

**PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA  
FACULTAD DE CIENCIAS DE LA INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y  
COMPUTACIÓN**



**“Tarea #6: Implementar software PaaS”**

**Presentado por:**

Diana Monegro – 10138847

**Presentado a:**

Carlos Alfredo Camacho

**Tarea #6 de la materia Programación Web**

**Santiago de los Caballeros**

**16 de Agosto, 2022**

## **Introducción**

El objetivo de esta tarea es poner en práctica los conocimientos adquiridos acerca de manejar Heroku y la base de datos que este implementa, la cual es PostgreSQL. Para ello utilizaremos la aplicación previamente hecha en la practica 4

Los objetivos de este proyecto son los siguientes:

- Implementar un proyecto de Java en una plataforma orientada a servicios – PaaS.
- Utilizar la plataforma de Heroku.
- Utilizar PostgreSQL como gestor de base de datos.

## **Desarrollo**

Para realizar esta práctica estarán utilizando su Práctica #3 y 4 Combinada, llamada Carro de Compra con ORM, en la cual tendrán que publicar la aplicación realizada en la plataforma como servicio (PaaS) de Heroku.

Para lograr el objetivo deben realizar las siguientes modificaciones:

- Utilizar como base de datos Postgresql bajo el servicio de Heroku. Ver documentación aquí.
- Cambiar o incluir el driver y configuración utilizada de H2 a PostgreSQL.
- Implementar método para pasar configuración de Heroku a JPA. Ver proyecto demo de ORM-JPA.

## **Respuestas y explicaciones:**

### **Heroku:**

Heroku es uno de los PaaS más utilizados en la actualidad en entornos empresariales por su fuerte enfoque en resolver el despliegue de una aplicación. En otras palabras, a Heroku solo le dices qué lenguaje de backend estás utilizando o qué base de datos vas a utilizar y te preocupas únicamente por el desarrollo de tu aplicación.

### **Características de Heroku**

- Soporta diferentes lenguajes de programación: Node, Ruby, Java, Clojure, Scala, Go, Python, PHP
- Tiene una versión gratuita fácil de usar
- Ejecuta las aplicaciones a través de sus contenedores, también conocidos como Dynos
- Tiene Dynos que pueden ser de tres tipos: web, worker o cron
- Ofrece más de 200 complementos con los que ampliar las aplicaciones al instante
- Ofrece varias características de seguridad, incluyendo SSL, autenticación y cumplimiento de PCI

### **PostgreSQL:**

PostgreSQL, es un sistema de código abierto de administración de bases de datos del tipo relacional, aunque también es posible ejecutar consultas que sean no relaciones. En este sistema, las consultas relacionales se basan en SQL, mientras que las no relacionales hacen uso de JSON.

Se trata de un sistema de código abierto y además gratuito, y su desarrollo es llevado adelante por una gran comunidad de colaboradores de todo el mundo que día a día ponen su granito de arena para hacer de este sistema una de las opciones más sólidas a nivel de bases de datos.

La documentación sobre como conectar nuestra aplicación en Heroku con PostgreSQL se encuentra en la siguiente página: <https://devcenter.heroku.com/articles/heroku-postgresql#local-setup>.

