Universidad de San Carlos de GuatemalaFacultad de Ingeniería
Escuela de Ciencias y Sistemas
Manejo e Implementación de ArchivosVacaciones Segundo semestre 2022
Catedrático: Ing. Oscar Paz
Tutor académico: Diego Andrés Obín Rosales

Proyecto 2

AviCar

Introducción:

El curso de Manejo e Implementación de Archivos busca que los estudiantes aprendan los conceptos sobre la administración de archivos, tanto en hardware como software, sistemas de archivos, particiones, entre otros conceptos, así mismo trata que los estudiantes apliquen estos conceptos en el desarrollo de un proyecto para que así de esta manera puedan aprender cada uno de los temas impartidos durante la clase magistral y el laboratorio para que luego se le pueda dar paso a los conocimientos que se impartirán en cursos posteriores como lo son los almacenamientos en línea.

Objetivos:

- Aprender a administrar archivos y estructuras en NodeJS
- Comprender la funcionalidad de un flujo de archivos JSON
- Aplicar la teoría de archivos JSON
- Utilizar un framework (Angular, React, Vue)
- Administrar los usuarios y permisos por medio de grupos
- Restringir y administrar el acceso a los archivos de modo administrador, cliente y recepcionista.
- Crear una aplicación visual
- Utilizar los servicios de una nube (para este proyecto utilizaremos AWS)

Descripción

“AviCar” es el sistema que se desarrollará para el gestionamiento de viajes de todo turista alrededor del mundo; con el fin de garantizar una completa y agradable experiencia en época de post pandemia. La finalidad es centralizar los datos y que el cliente haga la menor cantidad de validaciones al momento de planificar su viaje.

Usuarios

1. **Usuario Turista**: Toda persona que desea hacer un viaje a cualquier parte del mundo, puede visualizar todos los vuelos en la página de inicio y si desea; realizar la reservación de un boleto de avión, así como, el alquiler de un automóvil al momento de llegar a su destino.
2. **Usuario Recepcionista**: Este usuario será el encargado de aprobar o rechazar la solicitud de vuelo y renta del automóvil de los usuarios que lo soliciten. (Este tipo de usuario únicamente podrá aceptar o rechazar las solicitudes de los usuarios turistas).
3. **Usuario Administrador:** Este usuario será el único usuario que podrá agregar o eliminar usuarios turista o recepcionista, podrá revisar el historial de vuelos de cada usuario turista y agregar o eliminar viajes y automóviles.

Registros e inicio de sesión

La página web debe de tener una página de inicio y en ella se debe de poder iniciar sesión o generar nuevos registros de usuarios para poder acceder al sistema y tener las vistas correspondientes a su tipo de usuario. Desde la página web se pueden registrar únicamente turistas. El administrador es quien puede crear usuarios Turista y Recepcionista y Registro de Autos y Vuelos desde sus vistas correspondientes.

1. **Inicio de Sesión**:
    - **a.** Correo electrónico y/o usuario
    - **b.** Contraseña

2. **Registro de Usuario:**
    a. Nombre completo
    b. Usuario
    c. Foto de perfil
    d. Correo electrónico
    e. Contraseña
    f. Confirmación de contraseña
3. **Registro de Viajes**
    a. Nombre de la agencia
    b. Ciudad de origen
    c. Ciudad de destino
    d. Días de vuelo
    e. Precio de vuelo
4. **Registro de Autos**
    a. Nombre de la agencia
    b. Marca
    c. Placa
    d. Modelo
    e. Precio
    f. Ciudad en la que se encuentra el vehículo
5. **Registro de Recepcionista**
    a. Nombre completo
    b. Usuario
    c. Foto de perfil
    d. Correo electrónico
    e. Contraseña
    f. Confirmación de Contraseña

Renta de Vuelos y Autos

El turista debe de poder realizar una reserva de vuelo y/o automóvil que él desee en el momento. Para realizar la renta de cada uno de ellos podrá filtrarlos por sus características:

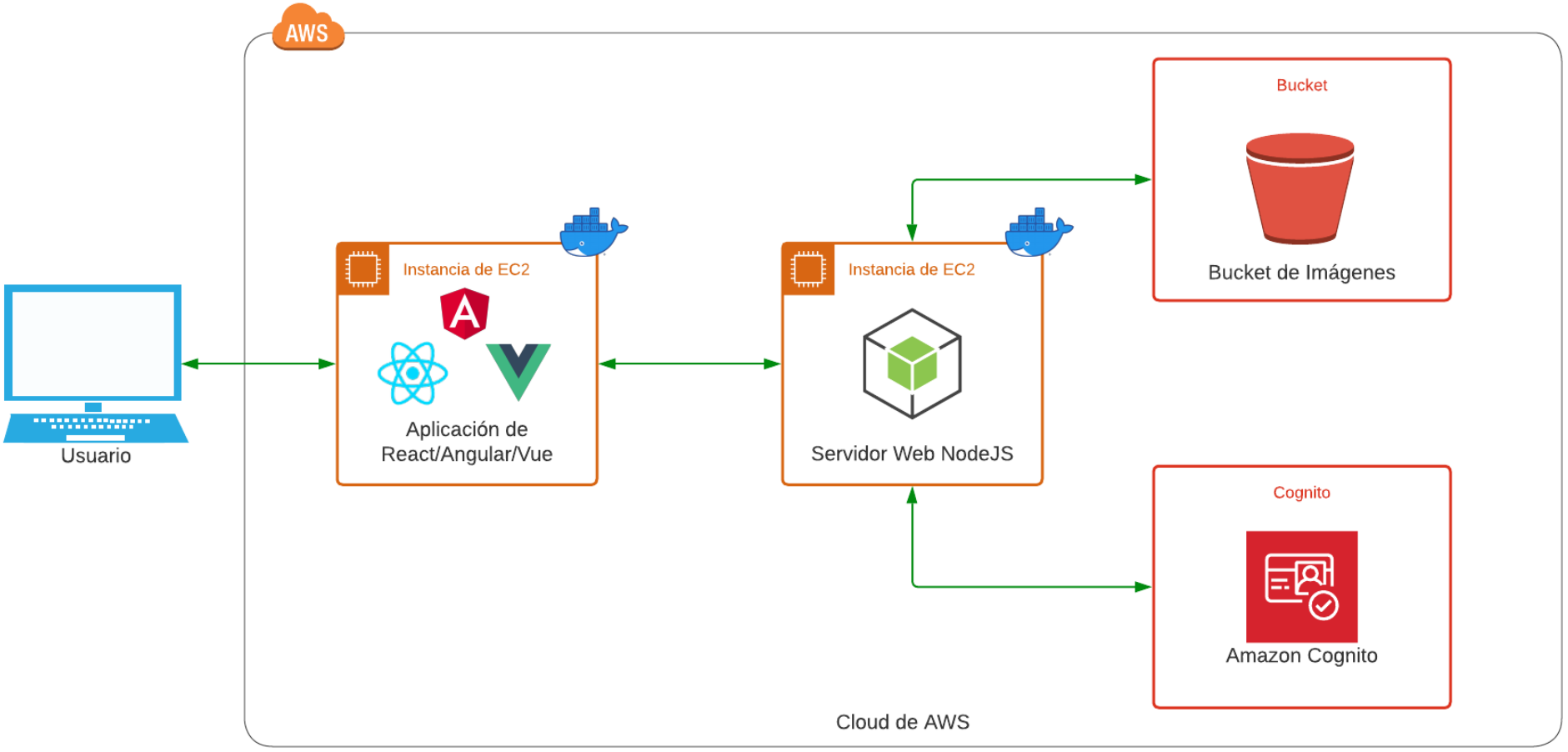
1. **Vuelos:**
    - **a.** Nombre de agencia
    - **b.** Ciudad de Origen
    - **c.** Ciudad de Destino
    - **d.** Días de vuelo
    - **e.** Precio de vuelo
2. **Autos:**
    - **a.** Nombre de la agencia
    - **b.** Marca
    - **c.** Modelo
    - **d.** Precio

Requisitos técnicos

1. “AviCar” debe de contar con todos los servicios requeridos para su funcionamiento: backend, frontend y gestor de datos con archivos JSON.
2. Para el registro de nuevos usuarios deberá de utilizar el servicio de Cognito de AWS de manera que se pueda verificar la identidad del nuevo usuario, queda a discreción del estudiante si desea generar un código de verificación o verificación por enlace.
3. Para el almacenamiento de las fotos de perfil de cada usuario deberá de utilizar el servicio de S3 de AWS. El nombre de este debe de ser **appweb-#carne-p2**, queda a discreción del estudiante la forma de implementarlo.
4. No deberá de utilizar ningún tipo de base de datos para el control de la información del programa, únicamente deberá de utilizar archivos en formato JSON dentro de su aplicación de Backend para el control de estos.
5. El programa deberá de ejecutarse en una EC2 de AWS, el estudiante puede desplegar la aplicación de la manera que mejor le convenga, se recomienda utilizar Docker para el levantamiento de las aplicaciones.
6. Deberá de crear los usuarios IAM que considere necesarios para el manejo y uso de los servicios de AWS que lo requieran con su política asociada. Se recomienda un usuario por cada servicio utilizado.
7. Git: uso de un repositorio para el control de versiones. Debe de agregar al auxiliar al repositorio y darle los permisos de propietario: @DiiAns23
8. Documentación: la documentación debe de guardarse en el repositorio en una carpeta llamada “Documentación”. Deberá de contener los archivos “manual técnico” y “manual de usuario” ambos archivos deberán de estar bien estructurado utilizando el formato MarkDown (.md)
    - **a.** Manual Técnico:
        - **i.** Objetivos
        - **ii.** Explicación de la arquitectura utilizada
        - **iii.** Descrición de cada usuario IAM creado con sus políticas asociadas
        - **iv.** Capturas y descripción de cómo se configuró cada servicio
        - **v.** Conclusiones
    - **b.** Manual de Usuario:
        - **i.** Objetivos
        - **ii.** Explicación y descripción de la aplicación
        - **iii.** Pasos con capturas de cómo utilizar la aplicación

Arquitectura de la aplicación

Se le recomienda utilizar la siguiente arquitectura para la implementación del proyecto:



El proyecto debe realizarse de forma individual, ***Se utilizará software para la detección de copias, las copias tendrán una nota de 0 y serán reportadas a la escuela.***

El lenguaje por utilizar es NodeJS. No se permite el uso de otro lenguaje.

Únicamente se calificará el proyecto sobre una instalación física de una distribución GNU/Linux.

NO se permite la modificación de código durante la calificación. El estudiante únicamente podrá utilizar el ejecutable entregado.

No se permite la utilización de bases de datos para el almacenamiento de la información.

ENTREGA: MIÉRCOLES 28 DE DICIEMBRE DE 2022 ANTES DE LAS 23:59 HORAS