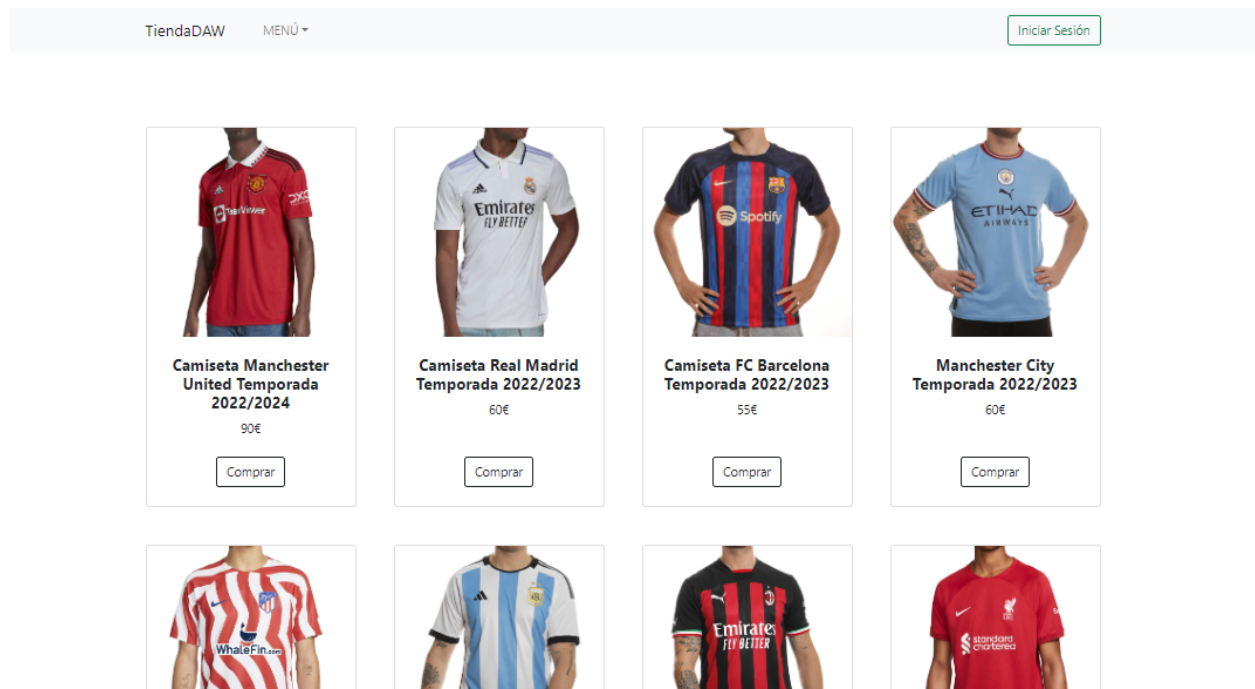


MEMORIA DEL PROYECTO 104: TIENDADAW

Desarrollo de Aplicaciones Web



Juan Manuel Valcárcel Sánchez

jmvs0006@red.ujaen.es

Juan Bautista Muñoz Ruiz

jbmr0001@red.ujaen.es

Curso Académico: 3

ÍNDICE

1- Descripción General del Proyecto:	3
2- Repositorio Gitlab:	3
3- Diagrama Entidad Relación del Modelo Final	4
4- Diagrama de Clases con la arquitectura MVC	6
5- Metodologías y tecnologías:	7
6- Manual de usuario:	14
7- Conclusiones sobre los resultados obtenidos:	15
8- Referencias bibliográficas y recursos on-line:	15

