



10 DE NOVIEMBRE DE 2023

# DOCUMENTACIÓN

PROGRAMACIÓN II

EDITH ALVAREZ HERNANDEZ 1990-22-17750

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA  
Chimaltenango, Guatemala



## Contenido

Documentación .....	2
Introducción.....	2
Manual de Usuario.....	2
Iniciar Sesión y Registro.....	2
Navegación del sitio .....	5
Gestión de Productos desde el administrador.....	8
Selección, Compra y Gestión de Pedidos .....	15
Soporte y Servicio al Cliente .....	18
Manual Técnico .....	19
Estructura del programa.....	19
Estructura de la base de datos.....	29
Conexión a la base de datos.....	32

# Documentación

## Introducción

- *Propósito del manual*

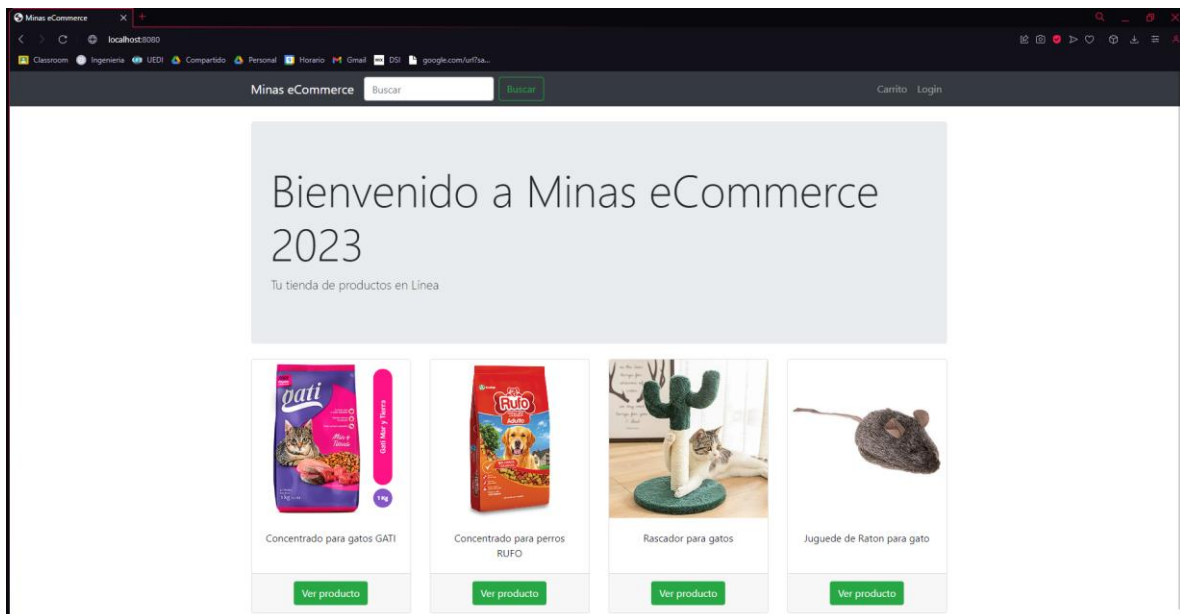
Proporcionar una guía detallada y accesible que ayude a los usuarios a navegar y utilizar eficazmente la plataforma de comercio electrónico. El Sistema eCommerce tiene como objetivo ayudar a gestionar su tienda online de manera eficiente.

## Manual de Usuario

### Iniciar Sesión y Registro

- *Vista del inicio del eCommerce*

Diríjase a su navegador de preferencia y dele clic a: <http://localhost:8080>



- *Acceso al Sistema*

En el botón "Login" y dele clic.

Minas eCommerce Login

### Ingresar

Email:  
Ingrese su email

Contraseña:  
Ingrese su contraseña

Ingresar

[Si aún no tiene cuenta aquí puede registrarse](#)

Copyright © Minas eCommerce 2023

Ya que se encuentra en el formulario ingrese sus credenciales, si en dado caso, no las tiene puede registrarse dándole clic donde dice “Si aun no tiene cuenta aquí puede registrarse”.

Ingresar

[Si aún no tiene cuenta aquí puede registrarse](#)

Ya que este en este formulario

Minas eCommerce Login

### Registro

Nombres:  
Ingrese sus nombres

Apellidos:  
Ingrese sus apellidos

Email:  
diego.obin23@gmail.com

Dirección:  
Ingrese su dirección

Contraseña  
\*\*\*\*\*

Guardar

Copyright © Minas eCommerce 2023

Proceda a llenar los campos con sus respectivos datos, si desea regresar ya sea al Inicio o el Login por x razón, de clic en el botón “Login”. Si fuera al inicio puede darle clic encima de donde dice: “Minas eCommerce”.

Ya creado su usuario puede iniciar Sesión.

Minas eCommerce

Login

Ingresar

Email:

diego.obin23@gmail.com

Contraseña:

.....

Ingresar

[Si aún no tiene cuenta aquí puede registrarse](#)

Copyright © Minas eCommerce 2023

Ya que haya ingresado debe aparecerle “USER”, en la parte superior derecha, donde se despliega un menú de navegación.

Minas eCommerce

Buscar


Buscar

USER ▾

Bienvenido a Minas eCommerce


2023

Tu tienda de productos en Línea




Concentrado para gatos GATI

Ver producto




Concentrado para perros RUFO

Ver producto



Rascador para gatos

Ver producto



Juguete de Raton para gato

Ver producto

Caso contrario si usted es un Administrador debe aparecerle la palabra “ADMIN” y la vista de los productos cambiara.

# Bienvenido a Minas Ecommerce 2023

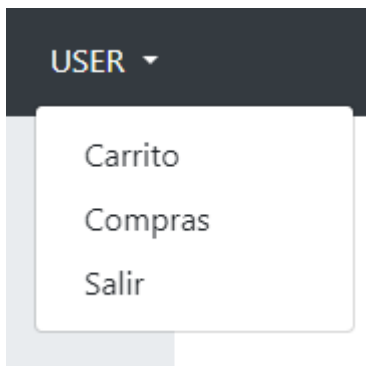
Tu tienda de productos en Línea



## Navegación del sitio

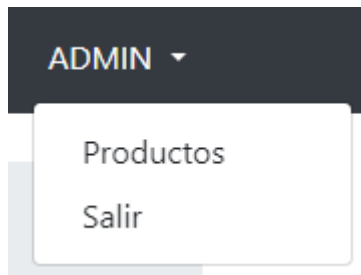
- *Menú desplegable del usuario*

Si usted da clic sobre “USER”, se desplegará un menú de opción para que pueda escoger.



