THE ENGLISH CUT

PRESENTACIÓN

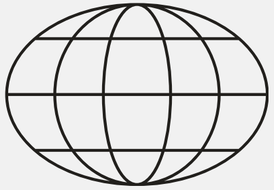
ÍNDICE

INTRODUCCIÓN A THE ENGLISH CUT

APLICACIÓN WEB DE COMPRA VENTA

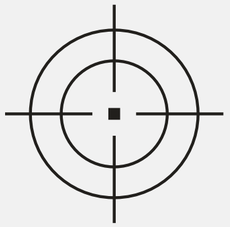
Compra venta

EN ENGLISH CUT, NOS ENCARGAMOS DE COMPRAR PRODUCTOS A EMPRESAS PARTICULARES QUE NOS GARANTICEN CIERTA CALIDAD Y VENDERLOS EN NUESTRA WEB



Internacional

NOSOTROS COMO THE ENGLISH CUT INTENTAMOS LLEGAR A LA MAXIMA GENTE POSIBLE, EMPEZANDO EN ESPAÑA Y EXPANDIENDONOS CADA VEZ MAS CON EL FIN DE LLEGAR A LA MAXIMA GENTE POSIBLE



Selección

CONTAMOS CON PERSONA PARA SELECCIONAR QUE EMPRESAS ESTAN CUALIFICADAS PARA VENDER A TRAVES DE NUESTRA APLICACION, BRINDANDO LA MAYOR CALIDAD A NUESTROS CLIENTES

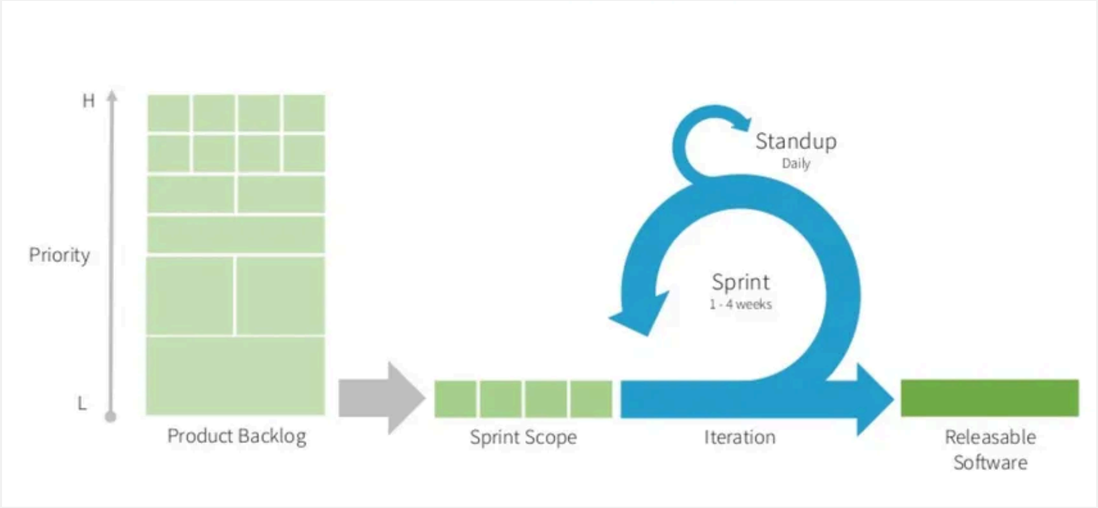
DESCRIPCIÓN DEL PROYECTO SOFTWARE

1- REUNIONES: 1 SEMANAL Y SPRINT
2- CONCRETAR EL PROYECTO: LENGUAJE Y DESARROLLO
3- ARQUITECTURA SOFTWARE UTILIZADA MVC
4- LA BASE DE DATOS ENTRADAS Y COMO LA HEMOS TRABAJADO
5- DISTRIBUCION DE LOS CASOS DE USO

