



Universidad Veracruzana

Licenciatura en Ingeniería de Software

Facultad de Estadística e Informática

Especificación Funcional

DESARROLLO DE APLICACIONES (83503)

Saúl Domínguez Isidro

Luis Ángel Elizalde Arroyo

Daniel Mongeote Tlachy

Ares Judda Rivera Soto

Alesis de Jesús Torres Osorio

Xalapa, Ver; miércoles 18 de Septiembre del 2024.

Contenido

I.	Introducción	4
a.	Propósito	4
b.	Alcance	4
c.	Definiciones, acrónimos y abreviaciones	5
II.	Requerimientos.....	6
a.	Contexto.....	6
b.	Clases de usuario.....	6
c.	Casos de uso	6
d.	Prototipo de UI	7
i.	Inicio de sesión	7
ii.	CU01 – Dar de alta cuenta	8
iii.	Vista Cliente	10
iv.	Vista vendedor.....	11
v.	Menú Principal.....	12
vi.	CU02 – Modificar Cuenta	14
vii.	CU03 – Publicar Artículo	16
viii.	CU04 – Modificar Artículo.....	19
ix.	CU05 – Consultar Inventario	22
x.	CU06 – Consultar Ventas	24
xi.	CU07 – Consultar Artículos	26
xii.	CU08 – Añadir producto al carrito de compras	28
xiii.	CU09 – Dar de alta pedido	30
xiv.	CU10 – Consultar historial de pedidos	33
e.	Requerimientos Funcionales	34
f.	Requerimientos No Funcionales.....	35
a.	Atributo de Calidad: Rendimiento	35
b.	Seguridad	36
c.	Atributo de Calidad: Usabilidad	36
g.	Restricciones.....	37
III.	Diseño	38
a.	Diseño Arquitectónico.....	38
i.	Vista de Casos de Uso (Diagrama UML de Casos de Uso)	38

ii.	Vista Lógica (Diagrama UML de Clases)	39
iii.	Vista de Implementación (Diagrama UML de Componentes)	39
iv.	Vista de Procesos (Diagrama UML de Secuencia).....	40
v.	Vista de Despliegue (Diagrama UML de Despliegue)	46
b.	Modelo de Datos (Diagrama UML de Entidad-Relación)	46
c.	Descripciones de Casos de Uso.....	47
i.	CU01 – Dar de alta cuenta	47
ii.	CU02 – Modificar cuenta	48
iii.	CU03 – Publicar artículo	49
iv.	CU04 – Modificar artículo	50
v.	CU05 – Consultar inventario	51
vi.	CU06 – Consultar ventas realizadas	52
vii.	CU07 – Consultar artículos	53
viii.	CU08 – Añadir producto al carrito de compras	54
ix.	CU09 – Dar de alta pedido	55
x.	CU10 – Consultar historial de pedidos.....	56
IV.	Construcción.....	58
a.	Selección justificada de la pila tecnológica	58
i.	Aplicación de clientes (móvil y web).....	58
ii.	Aplicación de servicios (API) – Servidor	58
b.	Estándar de codificación	59
c.	Prácticas de construcción realizadas	62
V.	Pruebas	63
a.	Plan de pruebas para las aplicaciones móvil y web.....	63
b.	Procedimiento de prueba (automatizado).....	65
c.	Casos de prueba (automatizados).....	66
d.	Resultados	70
VI.	Estrategia de Despliegue	74
VII.	Conclusiones	74
a.	Luis Ángel Elizalde Arroyo	74
b.	Daniel Mongeote Tlachy	75
c.	Ares Judda Rivera Soto	76

d. Alesis de Jesús Torres Osorio	76
VIII. Referencias	77

I. Introducción

a. Propósito

El presente documento constituye la Especificación Funcional para el proyecto final de la Experiencia Educativa de Desarrollo de Software. Esta especificación tiene como objetivo principal detallar los requisitos del sistema a desarrollar, sirviendo como guía fundamental para el equipo de desarrollo y demás partes interesadas.

Adicionalmente, en este documento se plasman los resultados del proceso de Ingeniería de Requisitos, incluyendo la elicitation, análisis y especificación de estos. Se abordan aspectos como el contexto del sistema, los casos de uso, los requerimientos funcionales y no funcionales, así como las restricciones del proyecto.

La estructura del documento incluye una descripción detallada del propósito y alcance del proyecto, definiciones relevantes, un análisis del contexto operativo, la identificación de las clases de usuario, y la presentación de un prototipo de interfaz de usuario de baja fidelidad. Además, se especifican los requerimientos funcionales y no funcionales, junto con las restricciones técnicas y de negocio aplicables.

Esta especificación funcional es esencial para establecer una comprensión compartida de los objetivos del proyecto entre todos los involucrados, sentando las bases para un desarrollo eficiente y efectivo del sistema propuesto.

b. Alcance

El sistema GOLF Clothing Shop es una plataforma de comercio electrónico diseñada específicamente para la compra y venta de prendas de vestir. Este producto está dirigido a público general quienes podrán adquirir ropa como camisetas y sudaderas. Además, la plataforma permitirá que tanto usuarios individuales interesados en vender sus productos como empresas ya establecidas puedan ofrecer sus prendas a través del sitio.

El alcance de GOLF se logrará a través de varias funcionalidades clave. Los vendedores, ya sean particulares o empresas, tendrán la capacidad de publicar sus artículos en la plataforma, gestionar sus productos y realizar un seguimiento de las ventas. Por otro lado, los compradores podrán explorar un catálogo completo con todas las prendas disponibles, ver información detallada de cada artículo y realizar compras de manera sencilla y segura.

Es importante destacar que, en esta fase inicial de desarrollo, el sistema GOLF Clothing Shop no incluirá un módulo de procesamiento de pagos integrado. Si bien los usuarios podrán navegar por el catálogo, seleccionar productos y agregarlos a su carrito de

compras, la plataforma no gestionará directamente las transacciones financieras. En su lugar, una vez que el cliente haya finalizado su selección de productos, el sistema generará un resumen detallado del pedido, incluyendo los artículos elegidos y el monto total.

c. Definiciones, acrónimos y abreviaciones

Palabra/Acrónimo/Abreviación	Definición
CU (Caso de uso)	Descripción de una secuencia de interacciones entre un sistema y uno o más actores externos que conducen a un resultado específico (Cockburn, 2000).
RF (Requerimiento Funcional)	Especificación de una función que un sistema debe ser capaz de realizar, describiendo las entradas, comportamientos y salidas (Sommerville, 2011).
RNF (Requerimiento No Funcional)	Requisito que especifica criterios para juzgar la operación de un sistema, en lugar de comportamientos específicos, como rendimiento, seguridad o usabilidad (Chung et al., 2009).
RE (Restricción)	Limitación o condición que debe ser respetada por el sistema, que restringe el grado de libertad en la construcción de una solución (Wieggers & Beatty, 2013).
Framework	Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software (Gamma et al., 1994).
Arquitectura de microservicios	Enfoque para desarrollar una aplicación como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso y comunicándose con mecanismos ligeros, a menudo una API de recursos HTTP (Newman, 2015).

II. Requerimientos

a. Contexto



b. Clases de usuario

Tipo de Usuario	Descripción
Vendedor/Empresa	Son tanto personas o pequeños negocios como medianos a grandes negocios que hacen uso del sistema para publicar sus artículos en venta. Solo pueden gestionar sus artículos publicados.
Cliente	Es el público al que está dirigido el negocio. Solo posee funcionalidades como consultar productos disponibles, dar de alta un pedido y cancelarlo.

c. Casos de uso

- CU01 – Dar de alta cuenta
- CU02 – Modificar cuenta
- CU03 – Publicar artículo
- CU04 – Modificar artículo
- CU05 – Consultar inventario
- CU06 – Consultar ventas
- CU07 – Consultar artículos
- CU08 – Añadir articulo al carrito de compras
- CU09 – Dar de alta pedido
- CU10 – Consultar historial de pedidos

d. Prototipo de UI

i. Inicio de sesión



Bienvenido
Inicio de sesión

Correo electrónico:


Contraseña:


Iniciar Sesión

¿No tienes una cuenta? [Regístrate aquí :\)](#)



Bienvenido
Inicio de sesión

Correo electrónico:


Contraseña:


Error: Usuario o contraseña incorrecta.
Por favor, verifica tus datos e intenta nuevamente.

Iniciar Sesión

¿No tienes una cuenta? [Regístrate aquí :\)](#)

ii. CU01 – Dar de alta cuenta

Register

Nombre(s):
Apellido(s):
Username:
Tipo de cuenta:
Correo electrónico:
Contraseña:
Crear Cuenta

The form includes a placeholder for each field: "Escriba su nombre" for Nombre(s), "Escriba su apellido materno y paterno" for Apellido(s), "Escriba su username" for Username, "Seleccione un tipo de cuenta" for Tipo de cuenta, "Escriba su correo electrónico" for Correo electrónico, and "Escriba su contraseña" for Contraseña.

Registro Éxito!
El usuario fue agregado
al sistema correctamente.

Aceptar

Nombre(s):
Apellido(s):
Tipo de cuenta:
Contraseña:
Crear Cuenta

The success message box contains the text "¡Registro Éxito!" and "El usuario fue agregado al sistema correctamente." with an "Aceptar" button. The background registration form shows filled-in fields: "alexis" for Nombre(s), "riveros" for Apellido(s), "Cliente" for Tipo de cuenta, and "contrasenaMUYseGUra1!" for Contraseña.

Regístrate



¡UPS ALGO SALIO MAL!

El correo con el que tratas
registrarste ya se encuentra en el
sistema.

Aceptar

Nombre(s):
s):
Tipo de cuenta:
Cliente

Contraseña:
contrasenaMUYseGUra1!

Crear Cuenta

Regístrate



¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de
conectar a la base de datos.
Inténtelo más tarde.

Aceptar

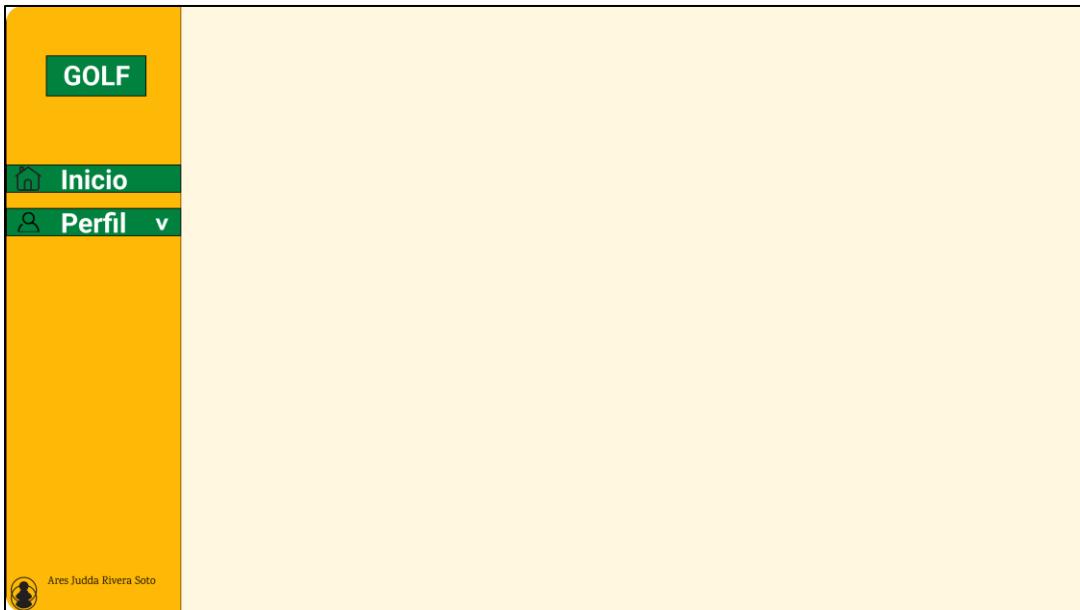
Nombre(s):
s):
Tipo de cuenta:
Cliente

Contraseña:
contrasenaMUYseGUra1!

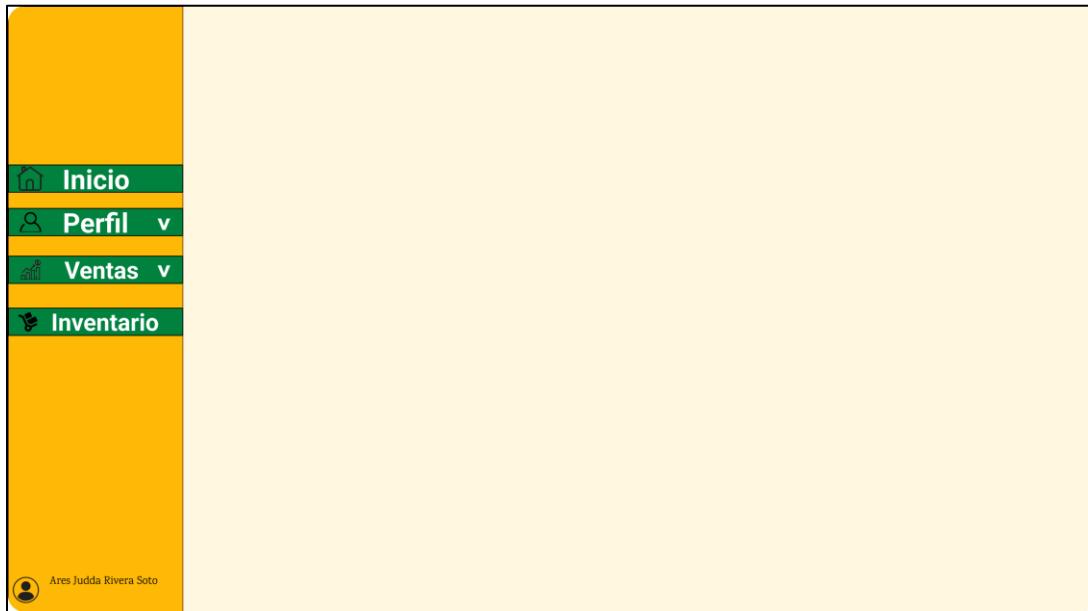
Crear Cuenta



iii. Vista Cliente



iv. Vista vendedor



v. Menú Principal

Artículos

Buscar articulo Filtrar por:

	Playera Golf Wang 01 \$299.90	Categoría: Playera Talla: Mediana Existencias: 20
	Playera Golf Wang 02 \$249.90	Categoría: Playera Talla: Grande Existencias: 10
	Sudadera Golf Wang 01 \$499.90	Categoría: Sudadera Talla: Mediana Agotada

Ares Judda Rivera Soto

Artículos

Buscar articulo Filtrar por:

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar a la base de datos.
Inténtelo más tarde.

Aceptar

Ares Judda Rivera Soto

GOLF

Inicio

Perfil ▾

Artículos



Filtrar por:



¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar

 Ares Judda Rivera Soto

vi. CU02 – Modificar Cuenta

GOLF

Actualiza tu perfil

Ares Judda Rivera Soto

Nombre(s): Donda
Apellido(s): Omari West

Username: LifeParty3000
Tipo de cuenta: Cliente

Fecha de nacimiento: N/A
Correo electrónico: escapeTokio94@outlook.es

Dirección: N/A
Código Postal: N/A

Cancelar **Actualizar Perfil**

GOLF

Actualiza tu perfil

Ares Judda Rivera Soto

Nombre(s): Donda
Apellido(s): Omari West

Username: LifeParty3000
Tipo de cuenta: Cliente

Fecha de nacimiento: N/A
Correo electrónico: escapeTokio94@outlook.es

Dirección: N/A
Código Postal: N/A

¡Proceso Éxito!
Los datos fueron actualizados en el sistema.

