**DOCUMENTO DE INSTALACIÓN DEL CHALLENGE DE EVERTEC**

### Por: Oscar Bohórquez

### Fecha: **19**-**Mayo**-202**2**

**Objetivo:**

El presente documento cuenta con las suficientes instrucciones para ejecutar el challenge de forma exitosa y fácil.

**Prerequisitos:**

1. git cliente o clonar el proyecto desde un navegador
2. Docker instalado de la pagina oficial “<https://docs.docker.com/get-docker>”, instalar la ultima versión de Docker en su Sistema Operativo.

**Instalación:**

Repositorio con el challenge se encuentra en el siguiente link de github: https://github.com/oscbohr/ddd-springboot-react-postgres.git

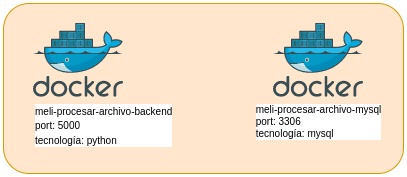
1. Clonar el repositorio dentro del directorio deseado y entrar al directorio descargado “mvc-python-rest-mysql”, revisar que se tengan la siguiente estructura de directorios y archivos:



comando: git clone https://github.com/oscbohr/ddd-springboot-react-postgres.git

comando: cd ddd-springboot-react-postgres

1. Crear la imagen y ejecutar los contenedores con docker-compose:



NOTA: la aplicación python está ejecutándose por el puerto: 5000 y la base de datos mysql en el puerto 3306; tener en cuenta que si la máquina donde se está ejecutando los contenedores ya tienen esos puertos en uso, debemos cambiarlos en el archivo Dokerfile de cada contenedor al igual que en el archivo “docker-compose.yml”

DockerFile: python

|  |
| --- |
| # set base image (host OS)  FROM python:3.8.10  # set the working directory in the container  WORKDIR /app  COPY /meli-procesar-archivo/codigo/ /app/  # copy the dependencies file to the working directory  COPY /meli-procesar-archivo/codigo/requirements.txt .  # install dependencies  RUN pip install -r requirements.txt  # copy the all content of directory to the working directory  #COPY . .  RUN ls /app  ENV FLASK\_ENV=development  CMD ["python3", "app.py"] |

docker-compose.yml

|  |
| --- |
| version: '3'  services:  backend:  build:  context: .  dockerfile: ./meli-procesar-archivo/codigo/Dockerfile  container\_name: meli-procesar-archivo-backend  hostname: backend  network\_mode: bridge  restart: always  expose:  - "5000"  ports:  - "5000:5000"  depends\_on:  - "db"  links:  - "db"  db:  image: mysql:8.0.29  network\_mode: bridge  hostname: db  container\_name: meli-procesar-archivo-mysql  restart: always  expose:  - "3306"  ports:  - "3306:3306"  volumes:  - ./db/init.sql:/docker-entrypoint-initdb.d/init.sql:ro  environment:  - MYSQL\_ROOT\_PASSWORD=my-secret-pw  - MYSQL\_DATABASE=testDB |

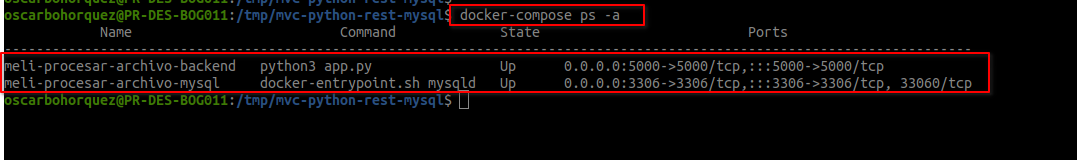
Ejecutar el comando: “**docker-compose up -d --build**”

|  |
| --- |
|  |
|  |

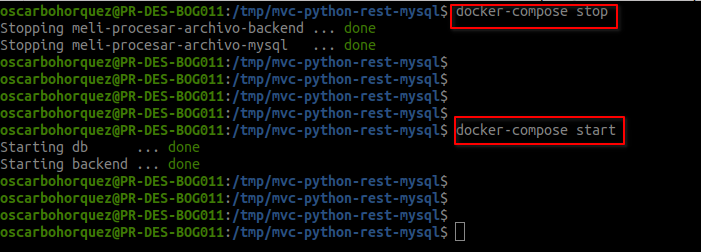
1. Verificar la correcta ejecución del contenedor de la aplicación frontend, aplicación backend y de la Base de Datos ejecutando:

comando: docker-compose ps -a

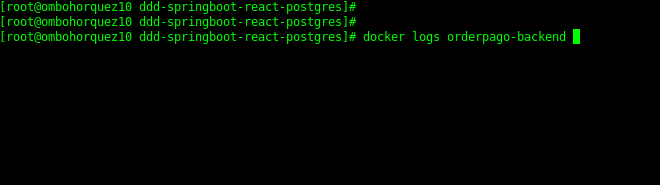
resultado:



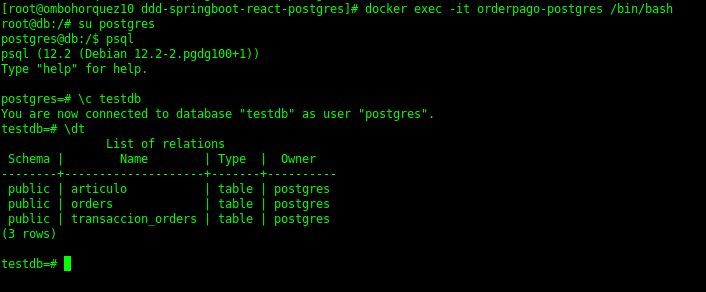
1. Detener e iniciar los contenedores para que tomen el nombre de host configurado en el archivo docker-componse.yml. Para ello ejecute: “docker-compose stop”, y seguido “docker-compose start”



1. Verificar los logs de los contenedores ejecutando los siguientes comandos:
   1. “docker logs -f meli-procesar-archivo-backend”
   2. “docker logs -f meli-procesar-archivo-mysql”

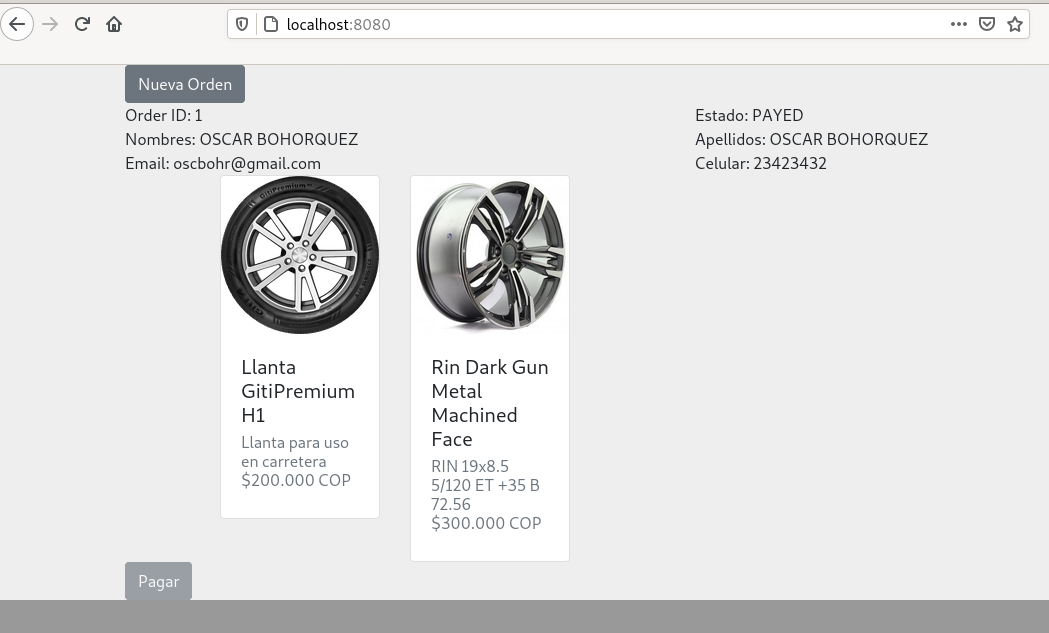


1. Ingresar al contenedor de la Base de Datos y verificar la estructura de tablas. Para ello ingrese los siguientes comandos:
   1. “docker exec -it orderpago-postgres /bin/bash”
   2. “su postgres”
   3. “psql”
   4. “\c testdb”
   5. “\dt”

