

Trabajo Final de Grado

Pablo Perea Faubel

Tutor: Enrique Tortajada

Curso 2021 – 2022

Desarrollo de
Aplicaciones Web

IES La Vereda



FAMILIA INFORMÁTICA Y COMUNICACIONES



Gestor para Pequeñas tiendas

Pablo Perea Faubel

Índice de contenidos

1. Introducción
 - 1.1. Resumen del contenido
 - 1.2. Descripción del proyecto
 - 1.3. Módulos que implica
2. Estudio previo
 - 2.1. Estudio de soluciones existentes
3. Plan de Trabajo
4. Diseño
 - 4.1. Diseño general
 - 4.2. Diseño detallado
5. Recursos
6. Conclusiones
 - 6.1. Grado de consecución de objetivos
 - 6.2. Problemas encontrados
 - 6.3. Mejoras
7. Manual de Usuario
8. Explicación contenidos GitHub.
9. Webgrafía
10. Índice de imágenes

1. Introducción

1.1. Resumen del Contenido

En esta memoria muestro el trabajo realizado para el proyecto de fin del grado de Desarrollo de Aplicaciones Web y su manual para el usuario.

1.2. Descripción del Proyecto

El proyecto es una Aplicación Web para llevar la gestión de una tienda pequeña, como podría ser una ferretería, una tienda de artículos deportivos o una tienda de libros. Para este ejemplo, he elegido una tienda de componentes de ordenador.

La aplicación cuenta con funciones para Añadir artículos, editar los mismos, añadir stock, imprimir ticket, registrar usuarios (serán los trabajadores de la tienda), ver ticket y un apartado accesible para el administrador de la tienda en el que podrá ver estadísticas de la tienda.

1.3. Módulos que implica

Para desarrollar la aplicación, he tenido que utilizar los conocimientos aprendidos en los siguientes módulos.

- Bases de Datos: Saber gestionar la base de datos de la aplicación.
- Diseño de interfaces: Hacer interfaz gráfica incluyendo que esta sea responsive.
- Lenguaje de Marcas: Tener una estructura de la web correcta.

- Desarrollo Web en Entorno Servidor: Desarrollar el funcionamiento interno de la aplicación.
- Desarrollo Web en Entorno Cliente: Desarrollar el funcionamiento de las vistas y comunicar con el funcionamiento interno (Comunicar PHP con JS, Jquery) para cubrir unas necesidades de funcionamiento del programa.
- Despliegue de aplicaciones: Desplegar la aplicación en un servidor para que sea accesible desde fuera de mi red local.
- Entornos de desarrollo: Saber en qué entorno me siento más cómodo para desarrollar mi aplicación.

2. Estudio previo

He estado estudiando el funcionamiento de una tienda que acaba de abrir, cercana a mi casa, y me he dado cuenta de que no tienen una aplicación para gestionar la tienda, por lo tanto, suceden los siguientes problemas:

- No hay buena gestión del stock
- Total del precio calculado con calculadora
- Tickets generados con una aplicación online en la que tienen que introducir todos los datos manualmente.
- Para ver las estadísticas de la tienda tienen que contar los tickets y los totales de forma manual.

2.1 Estudio de soluciones existentes

Diferentes aplicaciones de Escritorio para la gestión. Al ser de escritorio necesitan más potencia de procesamiento y también, las existentes actualmente tienen un diseño algo anticuado y no tienes posibilidad de ver las estadísticas.

3. Plan de Trabajo



En la creación de la estructura de la Base de Datos realicé el Entidad Relación y el Diagrama de Clases

Entidad Relación:

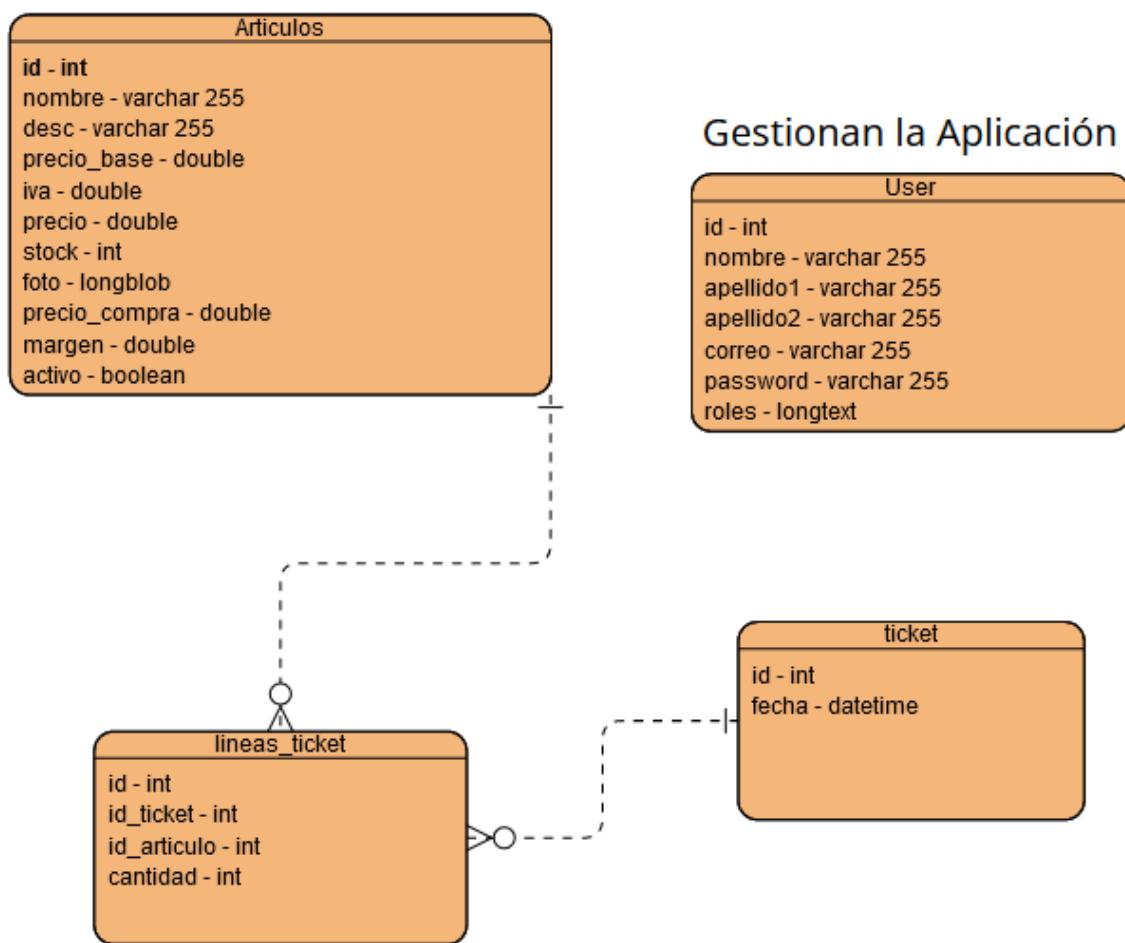


Ilustración 1: Entidad Relación

Diagrama de Clases:

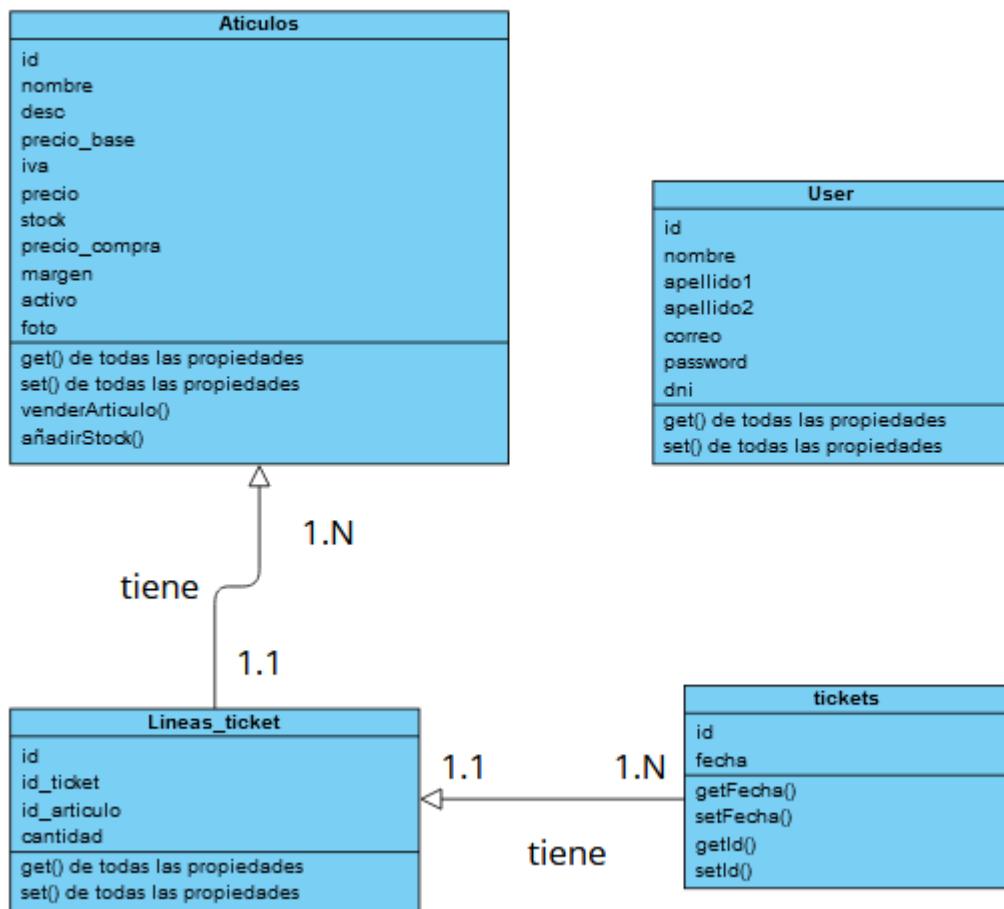


Ilustración 2: Diagrama de clases

4. Diseño

4.1. Diseño general

Una idea sobre cómo van a ser las vistas de la aplicación

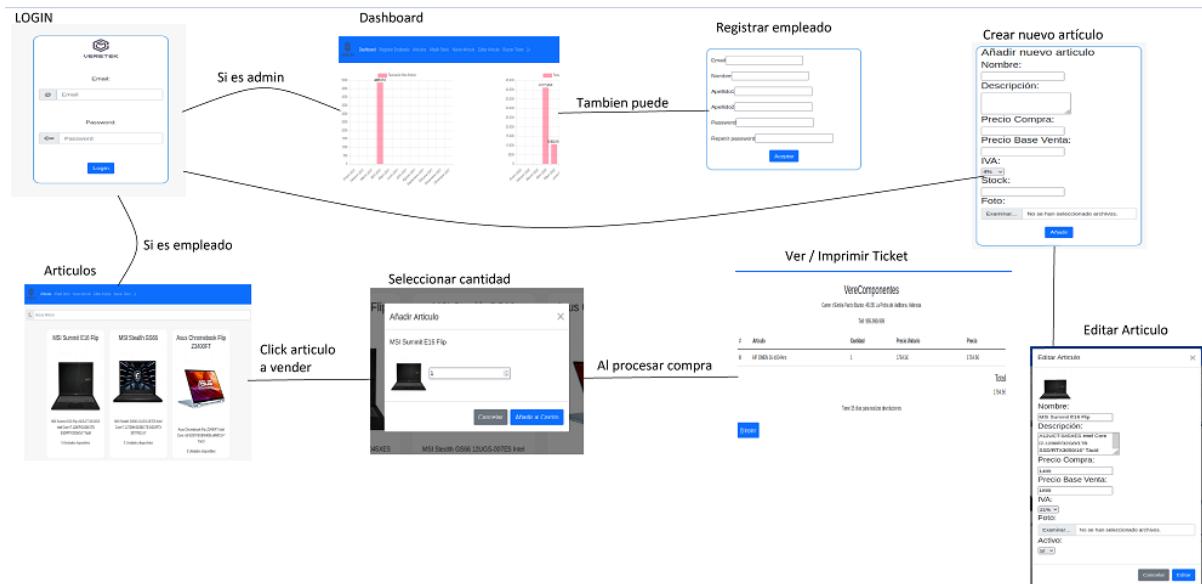


Ilustración 3: Mockup

4.2. Diseño detallado

Pequeña explicación sobre las vistas mas importantes

La primera vista que se le mostrará al usuario es la pantalla de inicio de sesión, ya que si no está logueado no puede realizar ninguna acción.

Vista simple, con el logo de la empresa y los campos de autenticación.

En caso de fallo de autenticación, se mostrará un mensaje de error en la parte inferior.

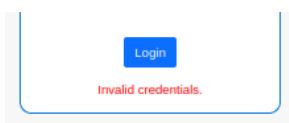


Ilustración 5: Error Login

A detailed view of the login screen. It features the 'VERETEK' logo at the top. Below it is a 'Email:' field with a placeholder 'Email' and a small icon. Underneath is a 'Password:' field with a placeholder 'Password' and a lock icon. To the right of the password field is a 'Login' button. The entire form is enclosed in a light gray rounded rectangle.

Ilustración 4: Login

```
/**
 * @Route("/login", name="app_login")
 */
public function login(AuthenticationUtils $authenticationUtils): Response
{
    $error = $authenticationUtils->getLastAuthenticationError();
    $lastUsername = $authenticationUtils->getLastUsername();

    return $this->render('gestor_tienda/login.html.twig', [
        'controller_name' => 'GestorTiendaController',
        'error' => $error,
        'last_username' => $lastUsername,
    ]);
}
```

Ilustración 6: Código Login

Una de las vistas a las que únicamente puede acceder el administrador es el Dashboard. Este será su inicio.

En este, se muestran datos relacionados con la facturación de la empresa y los productos más vendidos.

En esta vista, obtengo los datos haciendo llamadas AJAX al controlador, donde obtengo los datos necesarios para cada gráfico.

Una vez obtenidos los datos, en el caso de la facturación, ya que se muestra por meses, he desarrollado una función que separa los tickets por meses dependiendo su fecha y suma el valor total de ellos en cada mes.

```
for (var i = 0; i < result.length; i++) {
    for (var j = 0; j < etiquetasVisibles.length; j++) {

        var fechaComprobacion1 = new Date(result[i].fecha.date.split('-')[0], result[i].fecha.date.split('-')[1], '01');
        var fechaComprobacion2 = new Date(etiquetasVisibles[j].split(' ')[1], comprobarMes[j], '01');

        if (fechaComprobacion1.getTime() == fechaComprobacion2.getTime()) {
            datos[j] += result[i].total;
        }
        else {
            break;
        }
    }
    return [etiquetasVisibles, datos];
}
```

Ilustración 8: Código Dashboard

La autenticación se comprueba con las herramientas que trae Symfony y si es correcto se redirige al index, que dependiendo del rol será uno diferente.



Ilustración 7: Dashboard

Para poder separar los tickets en dicha función, he tenido que formatear las fechas que obtengo desde la base de datos en el mismo formato que cuando se crea una nueva fecha de JavaScript, para comprobar si el mes es el mismo y añadir el precio al total o no.

Formulario de alta de empleado:

- Email:
- Nombre:
- Apellido1:
- Apellido2:
- Password:
- Repetir password:

Ilustración 9: Registro

La segunda y última vista a la que solo puede acceder el Administrador, es una vista muy sencilla en la cual dará de alta a los empleados que tenga en la tienda .

Tendrá que asignar una contraseña al empleado, que luego podrá ser cambiada por el mismo.

Desde el controlador, recojo los datos del formulario, y compruebo que las contraseñas sean iguales. En caso de que sean diferentes se devuelve un error.

Si son iguales, se encripta la contraseña y se crea el usuario.

```
if ($form->isSubmitted() && $form->isValid()) {

    if (strcmp(strval($form->get('password'))->getData(),
        $rol = ['ROLE_USER'];
        $user->setRoles($rol);
        $hashedPassword = $PasswordHasher->hashPassword(
            $user,
            $form->get('password')->getData()
        );
        $user->setPassword($hashedPassword);
    }
}
```

Ilustración 10: Código registro

Vista de Artículos.

Aquí el usuario verá todos los artículos disponibles, pudiendo buscarlos por nombre y será la pantalla donde se vayan escogiendo los artículos y la cantidad del mismo que va a comprar un cliente.

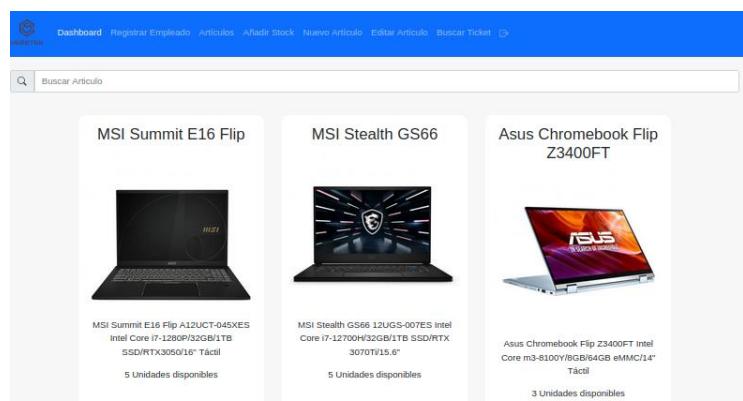


Ilustración 11: Artículos

Se envian los artículos a la vista y se crea el mostrado de estos mediante un bucle que va recorriendo todos los datos y comprueba si el artículo está activo o descatalogado

```

{# for key ,árticulo in artículos #}
{# if artículo.activo == 1 #}
<div class='artículo col-md-3 fondoBlanco m-3 text-center bordesRedondos' id='artículo-{{artículo.id}}'>
    <h3 class='nombre-{{artículo.id}} mt-3'>{{artículo.nombre}}</h3>
    
    <p>{{artículo.descripcion}}</p>
    <p class='stock-{{artículo.id}}'>
        {{artículo.stock}} Unidades disponibles</p>
</div>
{# endif #}
{# endfor #}

```

Ilustración 12: Código Artículos

Para añadir stock, se hará desde una ventana modal, que se abre a través de un artículo seleccionado.



Ilustración 13: Stock

En esta ventana únicamente se indicará cuantas unidades queremos añadir al stock del artículo.

Una vez presionamos el botón de añadir stock, se envían los datos a través de AJAX, se recupera el artículo al que vamos a actualizar el stock mediante su id y se hace la actualización.

```

$id = $_POST['id'];
$stock = $_POST['stock'];
$array = $em->getRepository(Articulos::class)->findBy(array('id' => $id));
$artículo = $array[0];

$artículo->addStock($stock);
$em->persist($artículo);
$em->flush($artículo);

```

Ilustración 15: Código Stock

El apartado de editar artículo tiene un funcionamiento interno muy similar al de actualizar stock, pero actualizando todos los datos del artículo.

Al seleccionar el artículo a editar, se obtienen todos los datos de este a través de otra llamada AJAX para poder llenar los campos del formulario con los datos correctos.

Esto facilita el código a la hora de hacer la actualización ya que los campos no quieren modificar el empleado se quedarán igual que están sin tener que realizar muchas más comprobaciones.

Ilustración 14: Editar Artículo

5. Recursos

Los recursos Hardware necesarios para el desarrollo de la aplicación no han sido muchos. Unicamente he necesitado un ordenador, con una potencia de procesación mas bien media baja, y como mínimo 4GB de ram, para poder lanzar una máquina virtual con al menos 2GB de ram y trabajar sobre ella.

El sistema operativo utilizado en la máquina real es Windows, pero el de la máquina virtual es Linux Mint, por lo tanto el sistema utilizado para el desarrollo ha sido Linux mint.