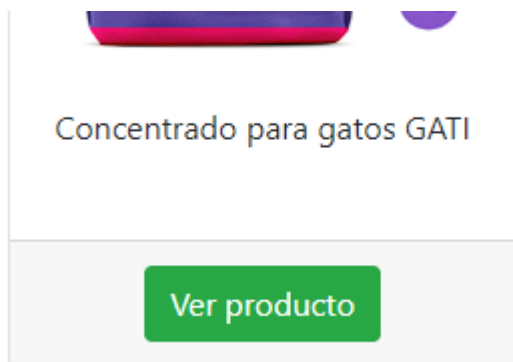
- *Menú desplegable del administrador*

Si usted da clic sobre “Admin”, se desplegará un menú de opción para que pueda escoger.



- *Ver productos como usuario*

En la página principal, usted puede visualizar los productos en existencia, para usted ver mas a detalle cada producto puede dirigirse al botón verde que dice “Ver producto”.



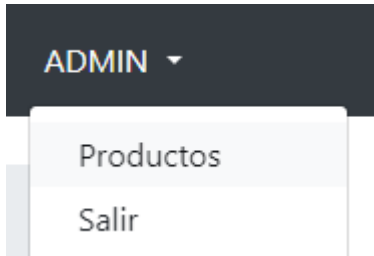
Al darle clic lo llevara a otra pagina donde puede ver el nombre, precio, descripción, la cantidad que usted desea comprar y un botón que añade al carrito.

## Minas eCommerce Producto



- *Ver productos como administrador*

En la página principal, usted puede visualizar los productos en existencia, para usted ver más a detalle cada producto puede dirigirse al botón que dice “ADMIN” y da clic en “Productos”



Al darle clic lo llevara a otra página donde puede ver en la parte superior 2 botones, uno que dice “Crear producto” y otro que dice “Generar Reporte”.

A continuación, puede ver en una tabla el nombre, descripción, la cantidad en inventario, precio y 2 acciones lo cual son “Editar” y “Eliminar”

## Minas eCommerce Productos

Home / Ver Productos

Crear Producto

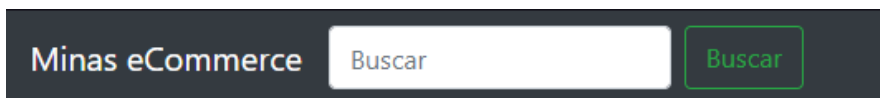
Generar Reporte

### Productos

Nombre	Descripción	Inventario	Precio	Acción	Acción
Concentrado para gatos GATI	Concentrado sabor mar y tierra para gatos de la marca gati de 1 kg	5	25.0	Editar	Eliminar
Concentrado para perros RUFO	Concentrado para perro adulto de la marca Rufo de 5 kg	10	50.0	Editar	Eliminar
Rascador para gatos	Rascador para gatos con forma de cactus de color verde lazo blanco	10	100.0	Editar	Eliminar
Juguete de Raton para gato	Juguete de peluche en forma de raton para gatos con catnip incluido	50	25.0	Editar	Eliminar

- *Buscar un producto*

En la parte superior puede usted visualizar una sección donde usted puede buscar un producto deseado.



Usted puede ingresar lo que esta buscando por ejemplo todo lo relacionado con la palabra “gato”.





Concentrado para gatos GATI



Rascador para gatos



Juguete de Raton para gato

## Gestión de Productos desde el administrador

- *Cómo añadir productos desde el administrador.*

Se ubica en “Productos” desde el menú desplegable, ya que se encuentre en la pagina de productos, da clic sobre el botón “Crear Producto”.

## Minas eCommerce Productos

[Home](#) / [Ver Productos](#)

Crear Producto

Generar Reporte

Se le desplegará un formulario donde deberá llenar con los respectivos datos del producto. Al momento de cargar la imagen de preferencia que no sea muy grande para que se mantenga la estética del recuadro de las fotos.

Si usted no carga ninguna foto, en la vista del inicio el producto saldrá con la imagen por default que se le dejo.

## Minas eCommerce Productos

[Home](#) / [Crear Productos](#)

### Crear Producto

Nombre:

Descripción:

Cantidad:

Precio:

Imagen:

 Sin archivos seleccionados

Puede guiarse por el siguiente ejemplo

## Minas eCommerce Productos

[Home](#) / [Crear Productos](#)

### Crear Producto

Nombre:

Descripción:

Cantidad:

Precio:

Imagen:

 JPP\_lazo1.jpg

Ya aparece en la vista productos de administrador

Nombre	Descripción	Inventario	Precio	Acción	Acción
Concentrado para gatos GATI	Concentrado sabor mar y tierra para gatos de la marca gati de 1 kg	5	25.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Concentrado para perros RUFO	Concentrado para perro adulto de la marca Rufo de 5 kg	10	50.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Rascador para gatos	Rascador para gatos con forma de cactus de color verde lazo blanco	10	100.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Juguede de Raton para gato	Juguete de peluche en forma de raton para gatos con catnip incluido	40	25.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>
Juguete de lazo para perro	Jugue de lazo de colores resistente para perro	10	25.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>



Por si no se cargo una imagen el producto se visualiza asi



- *Modificar productos desde el administrador.*

Ya ubicado el producto que se desea modificar, da clic al botón de color amarillo “Editar”

Juguete de lazo para perro	Jugue de lazo de colores resistente para perro	10	25.0	Editar	Eliminar
----------------------------	--	----	------	--------	----------

Le desplegara una pagina con un formulario con los datos del producto para que usted edite lo que tenga que modificar

## Minas eCommerce Productos

[Home](#) / Actualizar Producto

### Actualizar Producto

Nombre:

Juguete de lazo para perro

Descripción:

Jugue de lazo de colores resistente para perro

Cantidad:

10

Precio:

25.0

Imagen:

Seleccionar archivo Sin archivos seleccionados

Guardar

En este caso vamos a editarle el precio a 20

# Actualizar Producto

Nombre:

Juguete de lazo para perro

Descripción:

Jugue de lazo de colores resistente para perro

Cantidad:

10

Precio:

20

Imagen:

Seleccionar archivo Sin archivos seleccionados

Guardar

Si usted guarda el producto sin modificar la imagen, la que tenia desde el principio es la que seguirá mostrando, caso contrario usted puede cambiar la imagen cargando una nueva.