Aceptar

Cancelar **Actualizar Perfil**

GOLF

Inicio

Perfil ^

Actualizar

Salir

Ares Judda Rivera Soto

Actualiza tu perfil

Nombre(s): Username:

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar a la base de datos.
Inténtelo más tarde.

Aceptar

Cancelar Actualizar Perfil

This screenshot shows a user profile page with a sidebar on the left containing links for GOLF, Inicio, Perfil (with an upward arrow), Actualizar, and Salir. The main area has a yellow header 'Actualiza tu perfil'. Below it is a form with fields for Nombre(s) and Username, each with a dropdown arrow. A central modal window displays the error message '¡UPS ALGO SALIO MAL!' in large bold letters, followed by the text 'Ocurrió un error al tratar de conectar a la base de datos.' and 'Inténtelo más tarde.' at the bottom. At the bottom of the modal are 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons, and below the modal are 'Actualizar Perfil' (Update Profile) and 'Cancelar' buttons.

GOLF

Inicio

Perfil ^

Actualizar

Salir

Ares Judda Rivera Soto

Actualiza tu perfil

Nombre(s): Username:

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar

Cancelar Actualizar Perfil

This screenshot is identical to the one above, showing the same profile sidebar and 'Actualiza tu perfil' header. The central area features a yellow background with a modal window containing the error message '¡UPS ALGO SALIO MAL!', the connection details, and the 'Aceptar' and 'Cancelar' buttons. The bottom buttons are also identical.

vii. CU03 – Publicar Artículo

GOLF

Inicio
Perfil v
Ventas ^
Vender
Modificar
Ganancias
Inventario

Ares Judda Rivera Soto

Vende tu artículo



Nombre: Escriba el nombre del artículo

Tipo de artículo: Seleccione un tipo de artículo

Precio: Escriba la talla del artículo

Talla: Seleccione un tipo de artículo

Cantidad: Escriba la cantidad del artículo

Cancelar **Publicar**

GOLF

Inicio
Perfil v
Ventas ^
Vender
Modificar
Ganancias
Inventario

Ares Judda Rivera Soto

Vende tu artículo



Nombre: Sudadera Golf Wang

Tipo de artículo: Sudadera

Precio: 499.00

Talla: Mediana

Cantidad: 10

Cancelar **Publicar**

GOLF

- Inicio
- Perfil
- Ventas ▾
 - Vender
 - Modificar
 - Ganancias
- Inventario

Ares Judda Rivera Soto

Vende tu artículo

¡Proceso Éxito!
El artículo fue publicado en el sistema.

Aceptar

10 Cancelar Publicar

GOLF

- Inicio
- Perfil
- Ventas ▾
 - Vender
 - Modificar
 - Ganancias
- Inventario

Ares Judda Rivera Soto

Vende tu artículo

Nombre:
Tipo de artículo:

¡UPS ALGO SALIO MAL!
Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar

Cancelar Publicar

GOLF

Inicio

Perfil

Ventas ▾

Vender

Modificar

Ganancias

Inventario

Ares Judda Rivera Soto

Vende tu artículo

Nombre: Tipo de artículo:

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar **Cancelar** **Publicar**

viii. CU04 – Modificar Artículo

Artículos

Buscar Articulos

 Playera Golf Wang 01 \$299.90	Categoría: Playera Talla: Mediana Existencias: 20
 Playera Golf Wang 02 \$249.90	Categoría: Playera Talla: Grande Existencias: 10
 Sudadera Golf Wang 01 \$499.90	Categoría: Sudadera Talla: Mediana Agotada

Actualiza tu artículo

Nombre: Sudadera Golf Wang | Tipo de artículo: Sudadera

Precio: 499.00 | Talla: Mediana

Cantidad: 0 |

GOLF

- Inicio
- Perfil
- Ventas
- Vender
- Modificar
- Ganancias
- Inventario

Ares Judda Rivera Soto

Actualiza tu artículo

Nombre: Tipo de artículo:

¡Proceso Éxito!
El artículo fue actualizado en el sistema.

Aceptar **Eliminar** **Modificar**

GOLF

- Inicio
- Perfil
- Ventas
- Vender
- Modificar
- Ganancias
- Inventario

Ares Judda Rivera Soto

Actualiza tu artículo

Nombre: Tipo de artículo:

¡Proceso Éxito!
El artículo fue eliminado del sistema.

Aceptar **Modificar**

GOLF

- Inicio
- Perfil
- Ventas ▾
- Vender
- Modificar
- Ganancias
- Inventario

Ares Judda Rivera Soto

Actualiza tu artículo

Nombre: Tipo de artículo:

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar a la base de datos.
Inténtelo más tarde.

Aceptar **Modificar**

GOLF

- Inicio
- Perfil
- Ventas ▾
- Vender
- Modificar
- Ganancias
- Inventario

Ares Judda Rivera Soto

Actualiza tu artículo

Nombre: Tipo de artículo:

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar **Modificar**

ix. CU05 – Consultar Inventario

Inventario

Buscar Articulos Filtrar por:

PDF

Playera Golf Wang 01 \$299.90	Categoría: Playera Talla: Mediana Existencias: 20
Playera Golf Wang 02 \$249.90	Categoría: Playera Talla: Grande Existencias: 10
Sudadera Golf Wang 01 \$499.90	Categoría: Sudadera Talla: Mediana Agotada

Ares Judda Rivera Soto

Inventario

Buscar Articulos Filtrar por:

PDF

¡Proceso Éxito!

PDF del inventario fue generado correctamente.

Aceptar

Playera Golf Wang 01 \$299.90	Categoría: Playera Mediana Existencias: 20
Playera Golf Wang 02 \$249.90	Categoría: Playera Talla: Grande Existencias: 10
Sudadera Golf Wang 01 \$499.90	Categoría: Sudadera Talla: Mediana Agotada

Ares Judda Rivera Soto

Inventario

Buscar Articulos PDF

Plavera Golf Wang 01	Categoría: Playera
	Talla: S: 20 Talla: M: 10 Talla: L: 10
¡UPS ALGO SALIO MAL! Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde.	
Aceptar	
\$499.90	Talla: Mediana Agotada

GOLF

Inicio Perfil Ventas Inventario

Ares Judda Rivera Soto

Inventario

Buscar Articulos PDF

Sudadera Golf Wang 01	Categoría: Sudadera
	Talla: Mediana Agotada
¡UPS ALGO SALIO MAL! Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde.	
Aceptar	
\$499.90	

GOLF

Inicio Perfil Ventas Inventario

Ares Judda Rivera Soto

x. CU06 – Consultar Ventas

GOLF

- [Inicio](#)
- [Perfil](#)
- [Ventas](#)
- [Vender](#)
- [Modificar](#)
- [Ganancias](#)
- [Inventario](#)

Ares Judda Rivera Soto

Ganancias

Fecha de inicio ▼ Fecha de Corte ▼

 Playera Golf Wang 01 Vendidos: 5 Precio: \$299.90	Ventas realizadas por: Ares Judda Rivera Soto Fecha de Inicio: 01/10/2024 Fecha de Corte: 31/10/2024 Articulos vendidos: 10 GANANCIAS: \$3999.00
 Playera Golf Wang 02 Vendidos: 0 Precio: \$249.90	
 Sudadera Golf Wang 01 Vendidos: 5 Precio: \$499.90	

GOLF

- [Inicio](#)
- [Perfil](#)
- [Ventas](#)
- [Vender](#)
- [Modificar](#)
- [Ganancias](#)
- [Inventario](#)

Ares Judda Rivera Soto

Ganancias

Fecha de inicio ▼ Fecha de Corte ▼

¡UPS ALGO SALIO MAL! Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde.

Aceptar

 Playera Golf Wang 01 Vendidos: 5 Precio: \$299.90	Ventas realizadas por: Ares Judda Rivera Soto Fecha de Inicio: 01/10/2024 Fecha de Corte: 31/10/2024 Articulos vendidos: 10 GANANCIAS: \$3999.00
 Playera Golf Wang 02 Vendidos: 0 Precio: \$249.90	
 Sudadera Golf Wang 01 Vendidos: 5 Precio: \$499.90	

GOLF

- Inicio**
- Perfil**
- Ventas**
- Vender**
- Modificar**
- Ganancias**
- Inventario**

Ares Judda Rivera Soto

Ganancias

Fecha de inicio ▾ Fecha de Corte ▾

Diseño Golf Wang 01

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar

Sudadera Golf Wang 01

Vendidos: 5 Precio: \$499.90

GANANCIAS: \$3999.00

zadas por:
a Rivera
o

: 01/10/2024
: 31/10/2024
ndidos: 10

xi. CU07 – Consultar Artículos

The screenshots show a mobile application interface for a brand named 'GOLF'. The top screenshot displays a list of three products: 'Playera Golf Wang 01' (yellow t-shirt, \$299.90), 'Playera Golf Wang 02' (pink t-shirt, \$249.90), and 'Sudadera Golf Wang 01' (blue hoodie, \$499.90). The bottom screenshot shows an error message: '¡UPS ALGO SALIO MAL!' (An error occurred), 'Ocurrió un error al tratar de conectar a la base de datos.', and 'Inténtelo más tarde.' (Try again later).

GOLF

Artículos

Buscar artículo Filtrar por:

Playera Golf Wang 01
\$299.90

Categoría: Playera
Talla: Mediana
Existencias: 20

Playera Golf Wang 02
\$249.90

Categoría: Playera
Talla: Grande
Existencias: 10

Sudadera Golf Wang 01
\$499.90

Categoría: Sudadera
Talla: Mediana
Agotada

GOLF

Artículos

Buscar artículo Filtrar por:

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar a la base de datos.
Inténtelo más tarde.

Aceptar

GOLF

Inicio

Perfil ▾

Ares Judda Rivera Soto

Artículos

Buscar artículo

Filtrar por: Sudadera

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar

xii. CU08 – Añadir producto al carrito de compras

The screenshots show a mobile application interface for a golf apparel store named "GOLF".

Screenshot 1: Artículos (Articles) Screen

- Header:** Artículos
- Search Bar:** Playera Golf Wang 01
- Product Card:** Playera Golf Wang 01 (\$299.90)
 - Image:** Yellow t-shirt with "GOLF" logo.
 - Details:** Categoría: Playera, Talla: Mediana, Existencias: 20.
- Cart icon:** Shopping cart icon with 0 items.

Screenshot 2: Detalles del artículo (Article Details) Screen

- Header:** Detalles del artículo
- Product Image:** Yellow t-shirt with "GOLF" logo.
- Product Information:**

Nombre:	Tipo de artículo:
Sudadera Golf Wang	Sudadera
Precio:	Talla:
499.00	Mediana
Cantidad:	
20	Agregar al carrito

GOLF

Inicio

Perfil v

Ares Judda Rivera Soto

Artículos

Buscar artículo 

Filtrar por: 

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar a la base de datos.
Inténtelo más tarde.

Aceptar

GOLF

Inicio

Perfil v

Ares Judda Rivera Soto

Artículos

Buscar artículo 

Filtrar por: Sudadera 

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar

xiii. CU09 – Dar de alta pedido

The screenshot shows the GOLF app's shopping cart screen. On the left is a vertical sidebar with a yellow background and a green header labeled "GOLF". Below the header are two items: "Inicio" and "Perfil". At the bottom of the sidebar is a user profile icon with the name "Ares Judda Rivera Soto". The main content area has a yellow header "Carrito de compras". Below it is a yellow box titled "Historial de pedidos". Inside this box are two items:

- Playera Golf Wang 01**
\$299.90
Cantidad: 2
Talla: Mediana
Existencias: 20
- Playera Golf Wang 02**
\$249.90
Cantidad: 2
Talla: Grande
Existencias: Agotada

At the bottom of the cart screen, there is a summary: "Total A Pagar \$847.70" followed by "Cancelar" and "Realizar Pedido" buttons.

This screenshot shows the GOLF app's confirmation screen after a successful purchase. The layout is identical to the previous screenshot, with the yellow sidebar and "Carrito de compras" header. The main message in the center says "¡Proceso Éxito!" and "El pedido fue realizado correctamente.". Below this message is a green "Aceptar" button. The cart summary at the bottom shows "Total A Pagar \$847.70" and the same "Cancelar" and "Realizar Pedido" buttons as the previous screen.

Carrito de compras

Historial de pedidos




¡UPS ALGO SALIO MAL!

Uno de los artículos que quieres
pedir se encuentra agotado.

Aceptar

Total A Pagar
\$847.70

Cancelar Realizar Pedido

Camiseta: ⊖ 2 ⊕
Talla: Mediana
Existencias: 20

Camiseta: ⊖ 2 ⊕
Talla: Grande
Existencias: Agotada

Carrito de compras

Historial de pedidos




Playera Golf Wang 01 Cantidad: ⊖ 2 ⊕
Talla: Mediana
Existencias: 20

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de
conectar a la base de datos.
Inténtelo más tarde.

Aceptar

Total A Pagar
\$847.70

Cancelar Realizar Pedido

Camiseta: ⊖ 2 ⊕
Talla: Grande
Existencias: Agotada

GOLF

Inicio

Perfil

Ares Judda Rivera Soto

Carrito de compras

Historial de pedidos

Pleyers Golf Wong 01

¡UPS ALGO SALIO MAL!

Ocurrió un error al tratar de conectar al servidor del sistema.
Inténtelo más tarde.

Aceptar

Total A Pagar
\$847.70

Cancelar Realizar Pedido

Mediana
nencias: 20

Grande
s: Agotada

xiv. CU10 – Consultar historial de pedidos

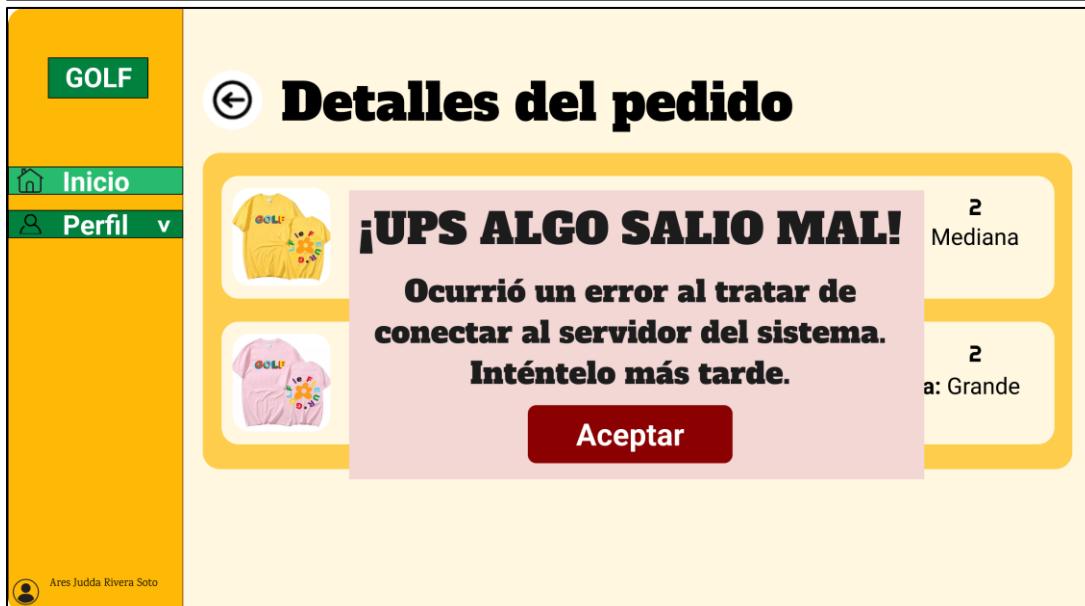
The screenshot shows two pages of a mobile application interface for 'GOLF'.

