UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS

SI807 U SISTEMAS DE INTELIGENCIA DE NEGOCIOS

“ Construcción de Pipeline ETL en Google Cloud Platform (GCP)”

PC3

Empresa: Cencosud

GRUPO N° 9:

LARICO CRUZ, DIEGO CÉSAR

CABANA CAZANI, GABRIEL ALESSANDRO

ÍNDICE

JUSTIFICACIÓN DE LA ELECCIÓN DE LA NUBE

Introducción y Decisión

Para la implementación de la arquitectura de datos, flujo ETL y Business Intelligence del presente proyecto, se ha seleccionado **Google Cloud Platform (GCP)**. Esta decisión es el resultado de un análisis comparativo técnico y comercial frente a los otros dos líderes del cuadrante de Gartner: **Amazon Web Services (AWS)** y **Microsoft Azure**.

La elección prioriza la arquitectura **"Serverless"** (sin servidor), la integración nativa de herramientas de Big Data y la eficiencia en costos para un proyecto piloto escalable.

Característica / Criterio	**Google Cloud (GCP) (Seleccionada)**	**Amazon Web Services (AWS)**	**Microsoft Azure**
1. Data Warehouse & Escalabilidad	**BigQuery:** Arquitectura 100% Serverless. Escala de gigabytes a petabytes automáticamente sin gestionar nodos ni clústeres. Rendimiento	**Redshift:** Tradicionalmente basado en clústeres provisionados. Aunque ofrece "Serverless", la configuración de red (VPC) y mantenimiento es más compleja.	**Synapse Analytics:** Potente integración empresarial, pero requiere aprovisionamiento de "Data Warehouse Units" (DWU) y su curva de configuración es alta.

Característica / Criterio	Google Cloud (GCP) (Seleccionada)	Amazon Web Services (AWS)	Microsoft Azure
	superior en consultas ad-hoc.		
2. Ecosistema de BI (Visualización)	**Looker Studio:** Solución nativa web, gratuita y con conexión directa a BigQuery. No requiere instalación de software ni drivers ODBC.	**QuickSight:** Modelo de pago por sesión/autor. Menos intuitivo para usuarios de negocio y con menos opciones de personalización gratuita.	**Power BI:** Líder del mercado, pero requiere licencias "Pro" para compartir reportes y configuración de Gateways para conectar con ciertas fuentes de datos.
3. Procesamiento Big Data (ETL)	**Dataproc & Dataflow:** Dataproc levanta clústeres de Spark en ~90 segundos (vs 10-	**EMR (Elastic MapReduce):** El estándar de la industria, muy robusto, pero	**HDInsight / Databricks:** Azure Databricks es excelente pero costoso; HDInsight

Característica / Criterio	Google Cloud (GCP) (Seleccionada)	Amazon Web Services (AWS)	Microsoft Azure
	15 min en otros), ideal para procesos efímeros y ahorro de costos.	complejo de configurar para cargas de trabajo simples o esporádicas.	es más complejo de administrar.
4. Almacenamiento (Data Lake)	**Cloud Storage (GCS):** Sistema de archivos unificado y simple. Maneja automáticamente las clases de almacenamiento y la redundancia regional.	**S3 (Simple Storage Service):** El más maduro y granular, pero la gestión de políticas de acceso (IAM) y buckets puede ser abrumadora al inicio.	**Data Lake Storage Gen2:** Requiere configuración de jerarquías de archivos y permisos POSIX, añadiendo complejidad a la ingesta simple.
5. Modelo de Precios (Pricing)	**Agresivo y Transparente:**	**Capa Gratuita**	**Corporativo:** $200 USD de

Característica / Criterio	Google Cloud (GCP) (Seleccionada)	Amazon Web Services (AWS)	Microsoft Azure
	Ofrece $300 USD de créditos iniciales y una capa gratuita generosa (ej. 1TB de consultas BigQuery/mes gratis). Cobro por segundo en muchos servicios.	**Limitada:** 12 meses de servicios limitados. Estructura de costos compleja y difícil de predecir sin calculadoras avanzadas.	crédito inicial. Modelo de licencias ventajoso solo si ya se posee un contrato Enterprise Agreement (EA) con Microsoft.
6. Seguridad y Gestión	**IAM Simplificado:** Gestión de identidades centralizada y orientada a proyectos.	**IAM Granular:** Extremadamente potente pero complejo; es fácil cometer errores de seguridad (ej.	**Active Directory:** Excelente si la empresa ya usa AD, pero añade una capa de complejidad innecesaria para

Característica / Criterio	Google Cloud (GCP) (Seleccionada)	Amazon Web Services (AWS)	Microsoft Azure
	Encriptación en reposo y en tránsito activada por defecto sin configuración extra.	buckets públicos) si no se es experto.	proyectos nativos de nube aislados.

