

Característica	Google Cloud (GCP) (Seleccionada)	Amazon Web Services (AWS)	Microsoft Azure
1. Data Warehouse (Motor SQL)	BigQuery: Totalmente Serverless. Separa almacenamiento de cómputo automáticamente. No requiere gestión de servidores.	Redshift: Potente, pero tradicionalmente requiere provisionar clústeres (aunque existe versión Serverless, es más compleja).	Synapse Analytics: Integración fuerte con el ecosistema Microsoft, pero con una curva de aprendizaje alta para configuraciones iniciales.
2. Ecosistema de BI (Visualización)	Looker Studio: Nativo, gratuito, web-based y conexión instantánea con BigQuery sin drivers.	QuickSight: Modelo de pago por sesión/usuario. Menos intuitivo para usuarios no técnicos.	Power BI: Líder del mercado, pero requiere licencias "Pro" para compartir y configuración de Gateways para datos on-premise/complexos.
3. Migración Big Data (Spark/Hadoop)	Dataprocc: Permite levantar clústeres de Spark en ~90 segundos. Ideal para trabajos efímeros.	EMR (Elastic MapReduce): El estándar de la industria, pero el tiempo de arranque y configuración es mayor (5-10 min).	HDInsight / Databricks: Muy potentes, pero orientados a arquitecturas empresariales complejas y costosas.

Característica	Google Cloud (GCP) (Seleccionada)	Amazon Web Services (AWS)	Microsoft Azure
4. Almacenamiento (Data Lake)	Cloud Storage (GCS): Funciona como un reemplazo directo de HDFS. Sistema de archivos plano y unificado.	S3 (Simple Storage Service): El más maduro del mercado, con múltiples niveles de almacenamiento, pero configuración de permisos compleja.	Blob Storage / Data Lake Gen2: Requiere jerarquías y configuraciones específicas para funcionar como HDFS real.
5. Curva de Aprendizaje	Baja/Media: Interfaz limpia, orientada a desarrolladores. Documentación muy clara para Data Engineering.	Alta: Cantidad abrumadora de servicios y configuraciones de red (VPC) obligatorias desde el inicio.	Media: Familiar para usuarios de Windows, pero el portal puede ser lento y denso.
6. Pricing (Costos y Capa Gratuita)	Modelo Agresivo: \$300 de crédito inicial + Capa "Siempre Gratuita" generosa (1TB de consultas en BigQuery al mes gratis).	Modelo Estándar: Capa gratuita por 12 meses limitada. Costos de salida de datos suelen ser más altos.	Modelo Corporativo: \$200 de crédito inicial. Muy conveniente si la empresa ya tiene contratos Enterprise Agreement.

Alineación de la Elección con los Requerimientos del Proyecto

Se eligió **GCP** porque sus fortalezas se alinean directamente con las necesidades críticas de este proyecto ETL:

1. Necesidad de una Arquitectura Serverless (BigQuery):

- *Requerimiento:* El proyecto requería pasar de archivos CSV a un modelo dimensional (Esquema Estrella) sin perder tiempo administrando infraestructura o memoria RAM (como sucedía en la VM local).
- *Solución GCP:* **BigQuery** permite cargar los datos y ejecutar transformaciones SQL complejas en segundos sin configurar ningún servidor. AWS Redshift hubiera requerido configurar nodos y clusters, lo cual es excesivo para este alcance.

2. Integración Nativa de BI (Looker Studio):

- *Requerimiento:* Visualizar los KPIs (Ticket Promedio, Ventas por Categoría) de forma rápida y compartible vía web.
- *Solución GCP:* La conexión entre BigQuery y **Looker Studio** es nativa. No se requieren drivers ODBC, Gateways ni instalaciones de escritorio (como Power BI Desktop), agilizando la entrega del Dashboard final.

3. Simplicidad en la Ingesta (GCS):

- *Requerimiento:* Un repositorio centralizado para los archivos "Raw" (CSV).
- *Solución GCP:* **Google Cloud Storage** actúa como el Data Lake. Su simplicidad permite subir archivos y que BigQuery los lea directamente como "Tablas Externas", eliminando pasos intermedios de carga.