El software utilizado para este desarrollo ha sido el siguiente:

- XAMPP: Para alojar la Base de Datos en mi propia máquina y tener un servidor web local.
- Terminal: Para lanzar los comandos de creación de la base de datos, las entidades y formularios que incluye el framework Symfony.
- Visual Studio Code: Entorno de desarrollo en el que he ido creando el código y las vistas. En él, tengo diferentes extensiones para facilitar el uso de este IDE. Algunos de ellos son: HTML,JS Text Formatter, SASS live, PHP intelephense y PHP extension Pack.

6. Conclusiones

6.1. Grado de consecución de objetivos

He consigo crear el funcionamiento básico para gestionar un tienda pequeña que no tiene que registrar clientes, aunque me hubiera gustado con un poco más de tiempo ir haciendo más completa la aplicación añadiendo funcionalidades útiles que normalmente no estén en programas de gestión de tiendas

6.2. Problemas encontrados

El planteamiento, en algunos casos no ha sido el correcto ya que a la hora de desarrollar no se podía seguir las ideas en mente por no saber o ser viable desarrollarlas.

En cuanto al código, he tenido errores que me han hecho perder bastante tiempo ya que he querido utilizar herramientas no utilizadas en clase y cuando me salía un error me ha costado saber resolverlo buscando por foros y documentación oficial.

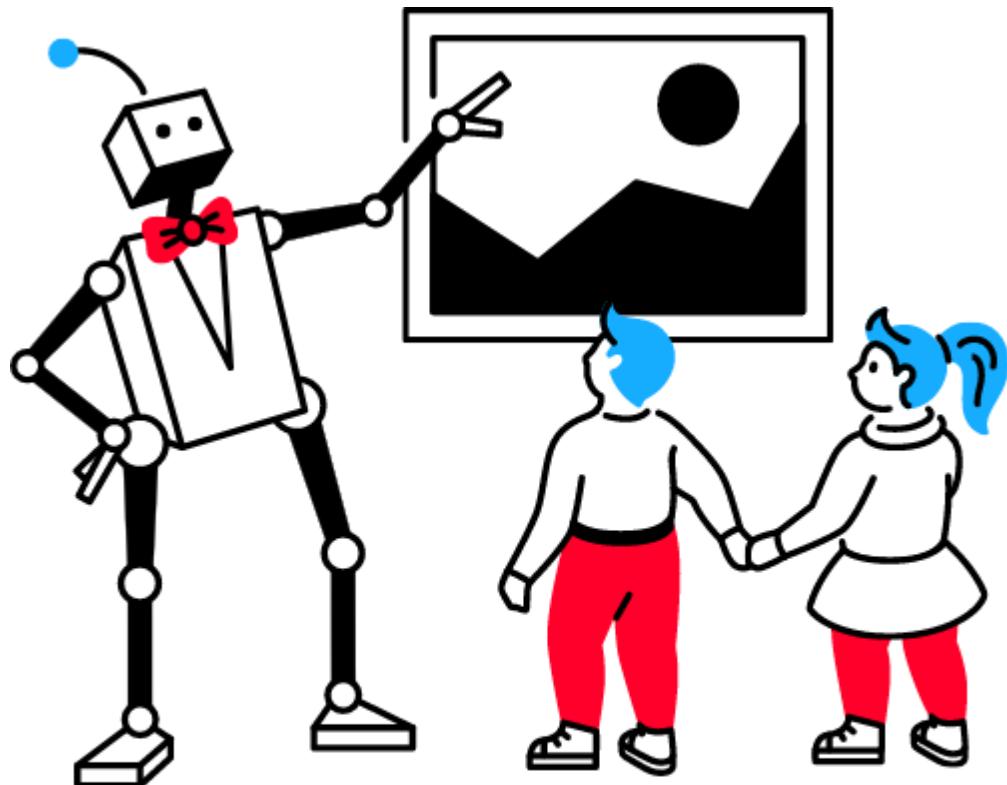
6.3. Mejoras

En caso de que la aplicación se comercializase, habría que aplicar un serie de mejoras, como por ejemplo implementar un módulo para la impresión de tickets que diera más herramientas de edición y hacer que se viera más profesional.

También debería implementar el funcionamiento de quitar un artículo antes de realizar el pago por si el cliente decide no comprar un artículo cuando ya lo hemos incluido en el carro.

Si tuviera un cliente real para la Aplicación, también se podrían ir añadiendo funcionalidades a medida dependiendo de las necesidades del cliente y su tienda.

7. Manual de Usuario



Manual de Usuario

1. INTRODUCCIÓN

La aplicación Gestor de Tiendas, está pensada y diseñada para facilitar a los administradores de una tienda pequeña, los mecanismos para poder gestionarla correctamente, teniendo los datos actualizados y permitiendo el registro de todos los artículos de la tienda.

Actualmente existen dos roles para acceder a la aplicación: Usuario, que será un empleado, y Administrador.

2. PORTAL DE ENTRADA

Para acceder a la aplicación, deberá entrar al acceso directo que el técnico le habrá puesto en el escritorio del ordenador en el que se va a gestionar la tienda, o en el enlace que le habrá facilitado.

Al acceder a la aplicación, encontrará la pantalla de Inicio de Sesión.

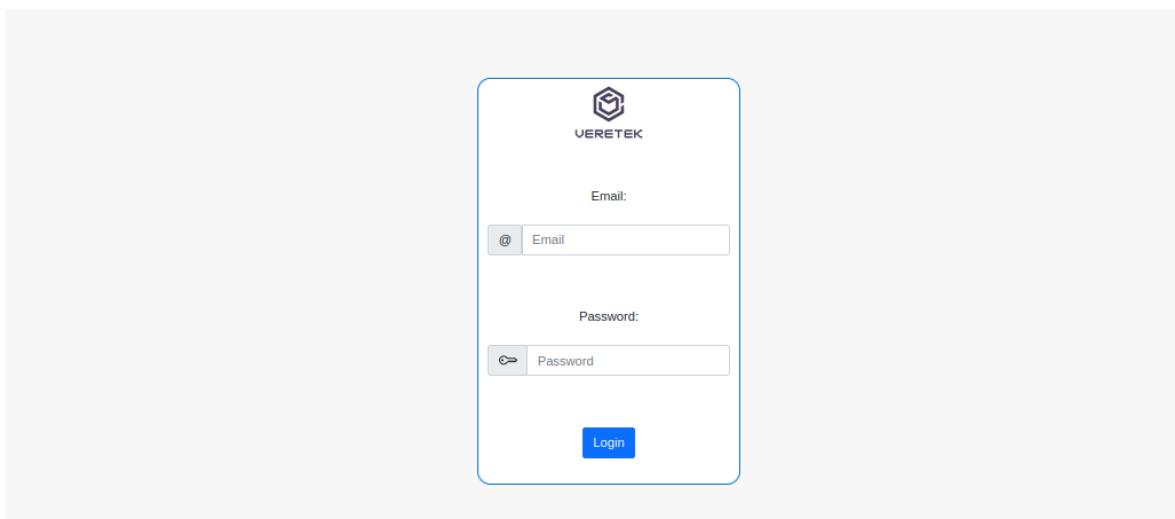


Ilustración 16: Login Manual

El Administrador de la tienda habrá sido registrado con el rol de Administrador comentado anteriormente. Para acceder a la pantalla de inicio tendrá que poner sus credenciales de acceso (email y contraseña).

Si es un trabajador, habrá sido registrado por el Administrador de la tienda.

3. INICIO

Una vez dentro de la aplicación, si el usuario es Administrador, se encontrará con el Dashboard, en caso de ser un usuario sin permisos, se encontrará con los artículos

4. DASHBOARD

En esta pantalla se muestran datos de la Facturación del año anterior, del año en curso y un TOP 10 de los artículos más vendidos.

El Dashboard es únicamente informativo y tiene acceso el Administrador.



Ilustración 17: Dashboard Manual

5. REGISTRO

Al igual que el Dashboard, la pantalla de registro es únicamente accesible para el Administrador, ya que nadie ajeno a la empresa puede registrarse en la aplicación.

Este será el que cree los usuarios que pueda necesitar en la tienda, en caso de que contrate a un dependiente.

Todos los campos de este formulario son obligatorios y deberá rellenarlos correctamente para poder completar el registro.

The screenshot shows a registration form titled 'Nuevo Artículo' (New Article) with a blue header bar containing the Veretek logo and navigation links: Dashboard, Registrar Empleado, Artículos, Añadir Stock, Nuevo Artículo, Editar Artículo, Buscar Ticket, and a search icon.

The form itself has a light gray background and a blue border. It contains six input fields labeled 'Email', 'Nombre', 'Apellido1', 'Apellido2', 'Password', and 'Repetir password'. Below these fields is a blue 'Aceptar' (Accept) button.

Ilustración 18: Registro Manual

6. ARTÍCULOS

En este apartado, es donde se muestran todos los artículos y donde se gestionaran las compras.

