

CS2032 - Cloud Computing (Ciclo 2025-2)

Proyecto Parcial

Semana 5 a Semana 7 (Exposición en Semana 8 y 9)

ELABORADO POR: GERALDO COLCHADO Y OSCAR MEJIA
CON COLABORACIÓN DE: MAYKOL MORALES (ACL CLOUD COMPUTING)

Agenda

Proyecto Parcial

1. Competencias a lograr
2. Enunciado y Rúbrica
3. Entregables y Plazo

Competencias a lograr

Por el alumno al finalizar el proyecto parcial

Competencia:

4.1: Crea, selecciona, adapta y aplica técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones. (nivel 3).

Agenda

Proyecto Parcial

1. Competencias a lograr
2. Enunciado y Rúbrica
3. Entregables y Plazo

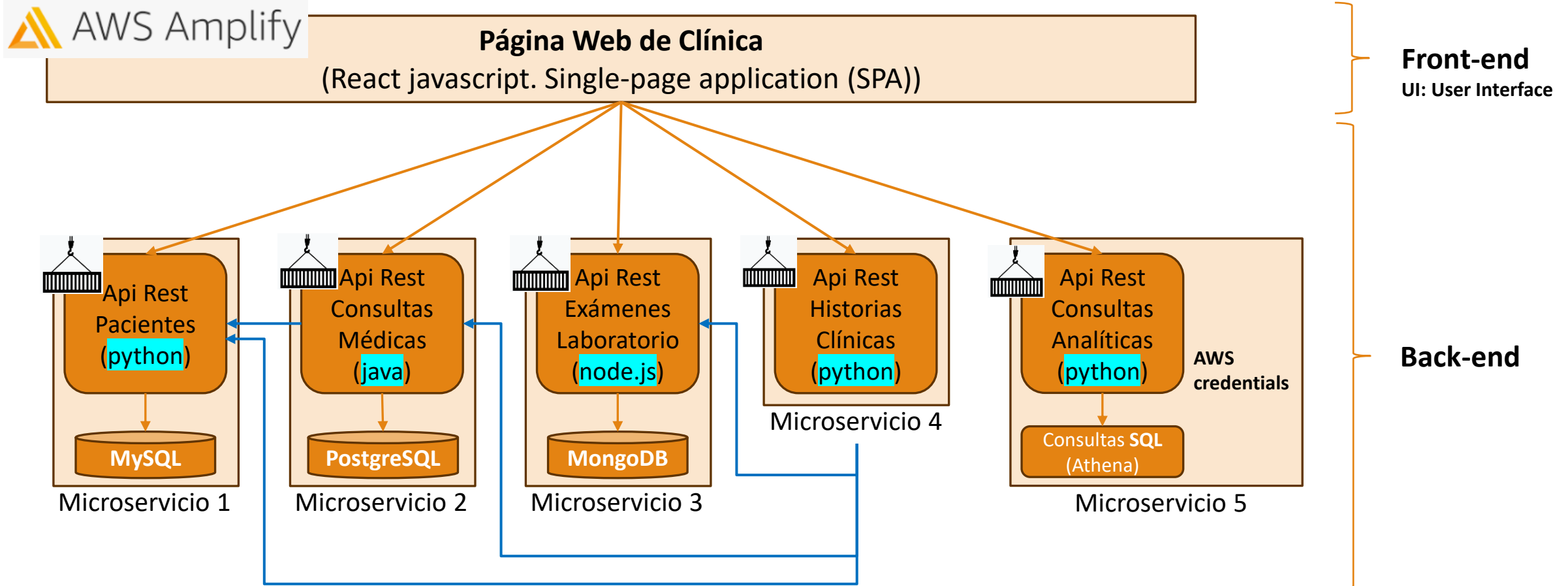
Enunciado y Rúbrica

Proyecto Parcial

- **Equipo de trabajo**
 - Cada proyecto debe desarrollarse en **grupos de 5 integrantes**.
- **Funcionalidad a implementar**
 - Los equipos tienen libertad para elegir la solución a desarrollar.
- **Avance obligatorio**
