Instituto Politécnico Nacional

Escuela Superior de Computo

Sistemas Distribuidos

Desarrollo de un sistema utilizando un servicio web

Profesor: Pineda Guerrero Carlos

Grupo: 3CV14

Bautista García Hadad

Introducción

¿Qué es una API de REST?

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST).

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones.Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta).

GSON

GSON es un API en Java, desarrollada por Google, que se utiliza para convertir objetos Java a JSON (serialización) y JSON a objetos Java (deserialización).

Este librería estructura los JSON de la siguiente manera:

- **JsonElement:** Esta clase representa cualquier elemento del Json que puede ser de alguno de los siguientes 4 tipos:
    1. **JsonObject:** Esta clase representa un objeto en el Json; es decir, un conjunto de pares clave-valor donde las claves son strings y los valores son cualquier otro tipo de JsonElement.
    2. **JsonArray:** Esta clase representa un array en el Json. Un array es una lista de JsonElements cada uno de los cuales puede ser de un tipo diferente. Se trata de una lista ordenada, por lo que el orden en que se añaden los elementos se conserva.
    3. **JsonPrimitive:** Esta clase representa un tipo de dato primitivo u objetos de datos simples (String, Integer, Double, etc.).
    4. **JsonNull:** Representa un objeto a null.

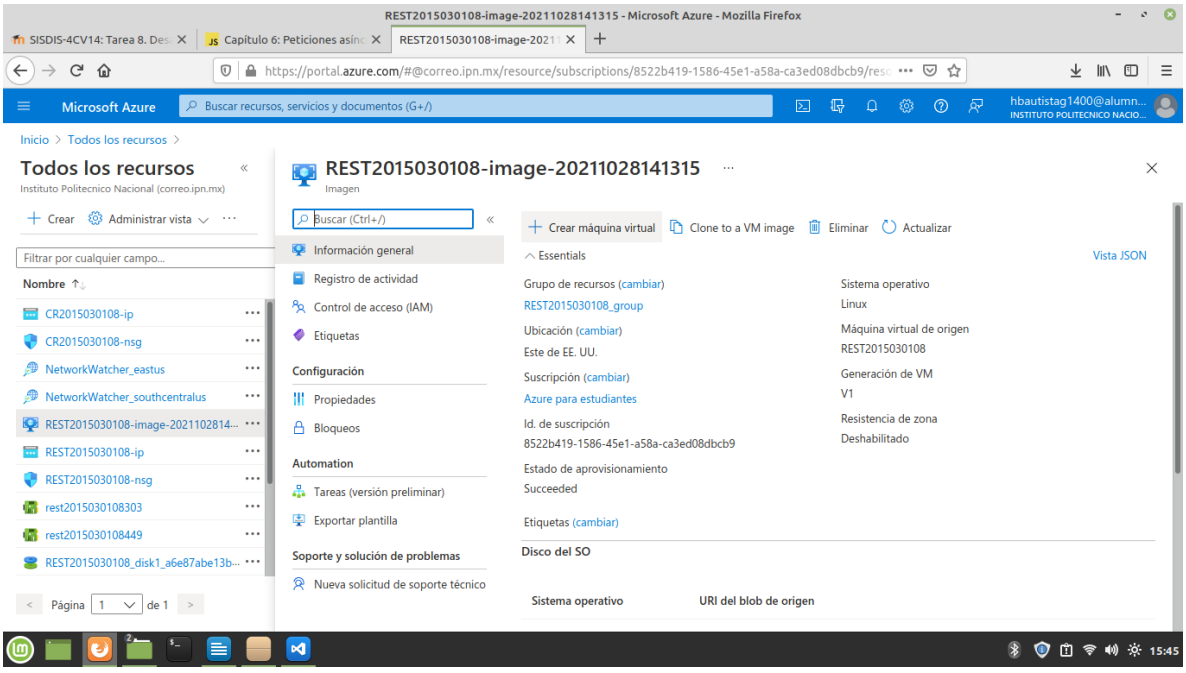
Peticiones asíncronas

Una petición asíncrona es una operación que, mientras esté siendo procesada, deja libre al navegador para que pueda hacer otras operaciones. Llamaremos peticiones asíncronas a las operaciones que tengan que ver con realizadas llamadas a servidores; sin embargo, existen muchas más operaciones asíncronas en JavaScript, como las que se realizan para leer y escribir en archivos, obtener la geolocalización de un navegador, o manejar base de datos.