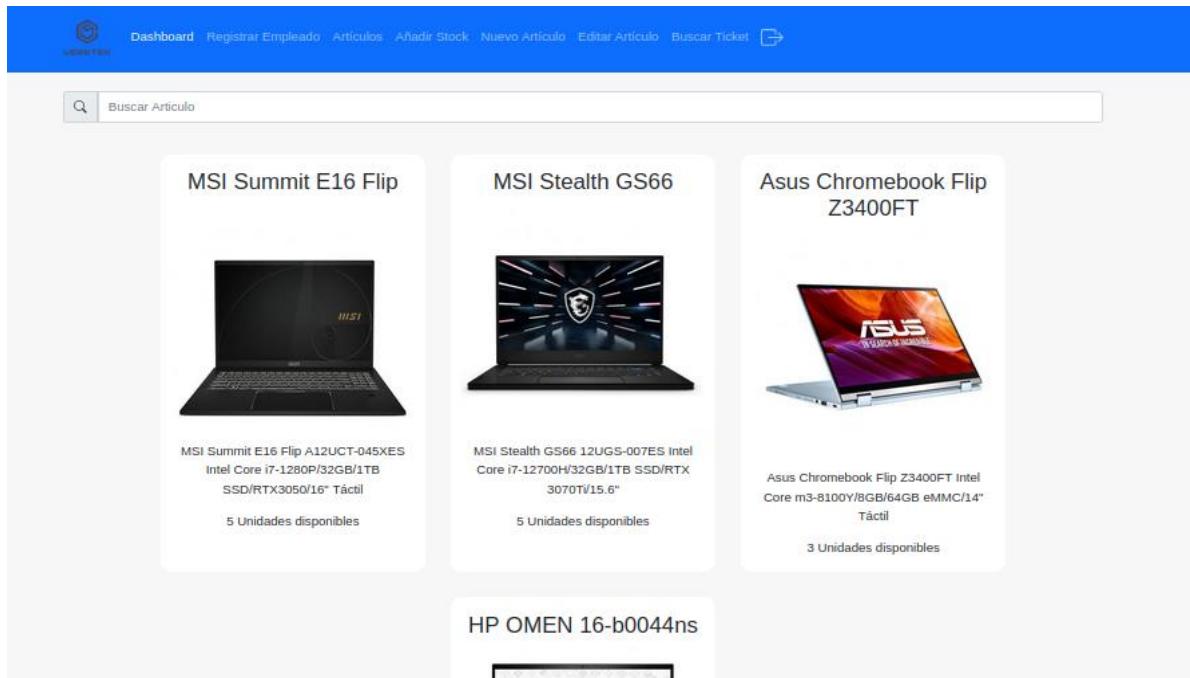


Ilustración 19: Artículos Manual

En la parte superior, disponemos de un buscador para poder encontrar el artículo más rápido. Este buscador encontrará coincidencias con los nombres de los artículos.

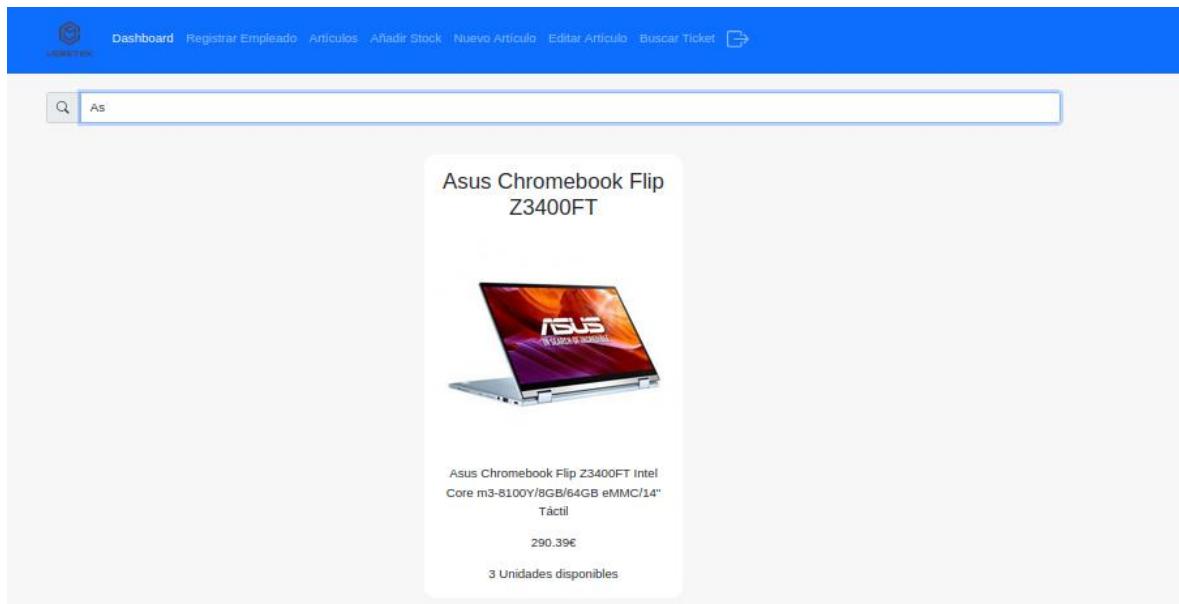


Ilustración 20: Buscador Manual

Para seleccionar el artículo que va a comprar el cliente, debemos hacer pulsar sobre ese artículo.

Al pulsar, se nos abrirá un POP-UP donde podremos seleccionar la cantidad de este mismo artículo que se va a comprar

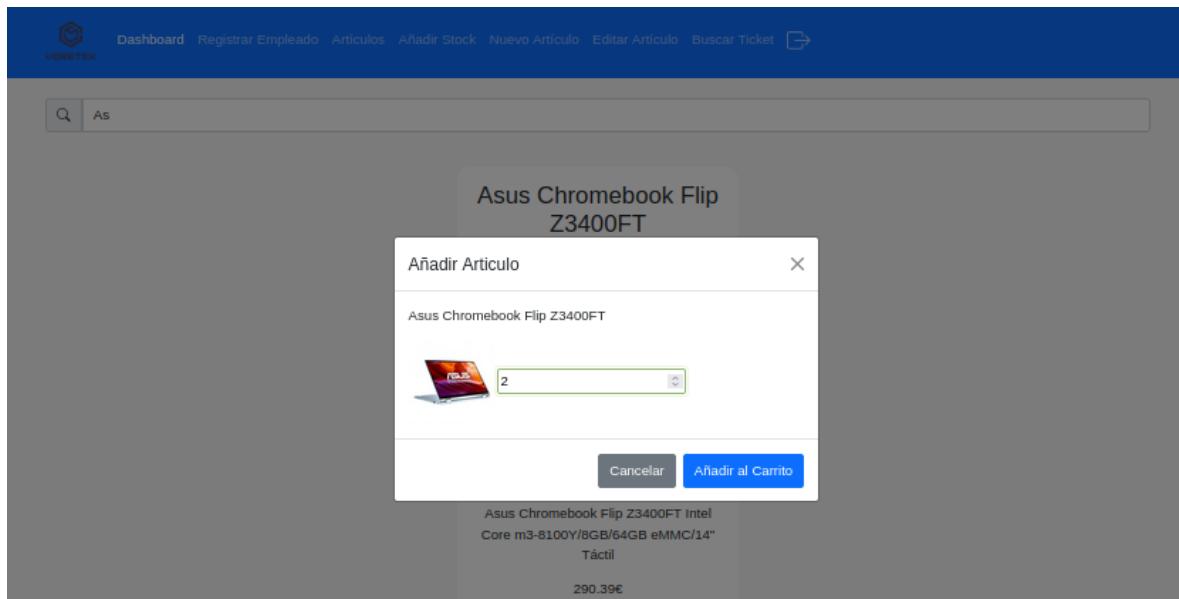


Ilustración 21: Stock Manual

Se selecciona la cantidad y lo añade al carrito.

En caso de seleccionar más stock del disponible, se mostrará un error y no añadirá el artículo al carrito.

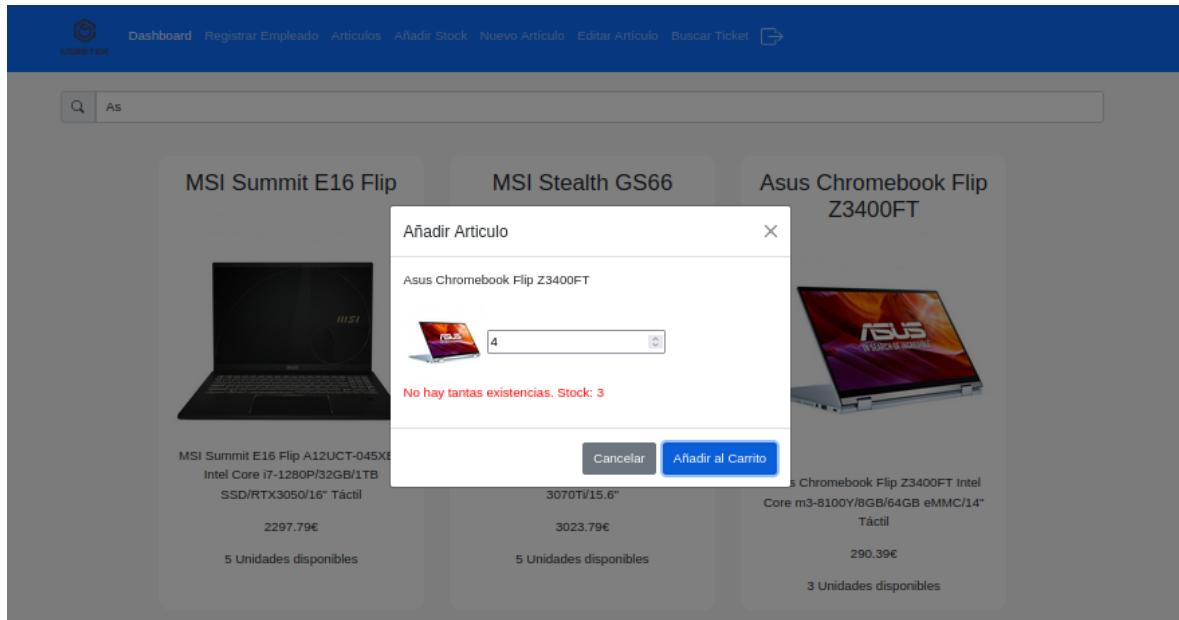


Ilustración 22: Error Stock Manual

Una vez que esté todo el carrito preparado, deberemos procesar la compra. Para ello, hay que pulsar el botón ‘Procesar Compra’ situado en la parte inferior de la pantalla.