Top Screen (Historial de compras):

- Header:** 'Historial de compras' with a 'Filtrar por fecha' dropdown.
- Items:**
 - Pedido del 10 de octubre del 2024:** \$847.70 (2 artículos comprados, Detalles)
 - Pedido del 08 de octubre del 2024:** \$120.10 (1 artículo comprado, Detalles)
 - Pedido del 10 de octubre del 2024:** \$1742.40 (4 artículos comprados, Detalles)
- User Info:** Ares Judda Rivera Soto.

Bottom Screen (Detalles del pedido):

- Header:** 'Detalles del pedido' with a back arrow icon.
- Items:**
 - Playera Golf Wang 01:** \$299.90 (Cantidad: 2, Talla: Mediana)
 - Playera Golf Wang 02:** \$249.90 (Cantidad: 2, Talla: Grande)
- User Info:** Ares Judda Rivera Soto.



e. Requerimientos Funcionales

RF – 01: El sistema posee 2 distintos tipos de usuarios (**Cliente y Vendedor**), por lo que cada uno posee diferentes privilegios y funcionalidades dentro del sistema.

RF – 02: El sistema permite a sus usuarios la creación de una cuenta ya sea (**Cliente o Vendedor**) a partir de su Nombre, Correo electrónico, dirección, numero de contacto y edad; para poder acceder a las funcionalidades correspondientes. (**CU01**)

RF – 03: El sistema debe permitir a los usuarios **Cliente y Vendedor** la modificación de los datos de su cuenta correspondiente Nombre, dirección, numero de contacto, foto de perfil. (**CU02**)

RF – 04: El sistema debe permitir al usuario del tipo **Vendedor** consultar la totalidad de artículos publicados y vendidos por dentro del sistema. (**CU05, CU06**)

RF – 05: El sistema debe permitir al usuario **Cliente** la búsqueda específica de artículos ya sea por los siguientes datos (nombre, el código, el nombre de usuario vendedor) sin importar si los caracteres ingresados en la búsqueda son mayúsculas o minúsculas. (**CU07**).

RF – 06: El sistema debe permitir al usuario **Vendedor** la publicación de un artículo con los siguientes datos nombre del producto, código de producto, código de barras (opcional), existencias, imagen, descripción y precio. (**CU03**)

RF – 07: El sistema debe permitir al usuario **Vendedor** la búsqueda específica de artículos que ha publicado en el sistema para vender ingresando el nombre, el código, o seleccionando el tipo de artículo. (**CU05**)

RF – 08: El sistema debe permitir al usuario **Vendedor** la modificación de los datos de todos los artículos publicados por el dentro del sistema (nombre del producto, código de producto, código de barras (opcional), existencias, imagen, descripción y precio). (**CU04**).

RF – 09: El sistema debe permitir al usuario **Vendedor** dar de baja todos los artículos que dicho usuario puso a la venta en el sistema. (**CU04, CU05**).

RF – 10: El sistema debe permitir al usuario **Cliente** consultar los artículos contenidos en su carrito de compras correspondiente. (**CU09**).

RF – 11: El sistema debe permitir al usuario **Cliente** agregar artículos a su carrito de compras correspondiente. (**CU9**).

RF – 12: El sistema debe permitir al usuario **Cliente** dar de baja artículos a su carrito de compras correspondiente. (**CU09**).

RF – 13: El sistema debe permitir a los usuarios de tipo **Cliente** realizar la compra de los artículos almacenados en su carrito de compras correspondiente. (**CU09**).

RF – 14: En caso de que se acaben las existencias de algún artículo el sistema debe hacerle saber al **Cliente**, que dicho artículo se encuentra agotado, ya sea que sea a través de búsqueda o del carrito de compras (**CU07, CU09**).

RF – 15: El sistema debe permitir al **Cliente** la consulta de su historial de pedidos. (**CU10**).

f. Requerimientos No Funcionales

a. Atributo de Calidad: Rendimiento

RNF – 01: El sistema debe garantizar que el cambio entre las funcionalidades y privilegios de los diferentes tipos de usuarios (**Cliente y Vendedor**) ocurra en menos de 2 segundos después del inicio de sesión. (**RF-01**)

RNF – 02: El proceso de registro de nuevos usuarios (*Cliente o Vendedor*) no debe exceder los 3 segundos una vez que se haya ingresado y verificado toda la información. (RF-02)

RNF – 03: Las búsquedas de artículos por parte de los *Clientes* deben arrojar resultados en menos de 2 segundos, independientemente de la cantidad de datos ingresados o si están en mayúsculas o minúsculas. (RF-05)

RNF – 04: La publicación de un artículo por parte de un *Vendedor* no debe tardar más de 4 segundos desde que se completa el formulario hasta que la confirmación es mostrada. (RF-06).

RNF – 05: La consulta de los artículos en el carrito de compras debe realizarse en un máximo de 2 segundos, incluso con hasta 100 artículos en el carrito. (RF-10)

RNF – 06: El sistema debe agregar un artículo al carrito de compras en menos de 2 segundos por cada artículo, incluyendo la actualización en la base de datos. (RF-11)

RNF – 07: El sistema debe notificar al *Cliente* sobre la falta de existencias de un artículo en menos de 1 segundo después de intentar agregarlo al carrito o consultararlo. (RF-14)

b. Seguridad

RNF – 08: Todos los datos sensibles deben ser cifrados usando un algoritmo seguro antes de almacenarse en la base de datos. (RF-02)

RNF – 09: Cualquier modificación de los datos de los artículos publicados debe generar una entrada en un registro de auditoría, indicando qué usuario realizó la modificación y a qué hora. (RF-08)

RNF – 10: Al eliminar un artículo del carrito, el sistema debe actualizar en tiempo real la base de datos para reflejar la disponibilidad de stock en menos de 2 segundos. (RF-12)

RNF – 11: En caso de una interrupción durante una modificación de datos, el sistema debe revertir automáticamente cualquier cambio 3 segundos después. (RF-03, RF-08)

RNF – 12: En caso de cualquier fallo del sistema o excepción ocurrida, el sistema debe generar una entrada en un registro de auditoría, indicando qué sucedió y a qué hora. (General).

c. Atributo de Calidad: Usabilidad

RNF – 13: Las interfaces para *Cliente y Vendedor* deben ser intuitivas y accesibles. Los usuarios no deben requerir más de 3 clics o menos de 5 segundos para acceder a las principales funcionalidades. (RF-01)

RNF – 14: El proceso de creación de cuenta debe ser sencillo y no debe tomar más de 2 minutos para completarse, excluyendo el tiempo de verificación de correo electrónico. (RF-02)

RNF – 15: En caso de errores en la interfaz (por ejemplo, campos faltantes en formularios), el sistema debe mostrar mensajes de error claros y específicos en menos de 2 segundos. **(General)**

g. Restricciones

RE – 01: El sistema debe diseñarse para poder ser compatible e implementado tanto para la web, como para dispositivos móviles con el sistema operativo Android.

RE – 02: El sistema debe ser capaz de comunicarse entre distintos lenguajes de programación, tales como Java, Python, C#, esto por medio del uso de tecnologías como Sockets, UDP, RMI, etc.

RE – 03: El sistema debe implementar bases de datos basadas en Structured Query Language (SQL).

RE – 04: El sistema tiene que ser implementado usando la arquitectura de microservicios.

RE – 05: El sistema debe ser distribuido, buscando ser altamente cohesivo.

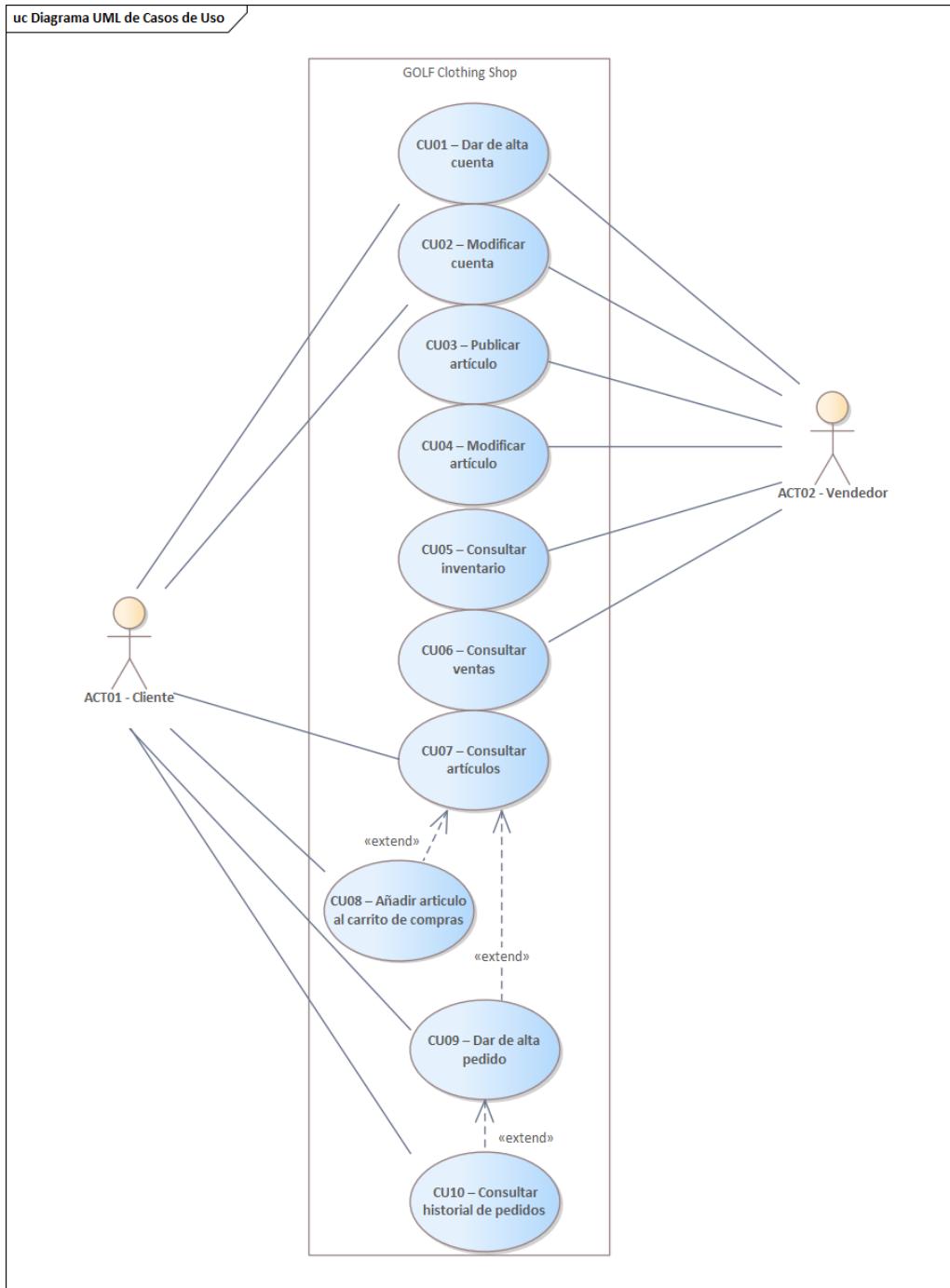
RE – 06: El sistema debe brindar concurrencia de usuarios, brindando distintas instancias paralelas de cliente.

III. Diseño

a. Diseño Arquitectónico

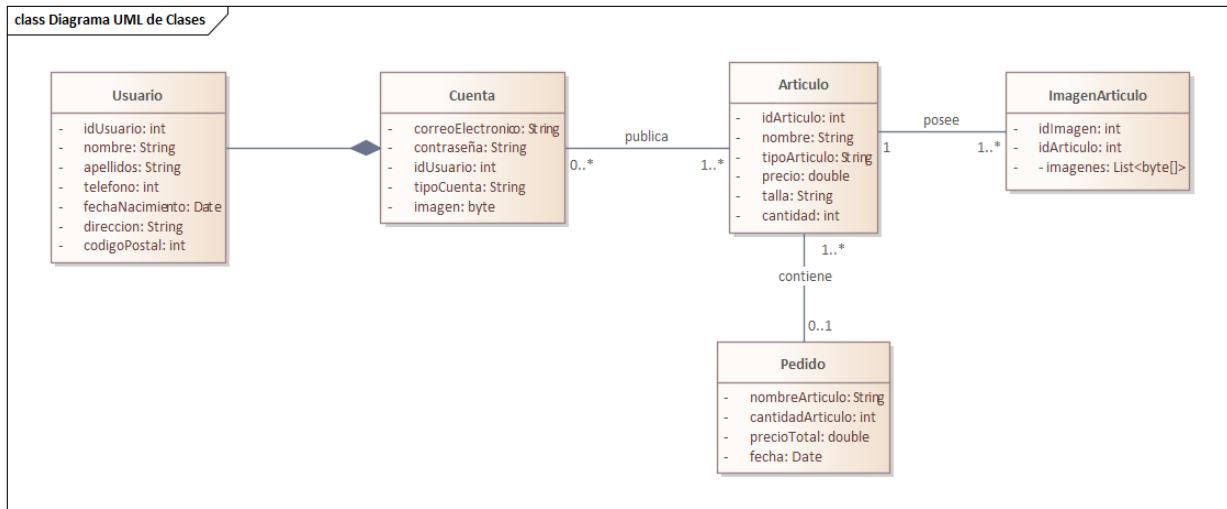
i. Vista de Casos de Uso (Diagrama UML de Casos de Uso)

En este apartado se detalla la lista de casos de uso por diseñar y construir para la lógica del sistema.