Juguete de lazo para perro	Jugue de lazo de colores resistente para perro	10	20.0	Editar	Eliminar
----------------------------	--	----	------	--------	----------

Concentrado para gatos GATI

Concentrado para perros  
RUFO



Juguete de lazo para perro

- *Eliminar productos desde el administrador.*

Una vez ubicado el producto a eliminar se le da clic sobre el botón “Eliminar”

Juguete de lazo para perro	Jugue de lazo de colores resistente para perro	10	20.0	Editar	Eliminar
----------------------------	--	----	------	--------	----------

Y automáticamente el producto será eliminado

## Minas eCommerce Productos

[Home](#) / Ver Productos

Crear Producto

Generar Reporte

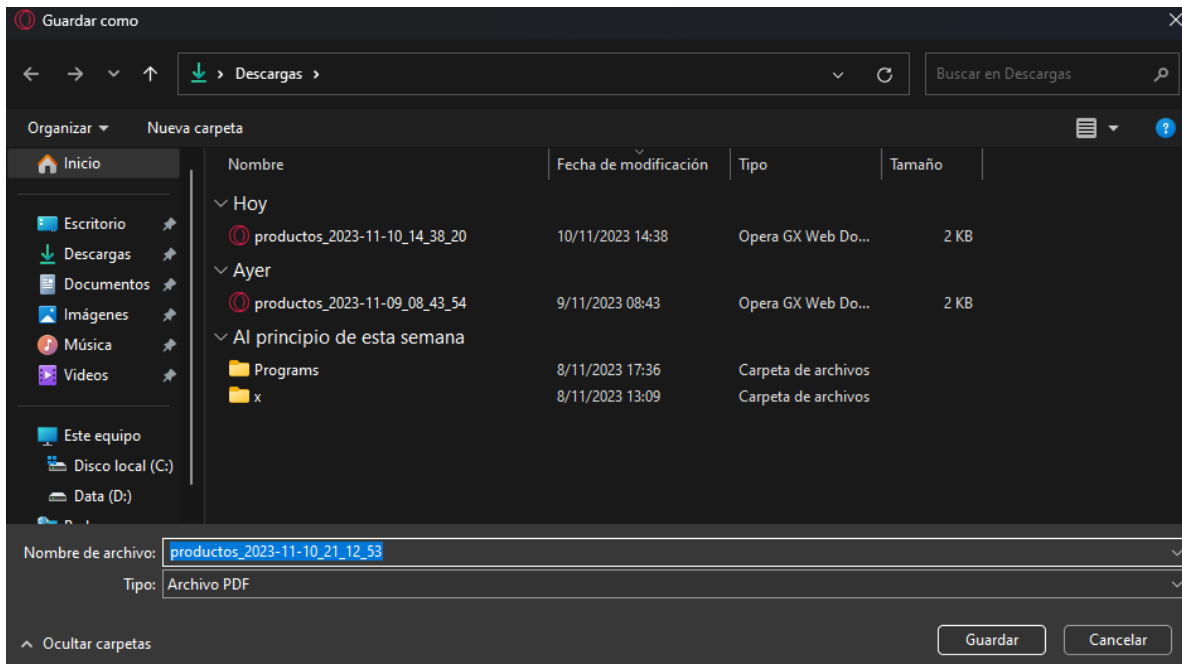
### Productos

Nombre	Descripción	Inventario	Precio	Acción	Acción
Concentrado para gatos GATI	Concentrado sabor mar y tierra para gatos de la marca gati de 1 kg	5	25.0	Editar	Eliminar
Concentrado para perros RUFO	Concentrado para perro adulto de la marca Rufo de 5 kg	10	50.0	Editar	Eliminar
Rascador para gatos	Rascador para gatos con forma de cactus de color verde lazo blanco	10	100.0	Editar	Eliminar
Juguete de Raton para gato	Juguete de peluche en forma de raton para gatos con catnip incluido	40	25.0	Editar	Eliminar

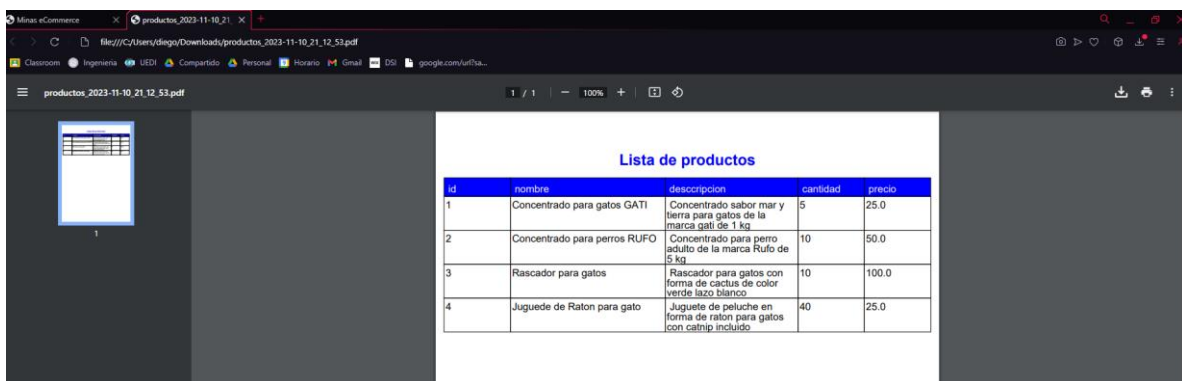
Copyright © Minas eCommerce 2023

- *Generar reporte de los productos existentes.*

Se ubica el botón de color gris oscuro que dice “Generar Reporte” y le da clic, esto le generara una ventana donde le solicita que usted ubique en su computador donde quiere que se guarde el archivo pdf.