REUNIONES

- **REUNIONES SEMANALES**

LA CARGA DEL SPRINT ERA REVISADA EN CADA REUNIÓN DEL EQUIPO, DONDE DECIDIAMOS QUE ACTIVIDADES REALIZAR PARA AVANZAR EL PROYECTO



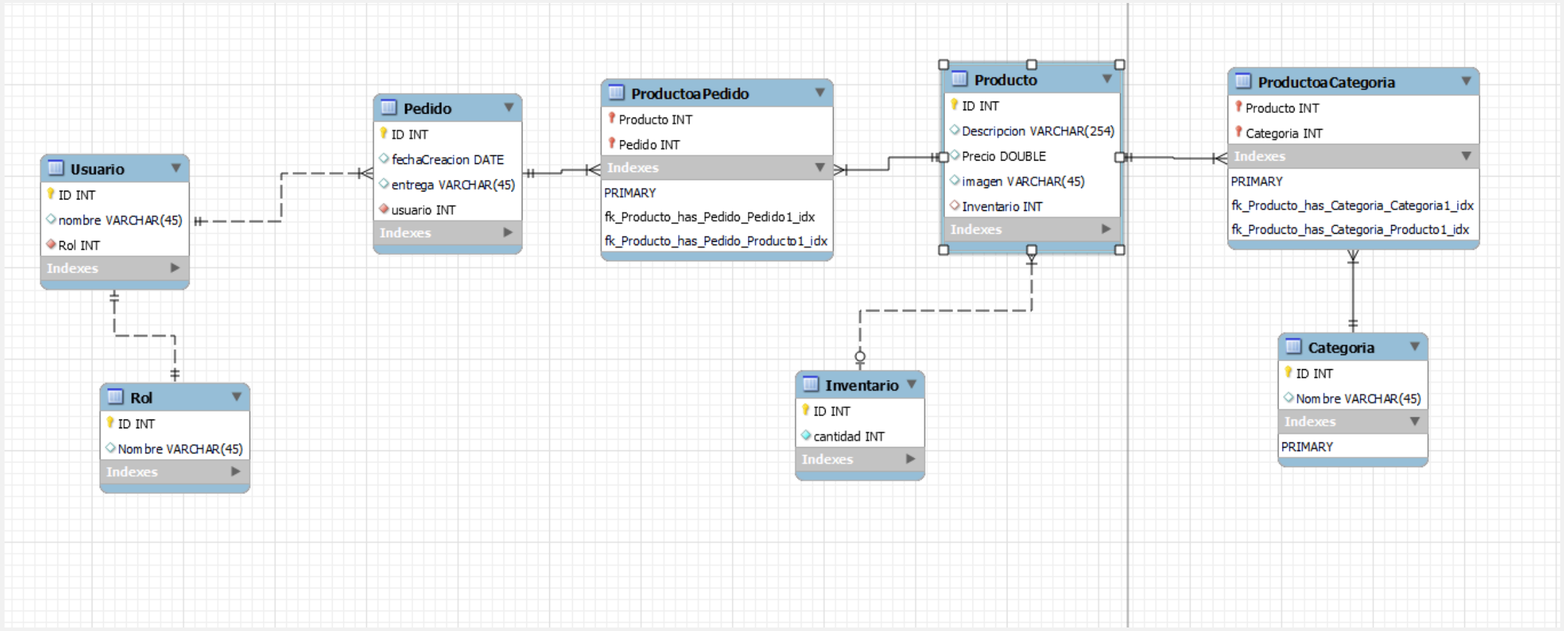
2- CONCRETAR EL PROYECTO: LENGUAJE Y DESARROLLO

- Para el desarrollo hemos nos hemos centro en un lenguaje que conocíamos.
- Como es java, el entorno hemos escogido intelliJ.
- Y para lo desconocido, hemos usado Spring boot

3- ARQUITECTURA SOFTWARE UTILIZADA MVC

HEMOS UTILIZADO JAVA SPRINT PARA EL DESARROLLO DEL PROYECTO QUE ES UN MODELO VISTA CONTROLADOR (MVC) DONDE LA VISTA QUE TIENE EL USUARIO DEL PROYECTO ES INDEPENDIENTE DEL RESTO DEL CODIGO.