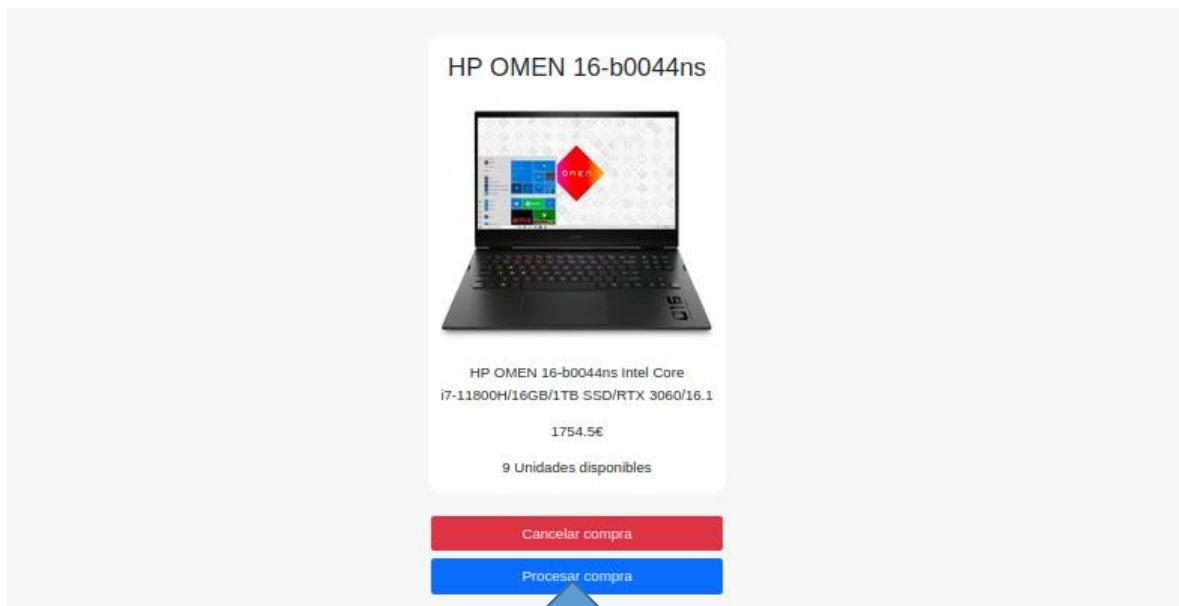


Ilustración 23: Procesar Compra Manual

Justo arriba del botón de ‘Procesar Compra’, se encuentra el de ‘Cancelar Compra’ que será utilizado en caso de que el cliente al final no vaya a realizar la compra o si se ha tenido un error al crear el carrito con los productos.

Una vez se haya finalizado el proceso de compra, se mostrará el ticket.

7. AÑADIR STOCK

Esta vista es igual que la de Artículos, pero al pulsar sobre uno, en vez de añadirlo al carrito, su función será añadir stock.

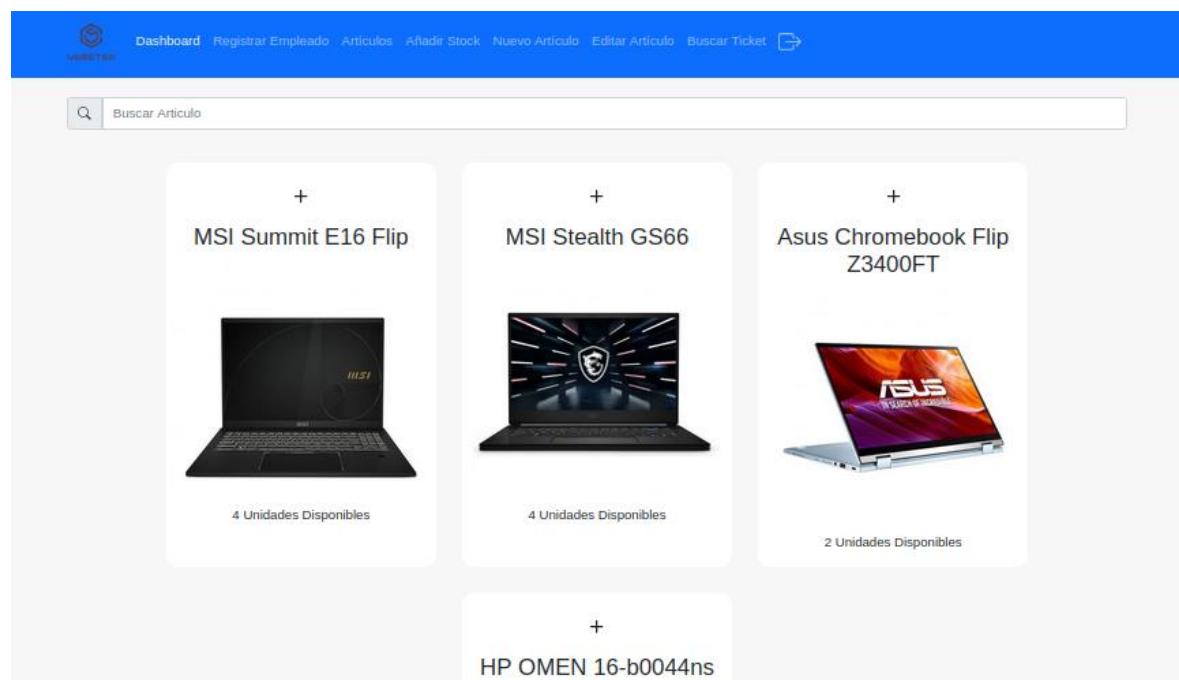


Ilustración 24: Añadir Stock Manual

Al seleccionar el artículo del cual se quiere añadir stock, se abrirá un POP-UP para que se pueda seleccionar la cantidad que se va a añadir

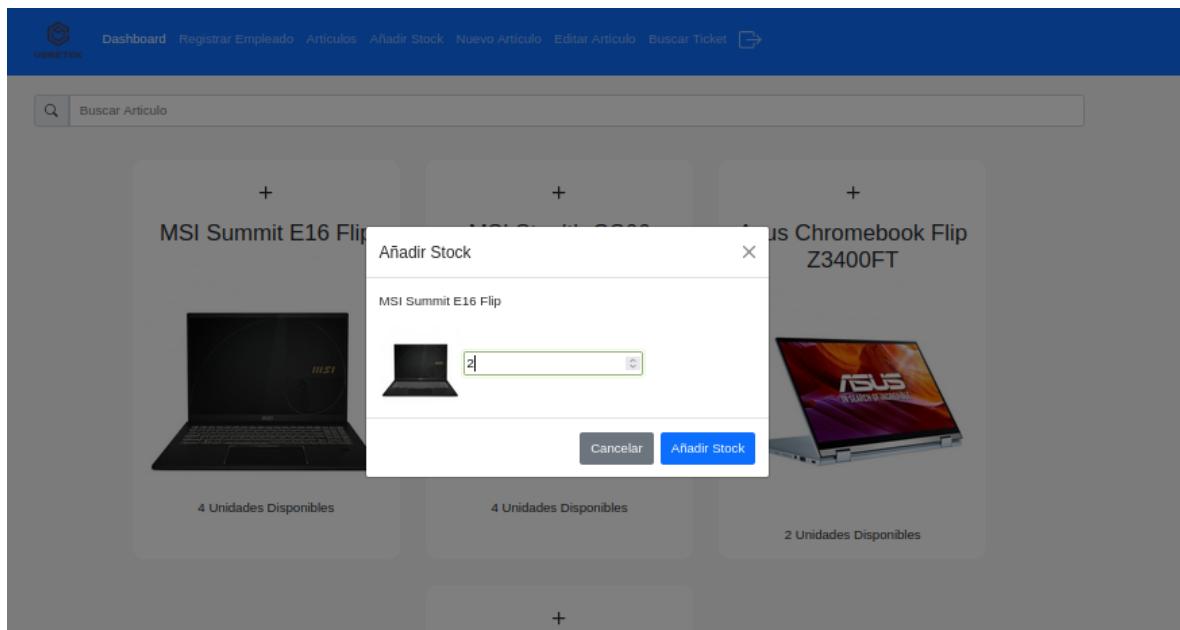


Ilustración 25: Añadir Stock Manual 2

Nada mas pulsar el botón de añadir stock, este será actualizado y la página será refrescada para mostrar el cambio.

8. AÑADIR ARTÍCULO

Aquí es donde se registran los nuevos artículos que se vayan a distribuir en la tienda. Es un formulario simple donde se tienen que llenar todos los datos relacionados con el artículo para poder registrararlo.

Ilustración 26: Añadir Artículo Manual

Precio de compra: Se indica el precio al que se compra el artículo al proveedor para poder calcular el margen de beneficios.

Precio Base Venta: Aquí se indica el precio al que hay que vender el artículo, pero sin IVA, ya que hay un desplegable para elegir el IVA y se calcula el precio de venta final automáticamente.

Al registrar el artículo se volverá a la página de artículos donde se mostrará el nuevo registro.

9. EDITAR ARTÍCULO

Junto con el apartado de añadir stock, este tiene el mismo funcionamiento que la vista de artículos.

