Proyecto Final - Pasantía de Ingeniería de Datos

Duración: 2 semanas (40 horas)
Modalidad: Individual
Presupuesto: Máximo $50 por estudiante
Evaluación: 70% Técnico | 20% Presentación | 10% Documentación

---

🎯 Objetivo

Construir una plataforma de análisis de datos end-to-end en AWS que demuestre tu capacidad para:

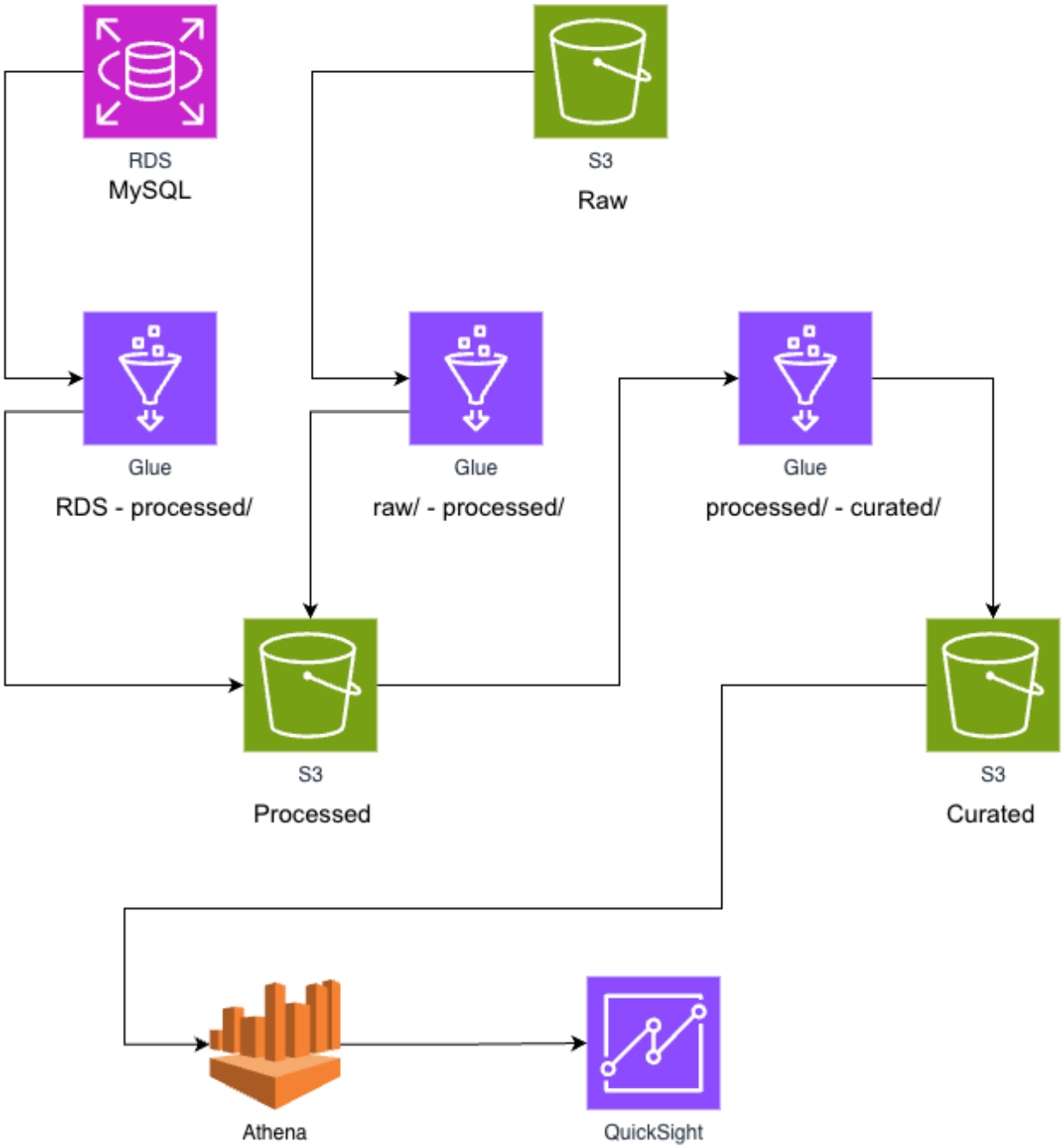
- Diseñar e implementar una arquitectura de data lake
- Construir pipelines ETL para transformar datos crudos
- Crear dashboards de inteligencia de negocios
- Seguir las mejores prácticas de AWS para optimización de costos y seguridad

