

Universidad de San Carlos de Guatemala  
Facultad de ingeniería  
Escuela de Ciencias y Sistemas  
Seminario de Sistemas 1  
Primer Semestre 2024  
Ing. Manuel Fernando López Fernández  
Aux. Pablo Andrés Axpuc Arévalo  
Ing. Neftali de Jesus Calderón Méndez  
Aux. Luis Enrique Rivera Najera



## Practica 1 **FAUNADEX**

### **Objetivos**

- **General**
  - Comprender la arquitectura de cloud computing y los principales servicios de AWS
- **Específicos**
  - Crear un entorno de desarrollo para una aplicación web
  - Implementar una arquitectura para microservicios
  - Integrar servicios de AWS

### **Descripción**

Amazon Web Services (AWS) es una plataforma de servicios de nube que ofrece potencia de cómputo, almacenamiento de bases de datos, entrega de contenido y otra funcionalidad para ayudar a las empresas a escalar y crecer.

Se desarrollará una aplicación web similar a una aplicación para almacenamiento de fotos, esta permitirá subir todo tipo de fotos. Tendrá las secciones para listar todas las fotos subidas, listarlas por álbumes o mostrar las fotos de perfil del usuario.

Mediante la ayuda de un balanceador de carga se tendrá resguardado lo que es la conexión de la aplicación, a la hora que uno de los 2 servidores sea apagado este debe de ser capaz de redirigir el tráfico al servidor aun en funcionamiento.

Para esta práctica se evaluará los conocimientos básicos de los siguientes servicios de AWS:

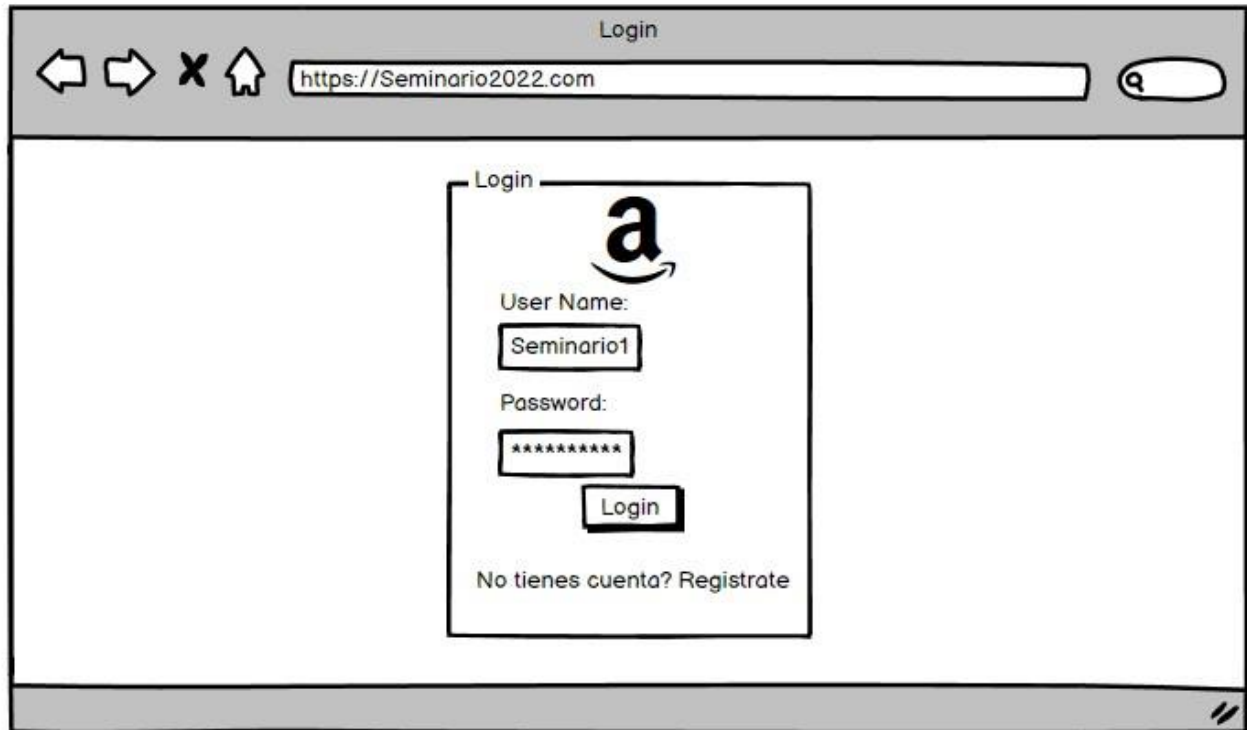
- Usuarios IAM
- Instancias EC2
- Security Groups
- Balanceador de Carga
- Buckets de S3
- RDS