1- Descripción General del Proyecto

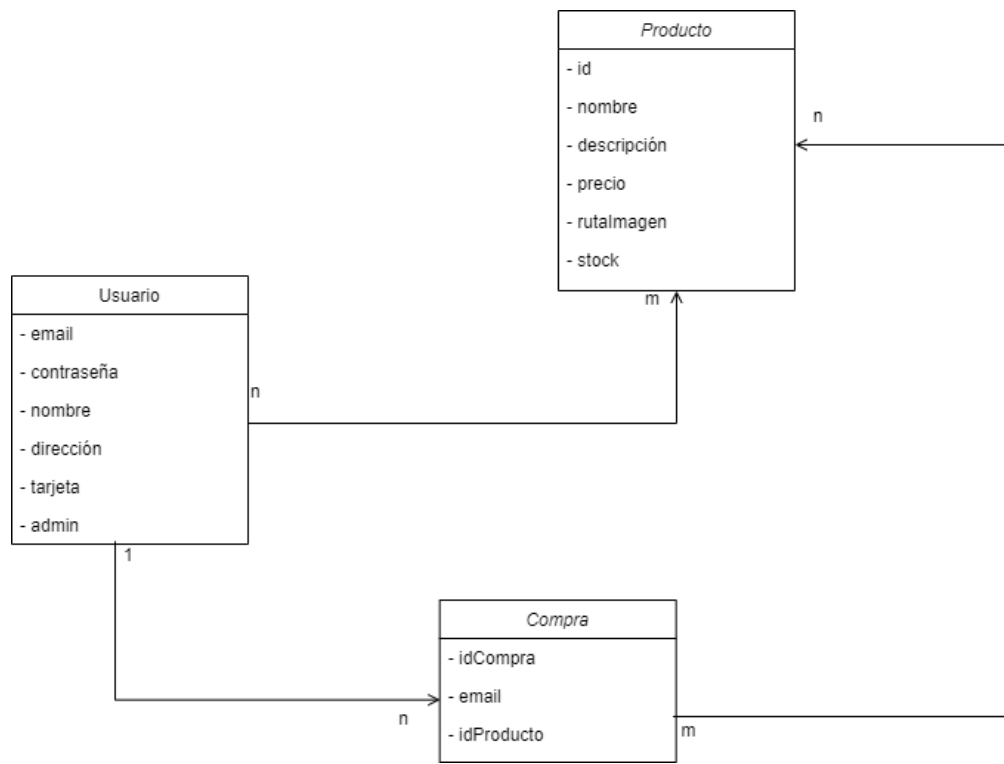
Tienda online donde comprar productos, concretamente camisetas de fútbol. Cuenta con dos tipos de usuario (admin y user). Estos usuarios se identificarán mediante un login.

- El Usuario “admin” tiene acceso a las operaciones CRUD y a las vistas de prueba que beben de la API. Además, el usuario admin podrá crear otros usuarios admin marcando el checkbox del formulario insertar.
- El Usuario “user” únicamente podrá comprar productos y ver sus compras.
- La página inicial mostrará todo el catálogo de productos. Aparte, hay secciones específicas para mostrar productos con ofertas y productos con stock mínimo.
- Al clicar en un producto específico este tendrá una página propia mostrando su información y posible compra.
- En cada página de producto hay una sección de productos recomendados.

2- Repositorio Gitlab

<https://gitlab.ujaen.es/DAW/2223/proyecto104/TiendaDAW>

3- Diagrama Entidad Relación del Modelo Final



Producto:

- **Id:** identificador interno para distinguir un producto de otro.
- **Nombre:** nombre del producto en cuestión.
- **Descripción:** texto informativo sobre el producto.
- **Precio:** precio en € del producto.
- **Ruta Imagen:** url de la imagen del producto.
- **Stock:** cantidad de unidades existentes en venta del producto.

Usuario:

- **Email:** email del usuario, es usado como identificador del usuario.
- **Contraseña:** clave para acceder a la cuenta del usuario.
- **Nombre:** nombre completo del usuario.
- **Dirección:** dirección del usuario para la entrega de los productos.

