Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sección A
Ing. Manuel Fernando López Fernández
Aux 1. Gerber Emerson Ordoñez Tucubal
Aux 2. Camilo Ernesto Sincal Sipac
Primer Semestre 2025

Practica 2

TaskFlow + CloudDrive

Objetivos

General

Comprender y desarrollar una aplicación de gestión de tareas y administración básica de archivos implementada con distintos proveedores de servicios en la nube con el fin de que los estudiantes adquieran experiencia práctica en el despliegue de soluciones Cloud, además de que comprendan las diferencias, similitudes y complejidades entre ambos proveedores.

Específicos

- Desplegar una aplicación completamente en la nube
- Familiarizarse con el desarrollo Cloud con distintos proveedores
- Comparar la complejidad de la implementación de servicios de **Azure** y **AWS**

Descripción

Amazon Web Services (AWS) y Microsoft Azure son proveedores de servicios de nube que ofrecen una gran cantidad de servicios de infraestructura que ayudan a las empresas a escalar y crecer, con esto en mente se desarrollará una aplicación que permita a los usuarios gestionar tareas y almacenar archivos en la nube.

TaskFlow + CloudDrive es una aplicación web que combina dos funcionalidades principales:

1. **Gestión de Tareas:** Los usuarios pueden crear, editar, completar y organizar tareas.
2. **Administrador de Archivos:** Los usuarios pueden subir y visualizar distintos tipos de archivos en la nube.

Los estudiantes implementarán la misma aplicación en **Azure** y **AWS**, utilizando servicios equivalentes en cada plataforma.

SERVICIOS REQUERIDOS

Los servicios de **AWS** que se solicita utilizar para la solución son los siguiente:

1. **IAM**
2. **EC2**
3. **Load Balancer**
4. **S3**
5. **RDS**
6. **Lambda**
7. **Api Gateway**

Los servicios de **Microsoft Azure** que se solicita utilizar para la solución son los siguiente:

1. **Azure VM**
2. **Azure Load Balancer**
3. **Azure Blob**
4. **Azure Functions**
5. **Azure Api Management**

FUNCIONALIDADES BASICAS REQUERIDAS

Registro de usuarios

Para el registro de usuarios nuevos se deberá llenar un formulario web donde se le requerirán a los usuarios los siguientes recursos:

- Nombre de usuario.
- Correo electrónico.
- Contraseña.
- Confirmación de la contraseña.
- Imagen de perfil.

Consideraciones:

1. *El nombre de usuario debe ser único en la plataforma, no pueden existir 2 usuarios con el mismo nombre de usuario.*
2. *Las contraseñas deben ser encriptadas antes de almacenarse en la base de datos.*

Inicio de Sesión

Todo usuario podrá iniciar sesión mediante las siguientes credenciales:

- Nombre de Usuario
- Contraseña

Consideraciones:

- *Cuando el usuario inicie sesión, la pantalla o vista de "inicio" puede ser la sección de tareas o la sección de archivos, esto queda a discreción del estudiante.*

SECCION DE TAREAS

Dentro de esta sección el usuario podrá administrar toda su lista de tareas, pudiendo crear, editar, eliminar y marcarlas como completadas.

Crear Tareas

El usuario podrá crear una tarea nueva, al momento de hacer esto deberá completarlo con lo siguiente:

- Título de la tarea
- Descripción de la tarea
- Fecha de creación, la cual puede crearse de manera automática o manual

Editar Tareas

El usuario podrá cambiar la información de cualquiera de sus tareas cuando quiera, pudiendo modificar lo siguiente:

- Título de la tarea
- Descripción de la tarea

Marcar Tareas Como Completadas

El usuario en cualquier momento podrá marcar cada una de sus tareas como completadas, la forma de representar dicha acción queda a discreción del estudiante.

Eliminar Tareas

Para mantener una mejor organización sobre sus listas, el usuario deberá ser capaz de eliminar las tareas que quiera, independientemente de si están o no completadas.

SECCION DE ARCHIVOS

Además de permitir organizar las tareas del usuario, la aplicación también incluye **un File Manager** básico que permite cargar y visualizar cualquier tipo de archivos.