A continuación, se detallarán las diferentes secciones de la aplicación web:

## Login

Para la autenticación se utilizará únicamente:

- Nombre de usuario •
- Contraseña.



The image shows a web browser window with the address bar displaying "https://Seminario2022.com". The page content is a login form titled "Login" with the Amazon logo. The form includes fields for "User Name:" (containing "Seminario1") and "Password:" (containing "\*\*\*\*\*"). Below these fields is a "Login" button. At the bottom of the form, it says "No tienes cuenta? Register".


## Registro


Para registrar un nuevo usuario obligatoriamente se pedirán los datos

- Usuario
  - No deben de existir 2 usuarios con el mismo nombre.
- Nombre Completo
- Contraseña
- Confirmación de Contraseña
  - Esta debe de coincidir con la contraseña.
- Foto
  - La foto puede ser tomada en ese momento o elegir un archivo de la computadora esto queda a criterio del estudiante.

Registro

https://Seminario2022.com

  
[Seleccionar Foto](#)

 Username

Username

Password

Confirm Password:

  
[Registrarse](#)  
[Ya tienes cuenta? Login](#)


## Página de Inicio


Únicamente se mostrará la información del usuario junto a su foto, y las opciones de subir foto, ver foto, editar perfil, editar álbumes y cerrar sesión.

Mi Perfil


https://Seminario2022.com

Datos Personales

  
[Ver fotos](#)

 Nombre de Usuario

 Nombre Completo

  
[Subir Foto](#)

[Editar Perfil](#)  
[Cerrar Sesión](#)  
[Editar Álbumes](#)

## Editar Perfil

En esta sección se puede editar el usuario, nombre y foto del usuario. Se pedirá la contraseña para poder confirmar la edición. (Esta no se puede cambiar)

The screenshot shows a web browser window titled "Editar Perfil" with the URL "https://Seminario2022.com". The page layout includes a profile picture placeholder on the left with a "Nueva Foto" button below it. To the right, there are form fields for "Username" (containing "Seminario1"), "Name" (containing "Seminario de Sistemas 1"), and "Confirm Password" (containing "\*\*\*\*\*"). Below these fields are two buttons: "Editar" and "Regresar".