Al darle clic a guardar se comienza la descarga y donde lo haya guardado ya puede visualizar el reporte. Abre el pdf y visualiza los productos



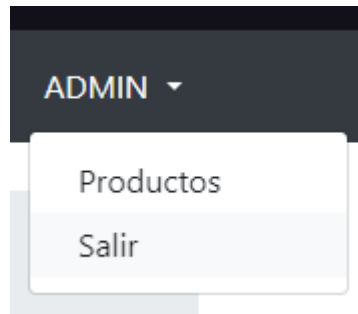
### Lista de productos

id	nombre	desccripcion	cantidad	precio
1	Concentrado para gatos GATI	Concentrado sabor mar y tierra para gatos de la marca gati de 1 kg	5	25.0
2	Concentrado para perros RUFO	Concentrado para perro adulto de la marca Rufo de 5 kg	10	50.0
3	Rascador para gatos	Rascador para gatos con forma de cactus de color verde lazo blanco	10	100.0
4	Juguete de Raton para gato	Juguete de peluche en forma de raton para gatos con catnip incluido	40	25.0

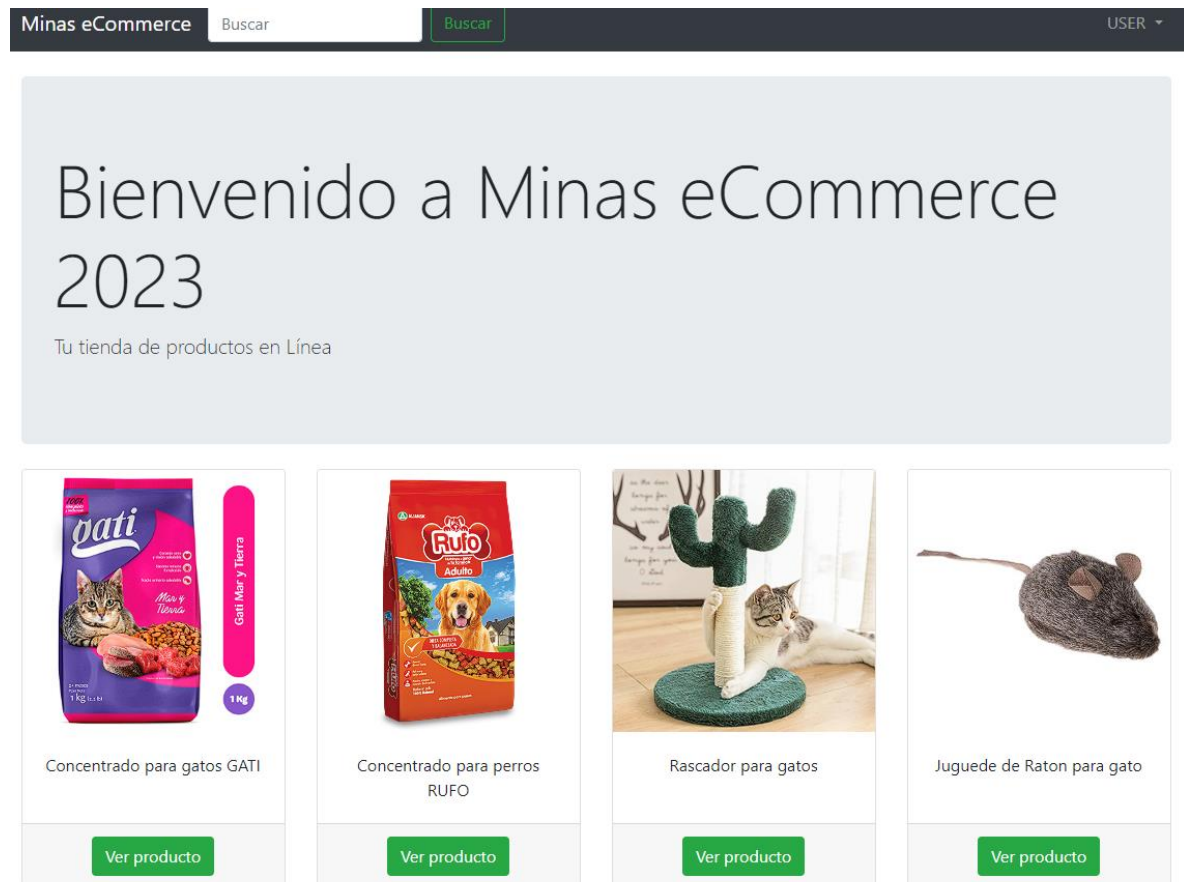
## Selección, Compra y Gestión de Pedidos

- *Cómo añadir productos al carrito.*

Solo los usuarios pueden ver y agregar cosas al carrito, por lo tanto, si está en la sesión como administrador. debe salir de su sesión. Diríjase al menú desplegable “ADMIN” y da clic sobre “Salir”.



Usted ya con su sesion iniciada puede visualizar los productos en existencia.



diríjase al botón “Ver Producto” y le da clic, ya que este en la visualización detallada del producto usted puede colocar la cantidad que desea adquirir.

En este caso en cantidad coloque 2 y le da clic a “Añadir al carrito”





**Gati Mar y Tierra**

**1 Kg**

### Concentrado para gatos GATI

Precio: 25.0

Descripción: Concentrado sabor mar y tierra para gatos de la marca gati de 1 kg

Cantidad:

Añadir al carrito

Ya que le dio clic al botón “Añadir al Carrito” se le desplegara la siguiente página:

Minas eCommerce

USER ▾

## Minas eCommerce Carrito

[Home](#) / [Carrito](#)

Producto	Precio	Cantidad	Total	Acción	SUBTOTAL
Concentrado para gatos GATI	25.0	2.0	50.0	<input type="button" value="Quitar"/>	

Copyright © Minas eCommerce 2023

Usted si lo desea puede quitar ese producto del carrito o seguir añadiendo otros productos.

Para añadir más productos puede dirigirse a “Home” o “Minas eCommerce” y darle clic.