 - Cada grupo debe presentar un **avance mínimo del 50% en cada parte del proyecto** durante una **asesoría con los ACLs**.
 - El avance esperado corresponde a:
 - **Back-End:** microservicios implementados parcialmente, con al menos una base de datos conectada y consultas básicas funcionando.
 - **Front-End:** página web inicial en AWS Amplify que consuma al menos un microservicio con un par de métodos REST.
 - **Data Science:** máquina virtual de ingesta configurada, bucket S3 creado y al menos un contenedor de ingesta funcionando con datos cargados en S3.
- **Rúbrica:**
 - **Backend:** Microservicios (6 puntos)
 - **Frontend:** Web (3 puntos)
 - **Data Science** (5 puntos)
 - Diagrama de Arquitectura Solución (1 punto)
 - Exposición presencial (1 punto)
 - Exposición virtual con ACL (4 puntos)

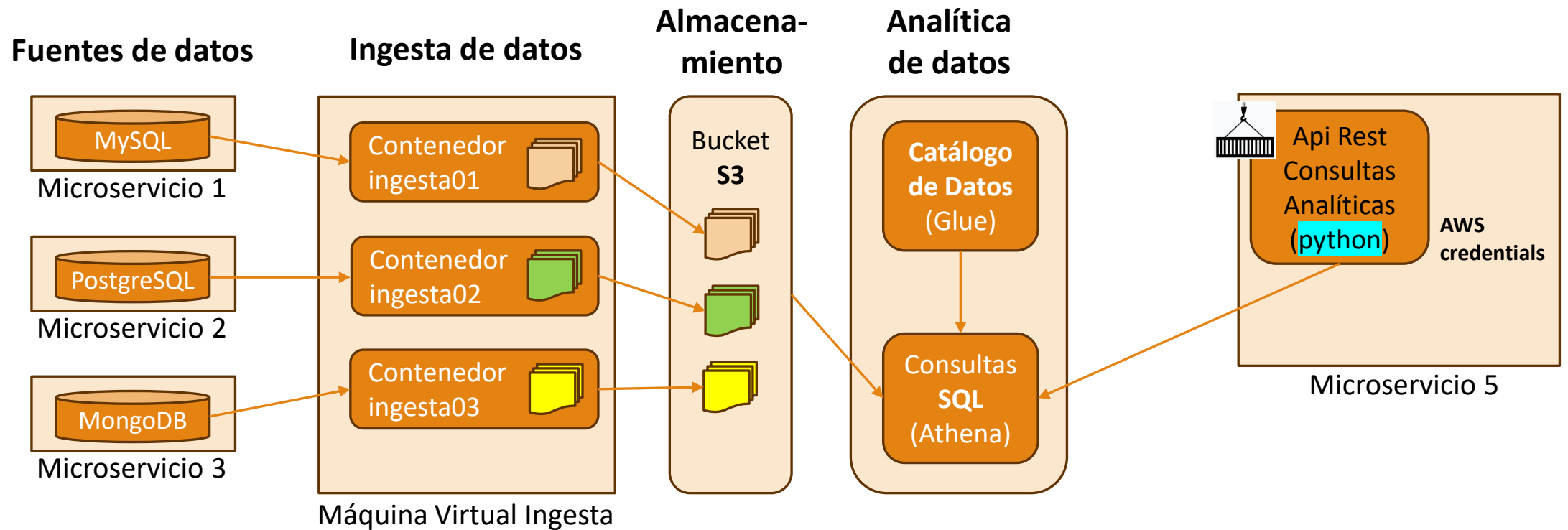
Si no se presenta a la exposición presencial su evaluación será desaprobatoria (sobre nota 10 como máximo).