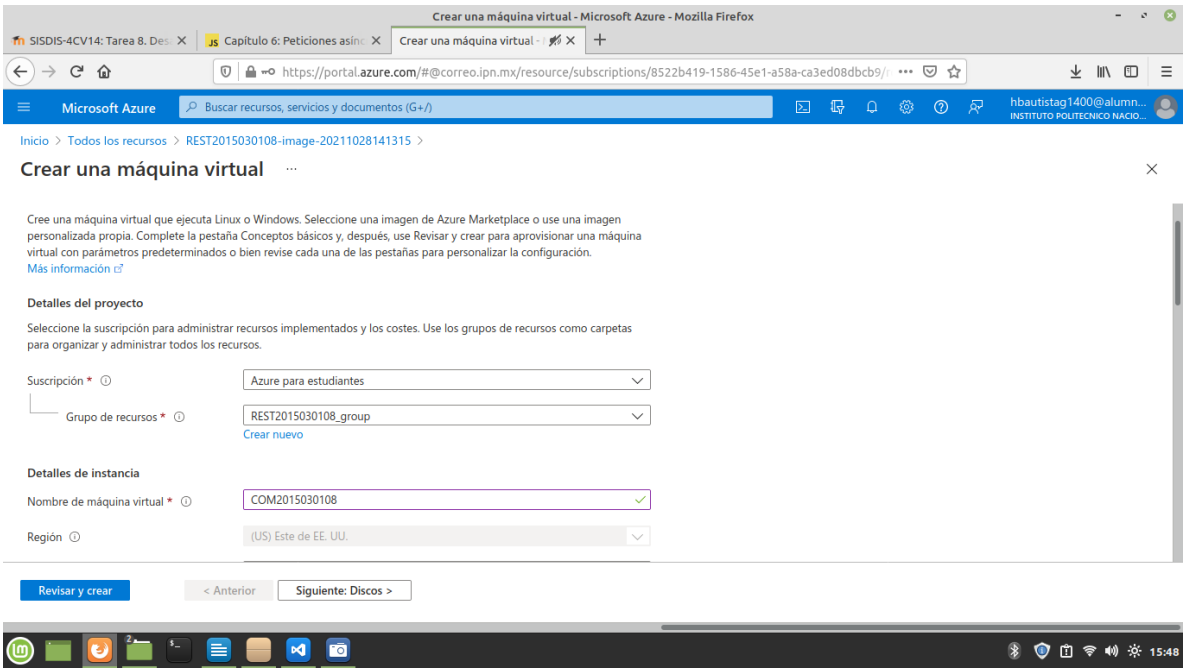
Desarrollo

Creación de la maquina virtual

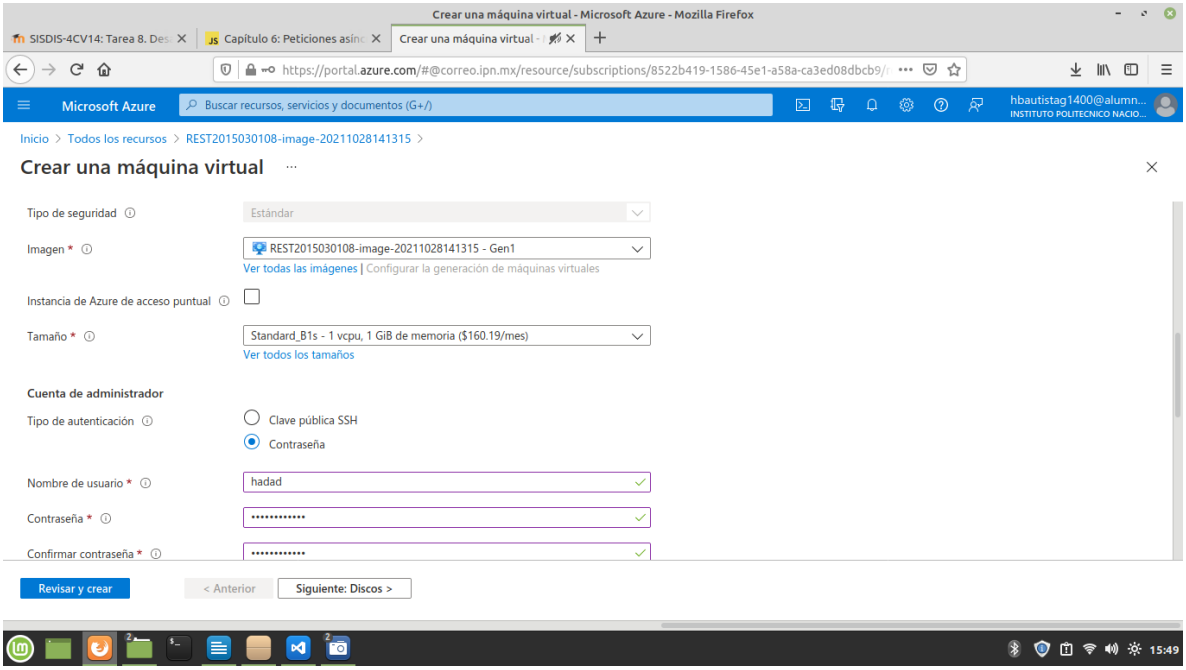
En esta ocacion crearemos la maquina virtual a partir de una imagen creada en una practica anterior , con esto nos ahorraremos un par de pasos para la ejecucion de esta practica en la captura podemos observar como seleccionamos la imagen para desde ahi empezar a crear la maquina virtual



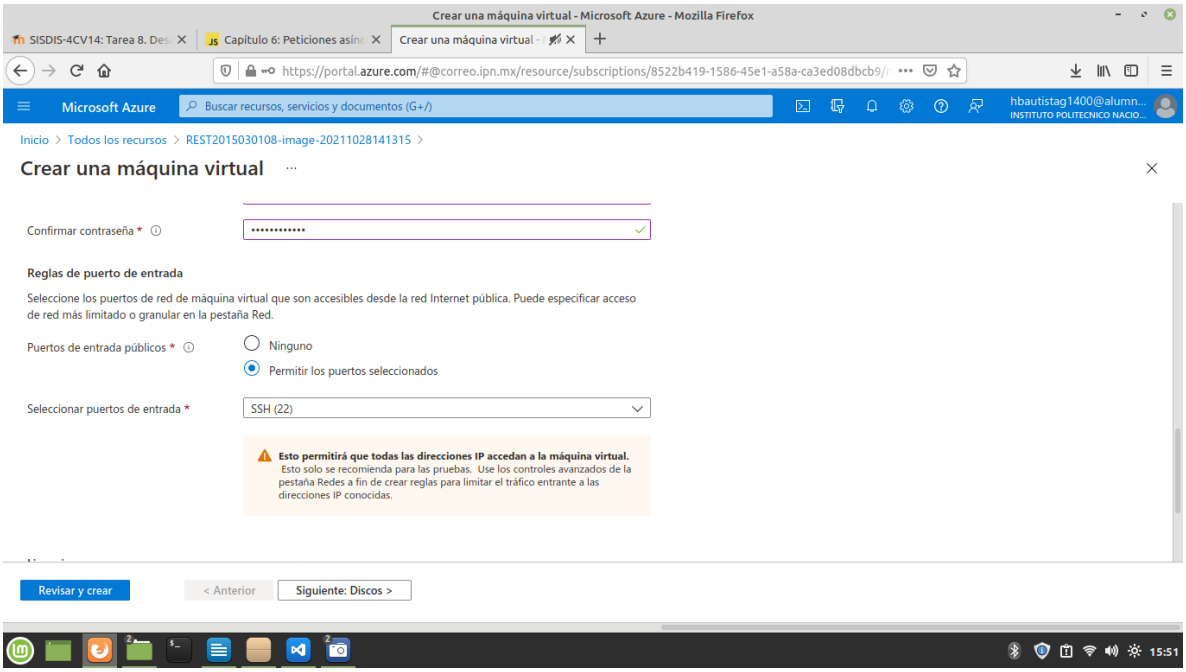
Una vez seleccionada la opción de crear maquina virtual se nos muestra un panel similar al panel que se nos muestra al crear una maquina virtual así que procedemos a crearla , inicialmente configuramos el campo de nombre de la maquina virtual