4- BBDD MYSQL




```
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `theenglishcut` DEFAULT CHARACTER SET
USE `theenglishcut` ;

-- -----------------------------------------------------
-- Table `theenglishcut`.`Inventario`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `theenglishcut`.`Inventario` (
  `ID` INT NOT NULL,
  `cantidad` INT NOT NULL,
  PRIMARY KEY (`ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `theenglishcut`.`Producto`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `theenglishcut`.`Producto` (
  `ID` INT NOT NULL AUTO_INCREMENT,
  `Descripcion` VARCHAR(45) NULL,
  `Precio` DOUBLE NULL,
  `imagen` VARCHAR(45) NULL,
  `Inventario` INT NULL,
  PRIMARY KEY (`ID`),
  INDEX `fk_Producto_Inventario1_idx` (`Inventario` ASC) VISIBLE,
  CONSTRAINT `fk_Producto_Inventario1`
    FOREIGN KEY (`Inventario`)
    REFERENCES `theenglishcut`.`Inventario` (`ID`)
```

El modelo, genera un script para las tablas.

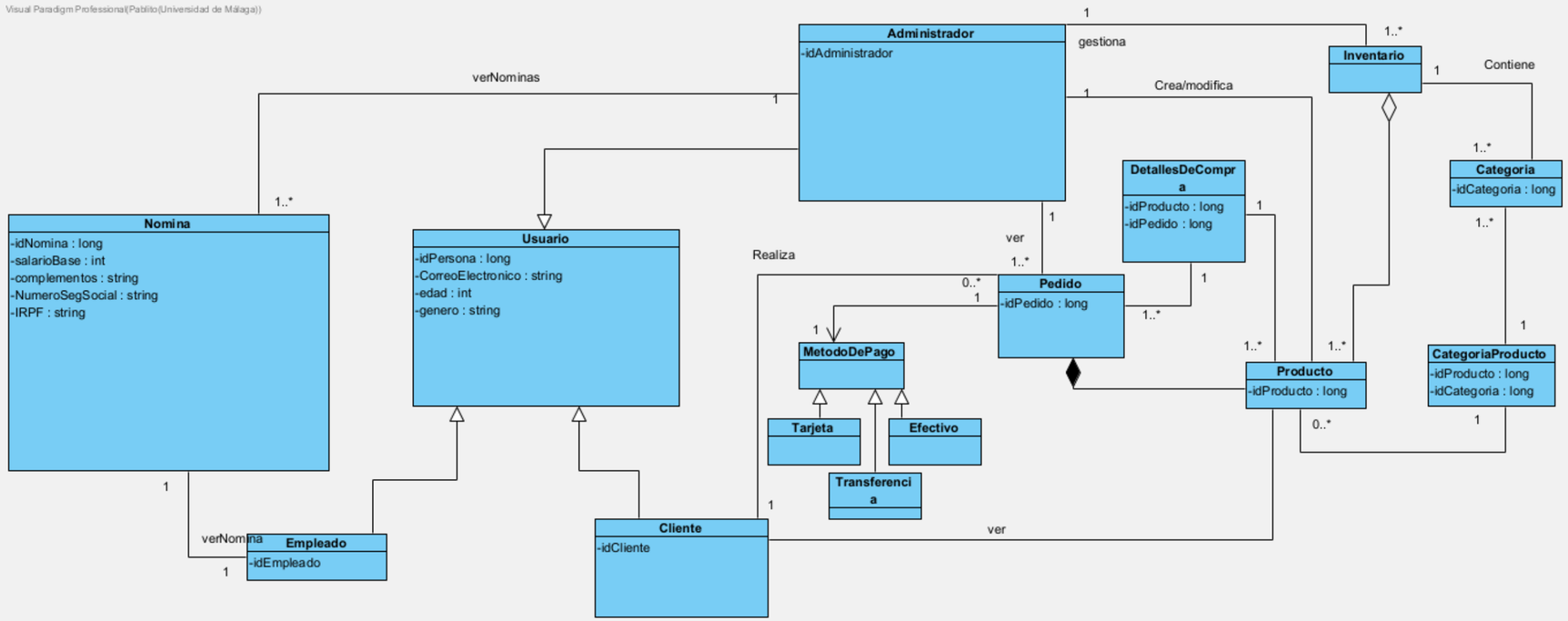
Que mas tarde nos genera las entidades con ayuda de JPA.