---

📋 Descripción del Proyecto

Elegirás UNO de cuatro conjuntos de datos de diferentes industrias (e-commerce, salud, finanzas o streaming) y construirás una solución completa de análisis que responda preguntas específicas de negocio usando servicios de AWS.

Arquitectura



📊 Opciones de Conjuntos de Datos

Elige UNA de las siguientes opciones antes del **final del Día 1**:

Opción 1: Análisis de E-Commerce (TechStore)

Industria: Retail
Objetivo de Negocio: Optimizar ventas y reducir la pérdida de clientes

Fuentes de Datos:

- **Tablas RDS:** customers, products, orders, order_items
- **Archivos S3:** web_clickstream.csv, customer_support_tickets.csv

Preguntas de Negocio:

1. ¿Cuáles son los 10 productos principales por ingresos?
2. ¿Cómo podemos segmentar clientes usando análisis RFM?
3. ¿Cuál es la tasa de conversión de visitas web a compras?
4. ¿Cuáles son las tendencias de ingresos mensuales?
5. ¿Qué categorías de productos generan más tickets de soporte?

Opción 2: Análisis de Salud (Hospital MediCare)

Industria: Salud
Objetivo de Negocio: Mejorar resultados de pacientes y eficiencia operacional

Fuentes de Datos:

- **Tablas RDS:** patients, doctors, appointments, treatments
- **Archivos S3:** lab_results.csv, patient_feedback.csv

Preguntas de Negocio:

1. ¿Qué especialidades médicas tienen los tiempos de espera más largos?
2. ¿Cuáles son las tasas de readmisión de pacientes por diagnóstico?
3. ¿Cómo se desempeñan los doctores en métricas clave?
4. ¿Cuál es el ingreso mensual por departamento?

5. ¿Cómo se correlaciona el tiempo de espera con la satisfacción del paciente?

Opción 3: Análisis de Servicios Financieros (Banco FinTech)

Industria: Banca/FinTech
Objetivo de Negocio: Detectar fraude y entender comportamiento del cliente

Fuentes de Datos:

- **Tablas RDS:** customers, accounts, transactions, loans
- **Archivos S3:** mobile_app_events.csv, customer_service_calls.csv

Preguntas de Negocio:

1. ¿Cuáles son los patrones de transacciones por segmento de cliente?
2. ¿Cuáles son las tasas de incumplimiento de préstamos por puntaje crediticio?
3. ¿Qué categorías de comerciantes son más populares por tipo de cuenta?
4. ¿Cuáles son las métricas de engagement de la app móvil?
5. ¿Cuáles son las tendencias de volumen de llamadas de servicio al cliente?

Opción 4: Análisis de Plataforma de Streaming (StreamFlix)

Industria: Medios/Entretenimiento
Objetivo de Negocio: Aumentar engagement y reducir cancelación de suscripciones

Fuentes de Datos:

- **Tablas RDS:** users, content, subscriptions, payments
- **Archivos S3:** viewing_history.csv, user_ratings.csv

Preguntas de Negocio:

1. ¿Cuáles son las tasas de finalización de contenido por género?
2. ¿Qué impulsa la cancelación de suscripciones por nivel?
3. ¿Cuándo son las horas pico de visualización y qué contenido se prefiere?
4. ¿Cuáles son las tendencias de ingresos por plan de suscripción?
5. ¿Qué series de contenido se ven más en maratón?

🌟 Componente Bonus Opcional (Crédito Extra)

¿Quieres destacarte? Elige UNO de los siguientes componentes opcionales para crédito extra:

Opción A: Infraestructura como Código (+10 puntos)

Si conoces CloudFormation o Terraform, despliega tu infraestructura como código en lugar de crear recursos manualmente.