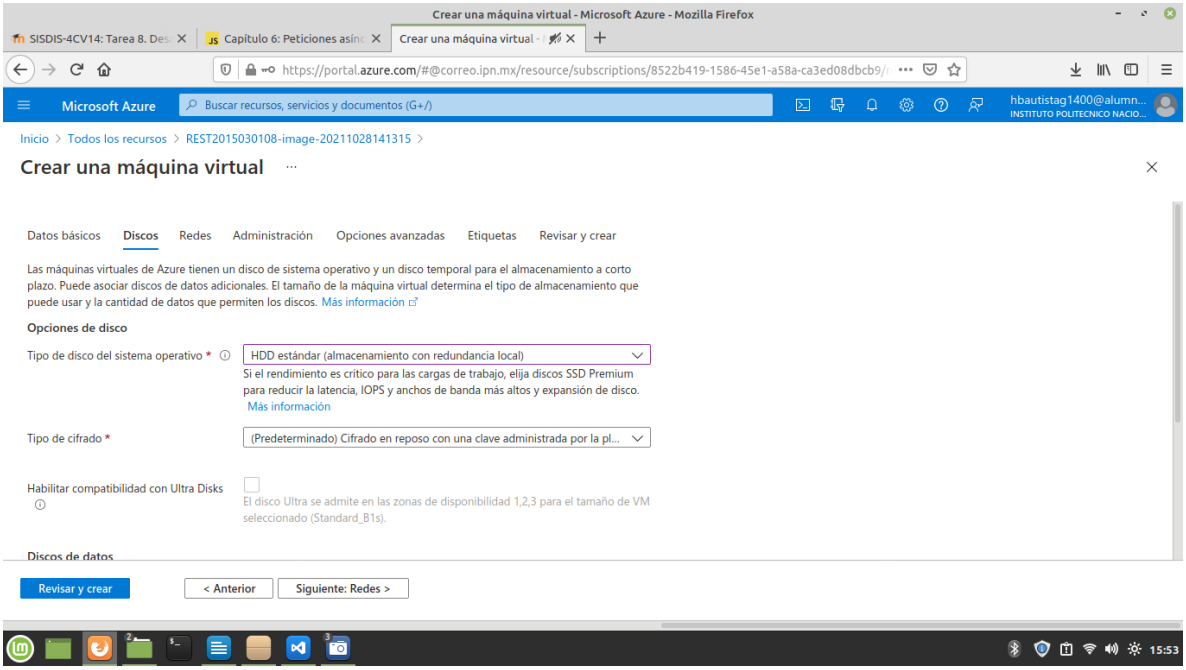
Después seleccionamos el tamaño de nuestra maquina así como el tipo de autenticación , en este caso se realizara mediante contraseña.



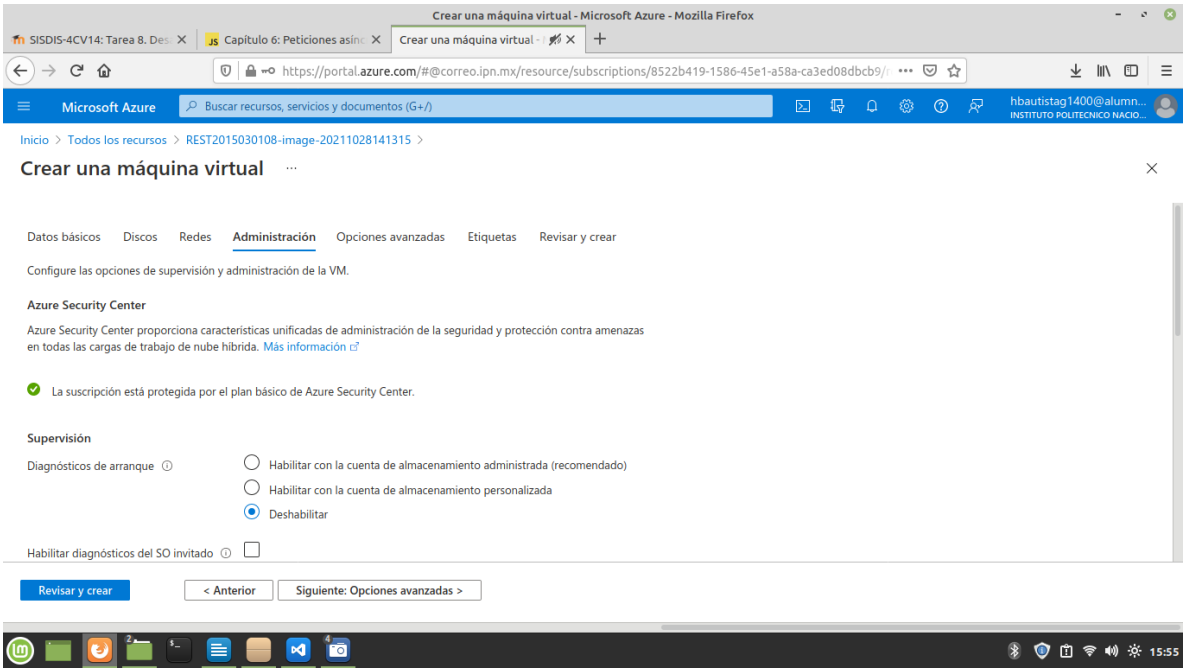
Para finalizar esta seccion selccionamos permitir los puertos seleccionados y elegimos el puerto de ssh y pasamos a la seccion de discos