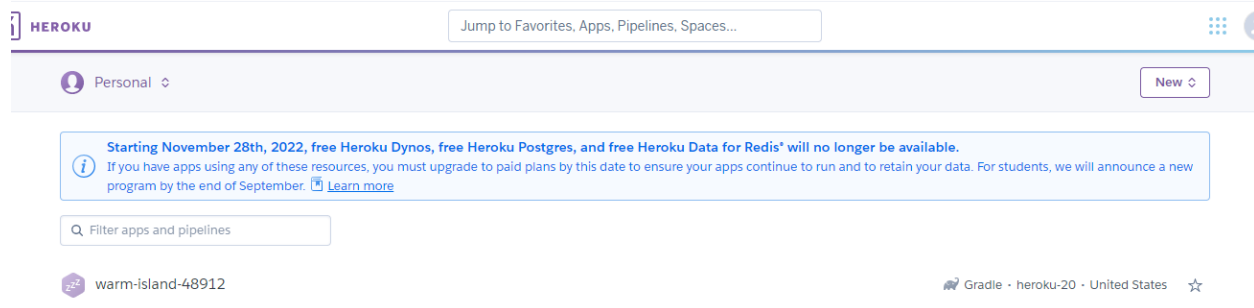
Asimismo, en el archivo *persistence.xml* es necesario declarar un nuevo *persistence-unit* con los datos de la base de dato que implementara Heroku:

```
<persistence-unit name="Heroku" transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
  <class>Entities.User</class>
  <class>Entities.Photo</class>
  <class>Entities.Product</class>
  <class>Entities.Cart</class>
  <class>Entities.CartProduct</class>
  <class>Entities.Review</class>
  <properties>
    <property name="hibernate.archive.autodetection" value="class"/>
    <property name="jakarta.persistence.jdbc.driver" value="org.postgresql.Driver" />
    <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQL90Dialect"/>
    <property name="hibernate.hbm2ddl.auto" value="update" />
    <property name="hibernate.show_sql" value="true"/>
    <!-- Indica como será la generación de las tablas en el motor-->
    <!-- ver la documentación: https://docs.oracle.com/javaee/7/tutorial/persistence-intro005.htm -->
    <property name="jakarta.persistence.schema-generation.database.action" value="drop-and-create"/>
    <!--<property name="eclipselink.logging.level" value="ALL" />-->
  </properties>
</persistence-unit>
```

Para subir la aplicación a Heroku, se utilizan los siguientes comandos principales:

- *Heroku login*
- *Heroku create*
- *Git push heroku master*

A utilizarlo, heroku crea una aplicación que se puede visualizar con el dashboard, con un nombre asignado al azar. En el caso de la aplicación para esta práctica, el nombre fue: *warm-island-48912*.



En el proyecto, es necesaria agregar la siguiente dependencia para utilizar postgresSQL:

```
implementation 'org.postgresql:postgresql:42.3.6'
```

Al acceder a la aplicación desde el dashboard de Heroku, podemos ver una vista general de la misma:

Personal > warm-island-48912 ☆ Open app More

Overview Resources Deploy Metrics Activity Access Settings

Installed add-ons **\$0.00/month** [Configure Add-ons](#)

There are no add-ons for this app  
You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation **\$0.00/month** [Configure Dynos](#)

This app is using **free** dynos

```
web java -jar build/libs/app.jar Heroku
```

**ON**

Collaborator activity [Manage Access](#)

Latest activity [All Activity](#)

- dxmd0001@ce.pucmm.edu.do: Deployed `a7c4b5c9`  
Jul 20 at 3:39 PM · v3
- dxmd0001@ce.pucmm.edu.do: Build succeeded  
Jul 20 at 3:38 PM · [View build log](#)
- dxmd0001@ce.pucmm.edu.do: Enable Logplex  
Jul 20 at 3:35 PM · v2
- dxmd0001@ce.pucmm.edu.do: Initial release  
Jul 20 at 3:35 PM · v1

Al correr la aplicación, podemos ver como la misma funciona correctamente implementando la base de datos de PostgreSQL y el link con el nombre asignado por Heroku.

warm-island-48912.herokuapp.com

Bienvenido/a

MENU

- Inicio

TIENDA

Carrito 0

## Productos Disponibles

Productos Disponibles

10 entries per page

Foto	Id	Producto	Precio	Cantidad a Agregar	Agregar	Descripcion
	1	Agua Opcion Dos	141.0189	<input type="text" value="1"/>	<button>Agregar al carrito</button>	<button>Ver Producto</button>
	2	Maiz Opcion Dos	264.49356	<input type="text" value="1"/>	<button>Agregar al carrito</button>	<button>Ver Producto</button>

## **Conclusión**

En esta práctica se lograron los objetivos especificados: se ejecutó la aplicación a través de Heroku, utilizando la base de datos PostgreSQL.

## Bibliografía

- *Github del proyecto:* <https://github.com/DianaMD21/programacion-web-dianamd-10138847.git>.
- *Documentación de utilizar Heroku en un app de Gradle:*  
<https://devcenter.heroku.com/articles/deploying-gradle-apps-on-heroku>
- *Documentación sobre cómo implementar PostgreSQL bajo el servicio de Heroku:*  
<https://devcenter.heroku.com/articles/heroku-postgresql#set-up-postgres-on-windows>