Requisitos:

- ☐ Crear plantillas IaC para infraestructura principal:
  - Bucket S3 con estructura de carpetas
  - Instancia RDS MySQL
  - Base de datos y conexiones Glue
  - Roles IAM para trabajos Glue
- ☐ Incluir instrucciones de despliegue en README
- ☐ Documentar parámetros y variables usadas
- ☐ Desplegar y eliminar infraestructura exitosamente

Entregables:

- Plantilla CloudFormation (`.yaml`) O archivos Terraform (`.tf`)
- Script o comandos de despliegue
- Documentación del enfoque IaC

Por qué esto importa:

- Muestra mentalidad DevOps
- Crítico para ambientes de producción
- Permite despliegues repetibles
- Demuestra conocimiento avanzado de AWS

Consejos:

- Comienza con una plantilla parcial, no intentes automatizar todo
- Enfócate en infraestructura principal (S3, RDS, Glue)
- Prueba el despliegue en un ambiente limpio
- Documenta cualquier paso manual aún requerido

Opción B: Optimización Avanzada de Costos (+10 puntos)

Ve más allá de la optimización básica e implementa técnicas avanzadas de ahorro de costos.

Requisitos:

- ☐ Implementar S3 Intelligent-Tiering o políticas de Lifecycle
- ☐ Usar job bookmarks de Glue para evitar reprocesar datos
- ☐ Implementar estrategia de carga incremental de datos
- ☐ Crear dashboard de costos en QuickSight mostrando gasto diario por servicio
- ☐ Documentar 5+ técnicas de optimización con impacto en costos

Entregables:

- Documento de optimización de costos (2 páginas)
- Comparación de costos antes/después
- Dashboard de costos en QuickSight

Opción C: Framework de Calidad de Datos (+10 puntos)

Implementar verificaciones automatizadas de calidad de datos en tu pipeline.

Requisitos:

- ☐ Agregar validaciones de calidad de datos en trabajos Glue:
  - Verificaciones de nulos en campos críticos
  - Validaciones de tipos de datos
  - Verificaciones de rangos (ej. precios > 0)
  - Verificaciones de integridad referencial
- ☐ Registrar métricas de calidad de datos
- ☐ Crear dashboard de calidad de datos
- ☐ Fallar pipeline si verificaciones críticas de calidad fallan

Entregables:

- Código de calidad de datos en trabajos Glue
- Métricas de calidad registradas en CloudWatch o S3
- Documentación de reglas de calidad

Nota: Los componentes bonus son opcionales. Enfócate en completar el proyecto principal primero. Solo intenta si terminas temprano y quieres mostrar habilidades adicionales.

📅 Cronograma y Milestones

Semana 1: Pipeline de Datos (20 horas)

Día 1-2: Configuración de Infraestructura (8 horas)

Tareas:

- ☐ Elegir tu opción de conjunto de datos
- ☐ Crear bucket S3: `[nombre-dataset]-[tunombre]-2025`
- ☐ Crear estructura de carpetas: `raw/`, `processed/`, `curated/`
- ☐ Crear instancia RDS MySQL (db.t3.micro)
- ☐ Cargar datos SQL proporcionados en RDS
- ☐ Subir archivos CSV a carpeta raw/ de S3
- ☐ Crear base de datos Glue Data Catalog
- ☐ Ejecutar Glue Crawler en datos raw de S3

Entregables:

- Bucket S3 con datos
- Instancia RDS con tablas pobladas
- Glue Catalog con tablas raw

Día 3-5: Pipeline ETL (12 horas)

Tareas:

- ☐ Crear conexión Glue a RDS MySQL (configurar VPC, security groups, probar conexión)
- ☐ **Trabajo Glue 1:** Extraer datos de RDS vía JDBC → Escribir a S3 como Parquet (processed/)
  - Leer tablas: customers, products, orders, order_items
  - Convertir a formato Parquet
  - Agregar conversiones básicas de tipos de datos