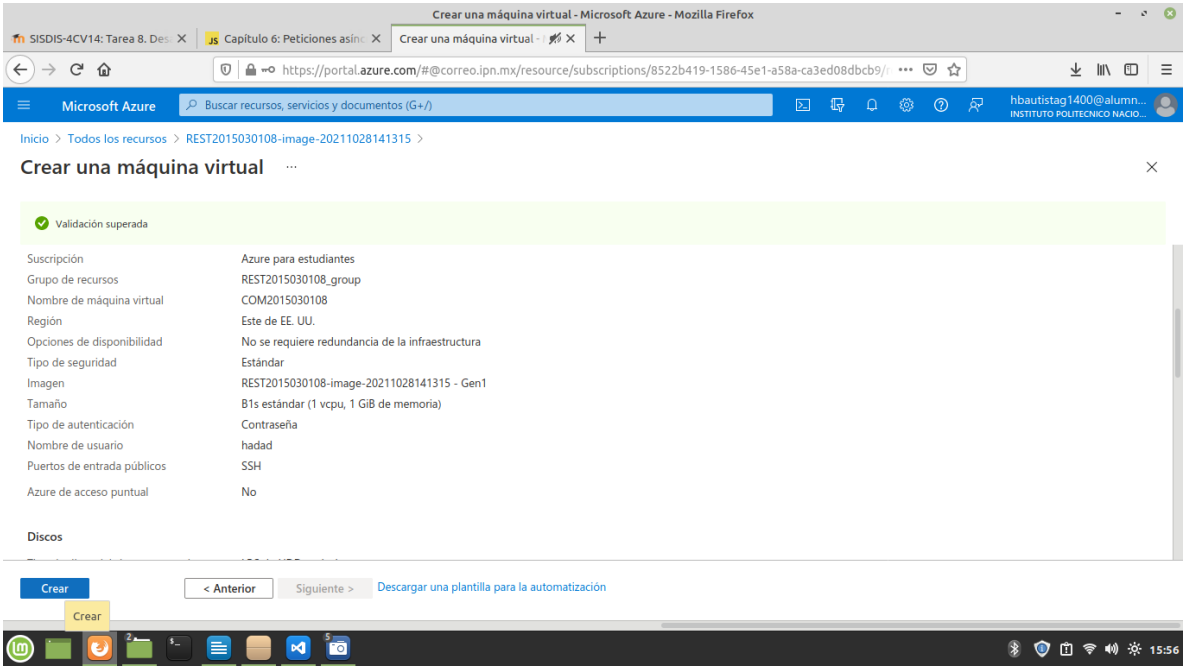
Seleccionamos el disco que utilizaremos para nuestra maquina virtual en este caso sera un disco HDD estandar



Posteriormente nos vamos a la pestaña de administración para deshabilitar el diagnostico de arranque damos click en revisar y crear.

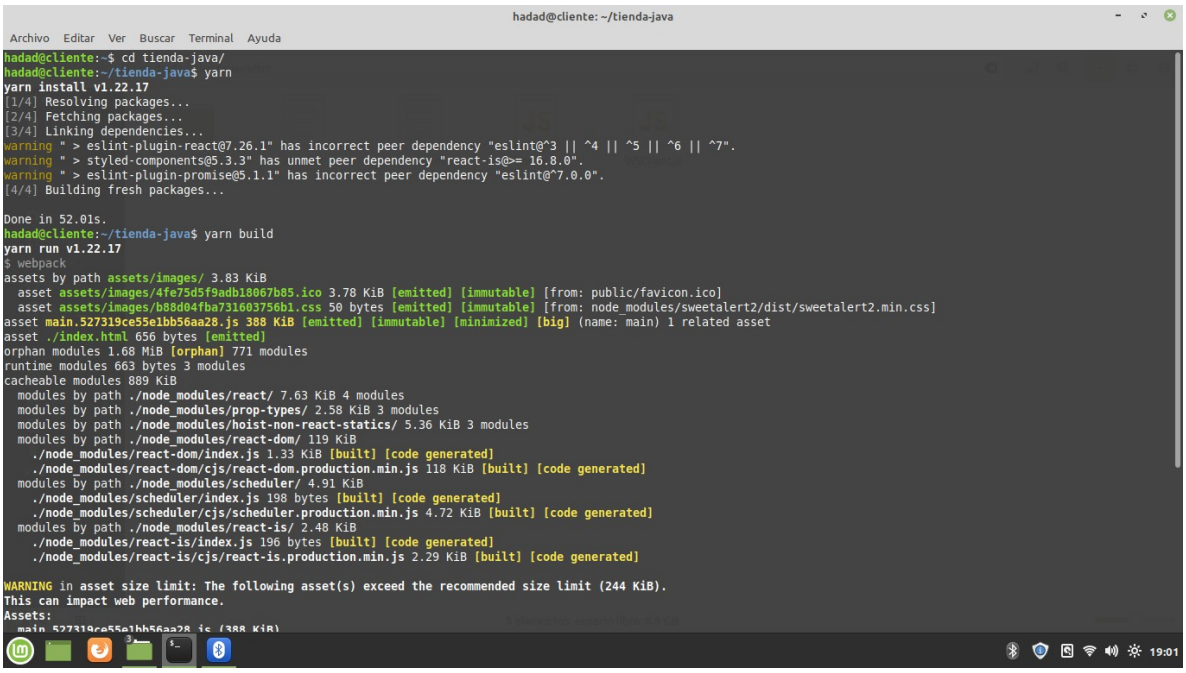


Finalmente revisamos los detalles de nuestra maquina a crear y le damos click en crear