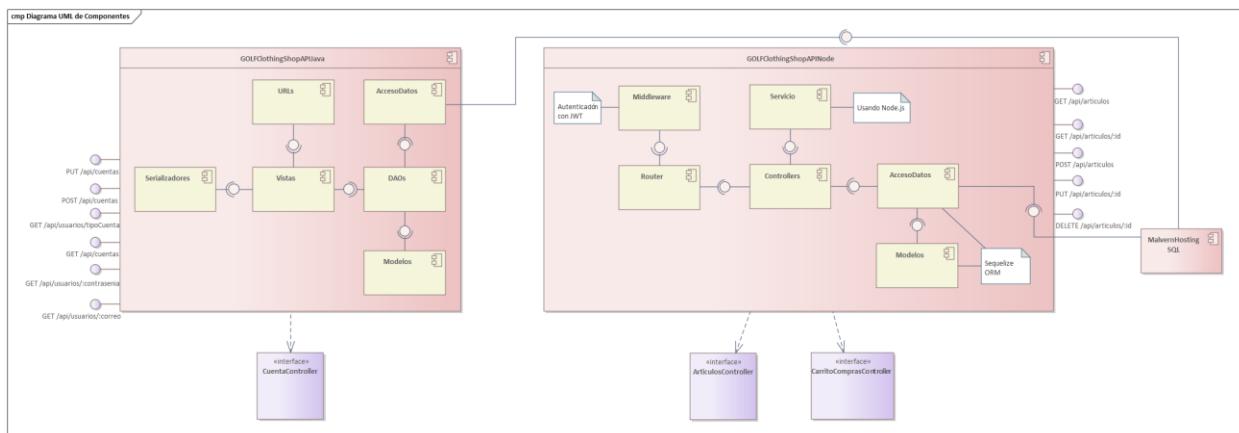
ii. Vista Lógica (Diagrama UML de Clases)

En este apartado se muestra el Diagrama UML de Clases, el cual detalla la estructura estática del sistema y las relaciones entre sus clases.



iii. Vista de Implementación (Diagrama UML de Componentes)

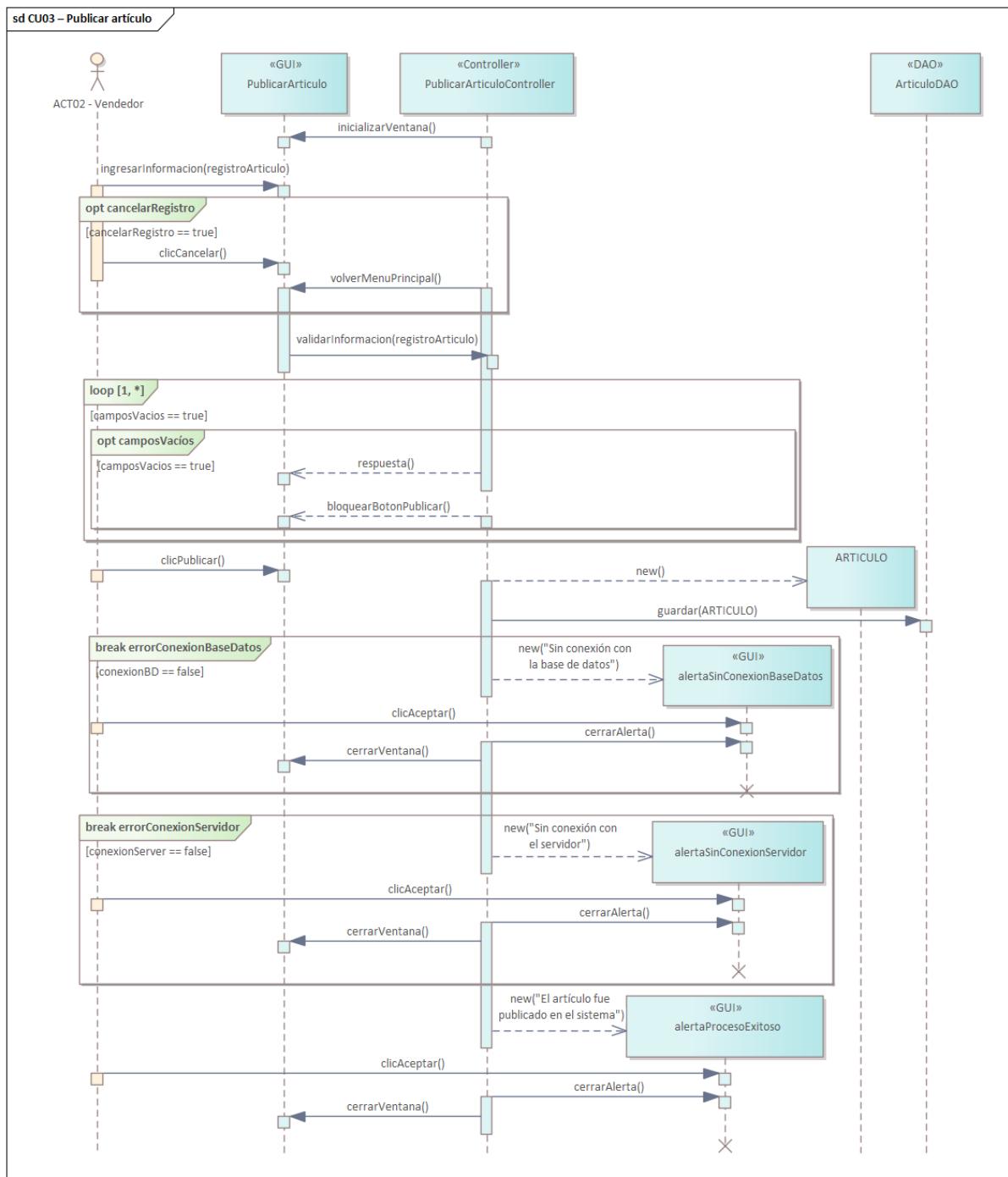
En este apartado se ilustra el Diagrama de Componentes, el cual describe la organización de los módulos de código fuente del sistema y sus dependencias entre estos.



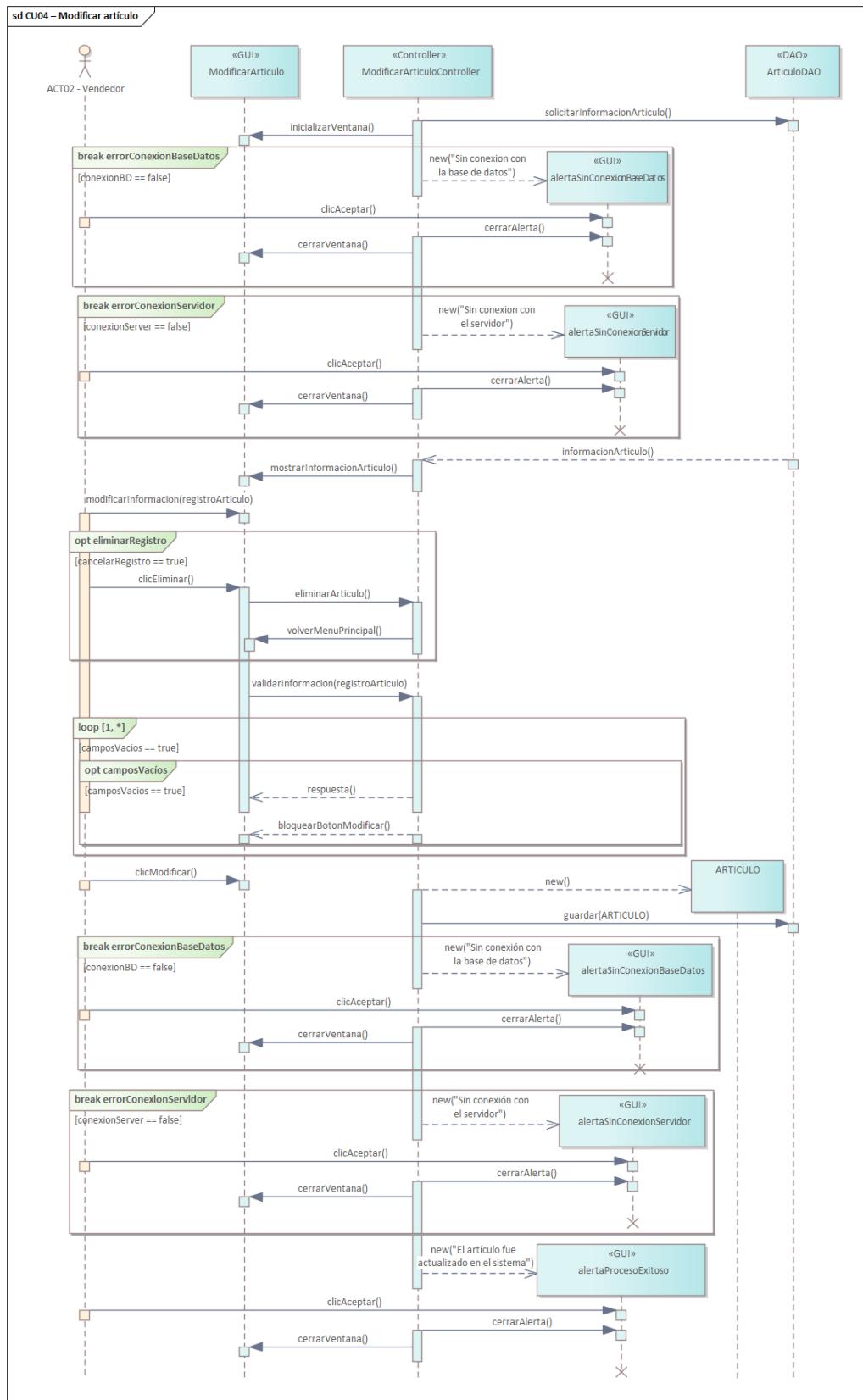
iv. Vista de Procesos (Diagrama UML de Secuencia)

En este apartado se plasma el Diagrama de Secuencia, el cual detalla las interacciones dinámicas entre los objetos del sistema. Para convención de la documentación, se decidió diseñar dichos diagramas únicamente de los procesos que se consideran más relevantes para la lógica de negocio del sistema.

