UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

SISTEMAS OPERATIVOS 1 SECCIÓN N

ING. SERGIO ARNALDO MENDEZ AGUILAR

AUX. GERMAN JOSÉ PAZ CORDÓN

SEGUNDO SEMESTRE 2022

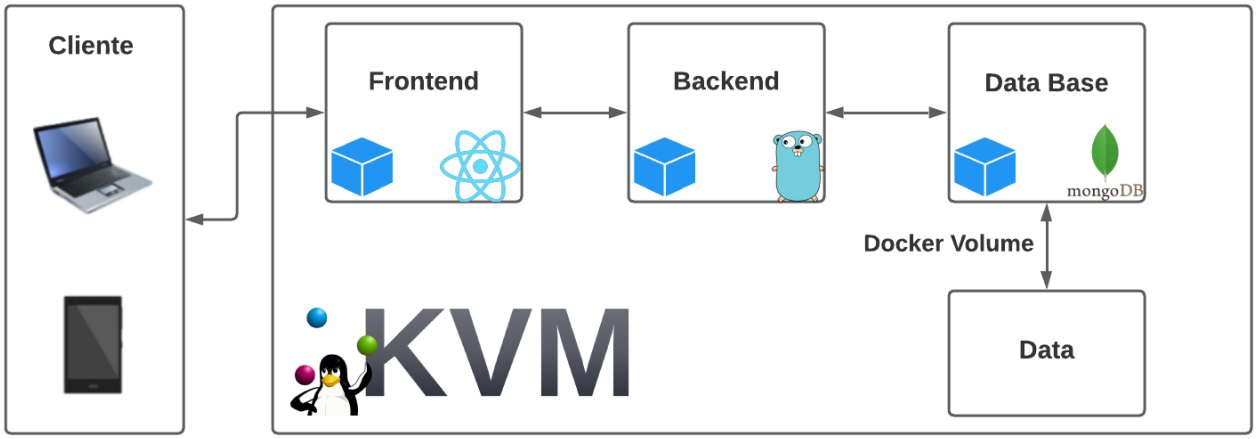
PRÁCTICA 1

KVM, Docker y Containers

OBJETIVOS

- Utilizar virtualización por hardware.
- Comprender cómo funcionan los contenedores.
- Practicar comandos de Docker.
- Implementar Docker-Compose.
- Generar persistencia de datos mediante Docker Volume.
- Utilizar Docker Hub.

ARQUITECTURA



DESCRIPCIÓN

Se solicita implementar una aplicación la cual contará con un frontend, backend y una base de datos, esta aplicación será desplegada mediante el uso de contenedores. Además, se solicita que utilizar virtualización por hardware por ello el desarrollo de la aplicación deberá hacerse utilizando del hipervisor de tipo 1 KVM.

FRONTEND

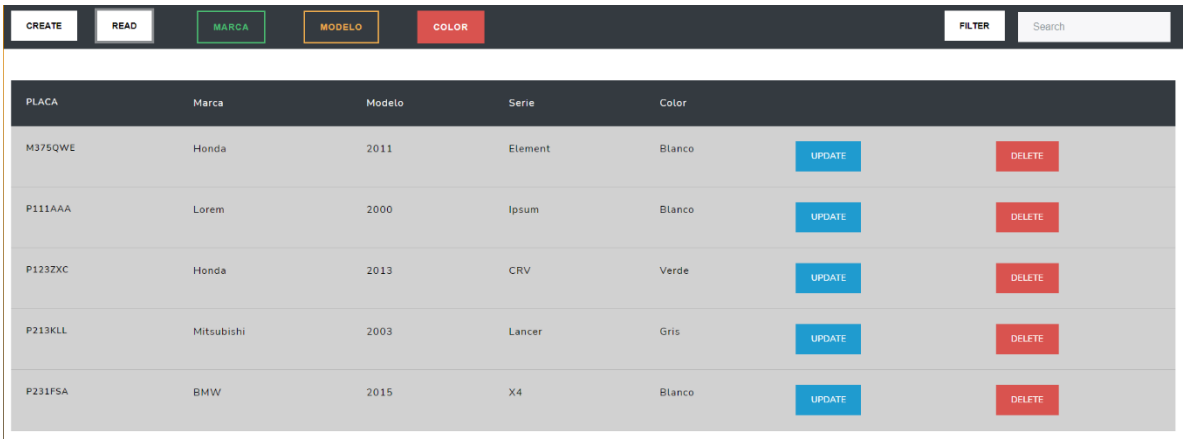
El estudiante deberá realizar una aplicación desarrollada en **React**, esta será un **CRUD** para el control de vehículos, el cual contará con las siguientes funciones:

- **Create:** Registrar un nuevo vehículo.
- **Read:** Visualizar todos los vehículos registrados.
- **Update:** Modificar el registro de un vehículo en específico.
- **Delete:** Eliminar registro de un vehículo en específico.