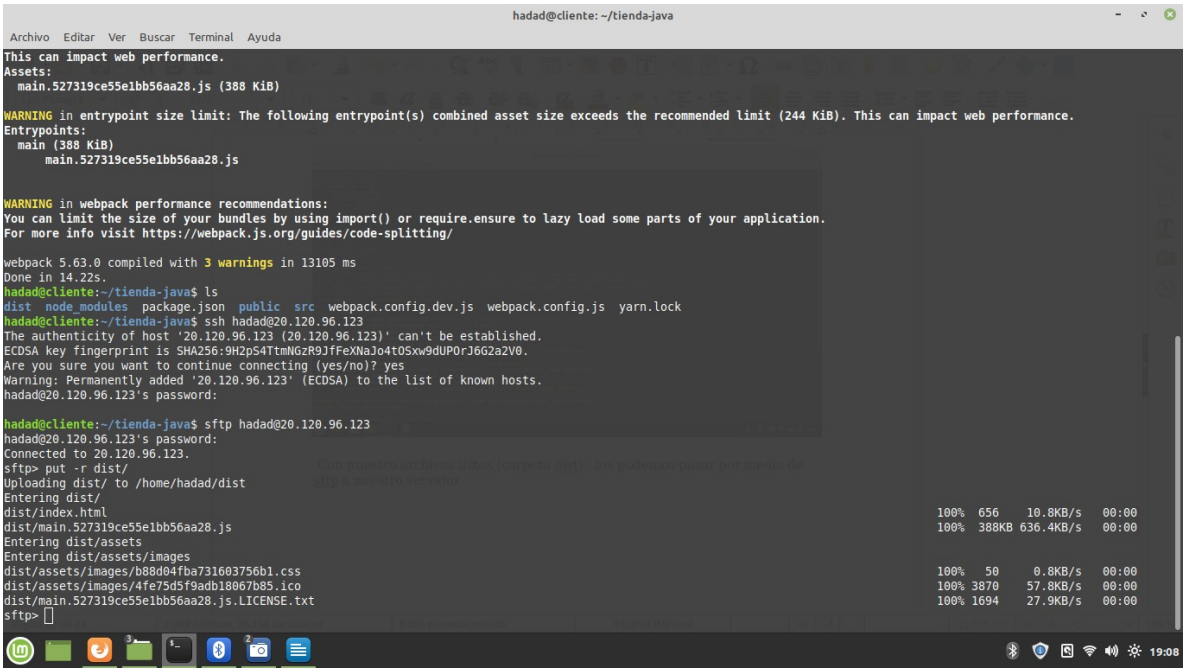


Compilación:

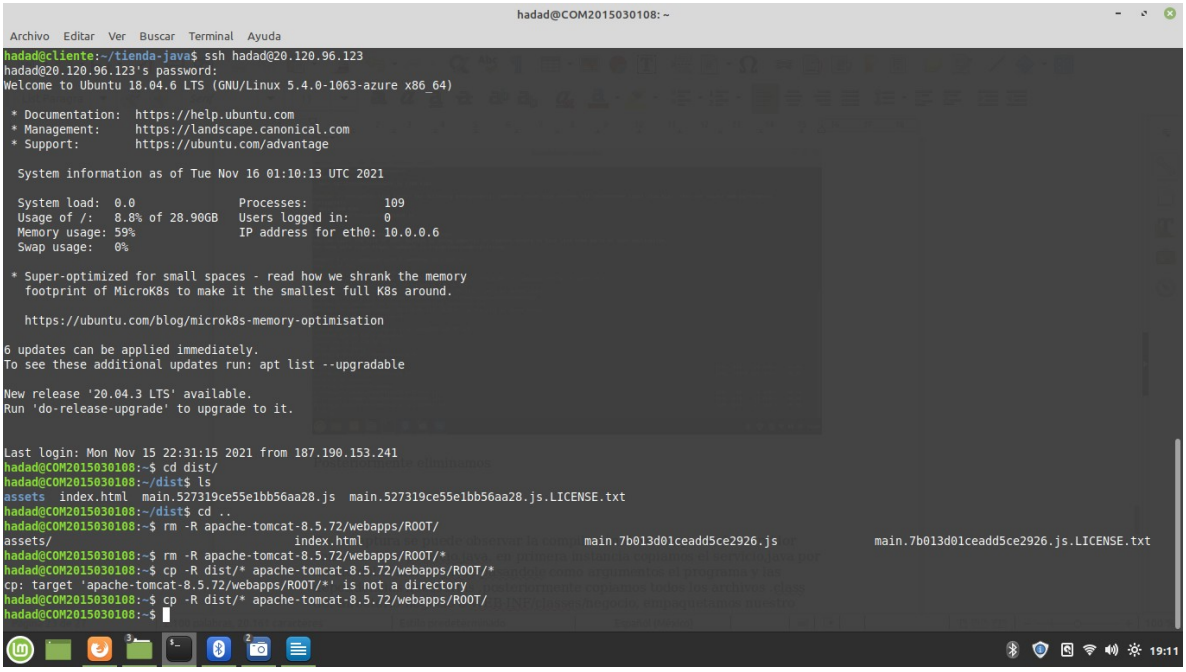
En primera instancia compilamos nuestro código fuente del front-end para generar nuestros archivos css ,js y html a partir de nuestro proyecto de react , como dependencias tenemos a node js que se puede instalar desde su pagina principal , yarn que puede ser instalado con npm install yarn -g y webpack el cual también puede ser instalado mediante npm install webpack -g , una vez con todas las dependencias podemos correr nuestro comando yarn build



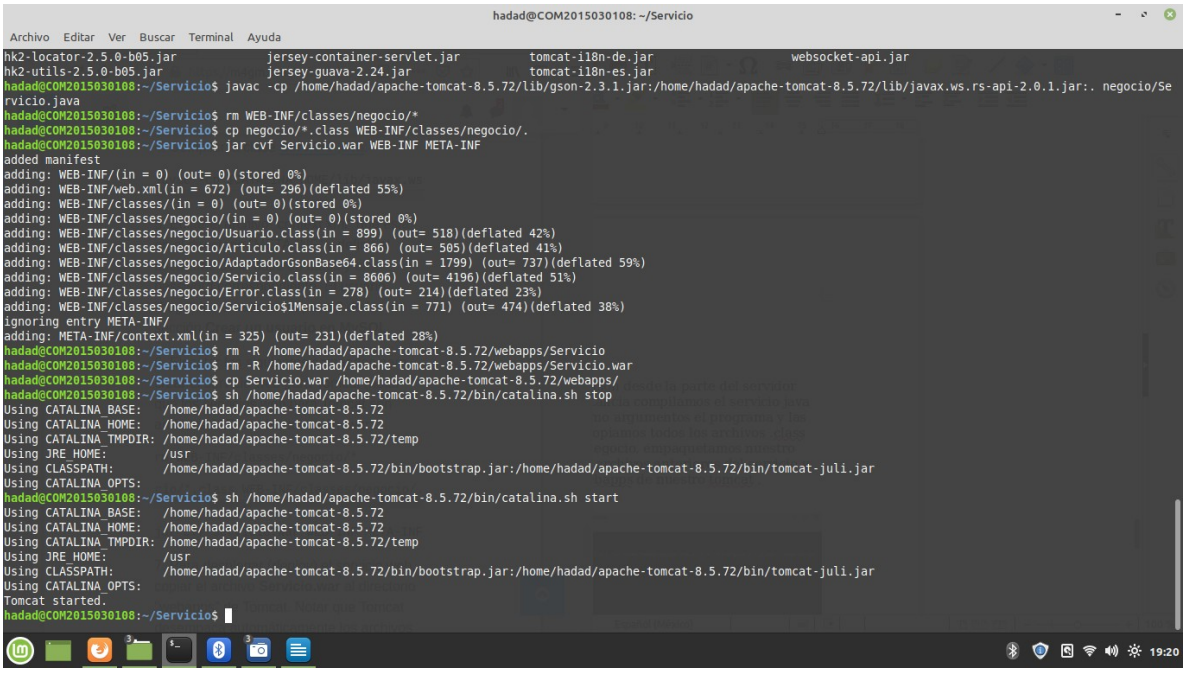
Con nuestro archivos listos (carpeta dist) , los podemos pasar por medio de sftp a nuestro servidor