- [ ] **Trabajo Glue 2:** Limpiar y transformar datos CSV de S3 raw/
  - Manejar valores nulos
  - Corregir tipos de datos
  - Agregar particiones (ej. por fecha)
  - Escribir a processed/ como Parquet
- [ ] **Trabajo Glue 3:** Crear tablas analíticas curadas
  - Unir múltiples fuentes (orders + products + customers)
  - Crear tablas agregadas/desnormalizadas para análisis
  - Escribir a curated/ con particiones
- [ ] Probar cada trabajo individualmente con muestras pequeñas de datos
- [ ] Ejecutar trabajos Glue manualmente en secuencia
- [ ] Ejecutar Glue Crawler en carpetas processed/ y curated/
- [ ] Verificar calidad de datos y conteos de filas

Entregables:

- Conexión JDBC de Glue configurada
- 3 trabajos ETL de Glue funcionando
- Archivos Parquet en processed/ y curated/
- Glue Catalog actualizado

Consejos:

- Prueba la conexión JDBC primero antes de escribir trabajos
- Usa formato Parquet para mejor rendimiento y ahorro de costos (80% de reducción en costos de escaneo de Athena)
- Agrega particiones (ej. por año/mes) para tablas grandes
- Comienza con `LIMIT 100` en tus consultas SQL para pruebas
- Revisa logs de CloudWatch para debugging
- Los trabajos Glue cobran por DPU-hora, así que optimiza tu código

---

Semana 2: Análisis y Visualización (20 horas)

Día 6-8: Análisis SQL con Athena (12 horas)

Tareas:

- [ ] Configurar workgroup de Athena con ubicación de resultados de consultas
- [ ] Escribir consultas SQL para responder las 5 preguntas de negocio
- [ ] Optimizar consultas (usar particiones, limitar escaneos)

- [ ] Crear vistas para consultas complejas
- [ ] Realizar análisis exploratorio adicional
- [ ] Exportar resultados de consultas a CSV
- [ ] Documentar insights y hallazgos

Entregables:

- 5+ consultas SQL con resultados
- Vistas de Athena creadas
- Documento de análisis con insights

Consejos:

- Usa `LIMIT` durante desarrollo para reducir costos
- Revisa "Data scanned" en resultados de consultas
- Usa Parquet + particiones para minimizar escaneos
- Guarda consultas para tu presentación

Día 9-10: Dashboards con QuickSight (8 horas)

Tareas:

- [ ] Registrarse en QuickSight (si aún no lo has hecho)
- [ ] Conectar Athena como fuente de datos (con CustomSQL)
- [ ] **Dashboard - Tab 1: Resumen Ejecutivo**
  - KPIs clave (ingresos, clientes, órdenes/usuarios)
  - Tendencia de ingresos en el tiempo
  - Principales productos/contenido/servicios
- [ ] **Dashboard - Tab 2: Análisis de Clientes/Usuarios**
  - Visualización de segmentación de clientes
  - Análisis de cohortes o métricas de retención
  - Distribución geográfica
- [ ] **Dashboard - Tab 3: Métricas Operacionales**
  - Tickets de soporte / citas / transacciones
  - Métricas de rendimiento
  - Tendencias operacionales
- [ ] Agregar filtros e interactividad
- [ ] Formatear y pulir dashboards
- [ ] Preparar presentación y documentación

Entregables:

- 3 dashboards interactivos de QuickSight
- Capturas de pantalla para documentación

Consejos:

- Comienza con visuales simples, luego mejora
- Usa campos calculados para KPIs
- Agrega filtros para rangos de fechas
- Prueba en diferentes tamaños de pantalla

📦 Entregables Finales

1. Repositorio de Código (Git)

Archivos requeridos:

```
project-root/
├── README.md                    # Instrucciones de configuración
├── architecture-diagram.png     # Arquitectura visual
├── glue-jobs/
│   ├── job1-rds-extraction.py
│   ├── job2-csv-cleaning.py
│   └── job3-curated-tables.py
├── athena-queries/
│   ├── query1-top-products.sql
│   ├── query2-segmentation.sql
│   └── ...
```