4. Eficiencia de Costos (Free Tier):

- *Requerimiento:* Viabilidad económica para un proyecto piloto/académico.
- *Solución GCP:* El proyecto cabe enteramente dentro de la capa gratuita de GCP (10 GB de almacenamiento y 1 TB de análisis mensual gratis en BigQuery), lo que lo hace costo-cero, a diferencia de mantener una instancia EC2 de AWS encendida.

Documentación de Arquitectura Pipeline ETL en GCP

1. Capa de Origen y Orquestación (Trigger)

Esta fase asegura que los datos lleguen al sistema y que el proceso se inicie automáticamente.

- **Archivos CSV (Origen):** Representan la data transaccional cruda generada por el negocio. Son depositados manualmente o por sistemas externos.
- **Cloud Scheduler (Orquestación):**
 - **Función:** Es el "cronómetro" de la arquitectura.
 - **Acción:** Envía una señal (Trigger) periódica (ej. diariamente a las 2:00 AM) para despertar al servicio de procesamiento (Dataproc), automatizando el ciclo sin intervención humana.
- **Cloud Storage (Data Lake / Landing Zone):**
 - **Función:** Almacenamiento de objetos altamente escalable.
 - **Acción:** Actúa como la zona de aterrizaje (/raw) donde residen los CSV originales y el repositorio de código (/scripts) donde se guarda tu lógica en Python.

2. Capa de Almacenamiento (Data Warehouse - BigQuery)

El corazón de los datos. Utilizas una estrategia de capas para asegurar calidad y gobierno.

- **BigQuery RAW (Bronce):**
 - **Tipo:** Tablas Externas.
 - **Función:** Permite consultar los CSVs que están en Cloud Storage usando SQL, sin mover los datos ni duplicarlos. Es la "vista pura" del archivo original.
- **BigQuery CURATED (Plata):**
 - **Tipo:** Tablas Nativas (Optimizadas).
 - **Función:** Almacena los datos limpios, tipados (fechas, floats correctos) y estandarizados. Aquí reside la "verdad única" de los datos a nivel de detalle.
- **BigQuery ANALYTICS (Oro):**
 - **Tipo:** Tablas Agregadas / Vistas Materializadas.

- **Función:** Contiene los datos listos para el consumo (KPIs, sumatorias, cubos OLAP). Está optimizada para que los reportes carguen rápido.

3. Capa de Procesamiento (ETL & Transformación)

Donde ocurre la magia de convertir datos crudos en información.

- **Dataproc Serverless (Motor Spark):**
 - **Función:** Procesamiento masivo y distribuido.
 - **Acción:** Recibe la orden del Scheduler, levanta un clúster efímero, lee los datos, aplica limpieza compleja con **PySpark**, escribe los resultados en la capa *Curated* y se apaga automáticamente (ahorrando costos).
- **Stored Procedure (Lógica SQL):**
 - **Función:** Transformación dentro del Warehouse.
 - **Acción:** Ejecuta lógica de negocio (Merge/Upsert) para tomar los datos limpios de *Curated*, calcular agregaciones (ventas totales, promedios) e insertarlos en la capa *Analytics*.

4. Capa de Monitoreo (Observabilidad)

Nota: En tu diagrama visualizas iconos de Oracle (OCI), pero en GCP los servicios equivalentes que funcionalmente realizarán esta tarea son:

- **Cloud Logging (equivalente a OCI Logging):**
 - **Función:** Auditoría y depuración.
 - **Acción:** Centraliza los logs de texto que generan Dataproc y el Scheduler. Si el script de Python falla, aquí es donde buscas el error ("Stack Trace").
- **Cloud Monitoring (equivalente a Oracle Monitoring):**
 - **Función:** Salud del sistema.
 - **Acción:** Visualiza métricas numéricas: ¿Cuánto tardó el job? ¿Cuánta CPU usó? ¿Falló el Scheduler? Envía alertas si algo se rompe.

5. Capa de Visualización (Business Intelligence)

- **Looker Studio:**
 - **Función:** Interfaz de usuario final.

- **Acción:** Se conecta nativamente a **BigQuery Analytics**. Visualiza los KPIs en tableros interactivos, permitiendo a los usuarios de negocio filtrar y analizar tendencias sin tocar código SQL.