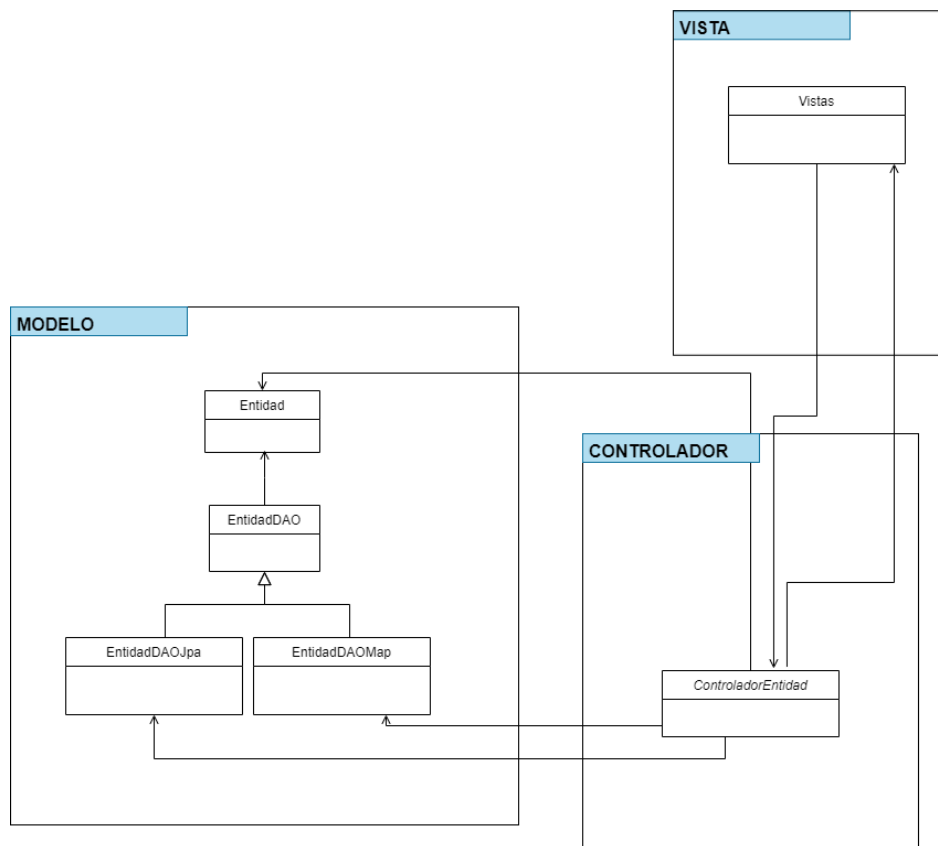
- **Tarjeta:** número de la tarjeta de crédito usada para la compra de productos.
- **Admin:** valor booleano para distinguir al administrador de los otros usuarios.

Compra:

- **IdCompra:** identificador interno para distinguir una compra de otra.
- **Email:** email del usuario que ha realizado la compra.
- **IdProducto:** id del producto que el usuario ha comprado.

4- Diagrama de Clases con la arquitectura MVC

En la siguiente imagen abstraemos de manera general para las tres entidades la arquitectura MVC de nuestro servicio web:



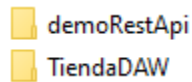
Para las tres entidades (Usuario, Producto y Compra) encontramos el mismo modelo MVC representando en la imagen. En este modelo las vistas se comunican con el controlador y este accede al modelo. Dentro del modelo encontramos, la propia clase de la entidad y la clase DAO para el almacenamiento de los datos.

5- Metodologías y tecnologías

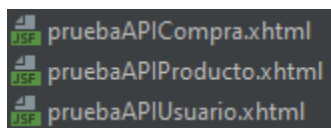
- HTML para la sintaxis básica junto con CSS y Bootstrap para el estilo de las vistas.
- Jakarta Enterprise Edition con Payara para la creación del proyecto.
- Java (JDK 17) usado como lenguaje en el servidor.
- Servlets (en las fases iniciales), semántica de peticiones HTTP y patrón Post-Redirect-Get.
- JakartaServerFaces (faces, facelets y tags) para el uso de diferentes etiquetas en las vistas JSF en archivos xhtml (c:if, ui:composition...).
- Jakarta Bean Validation para la validación de datos en el Back-End
- Unidad de persistencia JPA con una base de datos H2.
- JEE Security API para el login y registro de Usuarios.
- Inversión de control e inyección de dependencias para los DAO.
- JavaScript para la validación de formularios en el cliente.
- JQuery para acceder a los valores de los formularios en la validación con Javascript.
- Servicio web REST básico utilizando JAX-RS para el intercambio de JSON entre el cliente y el servidor.
- Peticiones AJAX con fetch a un servicio web REST para descargar y procesar información en formato JSON.
- Restricciones CORS y peticiones AJAX a un API REST desde una aplicación independiente VueJS.