├── sql/

│ └── rds-setup.sql

└── docs/

└── technical-document.pdf

---

2. Documentación Técnica

Secciones requeridas:

1. Resumen Ejecutivo

- Descripción general del proyecto
- Dataset elegido y por qué
- Hallazgos clave

2. Arquitectura y Diseño

- Diagrama de arquitectura con explicación
- Servicios AWS usados y por qué
- Descripción del flujo de datos
- Consideraciones de seguridad

3. Detalles de Implementación

- Diseño del pipeline ETL
- Transformaciones de datos aplicadas
- Desafíos enfrentados y soluciones
- Fragmentos de código (partes clave)

4. Análisis e Insights

- Respuestas a preguntas de negocio
- Insights clave descubiertos
- Visualizaciones y dashboards creados

5. Análisis y Optimización de Costos

- **Desglose de costos reales de AWS** (captura de pantalla del dashboard de facturación)

- **3+ técnicas de optimización implementadas** con impacto específico
    - Ejemplo: "Parquet redujo almacenamiento de 500MB a 120MB"
    - Ejemplo: "Particionamiento redujo costo promedio de consulta de $0.05 a $0.01"
- **Costo total del proyecto** y comparación con presupuesto de $50
- **Lecciones aprendidas** sobre gestión de costos
- **Qué harías diferente** para optimización de costos

Formato:

- Formato PDF
- Incluir capturas de pantalla donde sea relevante
- Formato profesional
- Revisar gramática y claridad

---

3. Presentación (15 minutos)

Estructura:

Introducción (2 min)

- Tu nombre y elección de dataset
- Descripción general del problema de negocio

Arquitectura (4 min)

- Mostrar diagrama de arquitectura
- Explicar flujo de datos
- Destacar decisiones clave de diseño

Demo en Vivo (6 min)

- Mostrar dashboards de QuickSight
- Ejecutar una consulta de Athena
- Explicar insights clave

Insights y Recomendaciones (3 min)

- Hallazgos clave del análisis
- Recomendaciones de negocio
- Impacto potencial

Lecciones Aprendidas (2 min)

- Desafíos técnicos
- Qué harías diferente
- Consejos de optimización de costos

Preguntas y Respuestas (5 min)

📊 Rúbrica de Evaluación

Implementación Técnica (70 puntos)

Arquitectura y Diseño (12 puntos)

- ☐ Estructura de data lake bien diseñada (4)
- ☐ Selección apropiada de servicios (4)
- ☐ Mejores prácticas de seguridad (2)
- ☐ Diagrama de arquitectura claro (2)

Data Lake y Catálogo (10 puntos)

- ☐ Estructura apropiada de carpetas S3 (3)
- ☐ Formato Parquet usado (3)
- ☐ Particionamiento implementado (2)
- ☐ Glue Catalog configurado apropiadamente (2)

Pipeline ETL (23 puntos)

- ☐ Todos los trabajos Glue funcionando correctamente (10)
- ☐ Conexión JDBC configurada apropiadamente (3)
- ☐ Verificaciones de calidad de datos (4)
- ☐ Manejo de errores (3)
- ☐ Calidad del código y comentarios (3)

Análisis (12 puntos)

- ☐ Las 5 preguntas de negocio respondidas (8)
- ☐ Optimización de consultas (2)
- ☐ Calidad de insights (2)

Visualización (8 puntos)

- [ ] 3 Tabs creadas dentro del dashboard (5)
- [ ] Interactividad y filtros (2)
- [ ] Calidad del diseño visual (1)

Optimización de Costos (5 puntos) ⭐ IMPORTANTE

- [ ] Formato Parquet usado en todo el proyecto (1)
- [ ] Particionamiento implementado en tablas grandes (1)
- [ ] Costo real del proyecto bajo $40 (1)
- [ ] Documentadas 3+ decisiones de optimización con justificación (1)
- [ ] Recursos detenidos cuando no están en uso (1)

Presentación (20 puntos)

- [ ] Comunicación clara (5)
- [ ] Demo en vivo funciona (8)
- [ ] Insights presentados (4)
- [ ] Gestión del tiempo (3)

Documentación (10 puntos)

- [ ] Documento técnico completo (5)
- [ ] README con instrucciones de configuración (3)
- [ ] Comentarios de código y organización (2)

Puntos Bonus (hasta +10)

- [ ] Implementación de Infraestructura como Código (+10)
- [ ] Optimización Avanzada de Costos (+10)
- [ ] Framework de Calidad de Datos (+10)

Nota: Solo UN componente bonus puede ser enviado para crédito extra.