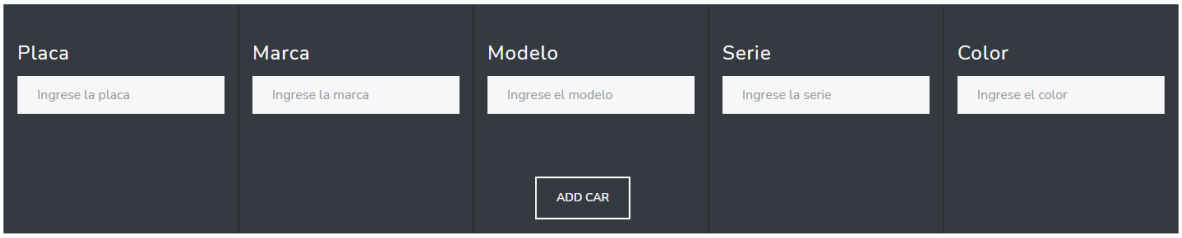
Además, se contará con una función de filtrado de datos mediante el cual se podrá visualizar todos los vehículos que cumplan con la característica de filtrado. El filtrado se podrá realizar mediante: **Marca**, **Modelo** y **Color**.

Vistas sugeridas de la aplicación:

Visualizar todos los vehículos



Registrar un vehículo



Actualizar datos de un vehículo

Visualizar vehículos filtrados

NOTA: **Las vistas del frontend quedan a discreción del estudiante, no necesariamente deben de ser como muestran las imágenes.**

BACKEND

Los datos para el funcionamiento de la aplicación serán obtenidos y almacenados desde el servidor, por lo que se le solicita al estudiante realizar un servidor desarrollado con el lenguaje Go.

DATA BASE

Se le solicita al estudiante realizar una base de datos con MongoDB para almacenar todos los datos de los vehículos registros. Adicionalmente se deberá llegar un registro de las funciones consultadas. Únicamente se agregará un log al momento de registrar, actualizar o eliminar un vehículo.

Datos que deberán ser almacenados sobre los vehículos.

```
{
  "Placa": "P123ABC",
  "Marca":  "Toyota",
  "Modelo": 2021,
  "Serie": "Corola",
  "Color": "Rojo"
}
```

Datos que deberán ser almacenados sobre el registro de actividad.

DATOS INCIALES

Por motivos de calificación se solicita que el estudiante tenga los siguientes datos ingresados en su base de datos:

- **2 vehículos de la misma marca.**
- **2 vehículos del mismo modelo**
- **2 vehículos del mismo color.**

En total deben haber registrados como mínimo 6 vehículos diferentes en la base de datos al iniciar la aplicación.

DOCKER

Para la creación y ejecución de los contenedores deber hacer uso de docker compose mediante los comandos:

- **docker-compose up -d**
- **docker-compose down**

Además, se solicita que el estudiante utilice dokcer hub para publicar las imágenes creadas para el funcionamiento de la aplicación. Deberá utilizar el siguiente formato para las imágenes:

- **<<user_dockerhub>>/frontend_p1_<<carnet>>**
- **<<user_dockerhub>>/backend_p1_<<carnet>>**

Se requiere que la base de datos tenga persistencia de manera que al realizar un **docker-compose down** y luego un **docker-compose up -d**, los datos de vehículos y funciones deben seguir existiendo en la base de datos.

RESTRICCIONES

- La práctica se realizará de manera individual.
- La aplicación debe tener un aspecto profesional.
- La interfaz gráfica debe ser realizada con React.
- El servidor debe ser realizado con Golang.
- Las máquinas virtuales deberán ser de Ubuntu, Centos o Rocky Linux.
- Utilizar un repositorio de GitHub, el cual debe ser privado con el nombre: **so1_practica1_<<carnet>>.**
- El código fuente debe ser administrado por medio de un repositorio de GitHub.
- Agregar al auxiliar al repositorio: GermanJosePazCordon
- Cualquier copia parcial o total será reportada a la Escuela de Ciencias y Sistemas para que proceda como indica el reglamento.

ENTREGABLES

- Código fuente de la solución.
- Enlace del repositorio de GitHub.
- Manual técnico con explicación de todos los componentes utilizados en la práctica. Este manual debe ser un archivo .md (Markdown), por lo que deberá estar en el README del repositorio.

FORMA DE ENTREGA

- Mediante UEDI, subiendo el enlace del repositorio con el código fuente y el manual técnico.

La entrega se debe realizar antes de las 23:59 del 21 de agosto de 2022.