Posteriormente eliminamos los archivos que tengamos en nuestro servidor de apache dentro de la carpeta webapps/ROOT y colocamo ahi nuestro archivos generados

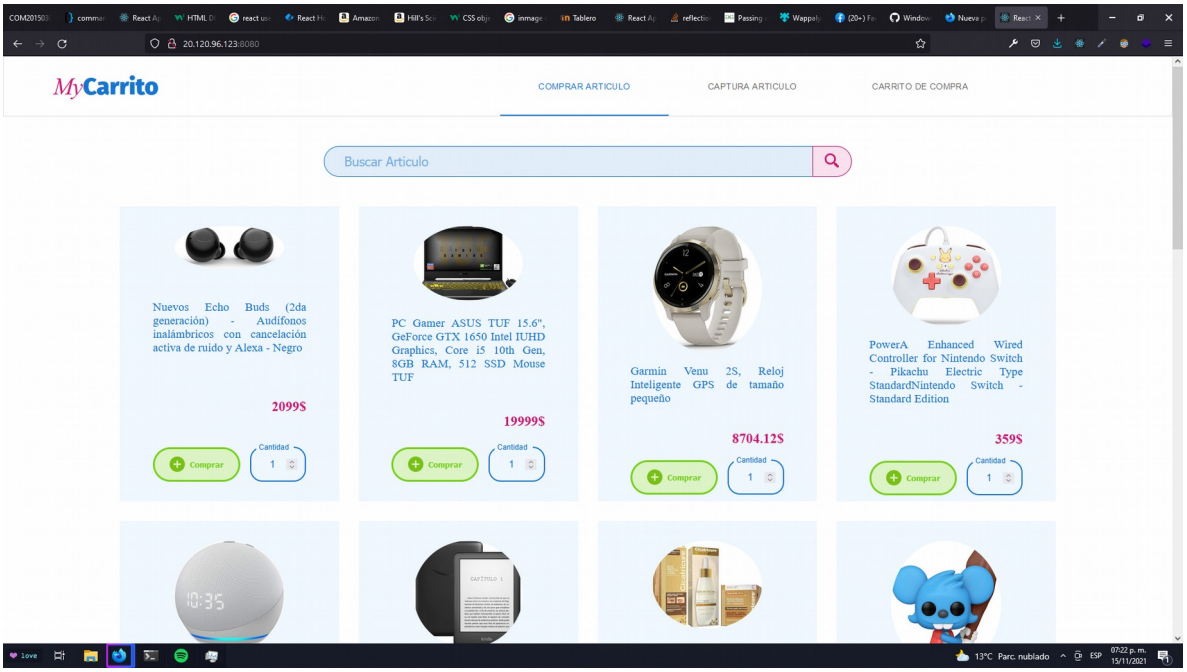


En la captura se puede observar la compilación desde la parte del servidor con el archivo servicio.java, en primera instancia compilamos el servicio java por medio del comando javac pasandole como argumentos el programa y las dependencias necesarias ,posteriormente copiamos todos los archivos .class del directorio negocio a WEB-INF/classes/negocio, empaquetamos nuestro servicio en un archivo .war, eliminamos los archivos anteriores del servicio y copiamos nuestro paquete en la carpeta webapps de nuestro tomcat , finalmente reiniciamos nuestro servidor

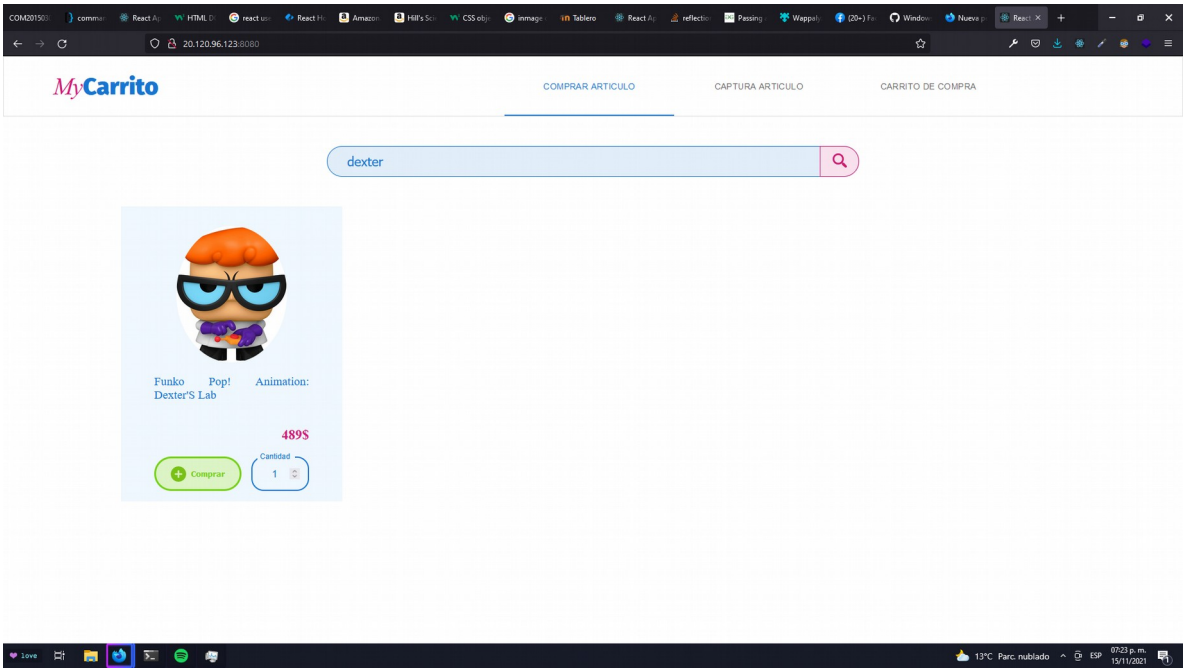


Ejecución:

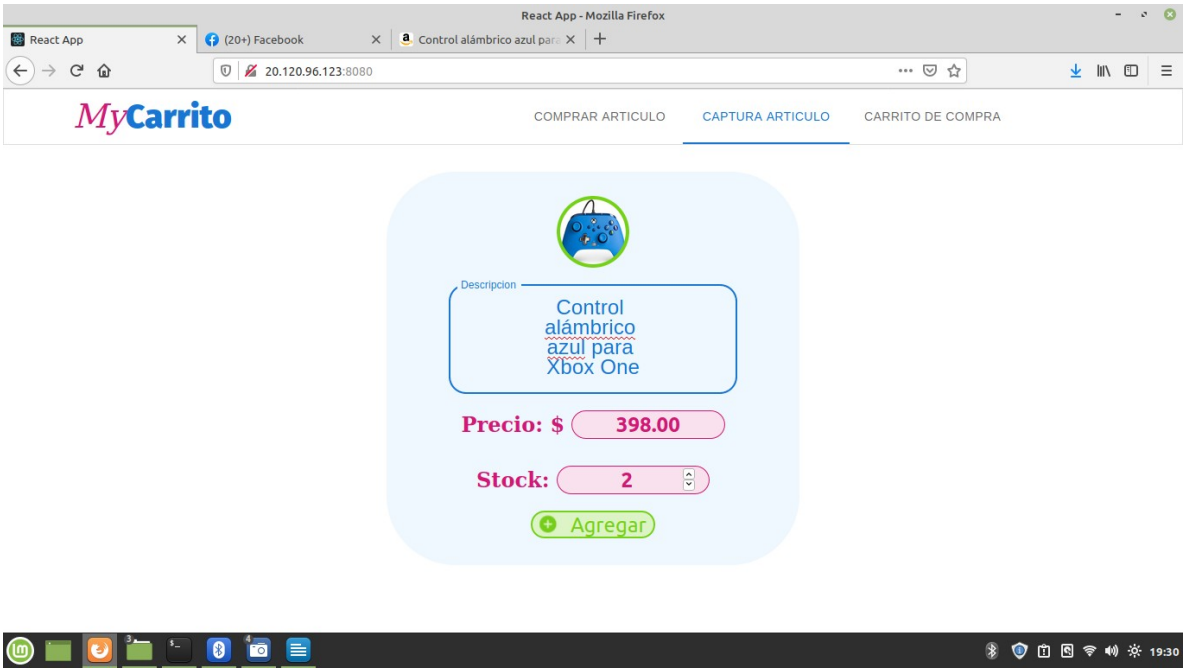
Para ver el comportamiento de nuestro programa debemos ingresar a la url formada por la ip de nuestro servidor junto con el puerto a utilizar en este caso el 8080



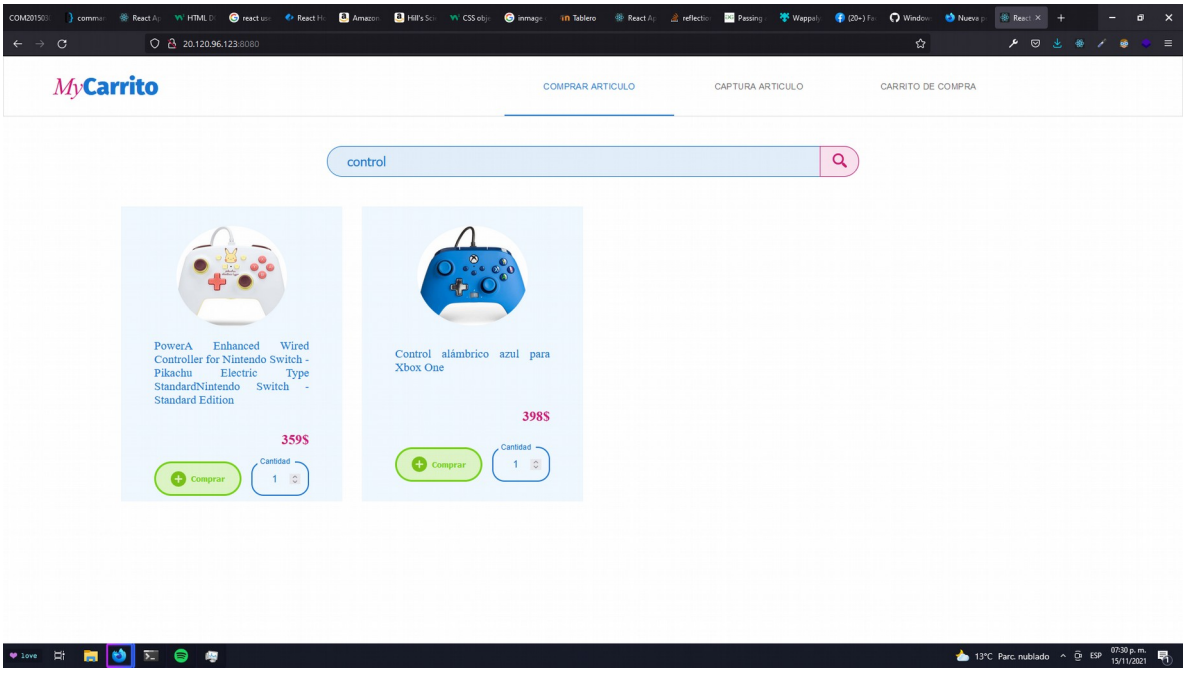
Para la primera prueba vamos a buscar un articulo que tenga la cadena “dexter”