6- Manual de Usuario

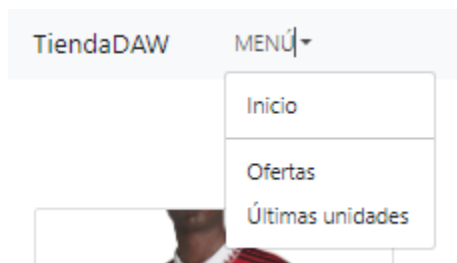
Encontramos dos proyectos:



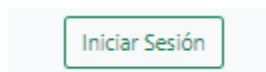
- demoRestApi se trata de un proyecto Vue para probar la API de usuarios.
- TiendaDAW se corresponde con el proyecto de nuestra tienda que lanzará la página y la API para las tres entidades. Cabe destacar que en TiendaDAW encontramos tres vistas que usan la API las cuales serán detalladas posteriormente.



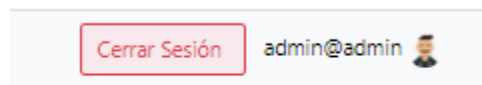
Al clicar en el nombre de la tienda se nos dirigirá a la página principal, desde cualquier vista. Además, disponemos de un menú desplegable para acceder a las diferentes secciones de la tienda:



Encontramos un botón de iniciar sesión el cual nos lleva a la página de login. Desde esta página de login se puede acceder a la de registro.