Cargar Archivos

El usuario debe tener la opción de cargar cualquier tipo de archivos que desee a la aplicación, siendo requisito mínimo poder cargar:

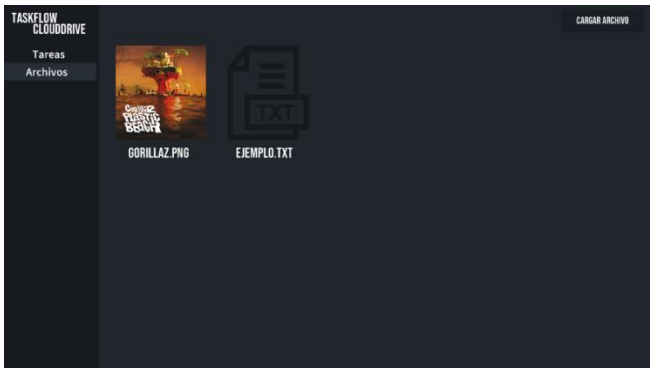
- Imágenes
- Archivos de texto

Listar Archivos

El usuario debe poder visualizar todos sus archivos en una sección dedicada para esto, la forma de mostrarlo queda a decisión del estudiante, pero debe mostrar como mínimo el nombre y tipo de archivo.

Visualizar Archivos

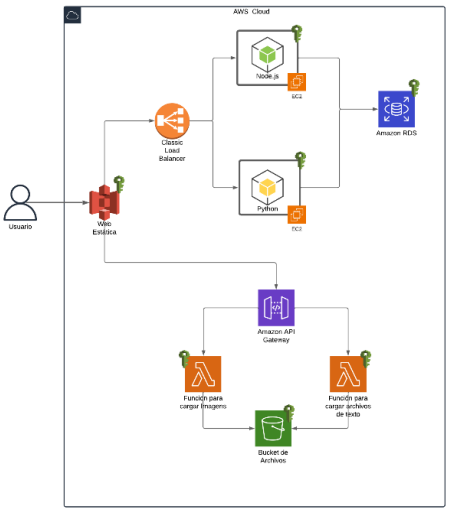
Para tener un control sobre sus cargas, el usuario deberá poder visualizar cada uno de los archivos que suba, la forma de hacerlo queda a discreción del estudiante.



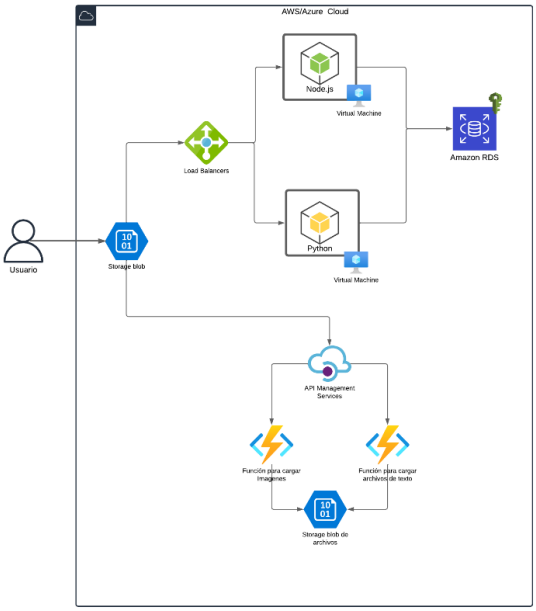
Implementación

Para la solución, se deberá implementar la siguiente arquitectura de la nube:

AWS



Microsoft Azure:



App web estática

Para la aplicación web estática, se requerirá de un cliente desarrollado en la tecnología seleccionada por el desarrollador.

El cliente deberá ser compilado y al obtener los archivos estáticos deberán de ser cargadas a un **bucket de Amazon S3 público** y a un **Blob Container de Azure** para que se pueda acceder a cualquier momento y pueda conectarse al balanceador de carga respectivo de cada proveedor.

Este bucket o blob container deberá llevar el nombre de **practica2Semi1<<Sección>>1s2025paginawebg#** (El # es el número de grupo).

Almacenamiento de Archivos

Para alojar los archivos del usuario, se debe de crear un **Bucket de Amazon S3 público** y un **Blob Container de Blob Storage de Azure** con el nombre **practica2semi1<<Sección>>1s2025Archivosg#** (El # es el número de grupo).

Se debe asegurar que todas las fotos sean públicas, para que se puedan acceder desde la aplicación mediante la dirección URL del objeto.

