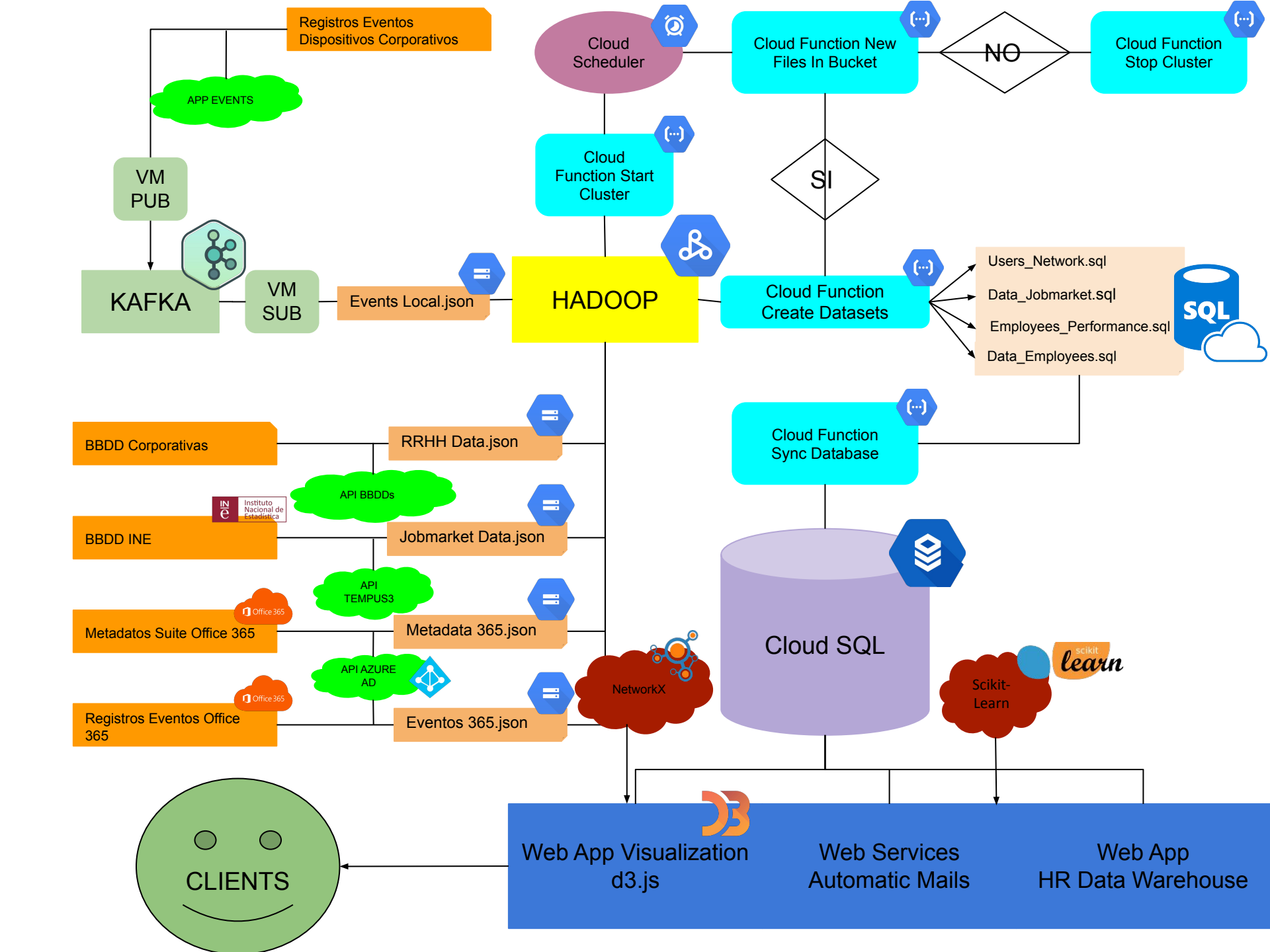
3. Employees_Perfomance: Dataset con los datos de rendimiento de los empleados con los siguientes campos:
   “UserID”: String
   “RatioClicksperMinute”: Float
   “RatioKeysperMinute”: Float
   “RatioWorkwhitFilesandDirectoryperMinute”: Float
   “RatioColaborationPerMinute_Mails_Chats_&_Share”: Float
   “Tiempstamp”: INT (UNIX)

4. Data_Employees: Dataset con la información de los empleados procedente de BBDDs de la compañía con los siguientes campos:
   “UserID”: String
   “Sector”: String
   “Área”: String
   “Puesto”: String
   “Manager”: Boolean
   “Salario”: Money
   “Corporate_Value_Produced”: Money
   “Días de antigüedad”: INT
   “Tiempstamp”: INT (UNIX)

Desarrollo de la plataforma DAaaS.

Todos estos datos se vuelcan en la BBDD SQL y sirve la información a dos Web APP y a un Web Services:

1. **WEB APP Visualization**: Web APP que muestra con las librerías d3.js y NetworkX para analizar y visualizar las interacciones laborales de User_Networks.
2. **Web App HR Data Warehouse:** Plataforma online para la consulta de los datos para el área de RRHH, contiene un modelo de BBDD relacional preconfigurado para recibir la información de los dataset en la BBDD Cloud SQL y acceden mediante permisos como cliente (Cloud SQL Client).
3. **Web Services Automatic Mails**: Analiza la información de User_Networks, Employees_Performance and Data_ Employees para mandar mails automáticos según sus patrones de trabajo y puesto determinado mediante el aprendizaje autómatico de patrones y KPI’s de resultados (Librería Python Scikit-Learn).


Registros Eventos Dispositivos Corporativos
APP EVENTS
VM PUB
KAFKA
VM SUB
Events Local.json
Cloud Scheduler
Cloud Function New Files In Bucket
NO
Cloud Function Stop Cluster
Cloud Function Start Cluster
SI
HADOOP
Cloud Function Create Datasets
Users_Network.sql
Data_Jobmarket.sql
Employees_Performance.sql
Data_Employees.sql
SQL
BBDD Corporativas
RRHH Data.json
Cloud Function Sync Database
API BBDDs
BBDD INE
Jobmarket Data.json
API TEMPUS3
Metadatos Suite Office 365
Metadata 365.json
Cloud SQL
API AZURE AD
Registros Eventos Office 365
Eventos 365.json
NetworkX
Scikit-Learn
CLIENTS
Web App Visualization d3.js
Web Services Automatic Mails
Web App HR Data Warehouse