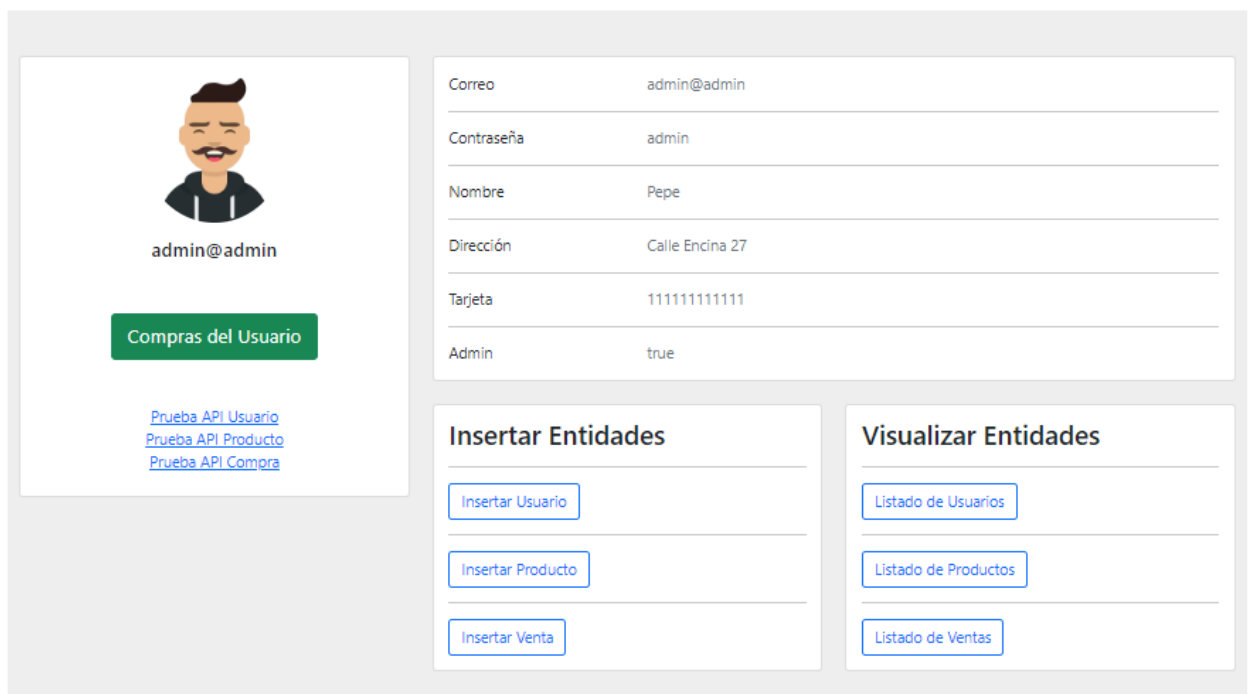


Una vez registrado aparecerá un botón para cerrar sesión así como el email del usuario, este último además, es un botón que nos lleva a nuestra página de usuario:

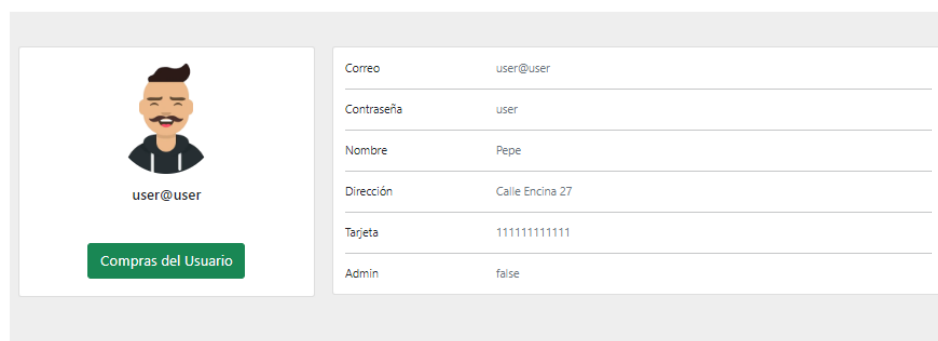


Si el usuario ha hecho login se le redirigirá a la página de usuario:

- En caso de ser usuario “admin” tendrá acceso a las operaciones CRUD y las vistas que prueban la API.



- En caso de no ser “admin” sólo tendrá acceso a visualizar sus compras así como sus datos:



Los usuarios “admin” realizan las operaciones de inserción haciendo click en los siguientes botones:

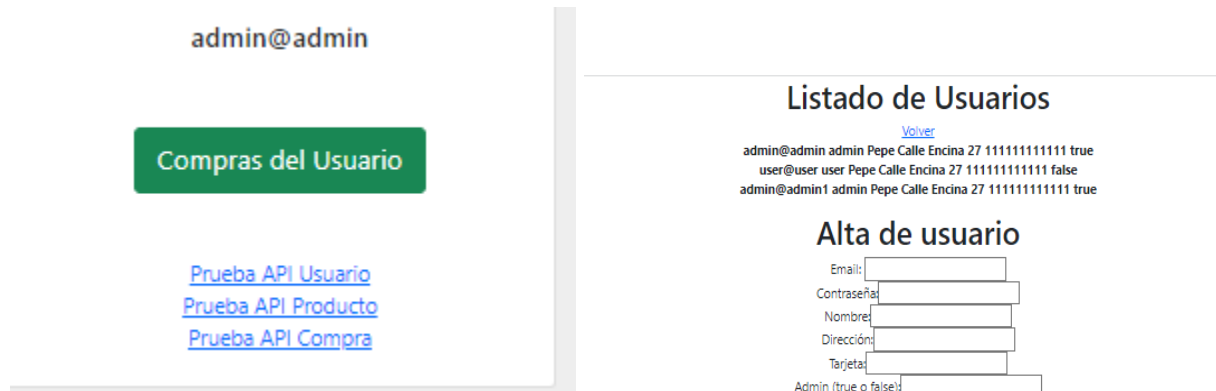


Los usuarios “admin” realizan las operaciones de listado, modificación y borrado haciendo click en los siguientes botones. Estos nos llevan a la página de listado, la cual tiene acceso al borrado y a la edición. Cabe destacar que el usuario admin@admin es imborrable.