## Subir foto (Animales)

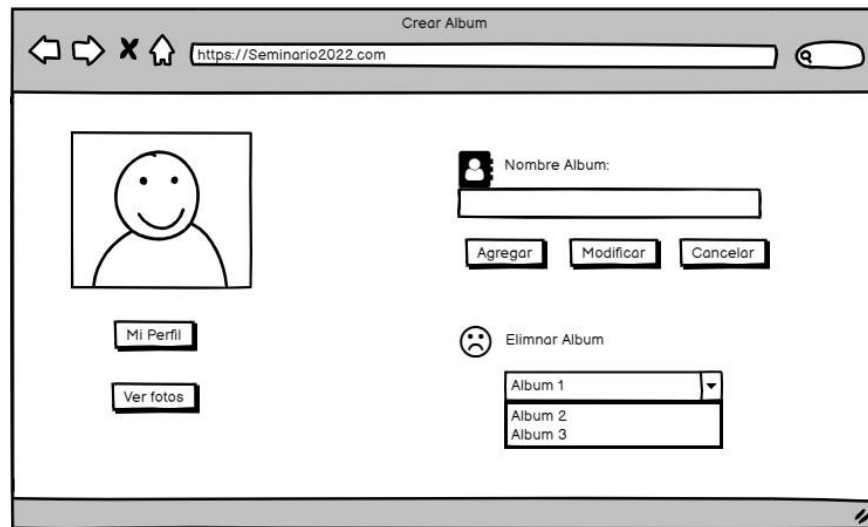
En esta sección se elegirá la foto a subir, se le asignará un nombre y el álbum en donde esta pertenecerá. Si en dado caso no exista el álbum deberá aparecer la opción para crearlo.

The screenshot shows a web browser window titled "Subir Fotos" with the URL "https://Seminario2022.com". The page layout includes a profile picture placeholder on the left. To the right, there are form fields for "Nombre Foto" and "Album" (a dropdown menu currently showing "Album 1"). A "Cargar Foto" button is positioned to the right of the "Nombre Foto" field.

## Editar Álbumes

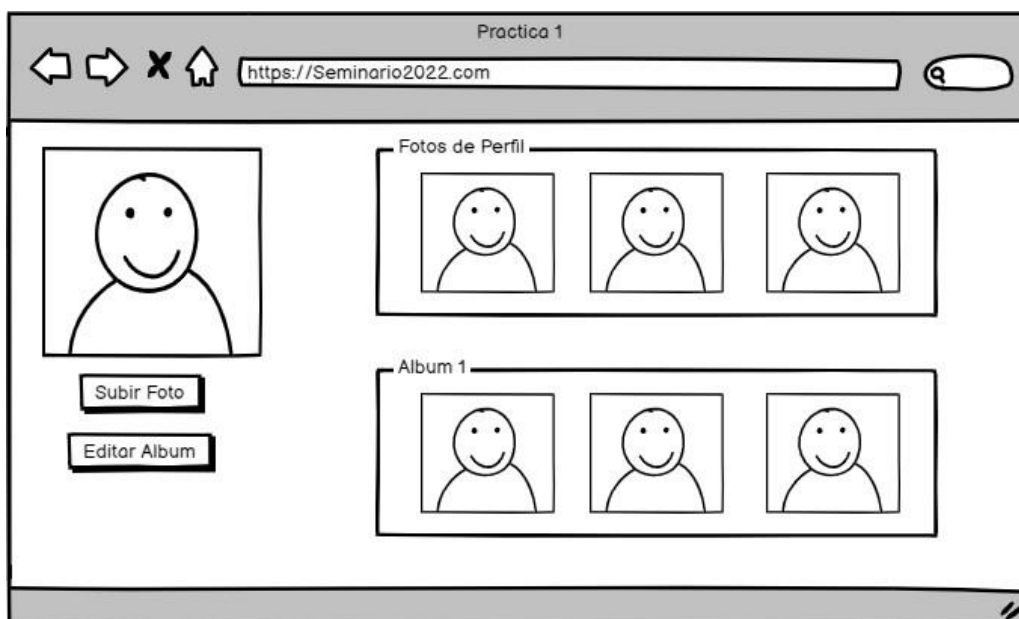
En esta sección se podrá crear, modificar y eliminar un álbum.

- Nota: a la hora de borrar el álbum, las fotos que están en este ya no deben de aparecer.



