

PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA

Departamento de Ciencias e Ingeniería.

Escuela de Ciencias En Computación y Telecomunicaciones



PROGRAMACION WEB

“Práctica 6”

Presentado por: Nicol Ureña (2018-1669)

Repositorio Documentos: <https://github.com/nicoluv/ProgWeb/tree/main/Documents>

Repositorio Principal: <https://github.com/nicoluv/ProgWeb>

Repositorio Práctica: <https://github.com/nicoluv/tienda>

Heroku: <https://gentle-mesa-17800.herokuapp.com/comprar>

Entregado a: Carlos Camacho

Fecha de entrega: julio del 2022

SANTIAGO DE LOS CABALLEROS, REPÚBLICA DOMINICANA

Introducción

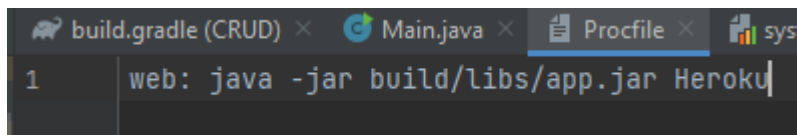
En el campo de la ciencia de la computación, la unión de conocimientos prácticos y teóricos no es sólo útil, si no también necesario. Debido a ello, tener claras las nociones relativas a seguridad, integración de herramientas, bases de datos, librerías, y servicios es un requisito para la realización de un buen sistema de computación. A continuación, se presentará una práctica en la cual se implementará un proyecto de Java en una plataforma orientada a servicios (PaaS), utilizando la plataforma Heroku y el gestor de datos PostgreSQL.

Desarrollo

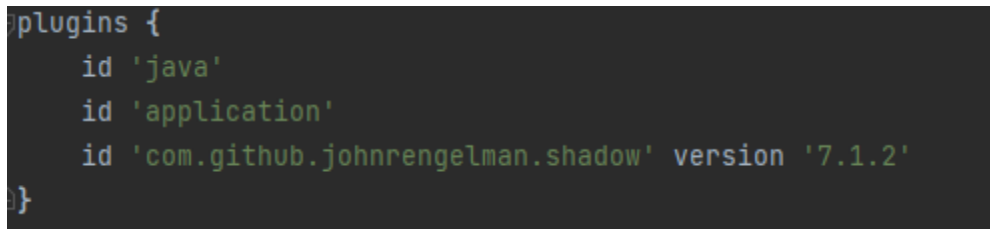
Anteriormente, se realizó la entrega de la práctica 3-4, la cual consistía en un carrito de compra en Javalin, integrando ORM y JDBC. En la práctica actual, se pretende continuar el trabajo implementando a PostgreSQL como gestor de base de datos y a Heroku como plataforma de servicios en la nube.

Repasando algunos de los cambios realizados a la práctica anterior para hacer posible utilizar PaaS:

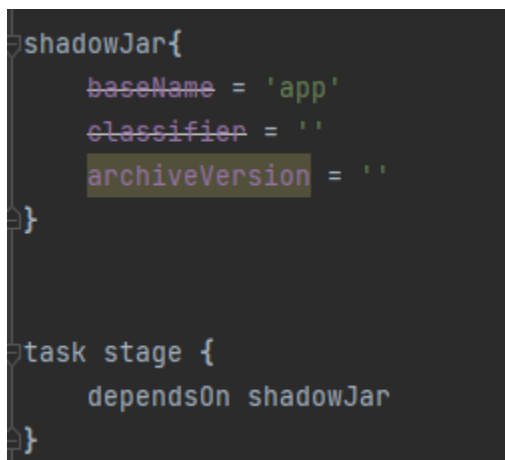
Se agregó un archivo Procfile con la siguiente línea:

A screenshot of an IDE window showing a file named 'Procfile'. The file contains a single line of code: `web: java -jar build/libs/app.jar Heroku`. The IDE has tabs for 'build.gradle (CRUD)', 'Main.java', 'Procfile', and 'sys'.

Fue fundamental cambiar de igual forma el archivo build.gradle, en donde se incluyeron las líneas referentes a shadow Jar, que ayudará con los Fat Jar.