MODELO DE DOMINIO

- USUARIO
- ADMINISTRADOR
- EMPLEADO
- NÓMINA
- CLIENTE

- INVENTARIO
- CATEGORÍA
- CATEGORÍA-PRODUCTO
- PRODUCTO
- PEDIDO
- DETALLES DE COMPRA

- MÉTODO DE PAGO
- TARJETA
- TRANSFERENCIA
- EFECTIVO



CASOS DE USO

GESTIÓN DE USUARIOS

- Registrar usuario
- Registrarse como usuario administrador
- Registrarse como empleado
- Mantener la sesión
- Actualizar la contraseña
- Añadir información
- Visualizar un listado de usuarios con cuenta en la página
- Mantener la sesión

GESTIÓN DE PRODUCTOS

- Crear productos
- Eliminar productos
- Modificar productos
- Ver productos

GESTIÓN DE PEDIDOS

- Añadir productos al carrito
- Realizar pedidos
- Ver pedidos
- Eliminar productos del carrito
- Aplicar código de descuento

NAVEGACIÓN POR LA WEB

- Ver categorías
- Ver ofertas
- Ver carrito
- Buscar por nombre
- Ver información de contacto

CASOS DE USO

GESTIÓN DE EMPLEADOS

- Acceder a listado de empleados y sus nóminas
- Añadir, eliminar o modificar empleados
- Gestionar la nómina de los empleados
- Visualizar nómina
- Añadir o modificar horario empleados
- Añadir o modificar los horarios de empleados
- Visualizar horarios de empleados

MODIFICACIÓN DE LA INTERFAZ

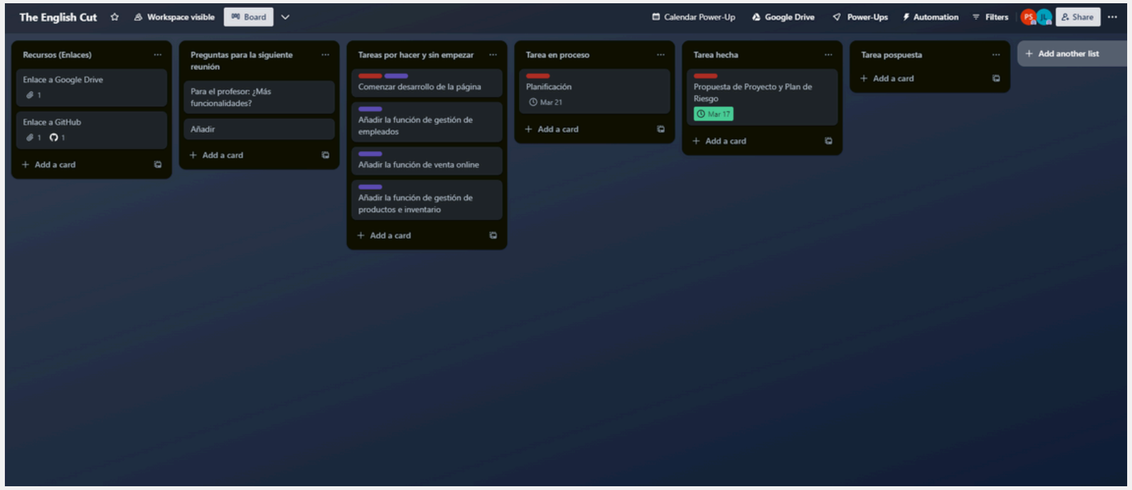
- Cambiar el color de la interfaz
- Cambiar color de la interfaz a una hora específica
- Aplicar un filtro para los productos de la página
- Botón de ayuda
- Support para problemas del sistema