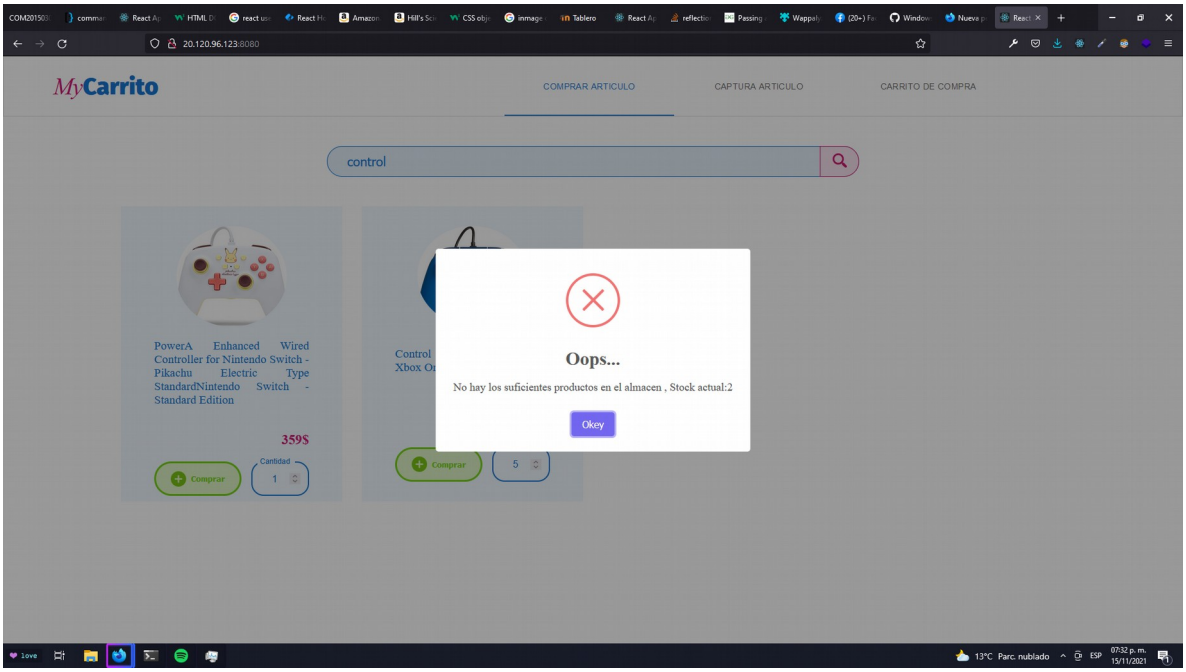
Posteriormente vamos añadir un articulo con su respectiva foto



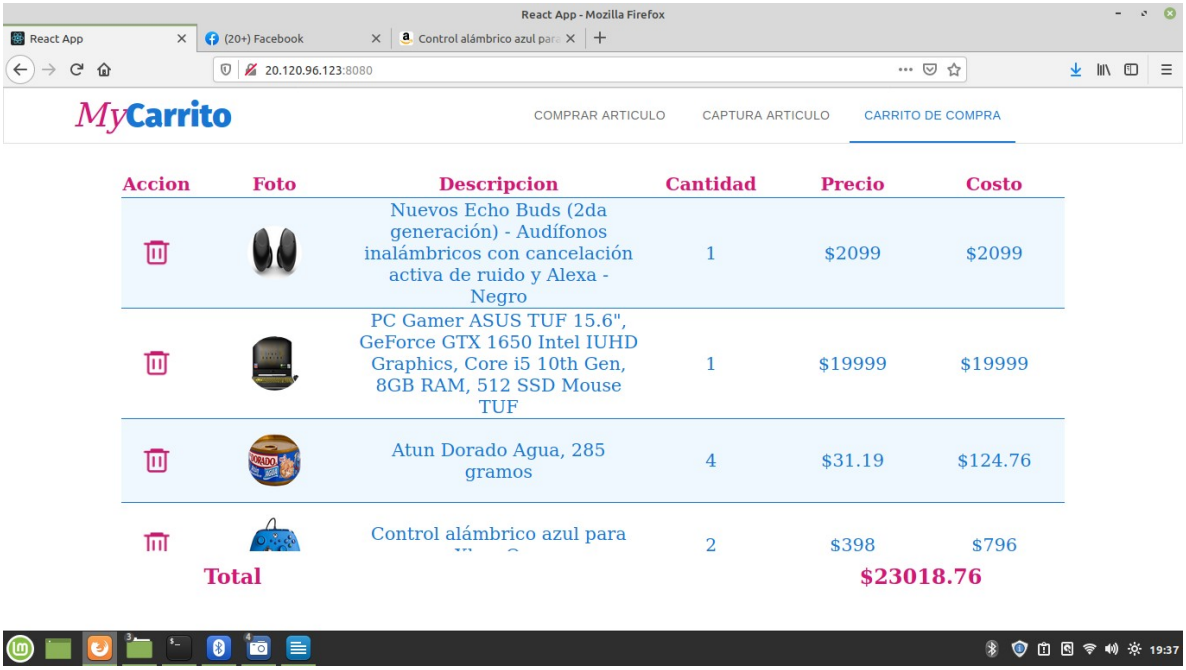
Posteriormente buscamos nuestro producto con la palabra clave control



Después intentaremos agregar 5 controles a nuestro carrito para hacer saltar el error



Agregaremos unos cuantos articulos a nuestro carrito para ver la siguiente demostracion, y poder ver nuestro carrito



Finalmente vamos a eliminar un objeto de nuestro carrito y podemos ver como el total se actualiza automáticamente

Conclusiones

Esta practica fue bastante entretenida de realizar ya que me permitió utilizar conocimientos adquiridos de otras tecnologías para poder implementarlo con un servidor de java rest el cual recibe y enviá peticiones , hubo varios inconvenientes al realizar la practica desde el planteamiento del diseño del front-end y la estructura de la base de datos pero debido a que se dio un tiempo suficiente para concluir la practica por lo que se pudo realizar con éxito la practica