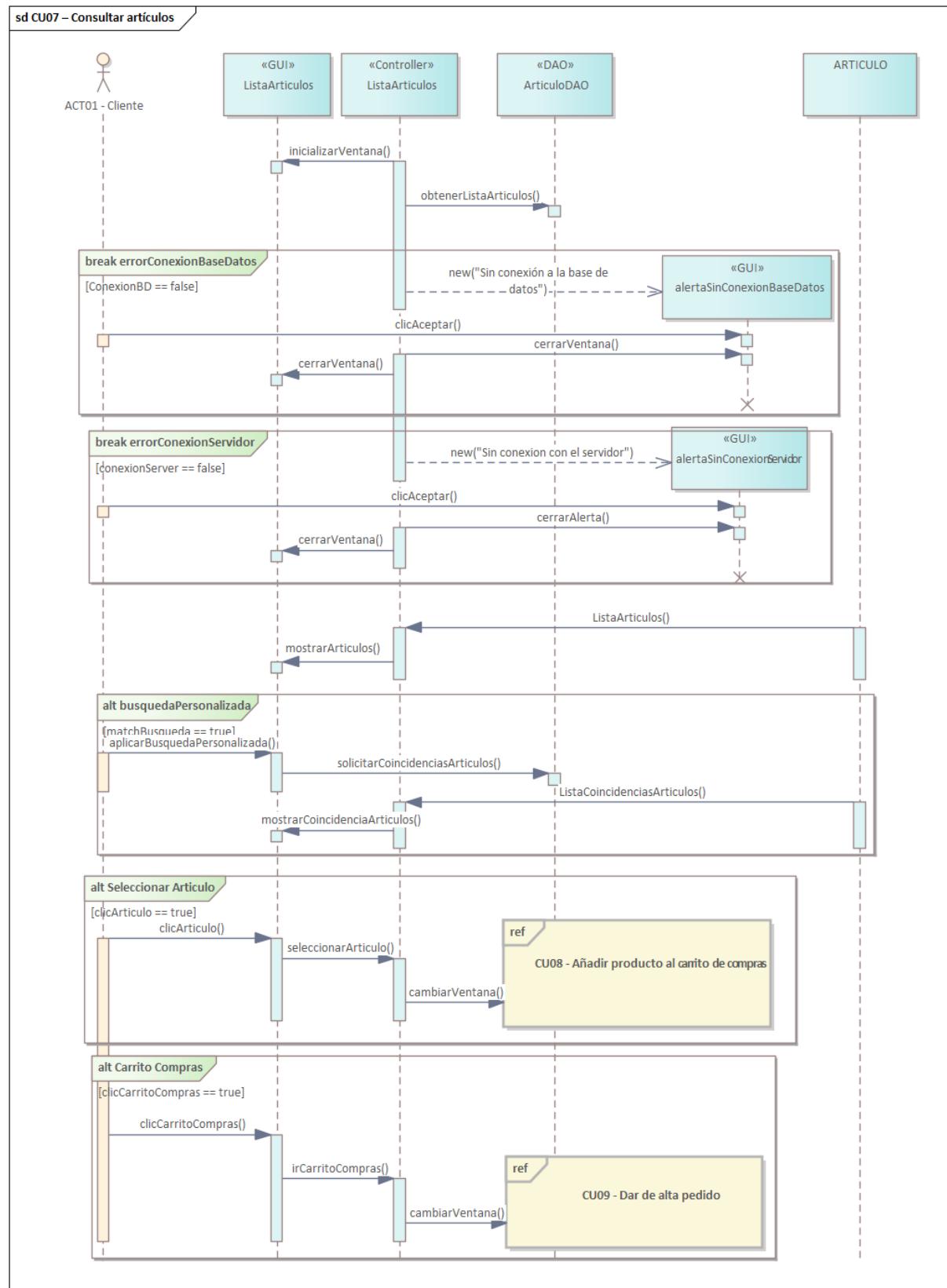
CU03.- Publicar Artículo



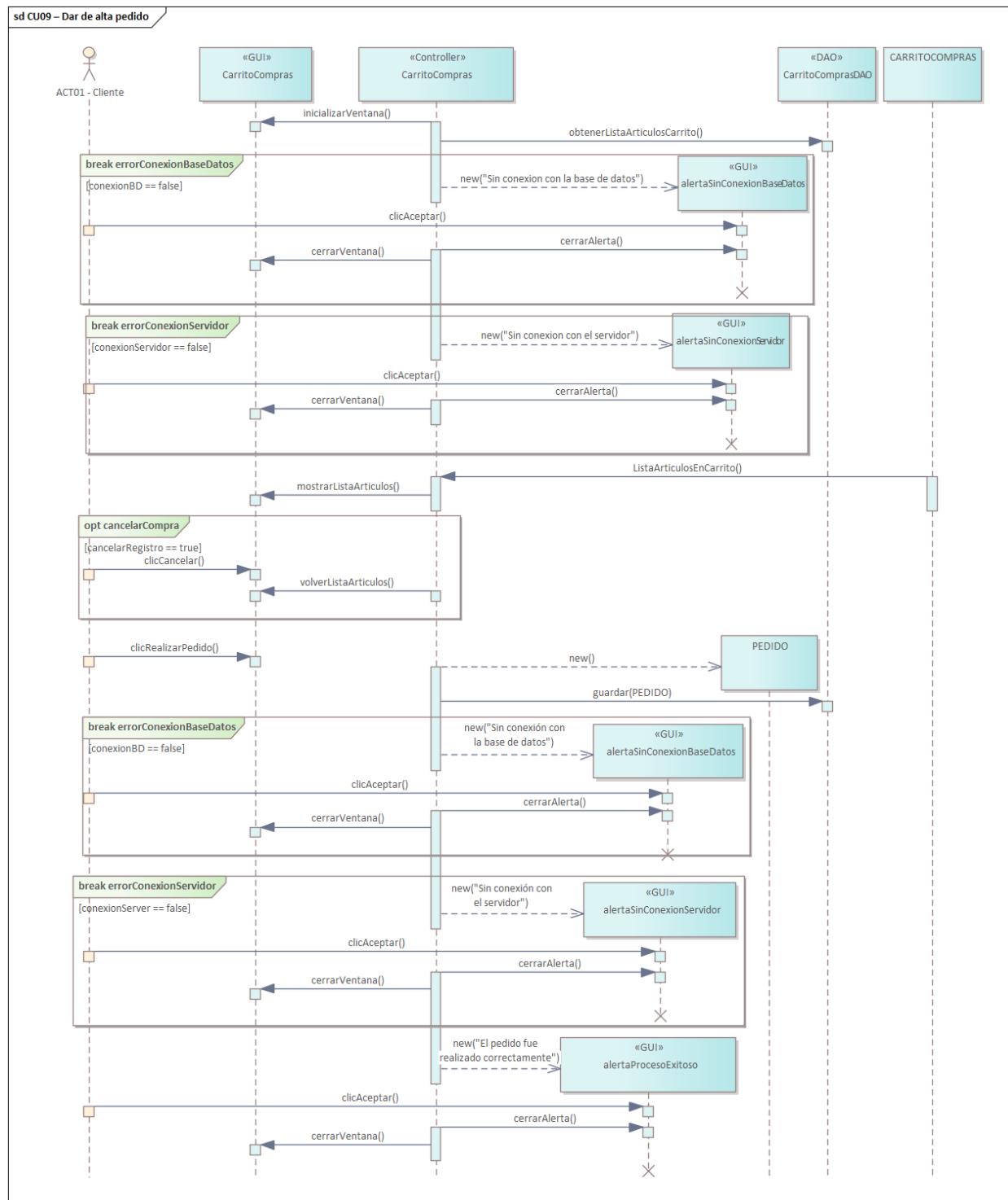
CU04.- Modificar Artículo



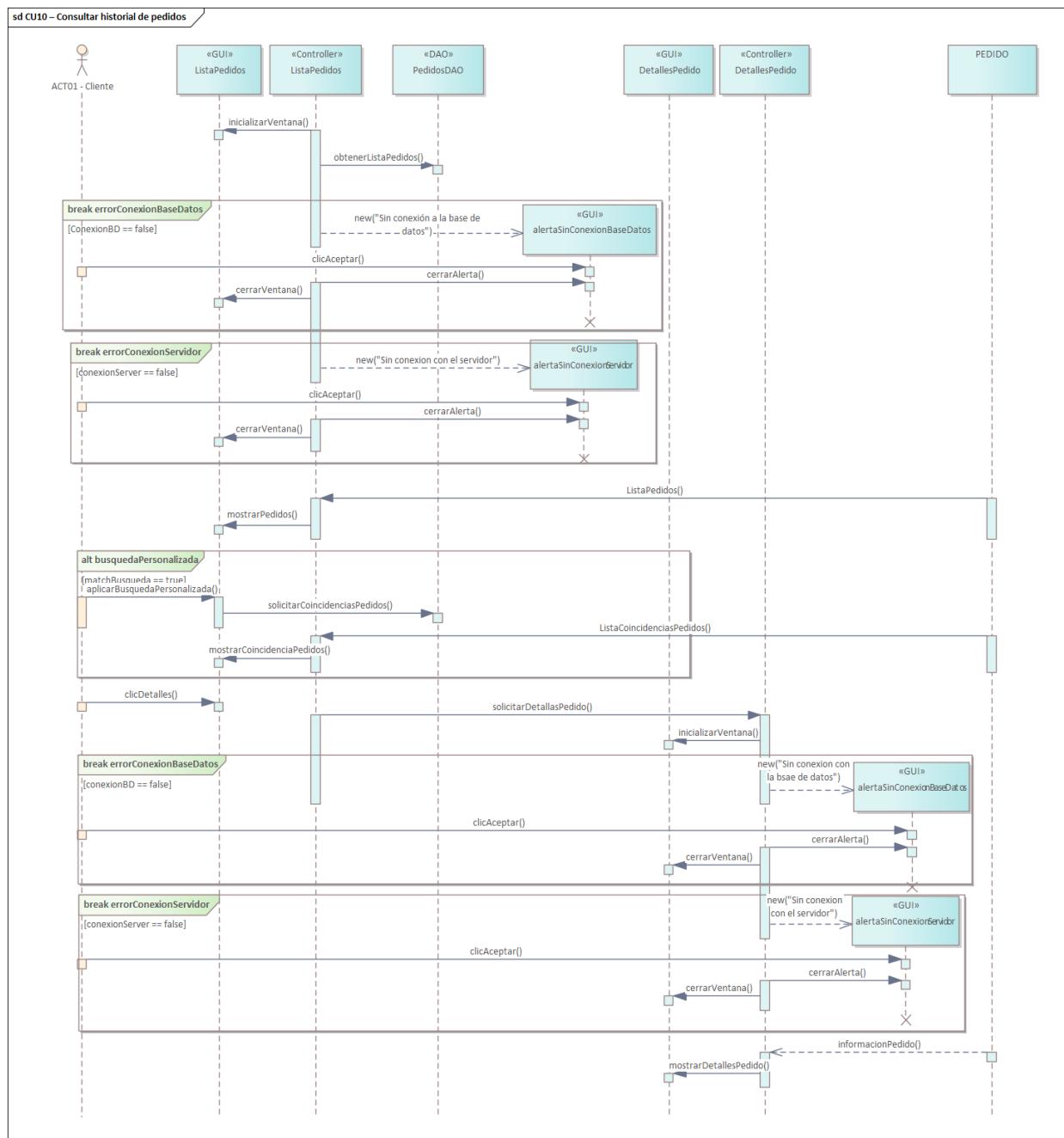
CU07.- Consultar Artículos



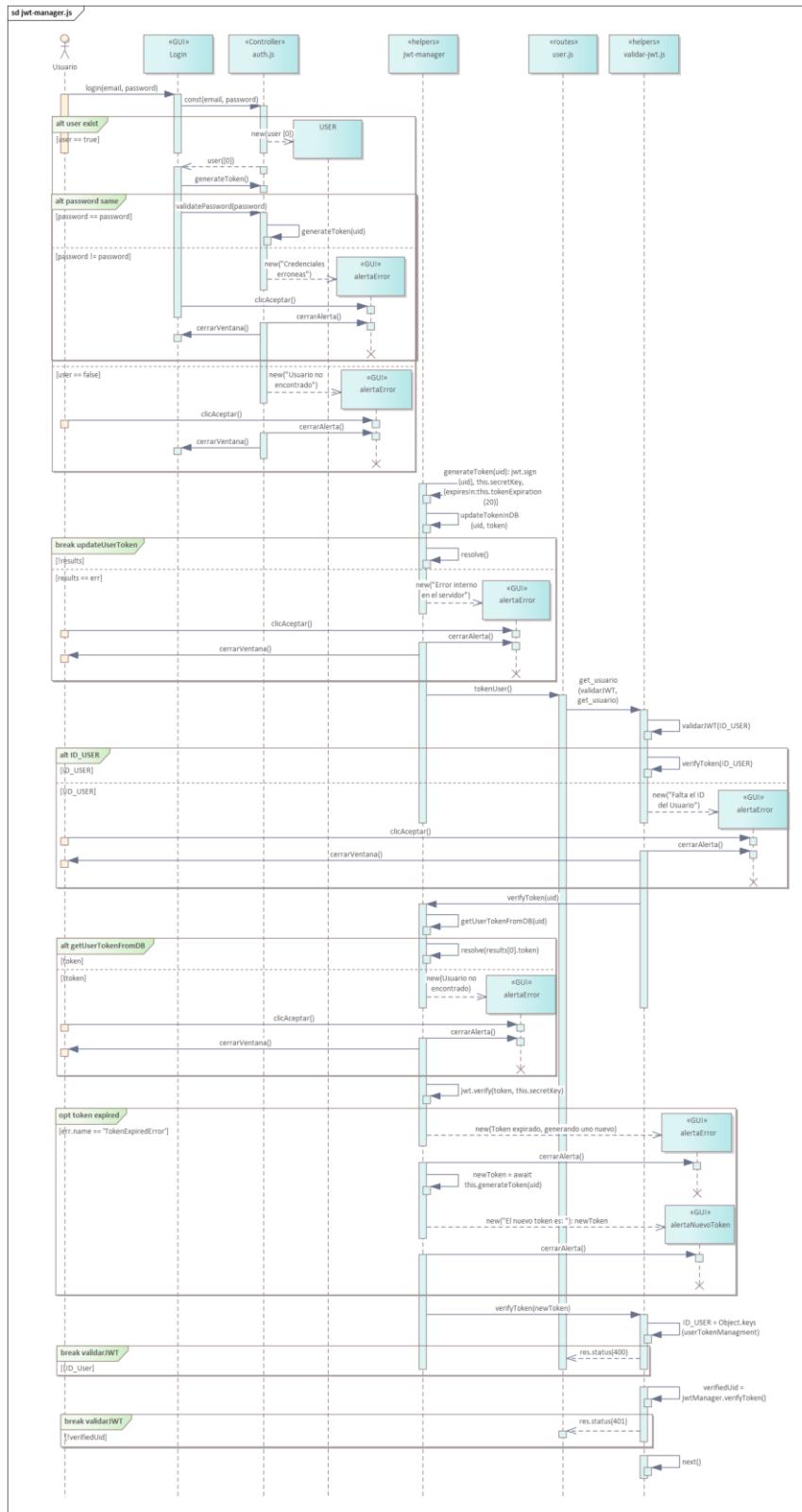
CU09 – Dar de alta pedido



CU10 – Consultar historial de pedidos

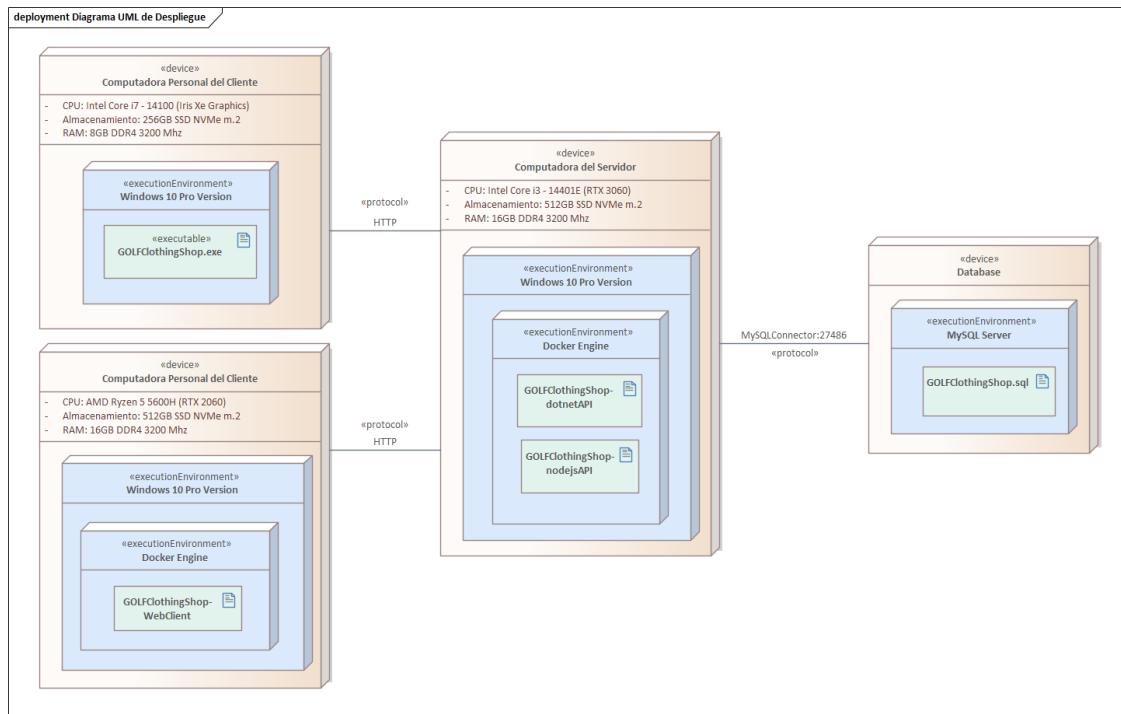


Adicionalmente consideramos modelar el siguiente proceso para inicio de sesión respecto a la generación del JWT:



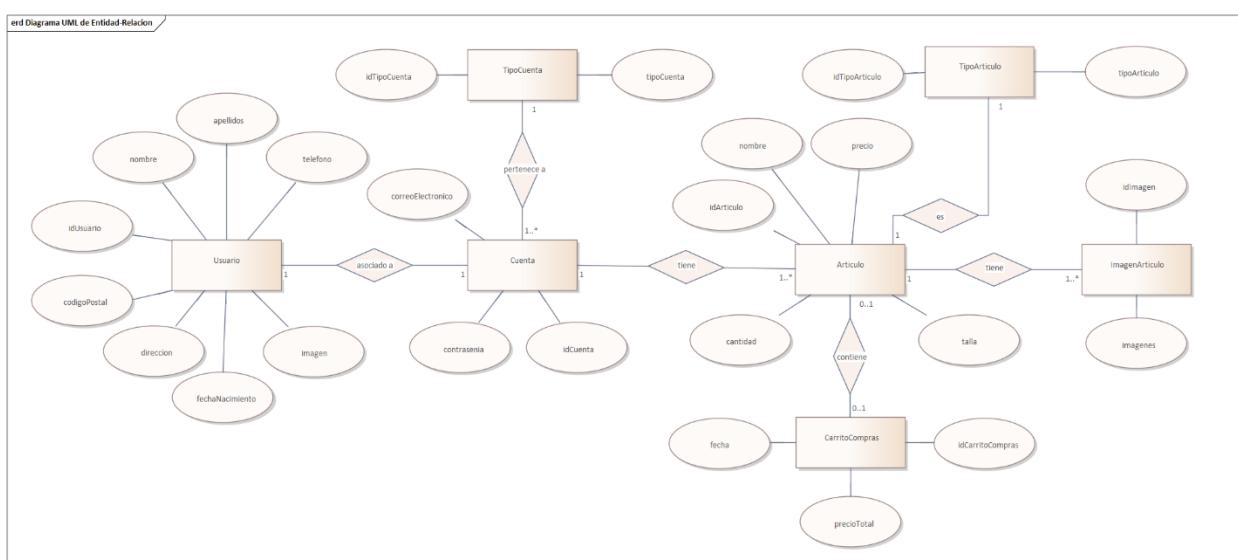
v. Vista de Despliegue (Diagrama UML de Despliegue)

En este apartado se especifica visualmente la distribución física de los componentes en el hardware del sistema.



b. Modelo de Datos (Diagrama UML de Entidad-Relación)

En este apartado se define la estructura y relaciones de la información almacenada para la lógica del sistema.



c. Descripciones de Casos de Uso

En este apartado se detallan las Descripciones de Casos de Uso, las cuales describen los escenarios de interacción entre los usuarios y el sistema, especificando el comportamiento desde la perspectiva del usuario final.

i. CU01 – Dar de alta cuenta

ID:	CU – 01
Nombre del CU:	Dar de alta cuenta
Fecha de actualización:	03/11/2024
Descripción:	El VENDEDOR O CLIENTE da de alta una nueva cuenta en el sistema.
Actor(es):	ACT01 - Cliente ACT02 - Vendedor
Disparador:	El Vendedor o Cliente selecciona el botón Regístrate aquí desde la ventana Inicio sesión.
Precondiciones:	PRE01 – No debe estar registrado un usuario en el sistema con los mismos datos que se planea tener el nuevo usuario.
Flujo Normal:	<ol style="list-style-type: none">1. El sistema muestra la ventana DarAltaUsuario con el formulario en donde se ingresa los siguientes datos: nombre(s), apellido(s), username, tipo de cuenta, correo electrónico y contraseña; junto con el botón Crear Cuenta.2. El Vendedor o Cliente ingresa la información y da clic en el botón Crear Cuenta (FA-01) (FA-02).3. El sistema registra la información del nuevo USUARIO en la base de datos y muestra la ventana RegistroExitoso con el mensaje “El usuario fue agregado al sistema” y el botón Aceptar (EX-01) (EX-02)4. El Vendedor o Cliente da clic en el botón Aceptar.5. Termina el caso de uso.
Flujos Alternos:	FA – 01. El Vendedor o Cliente deja campos en blanco. <ol style="list-style-type: none">1. El sistema no muestra el botón Crear Cuenta.2. Ir al paso 2 del flujo normal. FA – 02. El Vendedor o Cliente ingresa un correo ya registrado. <ol style="list-style-type: none">1. El sistema muestra la ventana CorreoRegistrado con el siguiente mensaje: “El correo con el que tratas registrarte ya se encuentra en el sistema” y marca en rojo el campo de correo electrónico.2. Ir al paso 2 del flujo normal.
Excepciones:	EX – 01. Base de datos sin conexión. <ol style="list-style-type: none">1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde”.2. El Vendedor o Cliente da clic en el botón Aceptar.3. Termina caso de uso. EX – 02. Servidor del sistema sin conexión.