GESTIÓN DEL PAGO

- Realizar un pago
- Guardar una tarjeta
- Devolución de un pago

5- DISTRIBUCION DE LOS CASOS DE USO

TRELLO PARA GESTION DE REQUISITOS



NOS HEMOS DIVIDIDO LOS CASOS DE USO POR EL GRADO DE DIFICULTAD, EL TIEMPO QUE SE INVIERTE EN CADA UNO Y POR EL CONOCIMIENTO DEL LENGUAJE

ORGANIZACION DE LOS CASOS DE USO:

1- ELIMINAR PRODUCTO -> PABLO GALVEZ
2- AÑADIR PRODUCTO -> DAVID LUQUE
3- HACER LOGIN -> PABLO ORTEGA
4- HACER REGISTER -> JANINE OLEGARIO
5- VER PEDIDOS -> JONATAN THORPE
6- VER CESTA-> MARCOS DIAZ
7- FILTRO -> ALEJANDRO LÓPEZ

CREAR PRODUCTOS

¿Quién lo puede realizar? Solamente el administrador podrá añadir nuevos productos

Secuencia de acciones:

- En la pantalla de inicio de la pagina web, pulsamos en el **controlador** de **crear producto**
- El sistema nos **redirigira al apartado** de crear un producto nuevo
- En este apartado tenemos un **formulario** con la **información** necesaria que tendremos que **rellenar** para añadir un producto nuevo
- Una vez todos los apartados **completos** pulsamos el **controlador de crear**
- Se **añadirá** correctamente el producto en el programa y el **sistema** nos lo **notifica**

Posibles errores:

- Debido a un **fallo** con la **base de datos**, el sistema nos informa del error

MODELO

```java
public class Producto {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer ID;
    private String nombre;  2 usages
    private String descripcion;  2 usages
    private Double precio;  2 usages
    private String imagen;  2 usages
    @ManyToOne  2 usages
    @JoinColumn(name = "Inventario")
    private Inventario inventario;
    @OneToMany(mappedBy = "producto")  2 usages
    private List<ProductoaPedido> pedidos;
    @OneToMany(mappedBy = "producto")  2 usages
    private List<ProductoaCategoria> categorias;
    public Integer getID() { return ID; }
    public void setID(Integer ID) { this.ID = ID; }
    public String getDescripcion() { return descripcion; }
    public void setDescripcion(String descripcion) { this.descripcion = descripcion; }
    public Double getPrecio() { return precio; }
    public void setPrecio(Double precio) { this.precio = precio; }
    public String getImagen() { return imagen; }
    public void setImagen(String imagen) { this.imagen = imagen; }
    public Inventario getInventario() {  2 usages  JonniThorpe
        return inventario;
    }
    public void setInventario(Inventario inventario) { this.inventario = inventario; }
    public List<ProductoaPedido> getPedidos() { return pedidos; }
    public void setPedidos(List<ProductoaPedido> pedidos) { this.pedidos = pedidos; }
    public List<ProductoaCategoria> getCategorias() { return categorias; }
    public void setCategorias(List<ProductoaCategoria> categorias) { this.categorias = categorias; }
    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }
```