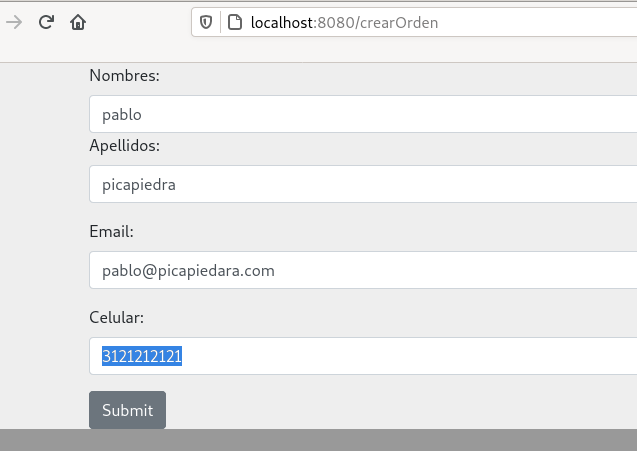


NOTA: Si la estructura de tablas no se encuentra, ejecutar sobre la misma terminal el script “init.sql” localizado en el directorio “db” del repositorio descargado.

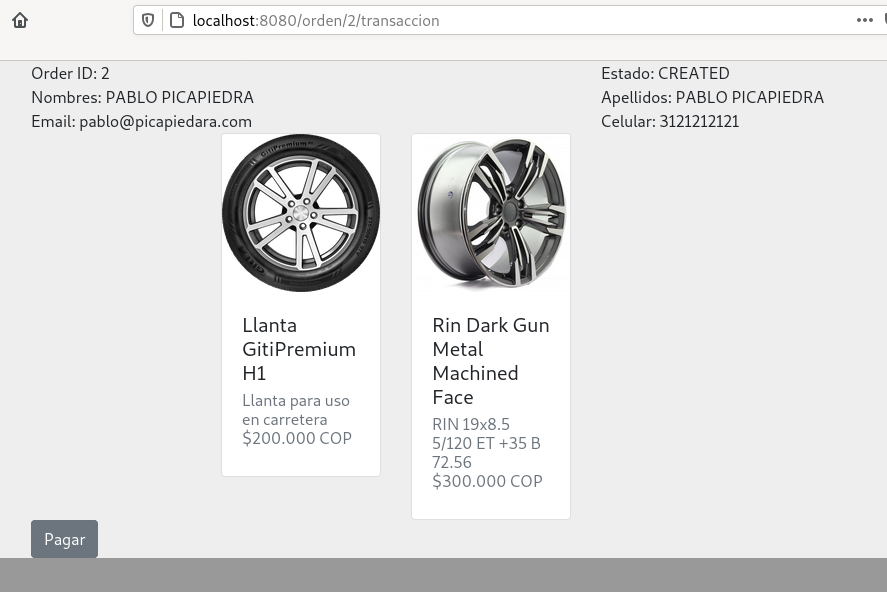
1. Ejecutar la aplicación: Abrir el browser de su preferencia (Firefox o Chrome) e ingresar la siguiente url “<http://localhost:8080/>”. Se visualizaran las ordenes emitidas por la tienda (o en blanco). Para crear una nueva Orden, oprimir el botón “Nueva Order”



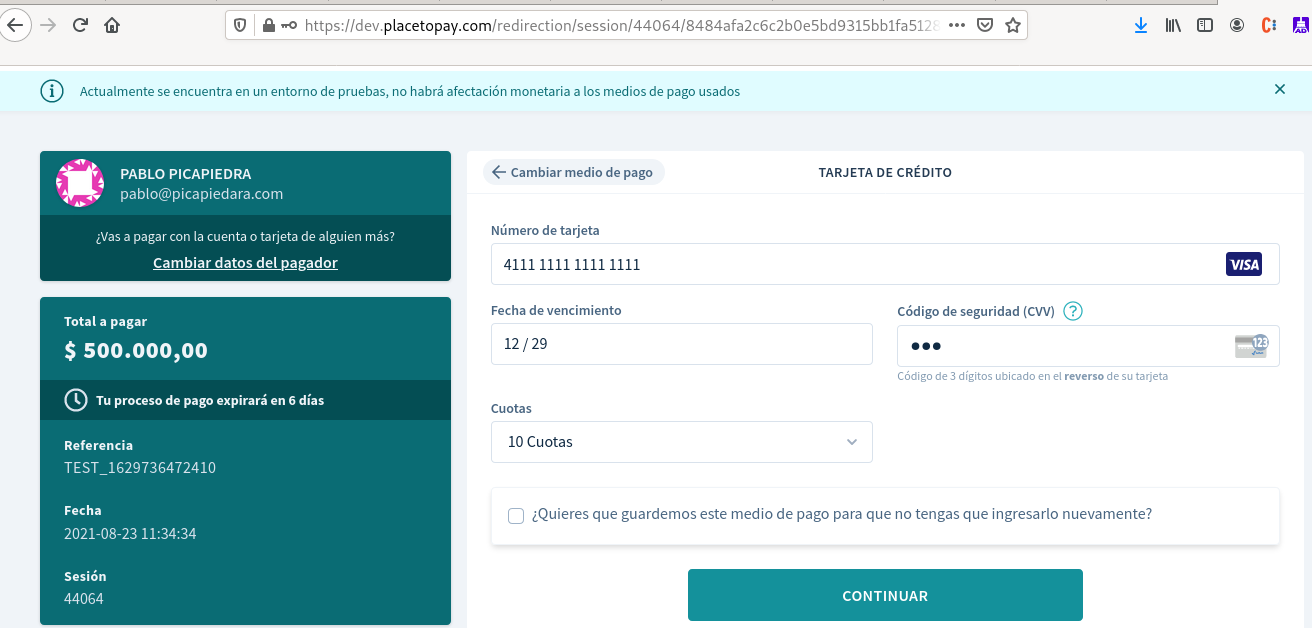
Ingrese la información del cliente y oprima el botón “submit”