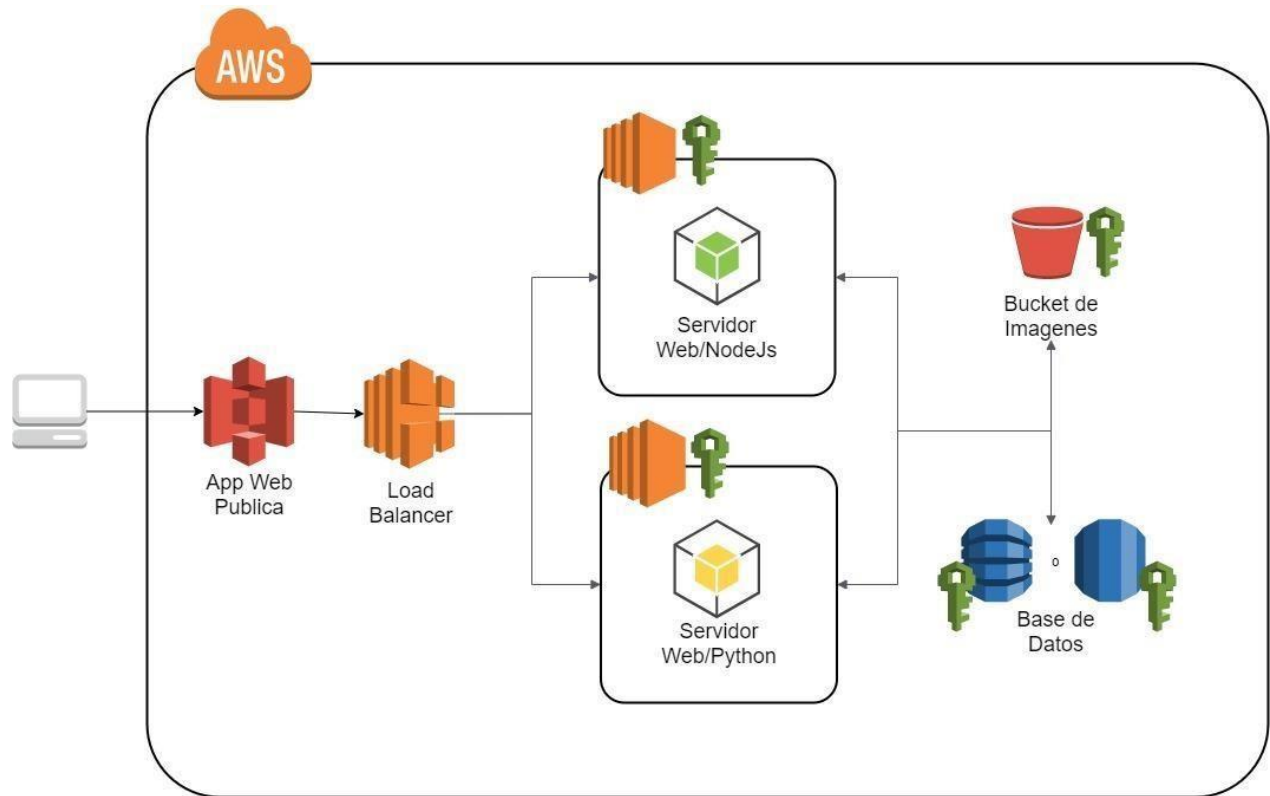
## Ver fotos

En esta sección se mostrarán todos los álbumes con las fotos dentro de ellos (Todas las fotos de perfil irán en un álbum por default o en una sección aparte esto queda a criterio del estudiante)



## Implementación

Para el desarrollo de esta aplicación web, se implementará en 1 arquitectura server la que se encuentra en la imagen a continuación.



### Base de datos

Para la base de datos de este proyecto se utilizará el servicio de **RDS**.