A screenshot of a code editor showing the 'plugins' block in a build.gradle file. The code is:

```
plugins {  
    id 'java'  
    id 'application'  
    id 'com.github.johnrengelman.shadow' version '7.1.2'  
}
```

A screenshot of a code editor showing the 'shadowJar' task configuration in a build.gradle file. The code is:

```
shadowJar {  
    baseName = 'app'  
    classifier = ''  
    archiveVersion = ''  
}  
  
task stage {  
    dependsOn shadowJar  
}
```

En cuanto al Main, se agregó un modoConexion, y se utilizó el getHerokuAssignedPort para obtener el puerto que Heroku asignará para la aplicación.

```
public static void main(String[] args) throws SQLException {
    if(args.length >= 1){
        modoConexion = args[0];
        System.out.println("Modo de Operacion: "+modoConexion);
    }

    if(modoConexion.isEmpty()) {
        DatabaseStarter.startDatabase();
    }

    Javalin app = Javalin.create(config ->{
        config.addStaticFiles(staticFileConfig -> {
            staticFileConfig.hostedPath = "/";
            staticFileConfig.directory = "/publico";
            staticFileConfig.location = Location.CLASSPATH;
        });
        config.enableCorsForAllOrigins();
    }).start(getHerokuAssignedPort());
}
```

En cuanto al manejador de la base de datos, se agregó una condición, la cual establece que si existe una conexión de Heroku, el EntityManagerFactory hará uso de la función getConfiguraciónBaseDatosHeroku().

```
public DBManager(Class<T> claseEntidad){
    if(Main.getConexion().equalsIgnoreCase("Heroku")){
        emf = getConfiguraciónBaseDatosHeroku();
    }else{
        emf = Persistence.createEntityManagerFactory("MiUnidadPersistencia");
    }

    this.claseEntidad= claseEntidad;
}
```

Persistence xml

En el archivo persistence.xml, se agregó un bloque de código para habilitar la aplicación.

```
<persistence-unit name="Heroku" transaction-type="RESOURCE_LOCAL">
```

En la terminal del computador se utilizaron los comandos siguientes, donde se observa que se clonó el repositorio de la aplicación

```
C:\Users\Nicole Urena\Desktop>git clone https://github.com/nicoluv/tienda.git
Cloning into 'tienda'...
remote: Enumerating objects: 105, done.
remote: Counting objects: 100% (105/105), done.
remote: Compressing objects: 100% (60/60), done.
Receiving objects: 66% (70/105)used 104 (delta 32), pack-reused 0
Receiving objects: 100% (105/105), 85.93 KiB | 956.00 KiB/s, done.
Resolving deltas: 100% (33/33), done.

C:\Users\Nicole Urena\Desktop>cd tienda

C:\Users\Nicole Urena\Desktop\tienda>git init
Reinitialized existing Git repository in C:/Users/Nicole Urena/Desktop/tienda/.git/
```

Se inició sesión en Heroku, y se utilizó la plataforma para que la aplicación pueda ser implementada en un PaaS.

```
C:\Users\Nicole Urena\Desktop\tienda>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/14b810a6-3fff-460c-825c-16dedc23b7b5?requestor=SFHyNTY.g2gdbQAAAsxODYuNy40LjEwMzI4GAMz1ASCAMIAAVGA.JV7ZkjkakPQq153tmyPyFFFWzopJ4YQ1E
9_JVK9FHyI
Logging in... done
Logged in as 20181669@e.purmm.edu.do
```

```
C:\Users\Nicole Urena\Desktop\g\tienda>heroku create
Creating app... done, ⬢ gentle-mesa-17800
https://gentle-mesa-17800.herokuapp.com/ | https://git.heroku.com/gentle-mesa-17800.git
```

```
C:\Users\Nicole Urena\Desktop\g\tienda>heroku git:remote -a gentle-mesa-17800
set git remote heroku to https://git.heroku.com/gentle-mesa-17800.git
```

```
C:\Users\Nicole Urena\Desktop\g\tienda>heroku addons:create heroku-postgresql:hobby-dev
Creating heroku-postgresql:hobby-dev on ⬢ gentle-mesa-17800... free
Database has been created and is available
! This database is empty. If upgrading, you can transfer
! data from another database with pg:copy
Created postgresql-spherical-90761 as DATABASE_URL
Use heroku addons:docs heroku-postgresql to view documentation
```

```
C:\Users\Nicole Urena\Desktop\tienda>git add .

C:\Users\Nicole Urena\Desktop\tienda>git commit -am "make it better"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

C:\Users\Nicole Urena\Desktop\tienda>git push heroku master
Enumerating objects: 105, done.
Counting objects: 100% (105/105), done.
Delta compression using up to 4 threads
Compressing objects: 100% (59/59), done.
Writing objects: 100% (105/105), 85.93 KiB | 14.32 MiB/s, done.
Total 105 (delta 33), reused 105 (delta 33), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: ----> Building on the Heroku-20 stack
remote: ----> Determining which buildpack to use for this app
remote: ----> Gradle app detected
remote: ----> Installing OpenJDK 17... done
remote: ----> Building Gradle app...
remote: ----> executing ./gradlew stage
remote:   Downloading https://services.gradle.org/distributions/gradle-7.4-bin.zip
remote:   .....10%.....20%.....30%.....40%.....50%.....60%.....70%.....80%.....90%.....100%
remote:   To honour the JVM settings for this build a single-use Daemon process will be forked. See https://docs.gradle.org/7.4/userguide/gradle_daemon.html#sec:disabling_the_daemon.
remote:   Daemon will be stopped at the end of the build
remote:
remote:   > Task :compileJava
remote:   Note: Some input files use unchecked or unsafe operations.
remote:   Note: Recompile with -Xlint:unchecked for details.
remote:
remote:   > Task :processResources
remote:   > Task :classes
remote:   > Task :shadowJar
remote:   > Task :stage
remote:
remote:   Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.
remote:
remote:   You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.
remote:
remote:   See https://docs.gradle.org/7.4/userguide/command_line_interface.html#sec:command_line_warnings
remote:
remote:   BUILD SUCCESSFUL in 38s
remote:   3 actionable tasks: 3 executed
remote: ----> Discovering process types
remote:   Procfile declares types -> web
```

```
remote:
remote:      BUILD SUCCESSFUL in 39s
remote:      3 actionable tasks: 3 executed
remote: -----> Discovering process types
remote:      Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:      Done: 93M
remote: -----> Launching...
remote:      Released v5
remote:      https://gentle-mesa-17800.herokuapp.com/ deployed to Heroku
remote:
remote: This app is using the Heroku-20 stack, however a newer stack is available.
remote: To upgrade to Heroku-22, see:
remote: https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/gentle-mesa-17800.git
* [new branch]      master -> master

C:\Users\Nicolé Urena\Desktop\g\tienda>heroku open
```