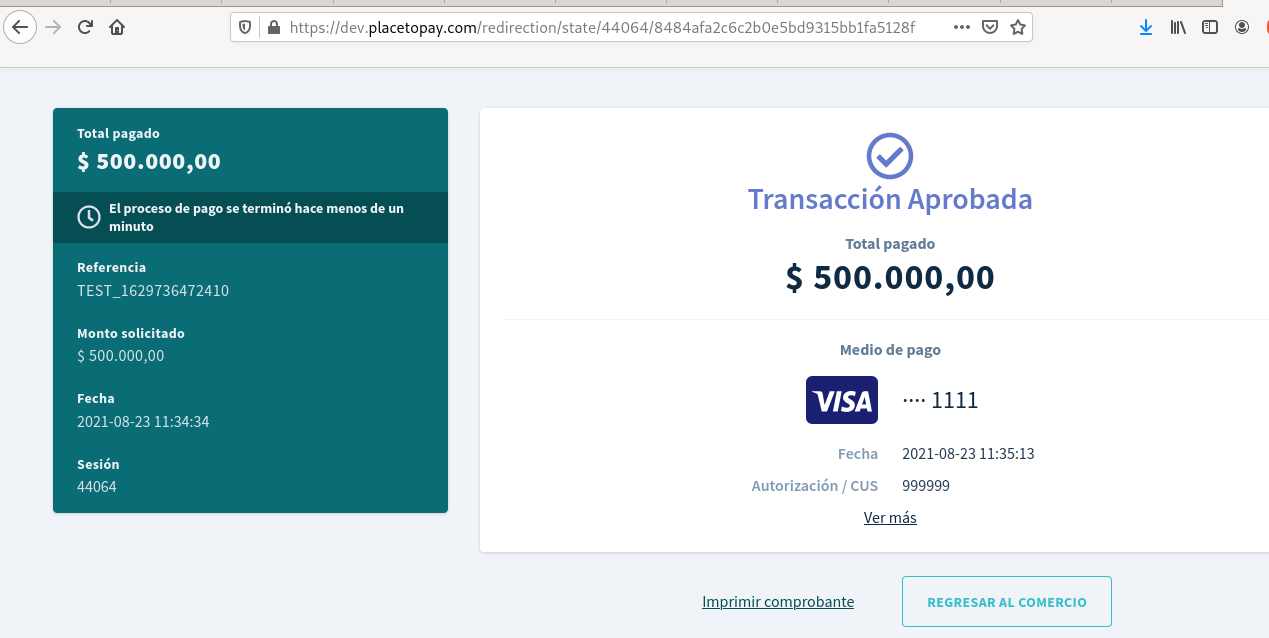


Se visualizará la Orden con su información y estado con la opción de “Pagar”



Al oprimir “Pagar” se hace el llamado a la pasarela de Pago





Al oprimir el botón “Regresar al comercio” se devuelve al Detalle de la Orden con la información del pago: Si el pago es satisfactorio se deshabilita el botón “Pagar” de lo contrario se habilita para posteriormente ejecutar el pago.