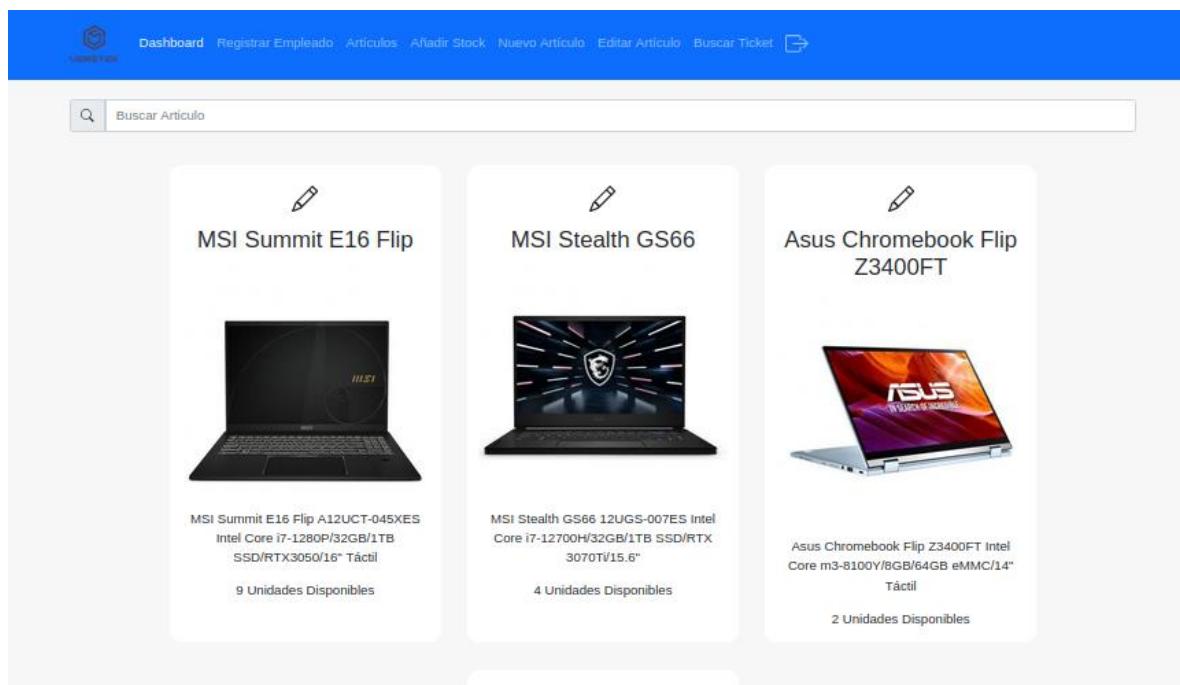


Ilustración 27: Editar Artículo Manual

Una vez seleccionado el artículo que se va a modificar, se mostrará un POP-UP con toda la información de este.

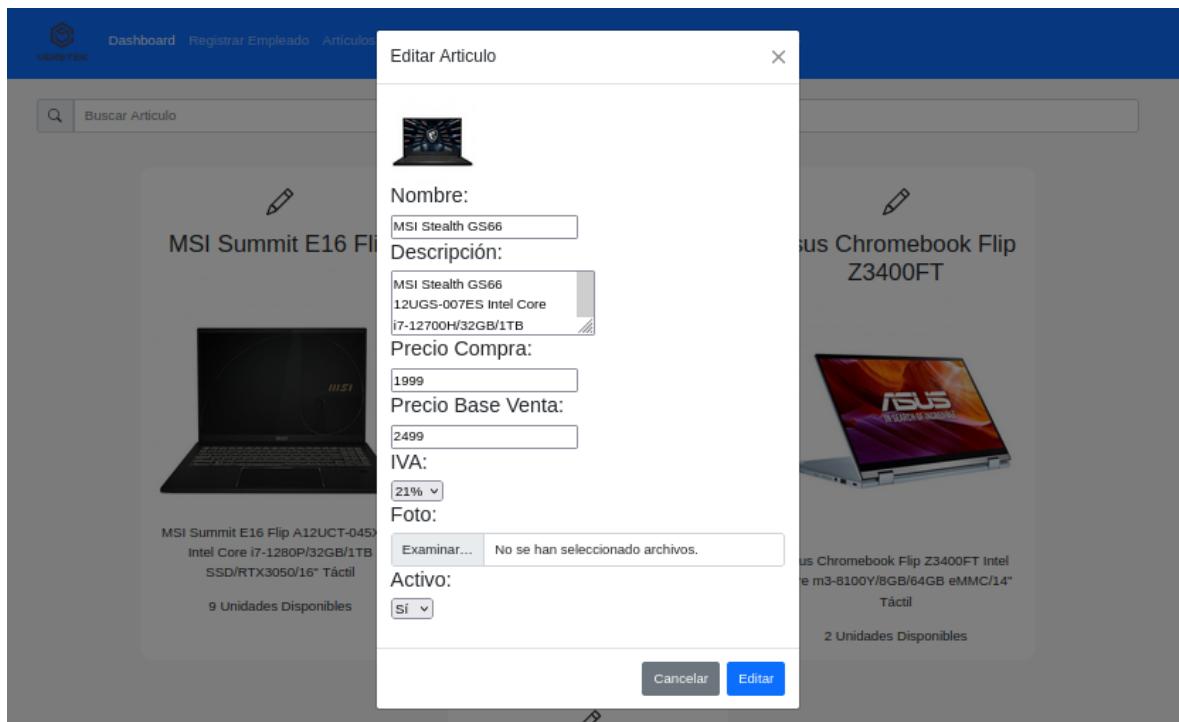


Ilustración 28: Editar Artículo Manual 2

Aquí se modificarán únicamente los campos que se van a cambiar. Son los mismo campos que al registrar un usuario, pero incluye un desplegable donde se selecciona si el artículo está activo o no.

En caso de que se haya descatalogado el artículo, se selecciona que no está activo y este dejará de mostrarse en la ventana de Artículos

10. TICKET

Es una vista en la que únicamente veremos el ticket de la compra que se ha realizado o bien hayamos seleccionado en Buscar Ticket.

#	Artículo	Cantidad	Precio Unitario	Precio
0	MSI Summit E16 Flip	1	2297.79€	2297.79€
1	MSI Stealth GS66	1	3023.79€	3023.79€
2	HP OMEN 16-b0044ns	1	1754.5€	1754.5€
				Total 7076.08€

Tiene 15 días para realizar devoluciones

[Imprimir](#)

Ilustración 29: Ver Ticket Manual

Para imprimir el ticket, pulsamos el botón de imprimir situado en la parte inferior derecha de la vista.

Se abrirá un POP-UP con el ticket y podremos mandarlo a nuestra impresora de tickets para entregárselo al cliente.

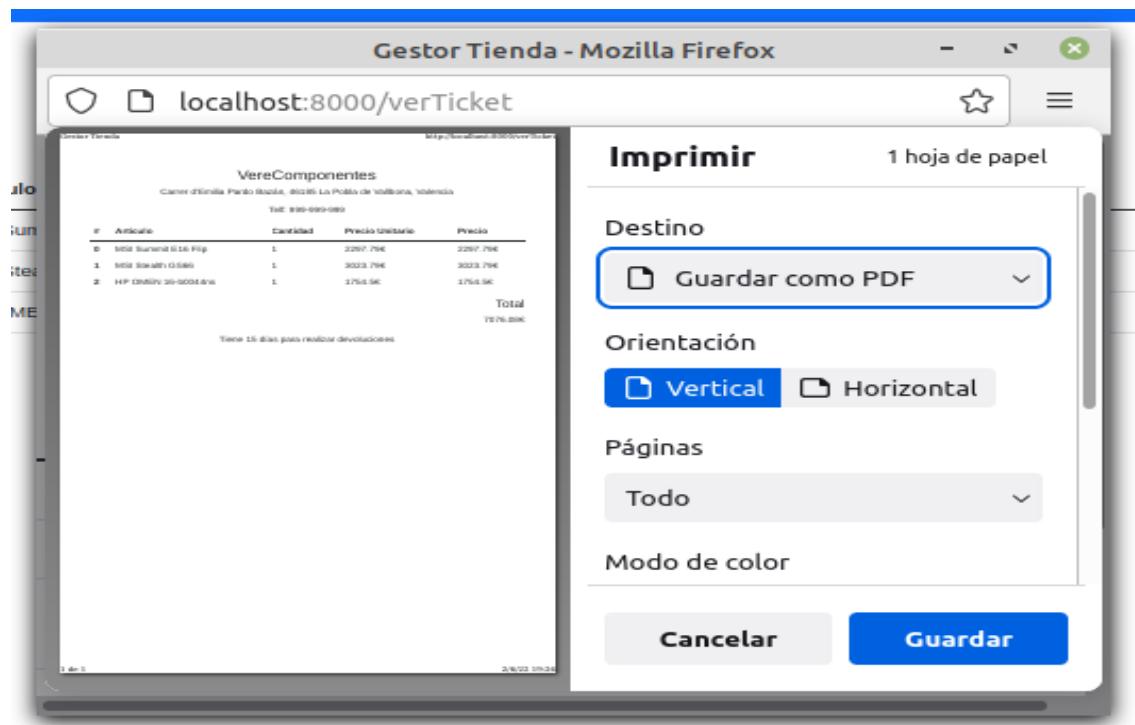


Ilustración 30: Imprimir ticket manual

11. BUSCAR TICKET

Es un apartado donde se puede buscar un ticket a través de su ID, por si en algún momento hace falta revisar un ticket o realizar un devolución.

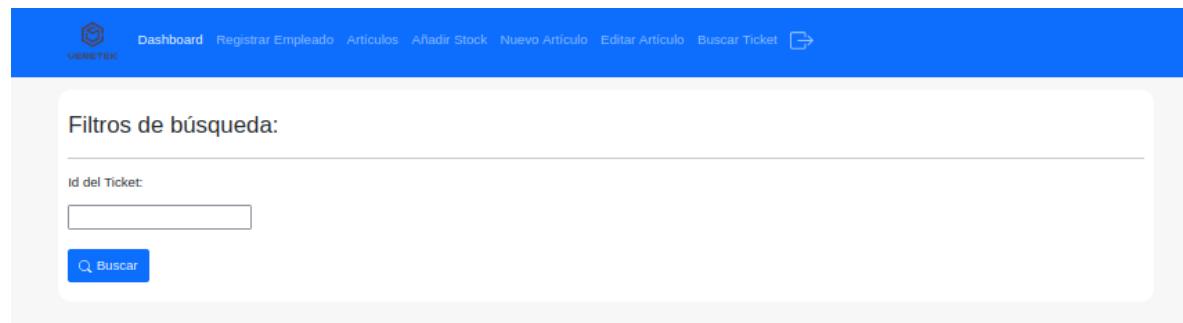


Ilustración 31: Buscar Ticket Manual

Se introduce el ID para buscarlo. En caso de no encontrar ningún ticket con el ID introducido se muestra un error.