VISTA

```jsp
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ taglib prefix="from" uri="http://www.springframework.org/tags/form" %>
<%@ page import="com.gt.theenglishcut.ui.producto" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%

    producto producto = (com.gt.theenglishcut.ui.producto) request.getAttribute("producto");
    int cantidadProducto = (int) request.getAttribute("cantidadProducto");

%>
<html>
<head>
    <title></title>
</head>
<body>
<%@ include file = "../componentes/Navbar.jsp" %>
<h1>Crear producto</h1>
<div class="container-sm">
    <form:form action="/guardarProducto" method="post" modelAttribute="producto">
        <div class="mb-3">
            <label >Nombre del producto</label>
            <from:input path="nombre" class="col-md-8 form-control input-md"/>
        </div>
        <div class="mb-3">
            <label>Descripcion del producto</label>
            <form:textarea path="descripcion" class="form-control" rows="3"></form:textarea>
        </div>
        <div class="mb-3">
            <label>Precio del producto</label>
            <form:input path="precio" class="col-md-8 form-control input-md" />
        </div>
        <div class="form-group mb-3">
            <label for="exampleFormControlSelect1">Cantidad</label>
            <form:select path="cantidad" class="form-control" id="exampleFormControlSelect1">
                <%
                    for (int i = 0; i < 100; i++) {
                    %>
                <option value="<%=i%>"><%=i%></option>
                        <%
                    }%>
            </form:select>
        </div>
        <div class="form-group mb-3">
            <label >imagen</label>
            <form:input path="imagen" type="text" class="form-control" />
        </div>
        <div class="input-group flex-nowrap mb-3">
            <button type="submit" class="btn btn-primary ">Enviar</button>
        </div>
        <a href="home.jsp" class="btn btn-primary ">Volver</a>
    </form:form>
</div>
</body>
</html>
```

CONTROLADOR

```java
@GetMapping("/CrearProducto")  JonniThorpe
public String crearProducto (Model modelo) {
    modelo.addAttribute( attributeName: "producto",  new producto());
    modelo.addAttribute( attributeName: "cantidadProducto", attributeValue: 6);
    return "/CrearProducto";
}

@PostMapping("/guardarProducto")  JonniThorpe
public String guardarProducto (Model modelo, @ModelAttribute("producto")producto producto) {
    Producto productoNuevo = new Producto();
    Inventario inventario = new Inventario();

    inventario.setCantidad(producto.getCantidad());
    inventarioRepository.save(inventario);

    productoNuevo.setInventario(inventario);
    productoNuevo.setNombre(producto.getNombre());
    productoNuevo.setDescripcion(producto.getDescripcion());
    productoNuevo.setImagen(producto.getImagen());
    productoNuevo.setPrecio(producto.getPrecio());

    productoRepository.save(productoNuevo);
    return "redirect:/listadoProductos?Categoria="+categoriaGlobal;
}
```

VER PEDIDOS 👀

¿Quién lo puede realizar? Tanto el **administrador** como el **usuario** (que se hayan registrado con anterioridad)

Secuencia de acciones:

- En la pantalla de inicio, se seleccionara el controlador de **ver pedidos**
- Una vez presionado se **redirigirá** al apartado de ver pedidos
- Aquí se mostraran el **conjunto** de **pedidos realizados**
- Si es **administrador**, además podrá **elegir** el **orden** de los mismos, por **identificador** o por **fecha de realización**

Posibles errores:

- El usuario **nunca ha realizado ningún pedido** por lo que se informara que esta vacío el apartado
- Hay **problemas** con la conexión a **Internet**

MODELO

```java
public class Pedido {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer ID;
    @Temporal(TemporalType.DATE)  2 usages
    private Date fechaCreacion;
    private String entrega;  2 usages
    @ManyToOne  2 usages
    @JoinColumn(name = "usuario")
    private Usuario usuario;
    @OneToMany(mappedBy = "pedido")  2 usages
    private List<ProductoaPedido> productos;
    public List<ProductoaPedido> getProductos() { return productos; }
    public void setProductos(List<ProductoaPedido> productos) { this.productos = productos; }
    public Usuario getUsuario() { return usuario; }
    public void setUsuario(Usuario usuario) { this.usuario = usuario; }
    public String getEntrega() { return entrega; }
    public void setEntrega(String entrega) { this.entrega = entrega; }
    public Date getFechaCreacion() { return fechaCreacion; }
    public void setFechaCreacion(Date fechaCreacion) { this.fechaCreacion = fechaCreacion; }
    public Integer getID() { return ID; }
    public void setID(Integer ID) { this.ID = ID; }
    // getters and setters
}
```