PASOS PARA CREACIÓN DEL ETL

Paso 1: Ingesta de Datos (Cloud Storage)

En lugar de subir archivos a HDFS mediante Ambari, usaremos "Buckets" (cubetas) en la nube.

1. **Crear el Bucket:**
    - Ve a la consola de GCP → Menú → **Cloud Storage → Buckets**.
    - Haz clic en **CREAR**.
    - Nombre único: etl-ventas-proyecto-[tu-nombre].
    - Configuración: Déjalo en "Regional" (us-central1) y "Standard". Crear.
2. **Crear Estructura de Carpetas:**
    - Dentro del bucket, crea 3 carpetas: raw, curated, analytics.
    - Dentro de raw, crea las subcarpetas igual que antes: dim_cliente, fact_hecho_venta, etc.
3. **Subir los CSVs:**
    - Entra a cada carpeta (ej. raw/dim_cliente/) y sube el archivo CSV correspondiente desde tu PC.

Paso 2: Capa Raw (BigQuery)

En lugar de CREATE EXTERNAL TABLE en Hive, crearemos tablas en BigQuery que "apunten" a tus CSVs en el Storage.

1. **Crear el Dataset (Base de Datos):**
    - Ve al Menú → **BigQuery**.
    - Al lado de tu proyecto, haz clic en los 3 puntos → **Crear conjunto de datos** (Create dataset).
    - ID: dwh_ventas. Ubicación: us-central1.
2. **Crear Tablas Externas:**
    - Selecciona el dataset dwh_ventas → **Crear Tabla**.
    - **Crear tabla desde:** Google Cloud Storage.
    - **Ruta GCS:** etl-ventas-proyecto-[tu-nombre]/raw/dim_cliente/*.csv
    - **Formato:** CSV.
    - **Nombre de tabla:** dim_cliente_raw.
    - **Tipo de tabla:** Externa (External table).
    - **Esquema:** Marca "Detectar automáticamente" (Auto detect). *¡Mucho más fácil que Hive!*
    - **Opciones avanzadas:** "Filas de encabezado que saltar": 1.
    - Haz clic en **Crear Tabla**.
    - *(Repite para las 6 tablas)*.

Paso 3: Transformación con Spark (Dataproc)

Aquí reemplazamos tu VM de Hortonworks con un clúster real y efímero.

1. **Habilitar API:** Busca "Dataproc API" y habilítala.
2. **Crear Clúster:**
    - Menú → **Dataproc → Clústeres → Crear Clúster**.
    - Nombre: cluster-etl-spark. Región: us-central1.
    - **Tipo:** "Un solo nodo" (Single Node) - *Para ahorrar costos en pruebas.*
    - **Componentes:** Marca "Jupyter Notebook" en la sección de Puerta de enlace de componentes.
    - **Crear**. (Tardará unos 90 segundos).
3. **Abrir JupyterLab:**
    - Una vez creado, haz clic en el nombre del clúster $\to$ Pestaña **Interfaces web → JupyterLab**.
4. **El Código PySpark (Adaptado a GCP):**
    - Crea un nuevo Notebook (PySpark).
    - El código es casi idéntico al anterior, pero leyendo/escribiendo en GCS.

Python
```
from pyspark.sql.functions import col, to_date, when, lit, upper, trim

# Rutas GCS (¡CAMBIA ESTO POR TU BUCKET!)
BUCKET = "gs://etl-ventas-proyecto-[tu-nombre]"

# 1. Leer desde BigQuery o CSV directo
# (Leemos CSV directo para simplificar configuración)
df_cliente = spark.read.csv(f"{BUCKET}/raw/dim_cliente/*.csv", header=True, inferSchema=True)

# 2. Transformaciones (¡Las mismas que hiciste antes!)
df_cliente_curated = df_cliente.withColumn(
    "fecha_alta_cliente_dt", to_date(col("fecha_alta_cliente"), "yyyy-MM-dd")
).na.fill("SIN TARJETA", ["numero_tarjeta_bonus"])

# ... (Repite la lógica para todas las tablas) ...

# 3. Escribir a GCS en formato PARQUET (Capa Curated)
df_cliente_curated.write.mode("overwrite").parquet(f"{BUCKET}/curated/dim_cliente/")
print("Dim Cliente procesada y guardada en GCS Curated")
```

Paso 4: Carga a Tablas Curated (BigQuery)

Ahora hacemos disponibles esos archivos Parquet limpios en BigQuery.