Ejemplo - Backend y Frontend



Ejemplo - Data Science

Incluye Api Rest Consultas Analíticas



Enunciado

Backend: Microservicios

- Debe implementar 5 microservicios en docker:
 - **3 microservicios cada uno con su propia base de datos.** Debe usar 3 lenguajes de programación diferentes y 3 bases de datos diferentes (2 SQL y 1 No SQL). Debe presentar un **diagrama Entidad/Relación** de todas las tablas por cada base de datos SQL y las **estructuras json** de la base de datos No SQL. Cada base de datos SQL debe tener como mínimo 2 tablas relacionadas. Al menos 1 microservicio debe consumir otro microservicio (Ejemplo: Api Rest Consultas Médicas)
 - **1 microservicio que no tenga base de datos** y sólo consuma otros microservicios (Ejemplo: Api Rest Historias Clínicas).
 - **1 microservicio analítico** (Ejemplo: Api Rest Consultas Analíticas) que ejecute queries con Athena.
- **Datos de prueba:** Debe insertar masivamente, por única vez, datos ficticios (fake data) en al menos 1 tabla de cada base de datos (Mínimo 20,000 registros).
- Debe realizar el despliegue con **docker compose** en 2 **Máquinas Virtuales de Producción** con **balanceador de carga**.
- Debe documentar las 5 apis para visualizarlas en **swagger-ui**.
- Debe incluir enlaces a repositorios públicos de github con los fuentes.

Enunciado

Frontend: Web

- **Aplicación Web**
 - Desarrollar una **página web (UI)** que consuma **los 5 microservicios** incluyendo el Api Rest Consultas Analíticas (Por ejemplo: Estadística de Exámenes de Laboratorio por Especialidad de Consultas Médicas y Rango de Edades de Pacientes)
 - De cada Microservicio consultado se deben invocar **al menos 2 métodos REST**.
 - La aplicación debe desplegarse en **AWS Amplify**.
- **Tecnología**
 - Se puede utilizar **cualquier lenguaje o framework web moderno** que permita integrar APIs y construir interfaces interactivas.
- **Entregables**
 - Incluir enlaces a los **repositorios públicos de GitHub** con el código fuente.

Enunciado

Data Science: Analytics

- Debe crear una máquina virtual “**MV ingesta**”.
- Debe crear un **bucket S3** para almacenar los archivos de la ingesta de datos.
- Debe implementar **3 contenedores docker** en python para la ingesta de datos con **estrategia pull** del 100% de los registros de las tablas. Cada contenedor ingestará la data de 1 microservicio y generará archivos csv o json que cargue en el bucket S3.
- Debe implementar un **catálogo de datos** en **AWS Glue** por cada archivo que cargue al bucket S3. Debe crear un diagrama **Entidad / Relación** que relacione **todas las tablas** del catálogo de datos.
- Debe mostrar evidencia de como mínimo 4 **consultas SQL** que unan varias tablas con **AWS Athena** y crear como mínimo 2 vistas.
- Debe incluir enlaces a repositorios públicos de github con los fuentes.

Enunciado

Diagrama de Arquitectura de Solución

- Debe elaborar un Diagrama de Arquitectura de Solución en draw.io que incluya:
 - Backend
 - Frontend
 - Data Science

Agenda

Proyecto Parcial

1. Competencias a lograr
2. Enunciado
3. Entregables y Plazo

Entregables y Plazo

Grupos de 5 personas

Entregables	Plazo (Fin de Semana 7)
<ul style="list-style-type: none">• Hito 1: Exposición virtual revisada por el ACL (4 puntos de la nota final).• Hito 2: Exposición presencial y demo revisada por el profesor (16 puntos de la nota final) - Semana 8 y 9• Informe en word o pdf con evidencia de todo lo solicitado.• Resumen en power point con todo lo solicitado.	<p>Hito 1: Máximo hasta Domingo-28-Septiembre-2025 23:59</p> <p>Hito 2: Máximo hasta Domingo-5-October-2025 23:59 (Subir en Canvas)</p>