

PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA

Departamento de Ciencias e Ingeniería.

Escuela de Ciencias En Computación y Telecomunicaciones



PROGRAMACION WEB

“Práctica 2”

Presentado por: Nicol Ureña (2018-1669)

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

Entregado a: Carlos Camacho

Fecha de entrega: junio del 2022

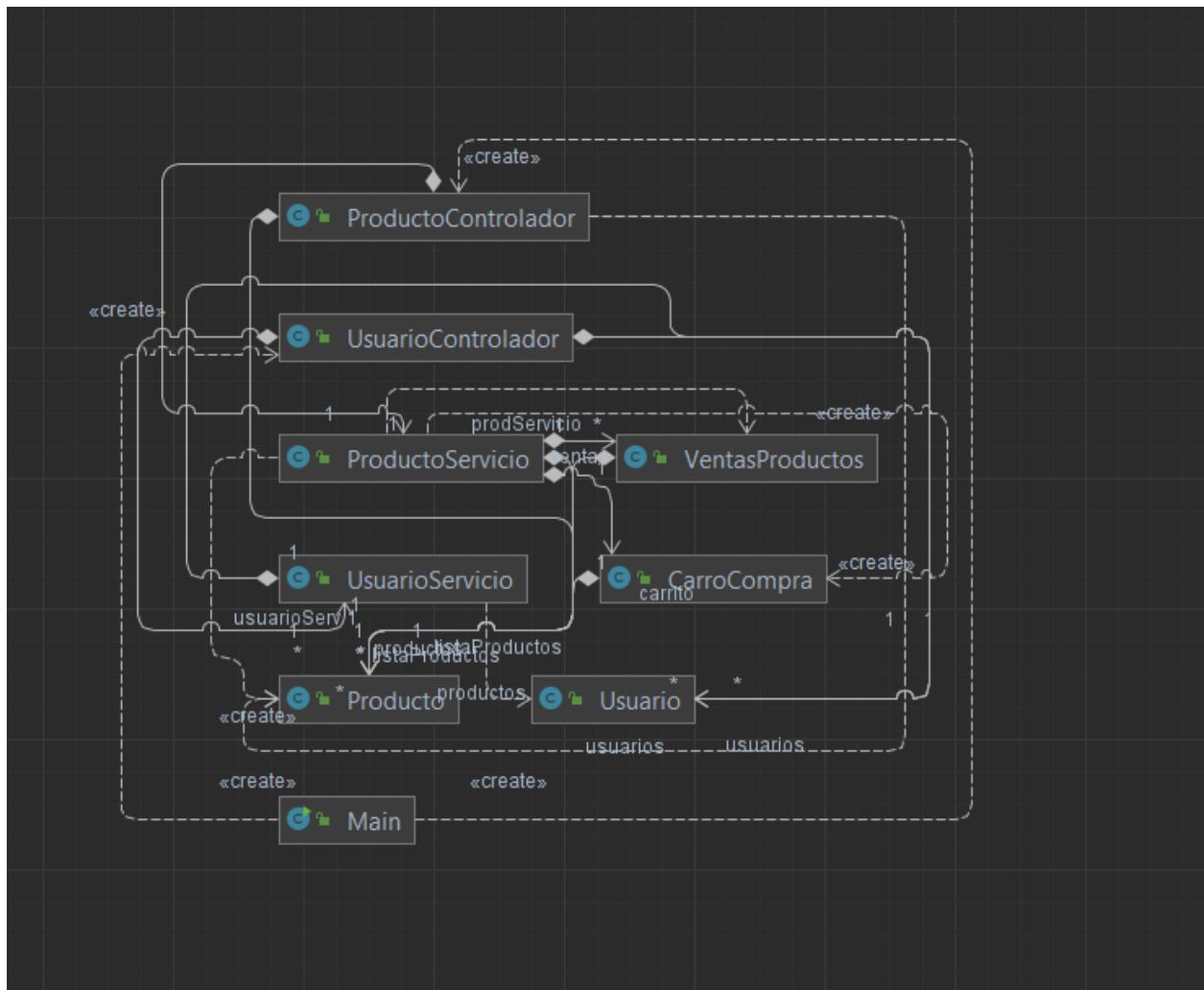
SANTIAGO DE LOS CABALLEROS, REPÚBLICA DOMINICANA

Introducción

En la actualidad, la seguridad en una aplicación web y su uso correcto de las herramientas disponibles es casi tan elemental como el aire para los humanos. Debido a ello, tener claras las nociones de estos conceptos no sólo resulta útil, sino también necesario. A continuación, se presentará una práctica acerca de la creación de un carro de compra en sesión, con los objetivos de manejar librerías Javalin en peticiones, respuesta, y uso de la plantilla, en el uso del contexto de sesión en aplicaciones, uso de plantillas y uso de seguridad en los recursos web. La misma contará con un apartado de desarrollo, donde se discutirán los métodos empleados y la implementación en general, seguido de los aprendizajes resultantes de la práctica.