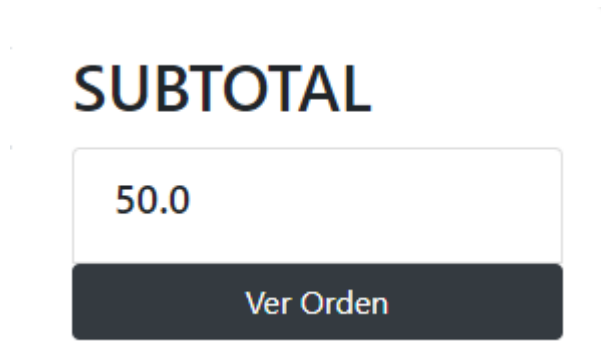
Minas eCommerce

[Home](#) / [Carrito](#)

Minas eCommerce

Y repetir el proceso explicado.

Si usted le da clic sobre el botón “Ver Orden”



Lo desplegara en una pagina donde esta más detallada la información de la orden, datos de usted como del pedido.

## Minas eCommerce Resumen de la orden

Home / Orden

**DATOS ORDEN**

Nombre: DiegoObin  
Correo: diego.obin23@gmail.com  
Dirección: Colonia Villa Sol, Guatemala

**PRODUCTOS**

Producto	Precio	Cantidad	Total
Concentrado para gatos GATI	25.0	2.0	50.0

**Resumen Orden**

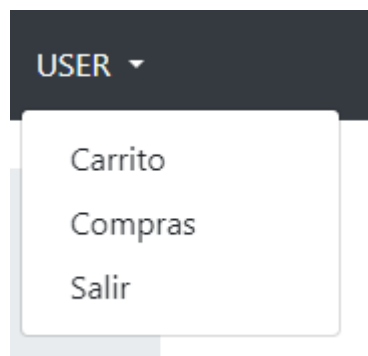
Total: 50.0

Generar

Si le da clic en generar usted estaría finalizando el pedido, realizando la petición para que se le cargue el pedido.

- [Como ver mis compras](#)

Ya que usted a generado su pedido y desea ver si se le cargo dicho asunto, puede verlo. Diríjase al menú desplegable “USER”



Y le saldrá una pagina llamada Compras donde le dice cuantos pedidos tiene, con la fecha de petición y el valor sobre dicho pedido.

## Spring eCommerce Compras

[Home](#) / [Compras](#)

No. de Orden	Fecha	Valor
00000001	2023-11-10 13:11:59.136	25.0
00000002	2023-11-10 13:32:34.326	25.0
00000003	2023-11-10 14:30:17.977	75.0
00000004	2023-11-10 21:29:42.061	50.0

En este caso la ultima es la que nosotros realizamos anteriormente.

### Soporte y Servicio al Cliente

- [Información de Contacto](#)

Envía un correo al email [edialva02@gmail.com](mailto:edialva02@gmail.com)

Contáctame en mis redes sociales

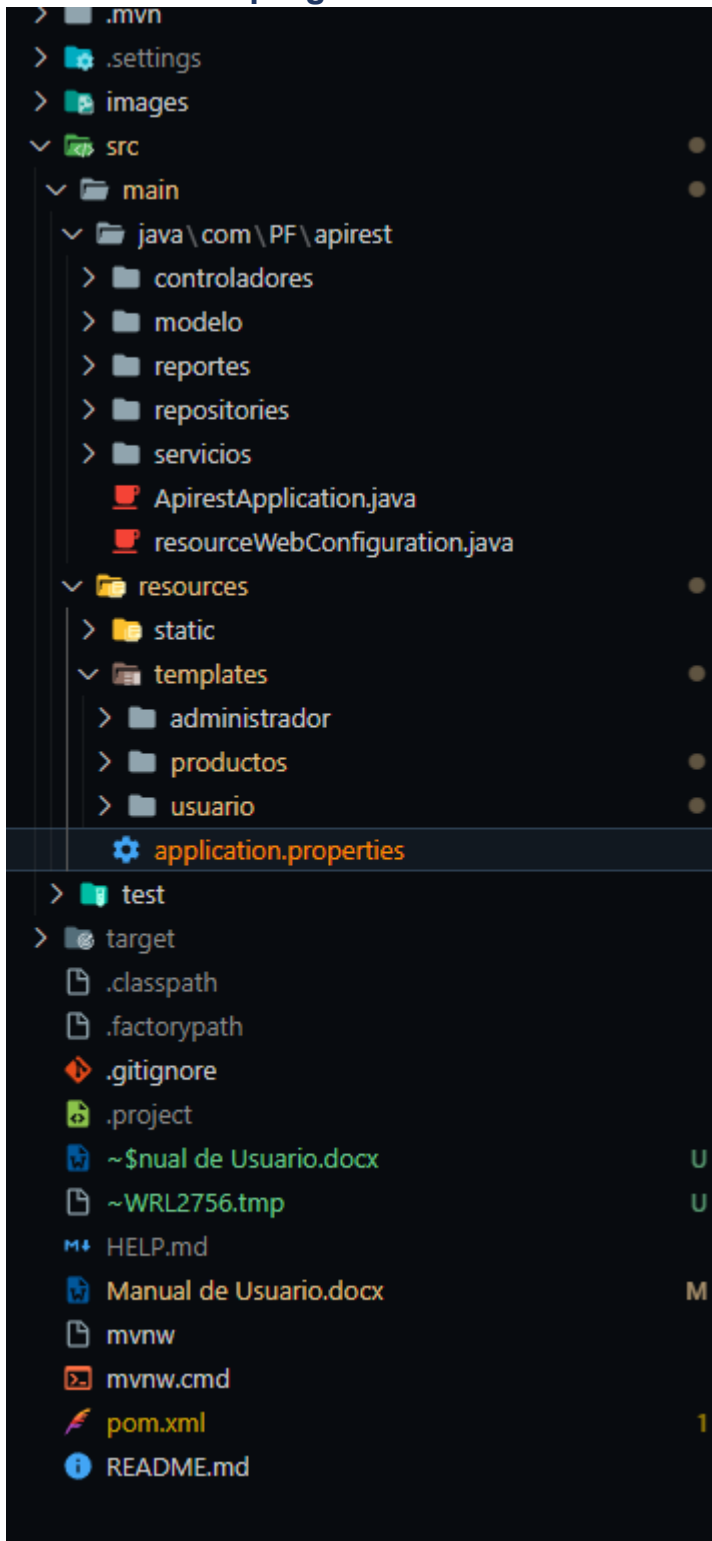
Instagram con el siguiente link: [alvarezedith123](#)