- La forma de trabajar queda a criterio del estudiante si utiliza una base de datos relacional o no relacional.
- La base de datos contiene toda la información necesaria para poder realizar las diferentes funciones de su página web
- NO almacenar las imágenes directamente para esto solo necesita almacenar la url donde esta almacenada la imagen en el bucket de S3. Por ejemplo:
  - Si la imagen esta almacenada en la carpeta **Fotos\_Perfil** y el nombre de la imagen es foto1 entonces solo se deberá almacenar el siguiente registro: "Fotos\_Perfil/foto1.jpg", como se enseñó en el laboratorio.
  - Esto con el fin de facilitar el trabajo de convertir imágenes.
- **La contraseña del usuario tiene que estar Encriptada con el algoritmo MD5.**

## Servidores

- Estos forman parte del backend de la aplicación.
- Se realizarán 2 servidores con exactamente las mismas funciones con la diferencia que uno estará programado en **NodeJs** y el otro en **Python**.
  - Utilizar siempre el **SDK** de aws.
  - A las funciones se refiere a los métodos Post, Get, Delete, etc.
- Cada uno de estos servidores estarán montados en instancias de **EC2** para esto tener en cuenta lo siguiente.
  - Configurar el Security group de ambas instancias únicamente con los puertos que necesite su servidor.
  - Utilizar un usuario que tenga las políticas para poder configurar únicamente este servicio.
- Únicamente debe de estar el servidor en cada instancia de EC2 y no la base de datos en la misma. Así los 2 servidores compartirán base de datos.

## Balanceador de Carga

Como se sabe se puede configurar el balanceador de carga de diferentes formas, en esta ocasión utilizaremos el servicio de AWS Load Balancing, este nos permitirá redirigir el tráfico de las peticiones a algunos de los servidores de las instancias EC2.

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

## Página web

- Puede ser realizada en cualquier lenguaje de programación al criterio del estudiante. Esta estará alojada en un Bucket de S3 **público** para que se pueda acceder a cualquier momento y poder conectarse al balanceador de carga. Este bucket debe de llevar el nombre de **practica1-G#-paginaweb** (El # es el número de grupo).

## Almacenamiento de Imágenes

- Para alojar las imágenes tanto las de perfil como la de los diferentes álbumes se debe de crear un bucket de S3 con el nombre **practica1-G#-imágenes**. (El # es el número de grupo)
  - Este debe tener 2 carpetas en las cuales se almacenarán las imágenes.
    - **Fotos\_Perfil:** Todas las fotos de perfil que se suban en la aplicación sin importar el usuario aquí se almacenaran todas.
    - **Fotos\_Publicadas:** Se almacenarán todas las fotos sin importar álbum o usuario, todas las que sean cargadas desde la aplicación web.
- Asegurarse que todas las fotos sean públicas para que se puedan acceder desde su aplicación mediante la dirección url que le proporciona el Bucket.

## Manual de Configuración

Se necesita que se realice un manual de configuración redactado en Markdown dentro del repositorio, lo único que debe de llevar este manual es lo siguiente:

- Datos de los estudiantes.
- Descripción de la arquitectura que utilizaron.
- Descripción de los diferentes usuarios de IAM que se utilizó con sus respectivas políticas.
- Capturas de Pantalla:
  - Buckets de S3
  - EC2
  - Base de Datos RDS en su defecto.
  - Aplicación Web

Pueden realizarlo en el README.md del repositorio.

## CONSIDERACIONES:

- Página Web debe estar cargada en S3
- Repositorio en github en modo privado y documentado con el formato Markdown (Manual de configuraciones).
- Agregar como colaborador en el repositorio al usuario dependiendo la sección
  - **andres-a2206 – Sección A**
  - **LuisRivera2016 – Sección B**
- La práctica debe ser en grupos.
- Usar los respectivos usuarios de IAM con sus respectivas políticas de acuerdo con el servicio que se está utilizando. Por ejemplo:
  - Si se está utilizando S3 crear un usuario con las políticas respectivas.
  - Si se está utilizando EC2 crear el usuario con las políticas para este servicio.
- Cualquier copia total o parcial será reportada a la escuela y se obtendrá una nota de 0 . •  
Fecha de entrega: 1/03/2024 a las 23:59 por **Uedi**