Desarrollo

En un principio, se creó un proyecto titulado CRUD, agregando las dependencias necesarias (la relativa a thymeleaf, Javalin, etc.) y se procedió a organizar la estructura del proyecto. En términos generales, la aplicación posee las siguientes clases:



En cuanto a la organización descrita, se utilizaron distintas carpetas: La de los modelos, que incluye las clases mencionadas en el documento de guía, como lo son **Usuario**, **Producto**,

CarroCompra, y VentasProductos. Estas clases contienen las variables de lugar según el formato sugerido.

Por otro lado, la carpeta de servicios contiene las clases de ProductoServicio y UsuarioServicio, mientras que la de controladores contiene a ProductoControlador y UsuarioControlador.

Recordando las pautas de la práctica, vemos algunos puntos como:

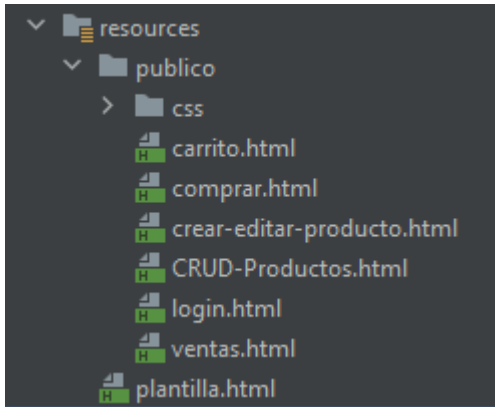
- En el manejo de datos, se utilizaron los servicios mencionados, y a la vez se empleó el patrón Singleton. Vemos que se crearon los productos de la tienda y de igual forma se creó el usuario administrador.

```
    productos.add(new Producto( nombre: "Kimchi enlatado", precio: 375));
    productos.add(new Producto( nombre: "Rice paper", precio: 120));
    productos.add(new Producto( nombre: "Pockys te verde", precio: 210));
}

1 usage
public static ProductoServicio getInstance(){
    if(instancia == null){
        instancia = new ProductoServicio();
    }
    return instancia;
}
```

- Para los controladores, se crearon 2 en su respectiva carpeta. El ProductoControlador que contiene los métodos de los productos y el UsuarioControlador que gestiona la autenticación del usuario y sus métodos.

- Para la parte visual se implementó el sistema de plantillas de thymeleaf, que es un moderno motor de plantillas Java del lado del servidor para entornos web e independientes.



- Los usuarios no autenticados, es decir, los usuarios directos, tendrán acceso a la lista de producto como página principal para seleccionar e indicar la cantidad y el producto que estarán comprando. Aquí observamos que cualquier usuario autenticado o no puede ver la lista de productos disponibles.

```
<tbody>
<tr th:each="producto : ${productos}">
  <td th:text="${producto.nombre}">Nombre del producto</td>
  <td th:text="${producto.precio}+'$'">Precio del producto</td>
  <form method="post" th:action="/comprar/'+${producto.id}">
    <td><input id="cant" name="cant" class="form-control" type="number" value="1" min="1"></td>
    <td><button class="btn btn-primary" type="submit">Agregar</button></td>
  </form>
</tr>
</tbody>
```

Así se muestran los productos. Se obtiene una lista desde el controlador de los productos y con la petición GET se obtienen de la clase service todos los productos disponibles.

- Visualizar la cantidad de elementos que el usuario tiene agregado en el carro de compras es posible gracias a que cuando un cliente compra algo, su carrito se guarda en el concepto de sesión.

Conclusión

Finalmente, al completar los puntos pautados para esta práctica, se denota el cumplimiento de los objetivos propuestos, puesto que se manejaron librerías Javalin en peticiones, respuesta, y uso de la plantilla, en el uso del contexto de sesión en aplicaciones, uso de plantillas y uso de seguridad en los recursos web. Se observó lo útil que resulta tener un motor de plantillas responsivas cuando de un entorno web se trata. Resultó interesante asimilar el uso y comportamiento del protocolo HTTP, y también algunas formas de mantener la seguridad en una aplicación. Entonces, se evidencia la relevancia de los frameworks y de las muchas utilidades y herramientas que junto con Javalin 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>

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

López Margaña, Luis Miguel. (12 de julio de 2018). Thymeleaf. OpenWebinars: Recuperado de <https://openwebinars.net/blog/que-es-thymeleaf/>

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