Facebook con el

siguientelink: <https://www.facebook.com/edith.alvarez.39142072?mibextid=ZbWKwL>

## Manual Técnico

### Estructura del programa



Se crearon paquetes que cada paquete llevara a cabo una determinada función

Por ejemplo en el paquete modelo, tenemos la estructura de lo que seria usuario, orden, producto y detalleorden



```
src > main > java > com > PF > apirest > modelo > usuario.java > usuario > usuario(Integer, Stri
You, hace 5 horas | 1 author (You)
1 package com.PF.apirest.modelo;
2
3 import java.util.List;
4 import jakarta.persistence.Entity;
5 import jakarta.persistence.GeneratedValue;
6 import jakarta.persistence.GenerationType;
7 import jakarta.persistence.Id;
8 import jakarta.persistence.OneToMany;
9 import jakarta.persistence.Table;
10
11
You, hace 5 horas | 1 author (You)
12 @Entity
13 @Table(name = "usuarios")
14 public class usuario {
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY )
17     private Integer id;
18     private String nombre;
19     private String apellido;
20     private String email;
21     private String password;
22     private String direccion;
23     private String rol;
24
25     @OneToMany(mappedBy = "usuario")
26     private List<producto> productos;
27
28     @OneToMany(mappedBy = "usuario")
29     private List<orden> ordenes;
```

También se le implemento al código que la relación y creación de las tablas se haga desde el código, así por si se desea dejar de usar un campo en la tabla se puede eliminar desde acá y ya no aparecería en la base de datos

```
detalleOrden.java  orden.java  producto.java X  usuario.java

src > main > java > com > PF > apirest > modelo > producto.java > producto > producto(Integer, String, String, String, double, int)
You, hace 5 horas | 1 author (You)
1 package com.PF.apirest.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.ManyToOne;
8 import jakarta.persistence.Table;
9
10 You, hace 5 horas | 1 author (You)
11 @Entity
12 @Table(name = "productos")
13 public class producto {
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Integer id;
17     private String nombre;
18     private String descripcion;
19     private String imagen;
20     private double precio;
21     private int cantidad;
22
23     @ManyToOne
24     private usuario usuario;
25
26     public producto() {
27     }
28
29     public producto(Integer id, String nombre, String descripcion, String imagen, double precio, int cantidad) {
30         this.id = id;
```

```
detalleOrden.java  orden.java x  producto.java  usuario.java
src > main > java > com > PF > apirest > modelo > orden.java > orden > toString()
15  @Table(name = "ordenes")
16  public class orden {
17      @Id
18      @GeneratedValue(strategy = GenerationType.IDENTITY)
19      private Integer id;
20      private String numero;
21      private Date fechaCreacion;
22      private Date fechaRecibida;
23      private double total;
24
25      @ManyToOne
26      private usuario usuario;
27
28      @OneToMany(mappedBy = "orden")
29      private List<detalleOrden> detalle;
30
31      public orden() {
32      }
33
34
35      public orden(Integer id, String numero, Date fechaCreacion, Date fechaRecibida, double total) {
36          this.id = id;
37          this.numero = numero;
38          this.fechaCreacion = fechaCreacion;
39          this.fechaRecibida = fechaRecibida;
40          this.total = total;
41      }
42
43
44      public Integer getId() {
45          return id;

```

```
detalleOrden.java x orden.java producto.java usuario.java
src > main > java > com > PF > apirest > modelo > detalleOrden.java > detalleOrden > detalleOrden(Integer, String, double, de
You, hace 5 horas | 1 author (You)
1 package com.PF.apirest.modelo;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import jakarta.persistence.ManyToOne;
8 import jakarta.persistence.Table;
9
You, hace 5 horas | 1 author (You)
10 @Entity
11 @Table(name = "detalles")
12 public class detalleOrden {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Integer id;
16     private String nombre;
17     private double precio;
18     private double cantidad;
19     private double total;
20
21     @ManyToOne
22     private orden orden;
23
24     @ManyToOne
25     private producto producto;
26
27     public detalleOrden() {
28     }
29
30 }
```

A estos se les creo un constructor, los getter and setter y toString.

En los controladores, como su palabra lo dice, nos ayuda a los redireccionamientos y ciertas funciones que se le crearon para que al momento de interactuar en la pagina realice las peticiones de los usuarios

```
controladores
├── administradorControlador.java
├── homeController.java
├── productoControlador.java
└── UsuarioController.java
```



```
administradorControlador.java x homeController.java productoControlador.java UsuarioController.java
src > main > java > com > PF > apirest > controladores > administradorControlador.java > ...
You, hace 3 semanas | 1 author (You)
1 package com.PF.apirest.controladores;
2
3
4 > import org.springframework.web.bind.annotation.GetMapping; ...
13
14
You, hace 3 semanas | 1 author (You)
15 @Controller
16 @RequestMapping("/administrador")
17 public class administradorControlador {
18
19     @Autowired
20     private productoService productoService;
21
22     @GetMapping("")
23     public String home(Model model) {
24
25         List<producto> productos= productoService.findAll();
26         model.addAttribute(attributeName:"productos", productos);
27
28         return "administrador/home"; // Devuelve la ruta del archivo HTML sin la extensión
29     }
30 }
31
```

Este se creó específicamente para el redireccionamiento de la home para el administrador

```
administradorControlador.java  homeController.java X  productoControlador.java  UsuarioController.java
src > main > java > com > PF > apirest > controladores > homeController.java > Language Support for Java(TM) by Red Hat > h
You, hace 5 horas | 1 author (You)
1 package com.PF.apirest.controladores;
2
3 > import java.util.Date; ...
29
You, hace 5 horas | 1 author (You)
30 @Controller
31 @RequestMapping("/")
32 public class homeController {
33
34     private final Logger log= LoggerFactory.getLogger(clazz:homeController.class);
35
36     @Autowired
37     private productoService productoService;
38
39     @Autowired
40     private InterfzUsuarioService usuarioService;
41
42     @Autowired
43     private InterfzOrdenService ordenService;
44
45     @Autowired
46     private InterfzDetalleOrdenService detalleOrdenService;
47
48     //para almacenar los detalles de la orden
49     List<detalleOrden> detalles = new ArrayList<detalleOrden>();
50
51     //datos de la orden
52     orden orden = new orden();
53
54     @GetMapping("")
```

Este se creo para el home, la parte donde se visualizan los productos.