En cuanto a la API:

- El usuario “admin” la puede probar accediendo a tres vistas. Por cada entidad podremos probar el listado, alta, modificación y borrado:



- El proyecto demoAPIRest contiene la prueba de la API mediante Vue y accederemos a él a través de <http://localhost:5173/>. Podemos testear el listado, alta, modificación y borrado de Usuarios.

```
cd src
src> cd main
src\main> cd front-end
src\main\front-end> npm run dev
```

Prueba de la API

Alta de Usuario

admin@admin	: admin	Pepe	Calle Encina 27	11111111111111	true
user@user	: user	Pepe	Calle Encina 27	11111111111111	false
admin@admin1	: admin	Pepe	Calle Encina 27	11111111111111	true

Email

Contraseña

Nombre

7- Memoria de la actividad

ITERACIÓN INICIAL

Creación del proyecto, puesta a punto de archivos , creación del storyboard y borradores.

- *Creación archivo .gitignore y CHANGELOG -> Juan Bautista Muñoz Ruiz.*
- *Creación archivo README -> Juan Manuel Valcárcel Sánchez.*
- *Edición archivo README (Planificación inicial y prototipado) -> Juan Manuel Valcárcel Sánchez y Juan Bautista Muñoz Ruiz.*
- **Diseño del Storyboard**
 - ***Pantalla inicial, pantalla de login y Pantalla 3 -> Juan Manuel Valcárcel Sánchez.***
 - ***Pantalla de producto -> Juan Bautista Muñoz Ruiz.***
 - ***Diseño del Diagrama Entidad-Relación -> Juan Bautista Muñoz Ruiz.***
- **Borradores de las vistas de aplicación (HTML y CSS)**
 - ***Pantalla inicial -> Juan Bautista Muñoz Ruiz***
 - ***Pantalla de login -> Juan Manuel Valcárcel Sánchez***
 - ***Pantalla de inserción, editado y borrado -> Juan Manuel Valcárcel Sánchez***
 - ***Pantalla de producto -> Juan Bautista Muñoz Ruiz***

ITERACIÓN 1

Conversión de vistas a JSF más la creación de servlets y controladores DAO.

- ***Plantillas JSF -> Juan Bautista Muñoz Ruiz***
- **Vistas JSF**
 - ***Index -> Juan Bautista Muñoz Ruiz***
 - ***Login y Registro -> Juan Bautista Muñoz Ruiz***
 - ***Página de Usuario -> Juan Bautista Muñoz Ruiz***
 - ***Añadir Producto -> Juan Manuel Valcárcel Sánchez***
 - ***Editar Producto -> Juan Manuel Valcárcel Sánchez***
 - ***Borrar Producto -> Juan Manuel Valcárcel Sánchez***

- **Producto** -> Juan Manuel Valcárcel Sánchez
- **Servlets**
 - **Servlet del login y Servlet del registro** -> Juan Bautista Muñoz Ruiz
- **Controladores, DAO y Modelo**
 - **Producto** -> Juan Manuel Valcárcel Sánchez
 - **Usuario** -> Juan Bautista Muñoz Ruiz

ITERACIÓN 2

Introducción clase compra y sus controladores, operaciones CRUD y modificación de partes anteriores.

- **Visualización productosDAO index** -> Juan Bautista Muñoz Ruiz
- **Operaciones CRUD Usuario** -> Juan Bautista Muñoz Ruiz
- **Nueva paginaUsuario** -> Juan Bautista Muñoz Ruiz
- **Nuevo login que llama al controlador** -> Juan Bautista Muñoz Ruiz
- **Operaciones CRUD Producto** -> Juan Manuel Valcárcel Sánchez
- **CompraDAO, controladorCompra y compra** -> Juan Manuel Valcárcel Sánchez
- **Visualización productosDAO recomendados en paginaProducto** -> Juan Manuel Valcárcel Sánchez
- **Nuevo registro que accede al controlador** -> Juan Manuel Valcárcel Sánchez

ITERACIÓN 3

Creación de los archivos Jpa sustitución de los DAO por DAOMap más autenticación de los usuarios.