	<ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: "Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde". 2. El Vendedor o Cliente da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	POST – 01. Se guarda la información del USUARIO en la base de datos.
Extiende:	Ninguno.
Incluye:	Ninguno.

ii. CU02 – Modificar cuenta

ID:	CU – 02
Nombre del CU:	Modificar cuenta
Fecha de actualización:	03/11/2024
Descripción:	El VENDEDOR O CLIENTE modifica los datos de un USUARIO que está registrado en la base de datos del sistema.
Actor(es):	ACT01 - Cliente ACT02 - Vendedor
Disparador:	El Vendedor o Cliente selecciona el botón Actualizar desde el apartado de Perfil.
Precondiciones:	PRE01 – Debe estar registrada la información del Usuario a modificar en la base de datos del sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema despliega la ventana ModificarCuenta con los siguientes campos registrados con información y que permiten ser modificados (a excepción del username, tipo de cuenta y correo electrónico): nombre(s), apellido(s), username, tipo de cuenta, fecha de nacimiento, correo electrónico, dirección y código postal; junto con el botón Actualizar perfil. 2. El Vendedor o Cliente edita la información de los campos del USUARIO y da clic en el botón Actualizar perfil (FA-01) (FA-02). 3. El sistema actualiza la información del USUARIO previo en la base de datos y muestra la ventana ProcesoExitoso con el mensaje "Los datos fueron actualizados en el sistema" y el botón Aceptar (EX-01) (EX-02). 4. El Vendedor o Cliente da clic en el botón Aceptar. 5. Termina el caso de uso.
Flujos Alternos:	<p>FA – 01. El Vendedor o Cliente deja campos en blanco.</p> <ol style="list-style-type: none"> 1. El sistema no muestra el botón Actualizar Perfil. 2. Ir al paso 2 del flujo normal. <p>FA – 02. El Vendedor o Cliente selecciona el botón Cancelar.</p> <ol style="list-style-type: none"> 1. El sistema regresa a la ventana MenuPrincipal. 2. Termina flujo alterno.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: "Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde".

	<p>2. El Vendedor o Cliente da clic en el botón Aceptar.</p> <p>3. Termina caso de uso.</p> <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde”. 2. El Vendedor o Cliente da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	POST – 01. Se actualiza la información del USUARIO en la base de datos.
Extiende:	Ninguno.
Incluye:	Ninguno.

iii. CU03 – Publicar artículo

ID:	CU – 03
Nombre del CU:	Publicar artículo
Fecha de actualización:	03/11/2024
Descripción:	El VENDEDOR publica un nuevo artículo por vender en el sistema.
Actor(es):	ACT02 - Vendedor
Disparador:	El Vendedor selecciona el botón Vender desde el apartado de Ventas.
Precondiciones:	PRE01 – No debe estar registrado un artículo en el sistema con los mismos datos que se planea tener el nuevo artículo.
Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema muestra la ventana PublicarArticulo con el formulario en donde se ingresa los siguientes datos: nombre, tipo de artículo, precio, talla y cantidad; junto con el botón Publicar. 2. El Vendedor ingresa la información y da clic en el botón Publicar (FA-01) (FA-02). 3. El sistema registra la información del nuevo ARTICULO en la base de datos y muestra la ventana ProcesoExitoso con el mensaje “El artículo fue publicado en el sistema” y el botón Aceptar (EX-01) (EX-02). 4. El Vendedor da clic en el botón Aceptar. 5. Termina el caso de uso.
Flujos Alternos:	<p>FA – 01. El Vendedor deja campos en blanco.</p> <ol style="list-style-type: none"> 1. El sistema no muestra el botón Publicar. 2. Ir al paso 2 del flujo normal. <p>FA – 02. El Vendedor o Cliente selecciona el botón Cancelar.</p> <ol style="list-style-type: none"> 1. El sistema regresa a la ventana MenuPrincipal. 2. Termina flujo alterno.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde”. 2. El Vendedor da clic en el botón Aceptar.

	<p>3. Termina caso de uso.</p> <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde”. 2. El Vendedor da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	POST – 01. Se guarda la información del ARTÍCULO en la base de datos. POST – 02. Se publica el ARTÍCULO en la ventana Artículos correctamente.
Extiende:	Ninguno.
Incluye:	Ninguno.

iv. CU04 – Modificar artículo

ID:	CU – 04
Nombre del CU:	Modificar artículo
Fecha actualización:	de 03/11/2024
Descripción:	El VENDEDOR modifica los datos de un ARTÍCULO que está registrado en la base de datos del sistema.
Actor(es):	ACT02 - Vendedor
Disparador:	El Vendedor selecciona el botón Modificar desde el apartado de Ventas.
Precondiciones:	PRE01 – Debe estar registrada la información del Artículo a modificar en la base de datos del sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema recupera una lista de ARTICULO(s) y los muestra en una tabla en la ventana ListaArticulos con los datos: imagen del artículo, nombre, precio, categoría, talla y existencias; además, se muestra una barra de búsqueda y una opción para filtrar el contenido (EX-01) (EX-02). 2. El Vendedor selecciona el ARTICULO que desea modificar. 3. El sistema muestra la ventana ModificarArticulo con los siguientes campos registrados con información y que permiten ser modificados: nombre, tipo de artículo, precio, talla y cantidad; junto con los botones Eliminar y Modificar (EX-01) (EX-02). 4. El Vendedor edita la información de los campos del ARTICULO y selecciona el botón Modificar (FA-01) (FA-02). 5. El sistema actualiza la información del ARTICULO previo en la base de datos y muestra la ventana ProcesoExitoso con el mensaje “El artículo fue actualizado en el sistema” y el botón Aceptar (EX-01) (EX-02). 6. El Vendedor da clic en el botón Aceptar. 7. Termina el caso de uso.
Flujos Alternos:	<p>FA – 01. El Vendedor deja campos en blanco.</p> <ol style="list-style-type: none"> 1. El sistema no muestra el botón Modificar. 2. Ir al paso 2 del flujo normal. <p>FA – 02. El Vendedor selecciona el botón Eliminar.</p>

	<ol style="list-style-type: none"> 1. El sistema muestra la ventana ProcesoExitoso con el mensaje “El artículo fue eliminado del sistema” y el botón Aceptar. 2. El Vendedor selecciona el botón Aceptar.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde”. 2. El Vendedor da clic en el botón Aceptar. 3. Termina caso de uso. <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde”. 2. El Vendedor da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	<p>POST – 01. Se actualiza la información del ARTÍCULO en la base de datos.</p> <p>POST – 02. Se publica el ARTÍCULO en la ventana Artículos correctamente.</p>
Extiende:	Ninguno.
Incluye:	Ninguno.

v. CU05 – Consultar inventario

ID:	CU – 05
Nombre del CU:	Consultar inventario
Fecha de actualización:	03/11/2024
Descripción:	El VENDEDOR consulta todos los ARTICULO(s) que este tiene registrado en la base de datos del sistema.
Actor(es):	ACT02 - Vendedor
Disparador:	El Vendedor selecciona el botón Inventario desde el mismo apartado.
Precondiciones:	PRE01 – Debe estar registrada la información de al menos un ARTICULO en la base de datos del sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema recupera una lista de ARTICULO(s) y los muestra en una tabla en la ventana ListaArticulos con los datos: imagen del artículo, nombre, precio, categoría, talla y existencias; además, se muestra una barra de búsqueda, una opción para filtrar el contenido y un ícono para exportar la lista de artículos en un archivo de formato PDF (FA-01) (EX-01) (EX-02). 2. Termina caso de uso.
Flujos Alternos:	<p>FA – 01. El Vendedor realiza una búsqueda personalizada.</p> <ol style="list-style-type: none"> 1. El Vendedor ingresa una o varias palabras clave en el campo de búsqueda o aplica un filtro a mostrar en el contenido de la tabla. 2. Si hay coincidencias, el sistema muestra los elementos de ARTICULO(s) que coincidan con el filtro; en caso de no encontrar coincidencias, el sistema no actualiza la lista.

	<p>FA – 02. El Vendedor selecciona el icono de Exportar a PDF.</p> <ol style="list-style-type: none"> 1. El sistema abre la ventana de ExploradorArchivos del Vendedor. 2. El Vendedor selecciona la ubicación en donde quiere que se guarde dicho archivo PDF (FA-03). 3. El sistema genera el reporte de inventario en formato PDF con los siguientes datos: nombre, código, categoría, talla y lo guarda en la ubicación definida; además, muestra la ventana ProcesoExitoso con el siguiente mensaje: "PDF del inventario fue generado correctamente" (EX-01). 4. El Vendedor selecciona el botón Aceptar. 5. Termina flujo alterno.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: "Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde". 2. El Vendedor da clic en el botón Aceptar. 3. Termina caso de uso. <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: "Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde". 2. El Vendedor da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	POST – 01. El sistema genera el Reporte de Inventario en formato PDF correctamente.
Extiende:	Ninguno.
Incluye:	Ninguno.

vi. CU06 – Consultar ventas realizadas

ID:	CU – 06
Nombre del CU:	Consultar ventas realizadas
Fecha de actualización:	03/11/2024
Descripción:	El VENDEDOR consulta todos las VENTA(s) realizadas y que están registradas en la base de datos del sistema.
Actor(es):	ACT02 - Vendedor
Disparador:	El Vendedor selecciona el botón Ganancias desde el apartado Ventas.
Precondiciones:	PRE01 – Debe estar registrada la información de al menos una VENTA en la base de datos del sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema recupera una lista de VENTA(s) referentes a un ARTICULO y los muestra en una tabla en la ventana ListaVentas con los datos: imagen del artículo, nombre, número de artículos vendidos, precio, fecha de inicio, fecha de corte, numero de artículos totales vendidos y total de ganancias;

	<p>además, se muestra una barra de búsqueda, una opción para filtrar el contenido y un ícono para exportar la lista de ventas en un archivo de formato PDF (FA-01) (EX-01) (EX-02).</p> <ol style="list-style-type: none"> 2. Termina caso de uso.
Flujos Alternos:	<p>FA – 01. El Vendedor realiza una búsqueda personalizada.</p> <ol style="list-style-type: none"> 1. El Vendedor ingresa aplica un filtro a mostrar en el contenido de la tabla. 2. Si hay coincidencias, el sistema muestra los elementos de ARTICULO(s) que coincidan con el filtro; en caso de no encontrar coincidencias, el sistema no actualiza la lista.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde”. 2. El Vendedor da clic en el botón Aceptar. 3. Termina caso de uso. <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde”. 2. El Vendedor da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	POST – 01. El sistema genera el Reporte de Inventario en formato PDF correctamente.
Extiende:	Ninguno.
Incluye:	Ninguno.

vii. CU07 – Consultar artículos

ID:	CU – 07
Nombre del CU:	Consultar artículos
Fecha de actualización:	03/11/2024
Descripción:	El Cliente consulta todos los ARTICULO(s) disponibles para comprar en el sistema.
Actor(es):	ACT01 - Cliente
Disparador:	El Cliente selecciona el botón Inicio desde el mismo apartado.
Precondiciones:	PRE01 – Debe estar registrada la información de al menos un ARTICULO en la base de datos del sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema recupera una lista de ARTICULO(s) y los muestra en una tabla en la ventana ListaArticulos con los datos: imagen del artículo, nombre, precio, categoría a la que pertenece, talla y cantidad en existencia; además, se muestra una barra de búsqueda, una opción para filtrar el contenido y un ícono para dirigirse al Carrito de Compras (EX-01) (EX-02). 2. El Cliente visualiza la lista de artículos disponibles para comprar (FA-01) (FA-02) (FA-03).

	<p>3. Termina caso de uso.</p>
Flujos Alternos:	<p>FA – 01. El Cliente realiza una búsqueda personalizada en el campo de búsqueda.</p> <ol style="list-style-type: none"> 1. El Cliente ingresa una o varias palabras clave en el campo de búsqueda o aplica un filtro a mostrar en el contenido de la tabla. 2. Si hay coincidencias, el sistema muestra los elementos de ARTICULO(s) que coincidan con la(s) palabra(s) clave; en caso de no encontrar coincidencias, el sistema no actualiza la lista. <p>FA – 02. El Cliente selecciona un ARTICULO para comprar.</p> <ol style="list-style-type: none"> 1. Ir al CU08 – Añadir producto al carrito de compras. 2. Termina caso de uso. <p>FA – 03. El Cliente selecciona el ícono de Carrito de Compras.</p> <ol style="list-style-type: none"> 1. Ir al CU09 – Dar de alta pedido. 2. Termina caso de uso.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde”. 2. El Cliente da clic en el botón Aceptar. 3. Termina caso de uso. <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde”. 2. El Cliente da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	<p>POST – 01. El sistema recupera la información de los PEDIDO(s) de PROVEEDOR correctamente.</p> <p>POST – 02. El sistema limpia el campo de búsqueda correctamente.</p>
Extiende:	<p>CU08 – Añadir producto al carrito de compras</p> <p>CU09 – Dar de alta pedido</p>
Incluye:	Ninguno.

viii. CU08 – Añadir producto al carrito de compras

ID:	CU – 08
Nombre del CU:	Añadir producto al carrito de compras extendido de CU07 – Consultar artículos
Fecha de actualización:	03/11/2024
Descripción:	El Cliente añade un ARTICULO al CARRITO DE COMPRAS para proceder a comprarlo.
Actor(es):	ACT01 - Cliente
Disparador:	El Cliente selecciona un artículo desde la tabla ListaArticulos.
Precondiciones:	PRE01 – Debe estar registrada la información de al menos un ARTICULO en la base de datos del sistema.