Total Posible: 100 puntos + 10 bonus = 110 puntos (limitado a 100)

💰 Gestión de Presupuesto

Presupuesto máximo: $50 por estudiante

Costos Esperados (2 semanas)

- RDS MySQL (db.t3.micro): ~$7
- Almacenamiento y solicitudes S3: ~$2
- Trabajos ETL Glue: ~$12
- Glue Crawlers: ~$2
- Conexión JDBC Glue: ~$1
- Consultas Athena: ~$1
- QuickSight: ~$5
- CloudWatch: ~$1
- **Buffer:** ~$19
- **Total estimado:** ~$50

Estrategias de Control de Costos

CRÍTICO - Hacer esto diariamente:

- ☐ **DETENER instancia RDS** cuando no estés trabajando (ahorra ~$0.40/día)
- ☐ Eliminar ejecuciones fallidas de trabajos Glue
- ☐ Monitorear dashboard de AWS Budgets

Técnicas de optimización (CALIFICADO - 5 puntos):

- Usar formato Parquet (reduce costos de escaneo de Athena en 80%)
- Particionar tablas grandes (reduce costos de consultas)
- Probar trabajos Glue con muestras pequeñas de datos primero (`LIMIT` en SQL)
- Usar `LIMIT` en Athena durante desarrollo
- Establecer timeout de trabajo Glue a máximo 30 minutos
- Usar reutilización de resultados de consultas Athena (24 horas)
- Habilitar job bookmarks de Glue para evitar reprocesamiento
- Comprimir datos (compresión Snappy con Parquet)
- Detener recursos inmediatamente cuando no estén en uso

Documenta tus decisiones de optimización: Debes documentar al menos 3 técnicas de optimización de costos que implementaste y su impacto. Esto vale 1 punto en tu evaluación.

Ejemplo:

- "Usé formato Parquet: Redujo escaneo de Athena de 500MB a 100MB (80% de ahorro)"
- "Particioné tabla de órdenes por año: Las consultas ahora escanean 1 partición en lugar de todos los datos"
- "Detuve RDS por las noches: Ahorré $5.60 en 2 semanas"

Alertas de presupuesto:

- Recibirás alertas por email a $30, $40, y $50
- Si te acercas a $40, contacta a tu mentor inmediatamente
- La cuenta puede ser suspendida si se excede el presupuesto
- **Mantenerse bajo $40 vale 1 punto en tu calificación**

🛠️🛠️ Recursos y Datos

Datos del proyecto:

Los estudiantes deben **buscar y utilizar sus propios datasets** que se ajusten a las opciones propuestas (e-commerce, salud, finanzas o streaming). Pueden usar:

- Datasets públicos (Kaggle, AWS Open Data, etc.)
- Datos sintéticos generados por ellos mismos
- Combinación de múltiples fuentes

Requisitos de los datos:

- Deben tener al menos 4 tablas relacionales para RDS
- Deben tener al menos 2 archivos CSV para S3
- Los datos deben permitir responder las 5 preguntas de negocio de la opción elegida

Documentación de AWS:

- AWS Glue Documentation
- Amazon Athena Documentation
- Amazon QuickSight Documentation
- AWS RDS MySQL Documentation

Nota: Este proyecto es completamente autónomo. Los estudiantes deben investigar, diseñar e implementar toda la solución por su cuenta.