- **Configuración de la persistencia** -> Juan Bautista Muñoz Ruiz
- **Sustitución de UsuarioDAO por UsuarioDAOMap** -> Juan Bautista Muñoz Ruiz
- **Creación de UsuarioDAOJpa** -> Juan Bautista Muñoz Ruiz

- **Autenticación (login, zona restringida usuarios y operaciones restringidas para admins) ->** Juan Bautista Muñoz Ruiz
- **Creación de Preferencias de Usuario ->** Juan Bautista Muñoz Ruiz
- **Sustitución de ProductoDAO por ProductoDAOMap ->** Juan Manuel Valcárcel Sánchez
- **Creación de ProductoDAOJpa ->** Juan Manuel Valcárcel Sánchez
- **Sustitución de CompraDAO por CompraDAOMap ->** Juan Manuel Valcárcel Sánchez
- **Creación de compraDAOJpa ->** Juan Manuel Valcárcel Sánchez

ITERACIÓN 4

Integración de las compras en la aplicación, borrado y modificación desde el listado de todos las clases y puesta a punto del trato a los usuarios.

- **Cambios en producto: Carga de imágenes de Producto por ruta, implementación del Stock e id autogenerado ->** Juan Bautista Muñoz Ruiz
- **Administrador Principal imborrable, mensajes de error y Creación de Usuarios admin mediante checkbox en inserción de Usuarios. ->** Juan Bautista Muñoz Ruiz
- **Nuevo Borrado (sin formulario) y Modificación (datos precargados) de Usuarios desde el listado en lugar de desde la página de Usuario ->** Juan Bautista Muñoz Ruiz
- **Nuevo Borrado (sin formulario) y Modificación (datos precargados) de Productos desde el listado en lugar de desde la página de Usuario->** Juan Manuel Valcárcel Sánchez
- **Insertar Compra y Modificar Compra + Nuevo Borrado y Modificación desde el listado->** Juan Manuel Valcárcel Sánchez
- **Visualización del listado de Compras de Usuario ->** Juan Manuel Valcárcel Sánchez

ITERACIÓN 5

Creación de APIs, validación en el cliente y sección de ofertas y últimos productos.

- **API Usuario** -> Juan Bautista Muñoz Ruiz
- **Validación en el cliente de Usuarios** -> Juan Bautista Muñoz Ruiz
- **Validación en el cliente de Compras** -> Juan Bautista Muñoz Ruiz
- **Uso de la API con vue** -> Juan Bautista Muñoz Ruiz
- **API Producto** -> Juan Manuel Valcárcel Sánchez
- **API Compra** -> Juan Manuel Valcárcel Sánchez
- **Validación en el cliente de Productos** -> Juan Manuel Valcárcel Sánchez
- **Sección Ofertas de Productos (Precio<=60) y Ultimas unidades (Stock==1)**
-> Juan Manuel Valcárcel Sánchez

Conclusiones:

El producto final es una página con funcionalidad total de atractiva apariencia que satisface los requisitos inicialmente propuestos y de gran utilidad para la venta de productos online.

Puntos fuertes: El desarrollo iterativo seguido, el aspecto visual, el funcionamiento y las consultas en la base de datos.

Funcionalidades y puntos débiles que se podrían mejorar: Implementación de un buscador, herramienta de filtrado y un almacenamiento de imágenes que no esté basado en rutas.

Referencias bibliográficas:

Temario y guiones de prácticas de la asignatura.

Repositorio de José Ramón Balsas: <https://github.com/jrbalsas>

StackOverflow y Bibliografía de los guiones de prácticas.