CRUD de Productos

Crear Producto

Producto	Precio	Acción
Ramen	33.0\$	<div>Editar</div> <div>Eliminar</div>

Personal > > gentle-mesa-17800

☆ Open app More

Overview Resources Deploy Metrics Activity Access Settings

Installed add-ons \$0.00/month Configure Add-ons

Heroku Postgres Hobby Dev postgresql-spherical-90761

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

20181669@ce.pucmm.edu.do 1 deploy

Latest activity All Activity

20181669@ce.pucmm.edu.do: Deployed ada71b2f Yesterday at 1:53 PM - v5

20181669@ce.pucmm.edu.do: Build succeeded Yesterday at 1:52 PM - View build log

20181669@ce.pucmm.edu.do: @ref:postgresql-spherical-90761 completed provisioning, setting DATABASE_URL Yesterday at 1:52 PM - v4

20181669@ce.pucmm.edu.do: Attach DATABASE (@ref:postgresql-spherical-90761) Yesterday at 1:52 PM - v3

20181669@ce.pucmm.edu.do: Enable Logplex Yesterday at 1:51 PM - v2

20181669@ce.pucmm.edu.do: Initial release Yesterday at 1:51 PM - v1

Conclusión

Finalmente, al completar los puntos pautados para esta práctica, se denota el cumplimiento de los objetivos propuestos, puesto que se implementó un proyecto de Java en una PaaS, específicamente empleando para ello Heroku y el gestor de base de datos PostgreSQL. Se observó lo útil que resulta utilizar plataformas como servicio, debido a que estas permiten un desarrollo rápido con una entrega de funcionalidad más rápida y frecuente, aplicando técnicas de integración continua y despliegue automático de aplicaciones, y aprovechando los beneficios más amplios de la nube. Resultó interesante asimilar el uso y comportamiento del protocolo HTTP, y también de lo ameno de implementar aplicaciones en La Nube. Entonces, se evidencia la relevancia de las PaaS y de las muchas utilidades y herramientas que pueden resultar en una aplicación que aunque funge como requisito de una asignatura, en un futuro podría ser base de un sistema para una empresa.

Bibliografía

Lum, Jeffry. (16 de marzo del 2022). Gradle tutorial. New York, EU. SE-EDU. : Recuperado de <https://se-education.org/guides/tutorials/gradle.html>

Centro de escritura Javeriano (Ed.). (2013). Normas APA. Cali, Colombia: Pontificia Universidad Javeriana. Recuperado de: <http://portales.puj.edu.co/ftpcentroescritura/Recursos/Normasapa.pdf>

w3schools.com. (2000). HTML Lists. Noruega. w3schools.com: Recuperado de https://www.w3schools.com/html/html_lists.asp

Seidornts. (21 DICIEMBRE, 2021). Heroku. España. Nts-solutioins.com: Recuperado de <https://www.nts-solutions.com/blog/heroku-que-es.html>