Para todo lo relacionado al carrito en homeController se hizo la estructura.

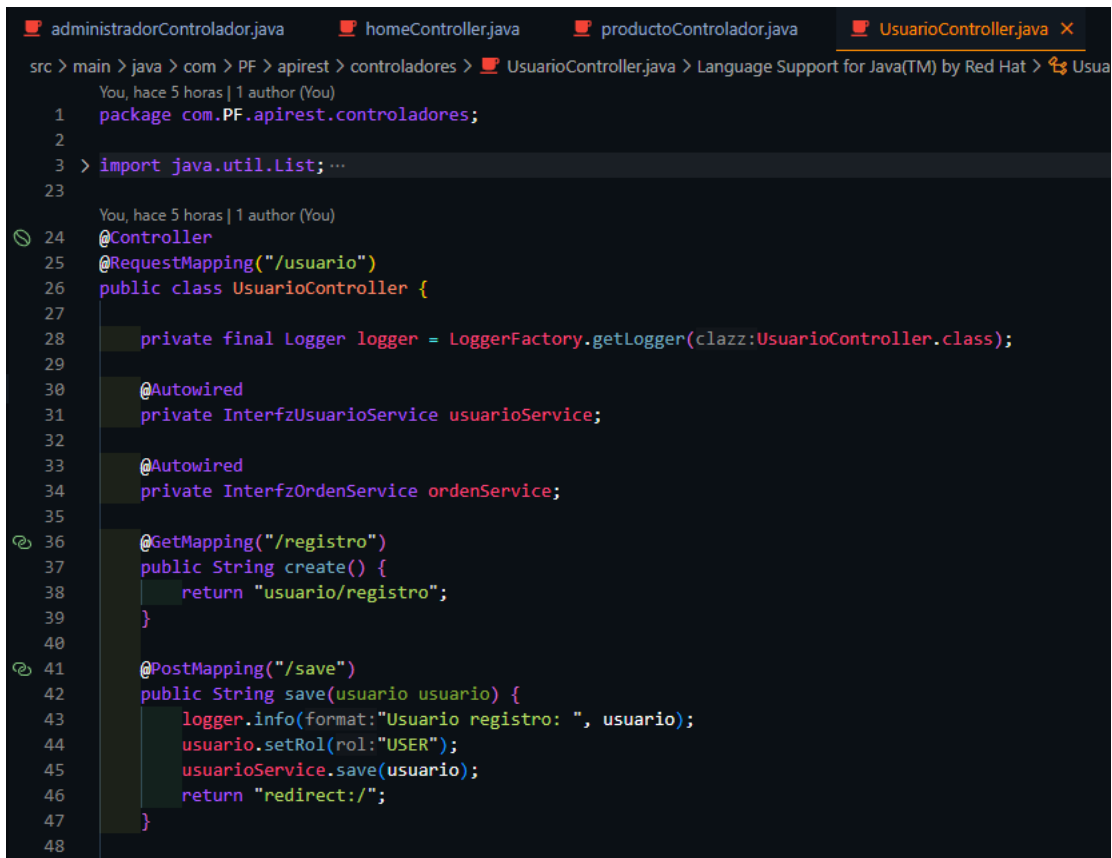
Para lo de la sesión, al momento que se inicia sesión y se realiza un pedido los datos del usuario que está abierta la sesión los mande a llamar para que pueda mostrar los datos del usuario cuando se da el detalle de la orden.

La funcionalidad de buscar producto se realizo aquí, se realizo una para administrador y otra para usuario.

```
administradorControlador.java  homeController.java  productoControlador.java X  UsuarioController.java
src > main > java > com > PF > apiREST > controladores > productoControlador.java > Language Support for Java(TM) by Red Hat > prod
You, hace 10 horas | 1 author (You)
1  package com.PF.apiREST.controladores;
2
3  > import java.io.IOException;...
31
32
You, hace 10 horas | 1 author (You)
33 @Controller
34 @RequestMapping("/productos")
35 public class productoControlador {
36
37     private final Logger LOGGER = LoggerFactory.getLogger(clazz:productoControlador.class);
38
39     @Autowired
40     private productoService productoService;
41
42     @Autowired
43     private InterfzUsuarioService usuarioService;
44
45     @Autowired
46     private uploadFileService upload;
47
48     @GetMapping("")
49     public String show(Model model){
50         model.addAttribute(attributeName:"productos", productoService.findAll());
51         return "productos/show";
52     }
53
54     @GetMapping("/create")
55     public String create(){
56         return "productos/create";
57     }
58 }
```

Aquí están los redireccionamientos para las funcionalidades en la que se involucra producto, como crear un producto, eliminar un producto, modificar. Esto también para lo de la imagen ya que se tuvieron que crear funciones específicas para las imágenes y luego relacionarlas para cuando editan, crean o eliminan un producto

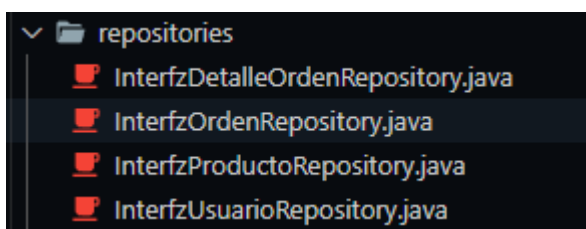
Y no olvidemos que aquí está lo de generar reporte de productos, la cual esta se vincula con el paquete reportes. Donde desde código se genera la estructura que tendrá la tabla donde se montarán los datos que se mandan a llamar a la base de datos.