CONTROLADOR

```java
public String confirmar_Producto_a_Pedido (HttpSession sesion) {
    String user = (String) sesion.getAttribute( s: "user");
    if (user == null) {
        // Manejar el caso en que el usuario no esté autenticado
        return "redirect:/login"; // Redirigir a la página de inicio de sesión
    }
    Usuario clientePedido = usuarioRepository.findByNombreUser(user);
    List<ProductoaPedido> productoaPedidoLista = new ArrayList<>();
    //Creamos el pedido
    pedidoCliente.setUsuario(clientePedido);
    pedidoCliente.setFechaCreacion(new Date());
    pedidoCliente.setEntrega("NO CONFIRMADO");
    pedidoRepository.save(pedidoCliente);
    for(Producto productoConfirmado:productosCarrito.keySet()){
        ProductoaPedido productoaPedido = new ProductoaPedido();
        productoaPedido.setProducto(productoConfirmado);
        productoaPedido.setPedido(pedidoCliente);
        PedidoaProductoRepository.save(productoaPedido);
        productoaPedidoLista.add(productoaPedido);
        productoConfirmado.setPedidos(productoaPedidoLista);
        productoRepository.save(productoConfirmado);
    }
    pedidoCliente.setProductos(productoaPedidoLista);
    pedidoRepository.save(pedidoCliente);
    //Limpiamos el carrito para nuevos productos y nuevo pedido
    productosCarrito.clear();
    pedidoCliente = new Pedido();
    return "redirect:/listadoProductos?Categoria="+categoriaGlobal;
}
```

VISTA

```jsp
<%@ page import="com.gt.theenglishcut.entity.Producto" %>
<%@ page import="java.util.*" %>
<%--
  Created by IntelliJ IDEA.
  User: Jonni
  Date: 21/05/2024
  Time: 19:47
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%

Map<Producto, Integer> productosCarrito = (Map<Producto, Integer>) request
%>
<html>
<head>
    <title>Title</title>
</head>
<!--
TODO Stackear producto en cantidad y precio si son el mismo tipo de produc
El listado de productos muestra cada producto de la lista productosCarrito
es necesario que aquel producto que este en la lista varias veces se muest
donde aparezca ese mismo producto y el numero de veces que aparece en el c
-->
<body>
```

```jsp
<body>
<%@ include file = "../componentes/Navbar.jsp" %>
    <form method="post" action="/confirmarPedido">

        <div class="container-fluid">
            <table class="table">
                <tr>
                    <th></th>
                    <th class="col">Nombre</th>
                    <th class="col">Precio/Unidad</th>
                    <th class="col">Número Unidades</th>
                    <th class="col">Precio Total</th>
                </tr>
                <%
                double total = 0;
                int num_productos;
                    for(Producto producto: productosCarrito.keySet()){
                        num_productos=productosCarrito.get(producto);
                double precio = producto.getPrecio();
                double precio_total=precio * num_productos;
                %>
                <tr>
                    <td><img src="../../img/<%=producto.getImagen()%>" width="64
                    <td><%=producto.getNombre()%></td>
                    <td><%=producto.getPrecio()%>€</td>
                    <td><%=num_productos%> uds</td>
                    <td><%=precio_total%></td>
                </tr>
                  <tr>
                      <td colspan="2">Total</td>
                      <td ><%=total%>€</td>
                  </tr>
            </table>
            <button type="submit" class="btn btn-primary">Realizar pedido </button>
        </div>
    </form>
</body>
</html>
```

LECCIONES APRENDIDAS

TRABAJO EN EQUIPO	SCRUM	TRABAJAR CON GIT
COMUNICACIÓN	JAVA SPRING	ORGANIZACIÓN CON DIVERSAS HERRAMIENTAS
DESARROLLO DE UNA APP WEB	USO DE FRAMEWORK	MANEJO DE BASE DE DATOS

MUCHAS GRACIAS