**SEGUNDO PARCIAL DE INFRAESTRUCTURA COMO CÓDIGO**

*Datos Obligatorios:*

Nombre y Apellido: Elena Castro

Dni: 25.466.551

Fecha: 21/6/2023

**Fila 2**

El examen consta de una parte teórica y una parte práctica, y se evaluará de acuerdo a

los siguientes criterios:

Criterios de evaluación para la parte teórica (30 puntos en total):

* Respuestas correctas y completas a cada una de las preguntas planteadas.
* Presentación clara y ordenada de las respuestas.
* La parte teórica del examen debe estar desarrollada en el mismo archivo de word proporcionado para el parcial, junto con el diseño del diagrama que se pide en la parte práctica.

Criterios de evaluación para la parte práctica (70 puntos en total):

* Proporcionar un archivo de Docker Compose completo y estructurado para la aplicación web y las bases de datos.
* Configurar el mapeo de puertos en Docker Compose para acceder a la aplicación web desde el puerto local de los desarrolladores.
* Establecer la configuración de volúmenes en Docker Compose para persistir datos de la aplicación web y las base de datos.
* Desarrollar el archivo Docker Compose siguiendo buenas prácticas de desarrollo, con código legible, estructurado y comentarios adecuados.
* Proporcionar un diagrama de arquitectura que muestre la interacción entre los componentes en el entorno Docker Compose.
* Todo el código debe estar subido a un repositorio de GitHub, con el nombre del título y el commit que pide la consigna, en el horario correspondiente de entrega. Al subirlo se deben ignorar los volumenes y el node\_modules.

**Teórico**

Explique la diferencia entre imagen y contenedor. (5 Puntos)

¿Ventajas de la Infraestructura como Código? (5 Puntos)

¿Qué es el Docker Engine? (5 Puntos)

¿Para qué sirve el Dockerfile? (5 Puntos)

¿A qué se refiere el término “Orquestar Contenedores” y para que se usa en el mundo IT? (5 Puntos)

¿Para qué sirve Docker Compose? (5 Puntos)

**Práctica**

En Aguas de Formosa SA, se ha anunciado recientemente que se dejará de brindar soporte a MySQL, el motor de base de datos utilizado en el sistema desde sus inicios. Como programador con más experiencia en la empresa, se te ha encomendado la tarea de preparar una nueva infraestructura de desarrollo utilizando Cloud Computing.

Antes de considerar la adopción de servicios en la nube como AWS, es necesario abordar la migración del sistema antiguo desarrollado en Delphi hacia tecnologías más modernas. En este sentido, se ha decidido utilizar MariaDB como sucesora de MySQL para evitar problemas a largo plazo. Además, se ha contratado un nuevo equipo de desarrolladores especializados en el stack MERN (MongoDB, Express.js, React.js y Node.js) para trabajar en otra parte del nuevo sistema.

Tu objetivo es configurar el entorno de desarrollo para hacer uso de MariaDB y MongoDB, y asegurarte de que ambas bases de datos estén en línea y accesibles. Esto permitirá a los nuevos desarrolladores y a los desarrolladores antiguos trabajar de manera eficiente en el sistema.

Se solicita implementar endpoints para verificar el estado de las bases de datos. Por ejemplo.

Endpoint de verificación de conexión a MariaDB:

URI: /check-mariadb-connection

Endpoint de verificación de conexión a MongoDB:

URI: /check-mongodb-connection

Recuerda que debes migrar el sistema existente a MariaDB y configurar MongoDB. Asegúrate de que los nuevos desarrolladores y los desarrolladores antiguos puedan acceder a las bases de datos y trabajar en conjunto de manera fluida.

Objetivos

* Diseñar un diagrama que permita entender a simple vista cómo funciona la arquitectura, agregando los puertos en los que escuchan, los endpoints y las imagenes que utilizan. (20 Puntos).
* Desarrollar e implementar los archivos y paquetes necesarios para dejar funcionando el entorno  (50 Puntos).

Una vez hecho toda la parte práctica se debe subir todo a un repositorio en github con las siguientes caracteristicas:

**Título repositorio:** *“SEGUNDO\_PARCIAL”*

**Commit**: *“NOMBRE\_APELLIDO\_ALUMNO-FECHA\_HORA”* .Ejemplo: ***“juan\_perez-21-06-2023:12:00”***

Cabe aclarar que solamente se permite un *commit* en el repositorio, se debe respetar el formato anterior, en caso de no respetar se le restaran puntos.

DESARROLLO:

1)UNA IMAGEN ES EL REFLEJO DE ALGO, ALGO SIMULADO FICTICIO ENCAMBIO UN CONTENEDOR ES AQUEL LUGAR DONDE SE CONTIENE ESE ALGO.

2) VENTAJAS DE INFRAESTRUCTURA COMO CODIGO SON:

ESCALABILIDAD,

ALTA DISPONIBILIDAD

PUEDO TENER VARIOS CONTENEDORES Y SI ALGUNO FALLA SE REDIRECIONA AUTOMATICAMENTE A OTRO QUE FUNCIONE PERFECTAMENTE.

GANO VELOCIDAD AL UTILIZAR COMANDOS.

EN EL CASO DE CONTENEDORES – SERVIDORES Y SI ALGUNO DE LOS CONTENEDORES FALLA REDIRECCIONA AUTOMATICAMENTE LA APLICACION O EL REQUERIMIENTO DEL USUARIO A OTRO QUE SE ENCUENTERE EN PERFECTO ESTADO SIN PROBOCAR CUELLAS DE BOTELLA.

BRIDA UNA INTERFAS AMIGABLE AL USUARIO.

PERMITE CREAR MAQUINAS VIRTUALES Y GESTIONAR TODO MAS RAPIDAMNETE ATRAVES DE COMANDOS MÁS RAPIDAMENTE.

3) DOCKER ENGINE ES UN CONJUNTO DE TRABAJOS Y HERRAMIENTAS QUE FACILITAN Y DA VELOCIDAD EN LA EJECUCIÓN DE LOS, REALIZACION DE LAS APLICACIONES ES UNA RED DE CONTENEDORES QUE NO PIERDE TIEMPO SI ALGUNO FALLA LO REDIRECIONA AUTOMATICAMENTE A OTRO QUE FUNCIONE.

4) DOCKERFILE SIRVE PARA LEVANTAR LAS APLICACIONES HACER CONECCIONES A LA BD Y DECIR EN QUE PUERTO QUIERO QUE SE EJECUTE.

5) ORQUETACIÓN DE CONTENEDORES: SE REFIERE A LA INTERCIONECCION DE LOS DISTINTOS CONTENEDORES Y COMO ELLOS TRABAJAN QUE COMO DIJE SI UNO FALLA LO REDIRECCIONA AUTOMATICAMENTE PARA OTRO QUE FUNCIONE PERFECTAMENTE SIN PRODUCIR DEMORAS O LO QUE SE LLAMA CUELLO DE BOTELLA PARA LOS REQUERI MIENTOS, APLICACIONES DE USUARIOS PRESENTANDOLES UNA INTERFAS AMIGABLE.

6) EL DOCKER COMPOSE SIRVE PARA TRABAJAR CON SERVIDORES

DESARROLLO DEL PRACTICO :

DESCARGO IMAGEN

IMAGEN MARIA DB:

Texto

Descripción generada automáticamente

IMAGEN MONGODB

seTexto

Descripción generada automáticamente