📝 Lineamientos Importantes

Integridad Académica

- Este es **100% trabajo individual**
- Puedes discutir conceptos con compañeros, pero no compartir código
- Todo el código debe ser tuyo
- El plagio resultará en falla del proyecto

Obtener Ayuda

- **Atascado < 30 minutos:** Intenta depurar tú mismo (revisa logs, Google)
- **Atascado > 30 minutos:** Pregunta a tu mentor
- **Preocupaciones de presupuesto:** Contacta al mentor inmediatamente
- **Bloqueos técnicos:** Documenta el problema y pide ayuda

Mejores Prácticas

- ☐ Hacer commit de código a Git frecuentemente (mínimo diario)
- ☐ Usar roles IAM, nunca hardcodear credenciales
- ☐ Etiquetar todos los recursos: `Owner:[tunombre]`, `Project:[nombre-dataset]`
- ☐ Documentar mientras avanzas (no esperes hasta el final)
- ☐ Probar incrementalmente (no construir todo y luego probar)
- ☐ Llevar registro de costos diariamente

Convenciones de Nomenclatura

Bucket S3: `[nombre-dataset]-[tunombre]-2025`
Ejemplo: `techstore-maria-2025`

Instancia RDS: `[nombre-dataset]-[tunombre]-db`
Ejemplo: `techstore-maria-db`

Base de datos Glue: `[nombre-dataset]_[tunombre]`
Ejemplo: `techstore_maria`

Trabajos Glue: `[nombre-dataset]-[tunombre]-[propósito-trabajo]`
Ejemplo: `techstore-maria-rds-extraction`

📅 Entrega

Fecha límite: Viernes 28/11, Semana 13

Qué entregar:

1. URL del repositorio Git (asegurar que el mentor tenga acceso)
2. Documento técnico (PDF)
3. Diagrama de arquitectura (PNG/PDF)
4. Diapositivas de presentación (PDF o PowerPoint)
5. ID de cuenta AWS (para verificación)

Presentaciones: Viernes 28/11

- 15 minutos de presentación + 5 minutos de preguntas y respuestas por estudiante
- El orden será aleatorizado
- Ten tu demo listo y probado

✅ Lista de Verificación Pre-Vuelo

Antes de comenzar:

- ☐ Acceso a cuenta AWS confirmado
- ☐ Alertas de presupuesto configuradas
- ☐ Repositorio Git creado
- ☐ Opción de dataset elegida
- ☐ Información de contacto del mentor guardada
- ☐ Ambiente de desarrollo listo

Antes de la entrega:

- ☐ Todos los recursos detenidos (RDS)
- ☐ Repositorio Git completo y organizado
- ☐ Documentación revisada para completitud
- ☐ Presentación probada y cronometrada
- ☐ Demo verificado funcionando
- ☐ Costos revisados (bajo $50)

🎓 Objetivos de Aprendizaje

Al completar este proyecto, demostrarás:

✅ Habilidades de Ingeniería de Datos

- Diseñar e implementar arquitecturas de data lake
- Construir pipelines ETL con AWS Glue
- Optimizar almacenamiento de datos y rendimiento de consultas
- Trabajar con múltiples fuentes de datos (RDS y S3)

✅ Habilidades de Análisis

- Escribir consultas SQL complejas
- Crear dashboards de inteligencia de negocios
- Derivar insights de datos
- Comunicar hallazgos efectivamente

✅ Habilidades de AWS Cloud

- Usar múltiples servicios AWS juntos
- Seguir mejores prácticas de seguridad
- Optimizar costos
- Monitorear y solucionar problemas

✅ Habilidades Profesionales

- Planificación de proyectos y gestión del tiempo
- Documentación técnica
- Presentación y comunicación
- Resolución de problemas bajo restricciones

🚀 ¿Listo para Comenzar?

Próximos pasos:

1. Revisar todas las opciones de datasets cuidadosamente
2. Elegir la que más te interese
3. Configurar tu ambiente de desarrollo

4. Comenzar con las tareas del Día 1
5. Hacer commit de tu primer código antes del final del Día 1

Recuerda: Este proyecto simula un escenario real de ingeniería de datos. Trátalo como si estuvieras construyendo esto para una empresa real. ¡Buena suerte! 🎉