Flujo Normal:	<ol style="list-style-type: none"> El sistema muestra la ventana DetallesArticulo con el formulario en donde se muestran los siguientes datos: nombre, tipo de artículo, precio, talla y cantidad; junto con el botón Agregar al carrito (EX-01) (EX-02). El Cliente selecciona el botón Agregar al carrito (FA-01). El sistema añade el ARTICULO a la cola del CARRITO DE COMPRAS y vuelve a la ventana ListaArticulos (EX-01) (EX-02). Termina el caso de uso.
Flujos Alternos:	<p>FA – 01. El Cliente selecciona el icono Regresar.</p> <ol style="list-style-type: none"> El sistema vuelve a la ventana ListaArticulos. Termina el caso de uso.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde”. El Cliente da clic en el botón Aceptar. Termina caso de uso. <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde”. El Cliente da clic en el botón Aceptar. Termina caso de uso.
Postcondiciones:	POST – 01. Se añade el artículo a la cola del carrito de compras.
Extiende:	Ninguno.
Incluye:	Ninguno.

ix. CU09 – Dar de alta pedido

ID:	CU – 09
Nombre del CU:	Dar de alta pedido extendido de CU07 – Consultar artículos
Fecha de actualización:	03/11/2024
Descripción:	El Cliente consulta el CARRITO DE COMPRAS y procede a pagar por los ARTICULO(s) que tiene agregados a la cola.
Actor(es):	ACT01 - Cliente
Disparador:	El Cliente selecciona el icono de Carrito de Compras.
Precondiciones:	PRE01 – Debe estar registrada la información de al menos un ARTICULO en la base de datos del sistema.
Flujo Normal:	<ol style="list-style-type: none"> El sistema recupera una lista de ARTICULO(s) agregados a la cola del CARRITO DE COMPRAS y los muestra en una tabla en la ventana CarritoCompras con los datos: imagen del artículo, nombre, precio, cantidad (con los botones + y -), talla y cantidad en existencia; junto a la cantidad total a pagar y los botones Realizar Pedido e Historial de pedidos (EX-01) (EX-02).

	<ol style="list-style-type: none"> 2. El Cliente selecciona el botón Realizar Pedido (FA-01). 3. El sistema registra la información del PEDIDO previo en la base de datos; además, muestra la ventana ProcesoExitoso con el siguiente mensaje: "El pedido fue realizado correctamente" (EX-01) (EX-02). 4. El Cliente selecciona el botón Aceptar. 5. Termina caso de uso.
Flujos Alternos:	<p>FA – 01. El Cliente selecciona el botón Cancelar.</p> <ol style="list-style-type: none"> 1. El sistema vacía la cola del carrito de compras y vuelve a la ventana ListaArticulos. 2. Termina el caso de uso. <p>FA – 02. Se agotó la cantidad disponible de un artículo de la cola del Carrito de Compras.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ArticuloAgotado con el siguiente mensaje: "Uno de los artículos que quieras pedir se encuentra agotado". 2. El Cliente da clic en el botón Aceptar.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: "Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde". 2. El Cliente da clic en el botón Aceptar. 3. Termina caso de uso. <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: "Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde". 2. El Cliente da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	POST – 01. Se guarda la información del PEDIDO en la base de datos.
Extiende:	CU10 – Consultar historial de pedidos.
Incluye:	Ninguno.

x. CU10 – Consultar historial de pedidos

ID:	CU – 10
Nombre del CU:	Consultar historial de pedidos
Fecha de actualización:	03/11/2024
Descripción:	El Cliente consulta los PEDIDOS que ha realizado en el sistema.
Actor(es):	ACT01 - Cliente
Disparador:	El Cliente selecciona el botón Historial de pedidos de la ventana CarritoCompras.
Precondiciones:	<p>PRE01 – Debe estar registrada la información de al menos un ARTICULO en la base de datos del sistema.</p> <p>PRE02 – Debe estar registrada la información de al menos un PEDIDO en la base de datos del sistema.</p>

Flujo Normal:	<ol style="list-style-type: none"> 1. El sistema recupera una lista de PEDIDOS(s) realizados y los muestra en una tabla en la ventana ListaPedidos con los datos: fecha de realización, costo total pagado, número de artículos compras y el botón Detalles; además, se muestra una opción para filtrar el contenido por fecha (EX-01) (EX-02). 2. El Cliente selecciona el botón Detalles de un ARTICULO (FA-01) (FA-01). 3. El sistema recupera la información del(los) ARTICULO(s) que contiene el PEDIDO seleccionado y lo muestra en una tabla en la ventana DetallesPedido con los datos: imagen del artículo, nombre, precio, cantidad seleccionada y talla (FA-01) (EX-02). 4. Termina caso de uso.
Flujos Alternos:	<p>FA – 01. El Cliente realiza una búsqueda personalizada en el campo de búsqueda.</p> <ol style="list-style-type: none"> 1. El Cliente ingresa una o varias palabras clave en el campo de búsqueda o aplica un filtro a mostrar en el contenido de la tabla. 2. Si hay coincidencias, el sistema muestra los elementos de ARTICULO(s) que coincidan con la(s) palabra(s) clave; en caso de no encontrar coincidencias, el sistema no actualiza la lista. <p>FA – 02. El Cliente selecciona el icono Regresar.</p> <ol style="list-style-type: none"> 1. El sistema vuelve a la ventana ListaPedidos. 2. Termina el caso de uso.
Excepciones:	<p>EX – 01. Base de datos sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana BaseDatosSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar a la base de datos. Inténtelo más tarde”. 2. El Cliente da clic en el botón Aceptar. 3. Termina caso de uso. <p>EX – 02. Servidor del sistema sin conexión.</p> <ol style="list-style-type: none"> 1. El sistema muestra la ventana ServidorSinConexion con el siguiente mensaje: “Ocurrió un error al tratar de conectar al servidor del sistema. Inténtelo más tarde”. 2. El Cliente da clic en el botón Aceptar. 3. Termina caso de uso.
Postcondiciones:	<p>POST – 01. El sistema recupera la información de los PEDIDO(s) correctamente.</p> <p>POST – 02. El sistema limpia el campo de búsqueda correctamente.</p>
Extiende:	Ninguno.
Incluye:	Ninguno.

IV. Construcción

a. Selección justificada de la pila tecnológica

i. Aplicación de clientes (móvil y web)

Para la creación del cliente móvil, se decidió utilizar Android Studio con el lenguaje de programación Java, mientras que para el cliente de escritorio se optó por Visual Studio con .NET y C#. Estas elecciones se justifican por las siguientes razones:

Cliente Móvil: Android Studio con Java + Kotlin

- **Compatibilidad y soporte:** Android Studio es la herramienta oficial para el desarrollo de aplicaciones Android, garantizando compatibilidad con la mayoría de los dispositivos y un alto soporte por parte de la comunidad y Google.
- **Flexibilidad y personalización:** El uso de Java como lenguaje principal permite un control detallado sobre la funcionalidad y diseño de la aplicación, siendo ideal para satisfacer los requisitos específicos del proyecto.

Cliente Web: Visual Studio con .NET y C#

- **Rendimiento:** .NET es conocido por su alto rendimiento y optimización en aplicaciones de escritorio, lo que garantiza que el cliente funcione de manera eficiente incluso en equipos de gama media-baja.
- **Herramientas avanzadas:** Visual Studio ofrece un entorno de desarrollo integrado con componentes gráficos robustos y bibliotecas adicionales, permitiendo la creación de interfaces avanzadas y funcionalidades específicas requeridas por los requisitos funcionales.
- **Experiencia previa del equipo:** Al tratarse de una tecnología que el equipo de desarrollo ya ha utilizado en proyectos anteriores, se espera un proceso de implementación más rápido y eficiente, lo que permite enfocar más recursos en el desarrollo de APIs web y otros componentes del sistema.

ii. Aplicación de servicios (API) – Servidor

Para la creación de la API necesaria para el proyecto, se optó por utilizar **Node.js** con **Gradle** como herramientas principales para el desarrollo, **MySQL Workbench** para la gestión de bases de datos, y **Postman** como herramienta para pruebas. Esta elección se justificó por las siguientes razones:

Node.js

- ✓ **Alto rendimiento y escalabilidad:** Node.js permite manejar múltiples solicitudes simultáneamente gracias a su modelo de programación asíncrono y basado en eventos. Esto lo convierte en una opción ideal para construir APIs rápidas y eficientes, incluso bajo alta carga.

- ✓ **Gran ecosistema de paquetes:** Con la gestión de paquetes de npm (Node Package Manager), Node.js ofrece una amplia gama de bibliotecas que facilitan la implementación de funcionalidades, desde la autenticación de usuarios hasta la manipulación de datos.
- ✓ **Experiencia previa del equipo:** Algunos miembros del equipo ya tenían experiencia trabajando con Node.js gracias a la Experiencia Educativa de Programación Segura, lo que permitió un aprendizaje más rápido y una implementación más eficiente de esta tecnología.

Gradle

- ✓ **Automatización del flujo de trabajo:** Gradle permite gestionar tareas repetitivas, como la instalación de dependencias, compilación y ejecución de pruebas, lo que optimiza el tiempo de desarrollo.
- ✓ **Flexibilidad y personalización:** Su capacidad para adaptarse a proyectos de cualquier tamaño y complejidad lo hace ideal para mantener un flujo de trabajo organizado en proyectos basados en Node.js.

MySQL Workbench

- ✓ **Estabilidad y rendimiento:** MySQL es conocido por su capacidad de manejar grandes volúmenes de datos de forma eficiente, garantizando una respuesta rápida a las solicitudes de la API.
- ✓ **Interfaz gráfica intuitiva:** MySQL Workbench ofrece una herramienta visual que simplifica la gestión de esquemas, consultas y datos, lo cual reduce la complejidad durante el desarrollo y mantenimiento de la base de datos.
- ✓ **Experiencia previa del equipo:** Todos los miembros del equipo ya habían trabajado con MySQL Workbench en las Experiencias Educativas de Principios de Construcción de Software y Tecnologías para la Construcción de Software, lo que facilitó el diseño e implementación de la base de datos del proyecto.

Postman

- ✓ **Facilidad de uso:** Proporciona una interfaz intuitiva para crear, enviar y depurar solicitudes HTTP, lo que agiliza la verificación de funcionalidad y desempeño de la API.
- ✓ **Capacidades avanzadas:** Ofrece funcionalidades como la creación de colecciones de pruebas y generación de documentación automática, mejorando la calidad del proceso de pruebas.

b. Estándar de codificación

Propósito	Promover prácticas de programación que resulten en un software seguro, fiable, verificable y de fácil mantenimiento.
Idioma	Se ha decidido usar el idioma Inglés para la nomenclatura utilizada en este estándar; la decisión fue tomada por todos los miembros del equipo con fines académico y de mejor nuestro aprendizaje de este.

Estructura de la API	<p>La aplicación de servicios (API) se ha estructurado en tres capas principales: Servicios, Negocios y Datos.</p> <p>La capa de Servicios actúa como el punto de entrada principal al sistema, implementando servicios RESTful y RPC que permiten abordar la heterogeneidad y apertura de los sistemas distribuidos. Además, esta capa incluye pruebas unitarias y de integración que garantizan al menos un 70% de cobertura funcional, asegurando la calidad de los servicios expuestos.</p> <p>La capa de Negocios centraliza las reglas y la lógica del dominio, procesando las solicitudes de forma eficiente y manejando retos clave como la concurrencia, el desempeño y la transparencia.</p> <p>Por su parte, la capa de Datos gestiona el acceso a las bases de datos, utilizando tecnologías relacionales como MySQL o NoSQL según los requerimientos específicos. Esta capa ha sido diseñada para optimizar el rendimiento y soportar alta concurrencia.</p> <p>Todo el sistema aborda requisitos de calidad críticos, como seguridad (implementada mediante autenticación y autorización), desempeño y disponibilidad, respaldado por pruebas de tolerancia a fallos.</p>
Nombrado de clases	<p>Las Clases deben tener como nombre un sustantivo o una frase que haga alusión al mismo (persona, cosa, etc.) y empezar con letra mayúscula. Se debe evitar nombre con verbos.</p> <p>Se ocupará como formato el PascalCamelCase para el nombrado de clases. Como ejemplo de nombrado de clases se tiene:</p> <pre>class User{...} class Clothes{...} class Cart{...}</pre>
Nombrado de componentes de interfaz	<p>Los componentes de una interfaz gráfica tales como botones, campos de texto, etiquetas, etc. deben de empezar con una abreviación del componente.</p> <p>Para abbreviar un componente de una sola palabra, se deben quitar sus vocales y ocupar las tres primeras consonantes.</p> <p>Para abbreviar un componente formado de dos palabras: de la primera palabra, se deben quitar sus vocales y ocupar las tres primeras consonantes, y juntar esto con la segunda palabra, empezando por mayúscula.</p> <p>Se ocupará el formato lowerCamelCase.</p> <p>En el lenguaje de programación C# usando .NET tenemos los siguientes ejemplos:</p> <pre>Label x:Name="txtTitleLogin" Label x:Name="txtPassword" Label x:Name="txtUsername" Label x:Name="txtEmail"</pre> <p>Para el caso del IDE de Android Studio usando Java tenemos los siguientes ejemplos:</p> <pre>Label x:Name="txtTitleLogin" Label x:Name="txtPassword" Label x:Name="txtUsername" Label x:Name="txtEmail"</pre>
Nombrado de constantes	<p>Las constantes del programa deben tener como nombre un sustantivo que haga alusión a el mismo y se escribirán con mayúsculas, además serán nombradas con el formato lowerCamelCase.</p> <p>En los lenguajes de programación de JavaScript, Java y C# tenemos los siguientes ejemplos:</p> <ul style="list-style-type: none"> - JavaScript: <code>const payload = { uid };</code> - C#: <code>private static final USERS = 3;</code> - Java: <code>private static final PRODUCT_NAME = "WolfGang Tshirt 01";</code>
Nombrado de métodos	<p>Los métodos de deben ser verbos o frases que hagan alusiones a verbos y se ocuparan prefijos en caso de ser necesario, además se utilizara el formato lowerCamelCase.</p>

	<p>En los lenguajes de programación de JavaScript y C# tenemos los siguientes ejemplos:</p> <ul style="list-style-type: none"> - JavaScript: <code>const validarJWT = (req, res, next) => {}</code> - C#: <code>public bool createUser(User newUser){...}</code> - Java: <code>public boolean createClothe(Clothes newClothe){...}</code>
Indentación	<p>La indentación (o mejor conocido como <i>sangría</i>) es utilizado para una mejor lectura y comprensión del código. Esta debería seguir las siguientes características:</p> <ul style="list-style-type: none"> • No hay un salto de línea antes de abrir una llave. • Hay un salto de línea después de la llave de cierre, solo si esa llave termina una declaración o el cuerpo de un método, constructor o clase con nombre. Por ejemplo, no hay salto de línea después de la llave si va seguido de un <code>else</code> o <code>";"</code>. • Poner una sola instrucción por línea. • Dejar un espacio entre cada uno de los elementos de una expresión. • Agregar una tabulación en las instrucciones que están dentro de estructuras de control, métodos o bloques específicos de código. <p>Como ejemplo se tiene el siguiente bloque de código:</p> <pre>public class example { public void method() { if(condition()) { try{ something(); }catch (ProblemException e) { recover(); } }else if (otherCondition()) { somethingElse(); }else { lastThing(); } } };</pre>
Comentarios	<p>Los comentarios en programación se utilizan para poner aclaraciones del código, y así es más fácil de entender lo que hace.</p> <ol style="list-style-type: none"> Explicar el por qué, no el cómo. Documentar si existe algo inusual o inesperado dentro del código. Evitar palabras altisonantes, chistes, bromas o cualquier comentario innecesario. <p>Un buen comentario describe el que hace cierta parte del código y no el cómo. Todos los comentarios deben ser claros y consistentes, utilizando mismo estilo.</p> <p>Como ejemplo se tiene el siguiente:</p> <pre>// Verify if user has access to every module. if ((user.Type == Client user.Type == Seller) && user.IsActive) { }</pre> <p>En un mal comentario se invierte mucho tiempo en generar cajas de comentarios “bonitas”, esto hace que se pierda mucho tiempo intentando acomodar cada caja cada vez que el comentario sea editado.</p> <p>Como ejemplo se tiene el siguiente:</p> <pre>// <summary> // User Class // </summary> public class User {</pre>