1. Vuelve a **BigQuery**.
2. En tu dataset dwh_ventas, crea nuevas tablas.
3. **Crear tabla desde:** Google Cloud Storage.
4. **Ruta:** etl-ventas-proyecto-[tu-nombre]/curated/dim_cliente/*.parquet
5. **Formato:** Parquet.

6. **Nombre:** dim_cliente_curated.
7. **Tipo de tabla:** Nativa (Native table) - *Esto cargará los datos para que sea súper rápido*.
8. Crear Tabla.

   (Repite para las 6 tablas).

Paso 5: Cubo OLAP (BigQuery SQL)

En lugar de esperar 5 minutos como en Hive, BigQuery hará esto en segundos.

1. En BigQuery, abre una pestaña de consulta (Query).
2. Ejecuta tu SQL de agregación (adaptando los nombres de tablas):

```
SQL
CREATE OR REPLACE TABLE `dwh_ventas.resumen_ventas_analytics` AS
SELECT
   p.anio, p.nombre_mes,
   t.ciudad, t.nombre_tienda,
   pr.categoria, pr.marca,
   SUM(f.monto_venta_neta) AS total_ventas_netas,
   COUNT(DISTINCT f.cod_ticket) AS total_tickets
FROM
   `dwh_ventas.fact_hecho_venta_curated` f
JOIN
   `dwh_ventas.dim_periodo_curated` p ON f.sk_periodo = p.sk_periodo
JOIN
   `dwh_ventas.dim_tienda_canal_curated` t ON f.sk_tienda = t.sk_tienda
JOIN
   `dwh_ventas.dim_producto_curated` pr ON f.sk_producto = pr.sk_producto
GROUP BY
   1, 2, 3, 4, 5, 6;
```

Paso 6: Dashboard (Looker Studio)

Aquí es donde verás la magia de la integración nativa.

1. Ve a **Looker Studio** (lookerstudio.google.com).
2. **Crear →Informe**.
3. Te pedirá conectar datos: Selecciona **BigQuery**.
4. Elige tu PROYECTO →Dataset dwh_ventas → Tabla resumen_ventas_analytics.
5. Haz clic en **Añadir**.
6. ¡Listo! Arrastra y suelta para crear tus KPIs (Ticket Promedio, Ventas por Categoría) tal como te enseñé en las respuestas anteriores.

Paso 7: Monitoreo y Limpieza (¡CRÍTICO!)

1. **Monitoreo:** Ve a la consola de GCP $\to$ **Operations → Logging**. Ahí verás los logs de Dataproc y BigQuery si algo falla.
2. **Limpieza (Ahorro de dinero):**
    - **Dataproc:** Apenas termines de ejecutar tu código Spark, **BORRA EL CLÚSTER**. Es lo que más cobra (aunque es barato, si lo dejas prendido días te consumirá el crédito).
    - **Storage y BigQuery:** Son baratísimos, puedes dejarlos.

EVALUACIÓN DE COSTOS

SERVICIO	ESPECIFICACIONES	PRIMER MES	PROYECCION 6 MESES
GCS STORAGE	Total amount of storage - 15 TiB Location type - Region Location - Iowa (us-central1) Storage class - Standard Storage Source region - North America Destination region - North America	$307,10	$1.842,60
BIG QUERY	Location type - Region Location - Iowa (us-central1) Edition - Enterprise Maximum slots - Small (100 slots) Baseline slotS - 100 Commitment - 1 Year Average utilization of autoscale slots - 50	$3.974,81	$23.848,86
DATAPROC SERVERLESS	Number of vCPUs - 4 Memory per vCPU (GiB) - 16 GiB Region - Iowa (us-central1) Usage time - 30 Hours Interactive - false Add GPUs - false Current cluster utilization % - 50	$416,10	$2.496,60
		TOTAL	$28.188,06