Filtros de búsqueda:

Id del Ticket:
1

Resultados:

El ID introducido no coincide con ningún ticket

Ilustración 32: Error Ticket Manual

Si hay un resultado, muestra la fecha y el total del ticket y las opciones de Ver Ticket o crear Devolución

Filtros de búsqueda:

Id del Ticket:
91

Resultados:

Ticket ID: 91
Fecha: 2022-06-01 19:14:55.000000
Total: 2297.79€

Ilustración 33: Resultado ticket Manual

Si se pulsa Ver Ticket, muestra Ticket como al realizar una compra

VereComponentes
Carrer d'Emilia Pardo Bazán, 46185 La Pobla de Vallbona, Valencia
Telf: 999-999-999
Ticket ID: 91

#	Artículo	Cantidad	Precio Unitario	Precio
0	MSI Summit E16 Flip	1	2297.79€	2297.79€

Total
2297.79€

Tiene 15 días para realizar devoluciones

Ilustración 34: Ver Ticket Manual 2

En el caso de seleccionar una devolución, pasaremos a una pantalla donde se mostraran los artículos que contiene el ticket y podremos seleccionar cual se va a devolver.

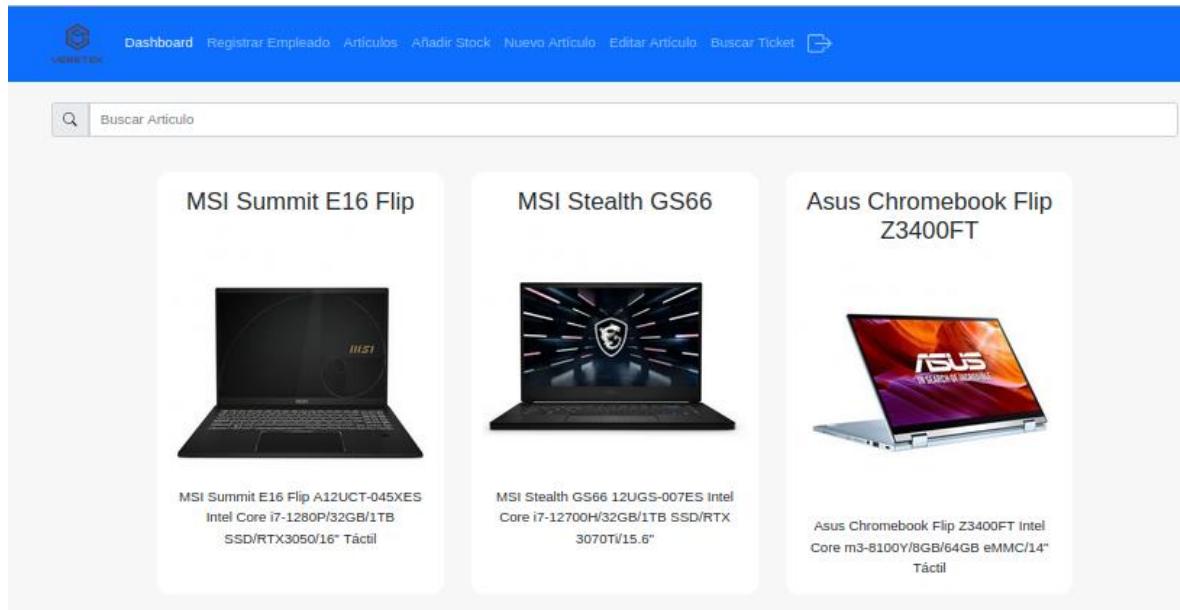


Ilustración 35: Devolución Manual

Al pulsar el artículo que se va a devolver, se mostrará un POP-UP para confirmar la devolución.

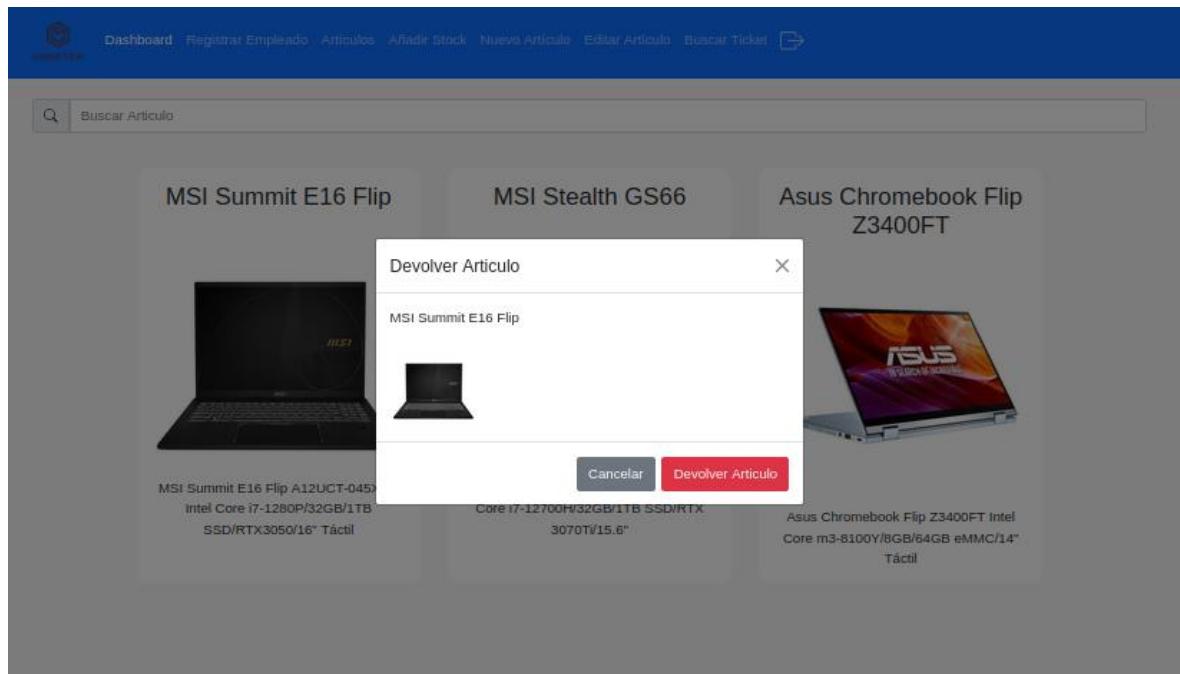


Ilustración 36: Artículo a devolver

En caso de confirmar la devolución, se generará un ticket con la fecha del ticket del que se ha devuelto el artículo, con el total en negativo, para que quede registro de que ha habido una devolución y quede reflejado en los datos que salen en el Dashboard.

Se mostrará un ticket con el total en negativo, el cual puede ser entregado al cliente para que quede constancia de la devolución.

The screenshot shows a software interface with a blue header bar containing the VereComponents logo and navigation links: Dashboard, Registrar Empleado, Artículos, Añadir Stock, Nuevo Artículo, Editar Artículo, Buscar Ticket, and a search icon. Below the header, the title "VereComponentes" is displayed, along with the address "Carrer d'Emilia Pardo Bazán, 46185 La Pobla de Vallbona, Valencia" and the phone number "Teléf: 999-999-999". The main content area is titled "Ticket Devuelto" and contains a table of returned items:

#	Artículo	Cantidad	Precio Unitario	Precio
0	MSI Summit E16 Flip	1	-2297.79€	-2297.79€

At the bottom of the table, it says "Total -2297.79€" and "Devolución de Artículo". There is a blue "Imprimir" button at the bottom left.

Ilustración 37: Ticket devuelto Manual

12. CAMBIAR CONTRASEÑA

Es una vista para poder cambiar la contraseña, ya que en caso de ser un empleado, la contraseña del usuario la ha puesto el Administrador del sistema. Es fácil que dicha contraseña sea genérica y requiera ser cambiada por el usuario.

The screenshot shows a software interface with a blue header bar containing the VereComponents logo and navigation links: Dashboard, Registrar Empleado, Artículos, Añadir Stock, Nuevo Artículo, Editar Artículo, Buscar Ticket, Cambiar Contraseña, and a search icon. Below the header, there is a modal dialog box with the VereComponents logo at the top. The dialog contains fields for "Nueva Contraseña:" (New Password) and "Repetir Contraseña:" (Repeat Password), each with a password input field and a "Copiar" (Copy) button. At the bottom of the dialog is a blue "Cambiar Contraseña" (Change Password) button.

Ilustración 38: Cambiar contraseña

En caso de que las contraseñas no coincidan, se mostrara un mensaje de error.

Si el cambio se completa correctamente, se redirigirá a la página de inicio.

8. Explicación contenidos GitHub

En la carpeta PUBLIC encontramos un árbol de directorios.

main - Gestor-Tienda / public /		
		Go to file Add file ...
...		
css	Commit inicial	2 minutes ago
img	Commit inicial	2 minutes ago
js	Commit inicial	2 minutes ago

Ilustración 39: Carpeta Public

Dentro de estas carpetas se encuentran:

- css: Hojas de estilos
- img: Imágenes
- js: Scripts

Todos los tipos de archivos mencionados que necesite utilizar la aplicación deben estar en la carpeta PUBLIC.

En la carpeta SRC se encuentra otro árbol de directorios.

main - Gestor-Tienda / src /		
		Go to file Add file ...
...		
Controller	Commit inicial	7 minutes ago
Entity	Commit inicial	7 minutes ago
Form	Commit inicial	7 minutes ago
Repository	Commit inicial	7 minutes ago

Ilustración 40: Carpeta SRC

- En la carpeta CONTROLLER se encuentra el controlador de la aplicación.
- En ENTITY, están todas las entidades que son necesarias para el funcionamiento.
- FORM, en esta carpeta están los formularios que han sido generados con la herramienta de Symfony.

- En REPOSITORY se encuentran los repositorios de los objetos conectados a la base de datos, donde podemos crear llamadas a la bbdd personalizadas.