Serverless

Para la carga y obtención de imágenes en S3 y Blob Storage se implementarán servicios Serverless, específicamente **Lambda con Api Gateway** para **AWS** y **Azure Functions con Api Management** para **Azure.**

Para esta configuración deberá crear como mínimo **2 funciones** para Lambda y Azure Functions:

1. Función para cargar imágenes.
2. Función para cargar documentos de texto.

Cada una de estas funciones deberá estar configurada en rutas en los servicios de Apis correspondientes a cada uno de los proveedores, las cuáles se consumirán en la web estática.

Base de datos

Para la base de datos de la solución se requerirá el uso del servicio **Amazon RDS** y se deberá tomar las siguientes consideraciones:

1. La base de datos deberá contener toda la información necesaria para almacenar la lógica de negocio de la solución.

2. La contraseña del usuario tiene que estar encriptada
3. NO se deberán almacenar las imágenes directamente en la base de datos, para ello se recomienda almacenar únicamente la URL del objeto de las imágenes almacenadas.

Servidores

Estos componen el Back-End de la solución, se deberá tomar en cuenta las siguientes consideraciones:

1. Se deberán realizar 2 servidores con exactamente las mismas funciones con la diferencia que uno deberá estar programado en **NodeJS** y el otro en **Python**.
2. Para conectarse a S3 se deberá utilizar el SDK de AWS.
3. Cada uno de los servidores deberá estar montado en una instancia de **EC2** y **VM de Azure**.
4. Se deberá configurar el grupo de seguridad de ambas instancias únicamente con los puertos que necesite su servidor.
5. Para el caso de Amazon se deberá crear un usuario que tenga las políticas para poder configurar únicamente este servicio.

Balanceador de carga

Para balancear la carga entre las instancias de **EC2** se deberá realizar un balanceador de carga en el servicio **AWS Load Balancing (ELB)** y para las instancias **de VM de Azure** se **utilizará el balanceador respectivo del proveedo**r, se deberá tomar en cuenta las siguientes consideraciones:

1. Se deberá de ser capaz de al apagar uno de los 2 servidores la aplicación web pueda seguir en funcionamiento.
2. Debe estar configurado para redirigir el tráfico a las 2 instancias que poseen los diferentes servidores.

Manual técnico o de configuración

Se requiere que se realice un manual redactado en formato Markdown dentro del repositorio, lo que se debe adjuntar al manual es lo siguiente:

1. Datos de los estudiantes.
2. Descripción de la arquitectura que utilizaron.
3. Descripción de los diferentes usuarios de IAM, para el caso de Amazon, que se utilizaron con sus respectivas políticas.
4. Capturas de pantalla de los recursos:
    a. **AWS:**

i. Buckets de Amazon S3.
ii. Instancias de EC2.
iii. Balanceador de carga de EC2.
iv. Base de datos de RDS.
v. Funciones Lambda
vi. Configuración de Api Gateway

b. **Azure:**

i. Blob Containers de Azure
ii. Instancias de VM de Azure
iii. Balanceador de carga de Azure
iv. Funciones de Azure Functions
v. Configuración de Api Management

5. Una conclusión sobre las diferencias percibidas entre el desarrollo de Azure y AWS

CONSIDERACIONES GENERALES

- Nombre del repositorio: **Semi1-Sección-1S2025-Grupo#-Practica2** (Donde # es su número de grupo).
- Repositorio de Github debe estar en modo privado y documentado con el formato Markdown (manual de configuraciones).
- La página web deberá estar cargada en **Amazon S3** y **Blob Storage**.
- **NO se aceptará ni calificará nada de forma local, todo deberá estar desplegado en la nube de AWS.**
- Agregar como colaborador en el repositorio al usuario del auxiliar del laboratorio dependiendo de la sección.
  - **Auxiliar 1**: Emerson-O
  - **Auxiliar 2**: CamiloSincal
- El proyecto debe desarrollarse en grupos.
- En el caso de AWS usar los respectivos usuarios de IAM con sus respectivas políticas de acuerdo con el servicio que se está utilizando.
- Cualquier copia total o parcial será reportada a la escuela y automáticamente se obtendrá una nota de 0.
- El entregable será la URL de su repositorio Git.

FECHA DE ENTREGA: 04/04/2025 a las 23:59 por UEDI.