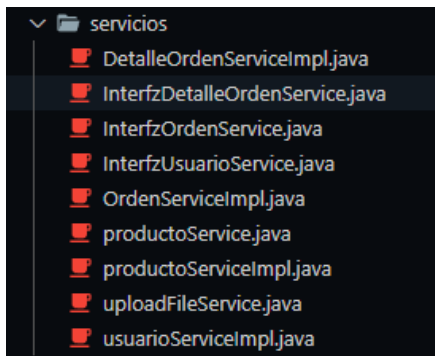
```
src > main > java > com > PF > apirest > controladores > UsuarioController.java > Language Support for Java(TM) by Red Hat > Usua
You, hace 5 horas | 1 author (You)
1 package com.PF.apirest.controladores;
2
3 > import java.util.List; ...
23
You, hace 5 horas | 1 author (You)
24 @Controller
25 @RequestMapping("/usuario")
26 public class UsuarioController {
27
28     private final Logger logger = LoggerFactory.getLogger(clazz:UsuarioController.class);
29
30     @Autowired
31     private InterfzUsuarioService usuarioService;
32
33     @Autowired
34     private InterfzOrdenService ordenService;
35
36     @GetMapping("/registro")
37     public String create() {
38         return "usuario/registro";
39     }
40
41     @PostMapping("/save")
42     public String save(usuario usuario) {
43         logger.info(format:"Usuario registro: ", usuario);
44         usuario.setRol(rol:"USER");
45         usuarioService.save(usuario);
46         return "redirect:/";
47     }
48 }
```

Aquí se realiza lo del registro y login, que al momento de que el correo exista en la base de datos esta guardara el id del usuario donde nosotros lo guardaremos dentro de una variable donde después la utilizaremos para las demás funciones para mostrar datos del usuario y cosas así. También esta lo de cerrar sesión.

En el paquete repositories, por lo general heredan los métodos para poder trabajar



En el paquete servicios se declaran 2 cosas, la lógica para que pueda aislarla de otras partes de la aplicacion y la definición de los métodos de los servicios que se van a manejar



Por ejemplo

```
InterfzDetalleOrdenRepository.java X DetalleOrdenServiceImpl.java InterfzDetalleOrdenService.java
src > main > java > com > PF > apirest > repositories > InterfzDetalleOrdenRepository.java > ...
You, ayer | 1 author (You)
1 package com.PF.apirest.repositories;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4 import org.springframework.stereotype.Repository;
5 import com.PF.apirest.modelo.detalleOrden;
6
7 @Repository
8 public interface InterfzDetalleOrdenRepository extends JpaRepository<detalleOrden, Integer>{
9
10 }
11
```

Esta interfaz extiende JpaRepository, que es una interfaz de Spring Data JPA. Al extender JpaRepository, InterfzDetalleOrdenRepository hereda varios métodos para trabajar con la base de datos, como guardar (save), buscar (find), eliminar (delete), etc.

```
InterfzDetalleOrdenRepository.java DetalleOrdenServiceImpl.java X InterfzDetalleOrdenService.java
src > main > java > com > PF > apirest > servicios > DetalleOrdenServiceImpl.java > ...
You, ayer | 1 author (You)
1 package com.PF.apirest.servicios;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Service;
5 import com.PF.apirest.modelo.detalleOrden;
6 import com.PF.apirest.repositories.InterfzDetalleOrdenRepository;
7
8 @Service
9 public class DetalleOrdenServiceImpl implements InterfzDetalleOrdenService{
10
11     @Autowired
12     private InterfzDetalleOrdenRepository detalleOrdenRepository;
13
14     @Override
15     public detalleOrden save(detalleOrden detalleOrden) {
16         return detalleOrdenRepository.save(detalleOrden);
17     }
18
19 }
20
```

Esta clase implementa la lógica de negocio relacionada con detalleOrden. Se define como un servicio (@Service) que es una parte crucial en la arquitectura de Spring Boot para separar la lógica de negocio de otras partes de la aplicación.

La anotación @Autowired se utiliza para inyectar automáticamente la dependencia de InterfzDetalleOrdenRepository. Esto significa que Spring se encarga de crear una instancia de InterfzDetalleOrdenRepository y la proporciona a esta clase.

El método save es una implementación del método definido en la interfaz InterfzDetalleOrdenService. Este método utiliza detalleOrdenRepository para guardar el objeto detalleOrden en la base de datos.

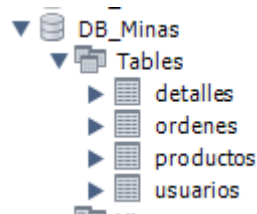


```
src > main > java > com > PF > apirest > servicios > InterfzDetalleOrdenService.java > ...
You, ayer | 1 author (You)
1 package com.PF.apirest.servicios;
2
3 import com.PF.apirest.modelo.detalleOrden;
4
5 You, ayer | 1 author (You)
6 public interface InterfzDetalleOrdenService {
7     detalleOrden save(detalleOrden detalleOrden);
8 }
9
```

Esta interfaz define los métodos que se deben implementar en el servicio que maneja detalleOrden.

Aquí se declara el método save, que luego será implementado en DetalleOrdenServiceImpl. Este método está destinado a guardar un objeto de tipo detalleOrden.

## Estructura de la base de datos





Result Grid						
Filter Rows: <input type="text"/>						
Edit: <input type="text"/>						
Export/Import: <input type="text"/>						
	id	fechaCreacion	fechaRecibida	numero	total	usuario_id
▶	1	2023-11-10 13:11:59.136000	NULL	00000001	25	2
	2	2023-11-10 13:32:34.326000	NULL	00000002	25	2
	3	2023-11-10 14:30:17.977000	NULL	00000003	75	2
	4	2023-11-10 21:29:42.061000	NULL	00000004	50	2
✱	NULL	NULL	NULL	NULL	NULL	NULL

[illegible]



## Conexión a la base de datos

⚙ application.properties ✕

src > main > resources > ⚙ application.properties

You, hace 3 semanas | 1 author (You)

```
1  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
2  spring.datasource.url=jdbc:mysql://localhost:3306/DB_Minas
3  spring.datasource.username=root
4  spring.datasource.password=password
5
6
7  spring.jpa.hibernate.ddl-auto=update
8  spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQLDialect
9  spring.jpa.show-sql=true
10 spring.jpa.open-in-view=false
11 spring.jpa.properties.hibernate.format_sql=true
12 #logging.level.org.hibernate.SQL=DEBUG
13
14 spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
15
16
17 # Configuración de Thymeleaf
18 spring.thymeleaf.prefix=classpath:/templates/
19 spring.thymeleaf.suffix=.html
20
```