En la carpeta TEMPLATES, está el archivo base.html.twig y otra carpeta donde están todas las vistas

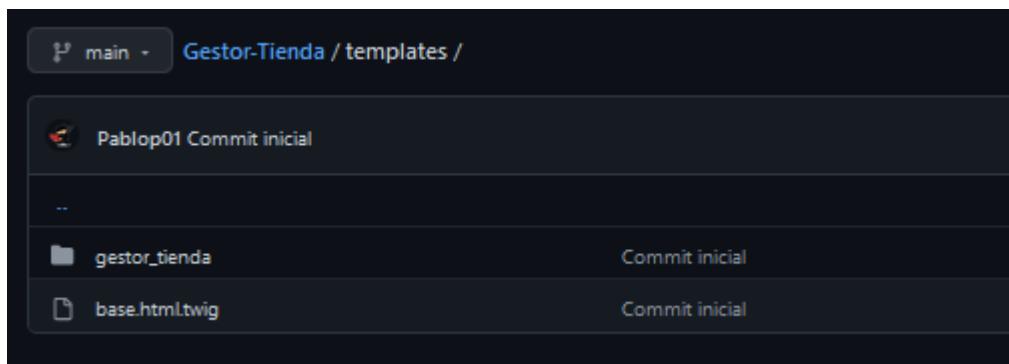


Ilustración 41: Carpeta Templates

Base.html.twig, es una vista de la que heredan todas mis otras vistas, en ella están todas las importaciones que voy a utilizar en todas las vistas.

Contenido de la carpeta gestor_tienda

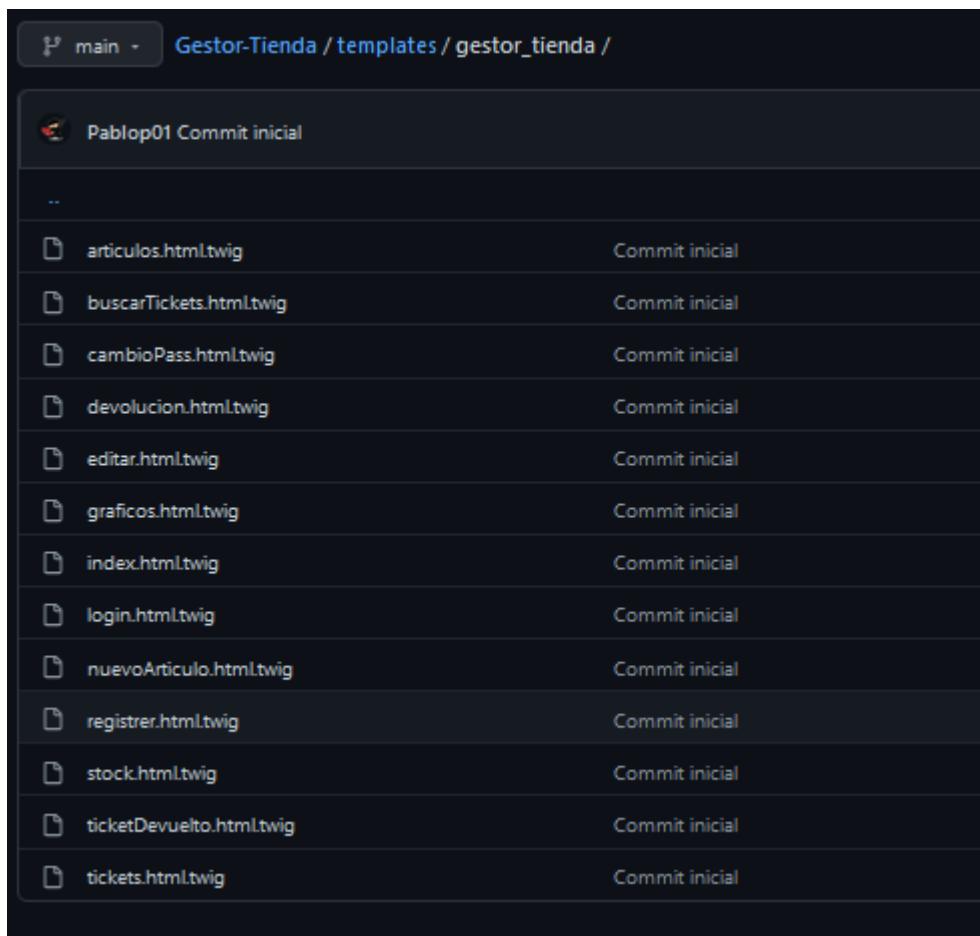


Ilustración 42: Templates/gestor_tienda

9. Webgrafía

Páginas web que han sido utilizadas para desarrollar la aplicación

Documentación de Symfony: <https://symfony.com/doc/current/index.html>

Stack Overflow: <https://es.stackoverflow.com/>

Documentación de Bootstrap: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

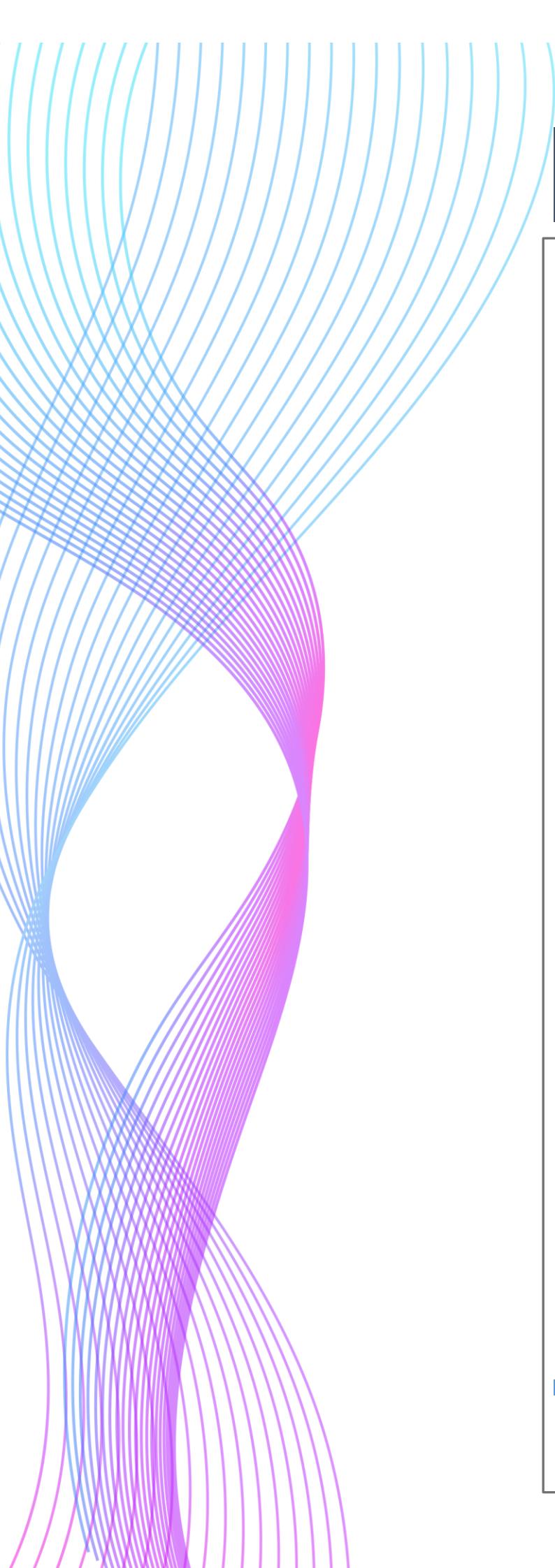
W3SCHOOLS: <https://www.w3schools.com/>

Documentación de PHP: <https://www.php.net/docs.php>

10. Índice de imágenes

Ilustración 1: Entidad Relación	7
Ilustración 2: Diagrama de clases.....	8
Ilustración 3: Mockup	9
Ilustración 4: Login.....	9
Ilustración 5: Error Login.....	9
Ilustración 6: Código Login.....	10
Ilustración 7: Dashboard	10
Ilustración 8: Código Dashboard.....	10
Ilustración 9: Registro	11
Ilustración 10: Código registro	11
Ilustración 11: Artículos	11
Ilustración 12: Código Artículos	12
Ilustración 13: Stock.....	12
Ilustración 14: Editar Artículo	12
Ilustración 15: Código Stock.....	12
Ilustración 16: Login.....	16
Ilustración 17: Dashboard	17
Ilustración 18: Registro	18
Ilustración 19: Artículos	18
Ilustración 20: Buscador.....	19
Ilustración 21: Stock.....	19
Ilustración 22: Error Stock.....	20
Ilustración 23: Procesar Compra.....	20
Ilustración 24: Añadir Stock	21
Ilustración 25: Añadir Stock 2	22
Ilustración 26: Añadir Artículo	22
Ilustración 27: Editar Artículo	23
Ilustración 28: Editar Artículo 2	24

Ilustración 29: Ver Ticket	24
Ilustración 30: Imprimir ticket	25
Ilustración 31: Buscar Ticket	25
Ilustración 32: Error Ticket	26
Ilustración 33: Resultado ticket	26
Ilustración 34: Ver Ticket 2	26
Ilustración 35: Devolución Manual	27
Ilustración 36: Artículo a devolver	27
Ilustración 37: Ticket devuelto	28
Ilustración 38: Cambiar contraseña	28
Ilustración 39: Carpeta Public	29
Ilustración 40: Carpeta SRC.....	29
Ilustración 41: Carpeta Templates	30
Ilustración 42: Carpeta Templates/gestor_tienda.....	30



El proyecto es una Aplicación Web para llevar la gestión de una tienda pequeña, como podría ser una ferretería, una tienda de artículos deportivos o una tienda de libros. Para este ejemplo, he elegido una tienda de componentes de ordenador.

La aplicación cuenta con funciones para Añadir artículos, editar los mismos, añadir stock, imprimir ticket, registrar usuarios (serán los trabajadores de la tienda), ver ticket y un apartado accesible para el administrador de la tienda en el que podrá ver estadísticas de la tienda.