	<pre> // <summary> // User Type // </summary> public string Type { get; set; } // <summary> // Verifies if user is active // </summary> public bool IsActive { get; set; } } </pre>
Manejo de excepciones	Para las excepciones se van a tomar en cuenta estos puntos: <ul style="list-style-type: none"> • En el Cliente, se va a utilizar en la medida de lo posible en los métodos manejadores de acciones de los componentes gráficos, como <code>selectUsersButtonOnClick</code>, <code>usersTextFieldOnTextChanges</code>, etc. • En la API, se van a utilizar en los métodos que van a ser expuestos para el uso de los clientes. • Se puede usar la clase <code>exception</code>.
Nombrado de pruebas unitarias	Para el nombrado de pruebas unitarias, simplemente se van a nombrar con el nombre del método a probar pero al final con la palabra “Test”. Ejemplos: <ul style="list-style-type: none"> • <code>public void getAccountByIdTest</code> • <code>public void modifyUserTest</code> • <code>public void deleteAccountTest</code>

c. Prácticas de construcción realizadas

A lo largo del desarrollo del proyecto se implementaron las siguientes prácticas de construcción:

- **Organización del Equipo:** El equipo alternaba entre trabajar juntos en un mismo lugar y de manera individual, pero siempre mantenía una comunicación constante. Esto garantizó una coordinación efectiva y ayudó a evitar problemas durante el desarrollo del proyecto.
- **Control de Versiones:** Se utilizó el sistema de control de versiones **Git**, acompañado de la plataforma **GitHub** para alojar el código. Cada vez que un miembro del equipo realizaba un cambio significativo, se hacía un `commit`, lo que permitió mantener un historial detallado de modificaciones. Esto facilitaba recuperar funcionalidades descartadas previamente en caso de ser necesario, simplemente consultando el historial de `commits`.
- **Revisión de Código:** Cada vez que un integrante subía cambios al repositorio y estos se integraban con los de otro miembro, se realizaban revisiones para detectar defectos, proponer modificaciones y asegurar la implementación correcta de funcionalidades. Estas revisiones se llevaban a cabo tanto de forma presencial, cuando el equipo trabajaba en un mismo lugar, como de manera remota, destacando su importancia para prevenir errores comunes.
- **Integración:** La integración del código se gestionó mediante **Git** y **GitHub Desktop**, utilizando comandos como `git pull`, `git fetch` y `git merge` a

través de **GitHub Desktop**. Estas integraciones se realizaban después de implementar funcionalidades importantes en el proyecto.

- **Gestión de la Configuración:** Para gestionar las configuraciones del código, como las conexiones entre la **API**, el **Ciente Rico** y la **base de datos**, se usaron archivos específicos como **.env**, **.properties** y **requirements.txt**. En cuanto a la documentación, se emplearon artefactos como diagramas de despliegue para abordar aspectos específicos de configuración.
- **Gestión de Dependencias:** En el **Cliente**, se utilizó el gestor de dependencias **Maven** para incluir las bibliotecas necesarias. Para la **API**, se emplearon herramientas como **npm** en **Node.js**, facilitando la instalación y administración de dependencias requeridas.
- **Reutilización de Código:** Dado que el proyecto utilizaba tecnologías y casos de uso similares a los de desarrollos previos, se reutilizó código de repositorios de GitHub de proyectos anteriores, adaptándolo al sistema actual. Esto redujo considerablemente el tiempo necesario para implementar algunas funcionalidades.
- **Programación Defensiva:** El equipo se enfocó en la implementación de medidas de seguridad, como validación de entradas, manejo de errores y excepciones, autenticación y protección de datos sensibles, asegurando un sistema robusto y confiable.
- **Depuración:** Para depurar, se utilizó el depurador integrado en **Visual Studio Code**, **Visual Studio** y **Android Studio**.
- **Pruebas:** Se llevaron a cabo tres tipos de pruebas: **pruebas unitarias** (utilizando para probar funciones individuales), **pruebas de integración** (para verificar los flujos entre distintos componentes del sistema) y **pruebas del sistema** (realizadas al final del desarrollo para evaluar flujos completos de los casos de uso).

V. Pruebas

a. Plan de pruebas para las aplicaciones móvil y web

i. Objetivo del plan:

El objetivo principal del plan de pruebas es garantizar que todas las funcionalidades del sistema, descritas en el apartado de Requerimientos, sean implementadas correctamente y cumplan con los estándares de calidad establecidos. Adicionalmente, se busca asegurar una experiencia de usuario fluida y accesible, fortaleciendo la usabilidad del sistema. En términos de seguridad, el plan garantizará la protección de los datos sensibles mediante la implementación de mecanismos robustos de autenticación y autorización.

ii. Alcance de las pruebas:

Las pruebas abarcarán todas las funcionalidades principales del sistema. Esto incluye procesos críticos como el inicio de sesión, la gestión de cuentas de usuario, la publicación

y modificación de artículos, la consulta de inventarios y ventas, así como las operaciones relacionadas con el carrito de compras y los pedidos.

iii. Tipos de pruebas:

El enfoque del sistema se basa en dos niveles de pruebas:

- **Pruebas unitarias por caso de uso (CU):** Estas validarán la funcionalidad individual de cada componente descrito en los requerimientos, asegurando que las unidades específicas del sistema cumplan con su propósito.
- **Pruebas de integración:** Este nivel permitirá evaluar la interacción entre los diferentes componentes del sistema para garantizar que funcione como un todo coherente, cumpliendo con los requerimientos funcionales y no funcionales.

iv. Entorno de pruebas:

El entorno de pruebas está diseñado para simular condiciones reales de uso, abarcando diferentes tipos de dispositivos y configuraciones de hardware y software. Esto permitirá evaluar el rendimiento y la funcionalidad del sistema en una amplia variedad de escenarios.

Dispositivos móviles

Las pruebas se realizarán en dos modelos representativos de diferentes gamas y características:

1. Hisense Rocks 6:

- Sistema operativo: Android 10.
- Propósito: Evaluar el rendimiento y la funcionalidad en dispositivos básicos, comúnmente utilizados por usuarios con necesidades estándar.
- Particularidades: Resolución de pantalla estándar y hardware modesto para probar la eficiencia del sistema en condiciones limitadas.

2. Poco Phone 2021:

- Sistema operativo: Android 12.
- Propósito: Comprobar la experiencia del usuario en un dispositivo de gama media-alta, evaluando funciones avanzadas y tiempos de respuesta.
- Particularidades: Procesador de alto rendimiento, capacidad gráfica mejorada y resolución Full HD+.

Laptops

Se seleccionaron tres modelos de laptops que cubren diferentes rangos de rendimiento y tipos de usuarios:

1. Acer Swift 3:

- a. Procesador: Intel Core i7 (12va generación).
- b. Memoria RAM: 8 GB.
- c. Sistema operativo: Windows 10 Pro.

2. Lenovo Ideapad 3:

- a. Procesador: AMD Ryzen 5.
- b. Memoria RAM: 16 GB.
- c. Sistema operativo: Windows 11 Pro.

3. ASUS TUF Gaming F15:

- a. Procesador: Intel Core i7 (11va generación).
- b. Tarjeta gráfica: NVIDIA GeForce RTX 3050.
- c. Memoria RAM: 16 GB.
- d. Sistema operativo: Windows 11 Pro.

Software y herramientas

- **Navegadores:** Las pruebas para la aplicación web incluirán los navegadores más utilizados, como Google Chrome, Mozilla Firefox y Microsoft Edge, en sus versiones actualizadas.
- **Base de datos:** La base de datos se ejecutará en un servidor local configurado para pruebas, utilizando MySQL.
- **Conexión de red:** Las pruebas se realizarán bajo diferentes condiciones de red (Wi-Fi de alta velocidad, conexión limitada y sin conexión), para garantizar la estabilidad del sistema en diversos entornos.

v. Herramienta por utilizar:

- **XUnit:** Esta biblioteca de pruebas se utilizó debido a su compatibilidad con .NET y su capacidad para realizar pruebas unitarias e integradas de manera eficiente. XUnit ofrece características avanzadas como la ejecución paralela de pruebas, lo que permite acelerar el proceso de validación del sistema. Además, su estructura modular y su integración con plataformas como Visual Studio facilitaron la creación y ejecución de los casos de prueba definidos en el proyecto.

b. Procedimiento de prueba (automatizado)

i. Configuración inicial:

El proceso comenzará con la configuración del entorno de pruebas, lo que incluirá la instalación de servidores, la configuración de la base de datos y la preparación de los dispositivos y herramientas necesarios. Una vez listo el entorno, se procederá a crear scripts automatizados para los casos de uso críticos, priorizando funciones esenciales como el inicio de sesión, la gestión de cuentas y el proceso de compra.

ii. Ejecución:

Los scripts se ejecutarán tanto en entornos de desarrollo como en simulaciones de producción. Esto permitirá identificar fallos antes del despliegue final. Los resultados de estas ejecuciones serán registrados en un sistema de seguimiento, facilitando el análisis y la resolución de errores.

iii. Reporte:

Esto documentará el desempeño del sistema, señalaraán áreas de mejora y permitirán dar seguimiento a los errores detectados. Este proceso sistemático asegura la confiabilidad del sistema y su preparación para el despliegue en entornos reales.

c. Casos de prueba (automatizados)

Los casos de prueba estarán definidos de manera clara para cada funcionalidad del sistema. Cada caso incluirá un identificador único, una descripción detallada, las entradas necesarias, los resultados esperados, los resultados obtenidos tras la ejecución, y un estado que indicará si la prueba fue exitosa o fallida.

Gestión de Usuarios

ID de Prueba	Descripción	Entradas	Resultado esperado	Resultado obtenido	Conclusión
TestUsers_001 GetAllUsuarios_Test_Success	Recupera todos los usuarios registrados.	ID_User: Id del usuario loggeado y autenticado	Lista de usuarios recuperada	Se recuperan los datos de los usuarios de manera exitosa.	Prueba pasada
TestUsers_002 User_GetUser_Test_Success	Recupera un usuario en específico	ID_User: Id del usuario loggeado y autenticado	Información del usuario dado y su ID_User (debe tener token activo)	Se recupera de manera exitosa el usuario.	Prueba pasada
TestUsers_003 User_GetUser_Test_Failure	Se intenta recuperar un usuario inexistente.	ID_Usuario invalido	El sistema rechaza la solicitud.	El sistema rechaza la solicitud de manera exitosa.	Prueba pasada
TestUsers_004 UploadImage_Test_Success	Se realiza el registro de una imagen de prueba sin datos.	Imagen generada sin pixeles.	Que el sistema registre y guarde exitosamente la imagen.	El sistema registra la imagen.	Prueba pasada
TestUsers_005 UploadImage_Test_Failure	Se realiza el registro de una imagen de prueba sin bytes.	Objeto imagen nula.	El sistema rechaza el registro del objeto nulo.	El sistema rechaza el registro.	Prueba pasada
TestUsers_006 Save_User_Success	Se realiza el registro de un usuario con correo y usuario no registrados.	Objeto de usuario con todos los datos válidos.	El sistema registra al nuevo usuario.	El sistema registra al nuevo usuario.	Prueba pasada (Registro doble, da falso negativo)

TestUsers_007 Save_User_Fail_Missing_Fields	Se intenta registrar un usuario con datos faltantes.	Objeto usuario sin password ni username.	El sistema rechaza la solicitud de registro.	El sistema rechaza la solicitud de registro.	Prueba pasada
TestUsers_008 Save_User_Fail_Email_Already_Registered	Se intenta registrar un usuario con email previamente utilizado.	Objeto usuario con email ya registrado en la base.	El sistema rechaza la solicitud de registro.	El sistema rechaza la solicitud de registro.	Prueba pasada

Gestión de Carrito de compras

ID de prueba	Descripción	Entradas	Resultado esperado	Resultado obtenido	Conclusión
TestShoppingCar_001 add_articulo_to_car_Success	Añadir cantidad de un producto valido al carrito.	ID_User, ID_Cloth y quota validos	El sistema añade el valor al carrito.	El token rechaza la solicitud directa de la operación.	Falso negativo
TestShoppingCar_002 add_articulo_to_car_Fail_ID	Probar compra con ID_Cloth incorrecto.	Cantidad válida, Producto inválido.	Error: articulo inválido.	El sistema rechaza la operación.	Prueba pasada
TestShoppingCar_003 add_articulo_to_car_Fail_User	Se intenta añadir una cantidad a un carrito inexistente.	Cantidad válida ID_Cloth válidos, usuario inválido.	El sistema rechaza la solicitud para añadir el artículo.	El sistema rechaza la solicitud para añadir el artículo.	Prueba pasada
TestShoppingCar_004 get_shopping_car_Success	Obtener el carrito de compras con usuario valido.	ID_User valido con un carrito cargado.	Se obtiene el carrito relacionado al usuario de manera exitosa.	El sistema retorna el carrito de compras del cliente.	Prueba pasada
TestShoppingCar_005 GetShoppingCar_Fail	Obtener historial de compras con token rechazado.	Token inválido o expirado.	Error: 'Token rechazado'.	El sistema maneja la solicitud como negativa	Falso Negativo
TestShoppingCar_006 Update_Quantity_Success	Actualizar la cantidad de un producto en el carrito.	ID_User, ID_Cloth, quota válidos	El sistema actualiza el carrito exitosamente.	El sistema rechaza la solicitud debido a autenticación.	Falso negativo
TestShoppingCar_007 UpdateQuantity_Fail_MissingFields	Actualizar la cantidad de un producto en el carrito sin datos válidos.	ID_User, ID_Cloth, válidos, quota inválida.	El sistema rechaza la solicitud de cambio	El sistema rechaza la solicitud de cambio pero retorna un mensaje	Falso negativo

TestShoppingCar_008 UpdateQuantity_Fail _ItemNotFound	Actualizar la cantidad de un producto en el carrito sin datos válidos.	ID_User, válido, ruta y ID_Cloth, invalido	El sistema rechaza la solicitud de cambio	El sistema rechaza la solicitud de cambio pero retorna un mensaje	Falso negativo
TestSales_001 get_purchase_histor y_by_client_Success	Recuperar el carrito del cliente de manera exitosa.	ID_Client válido	Se recupera la información del carrito de compra del usuario.	Se niega el acceso al carrito de compra del usuario por autenticaci ón.	Falso negativo
TestSales_002 GetPurchaseHistory ByClient_TokenRejected_Fail	Intenta recuperar el carrito de un cliente Inexistente	ID_Client inválido	El servidor rechaza la solicitud del cliente	El servidor rechaza la solicitud del cliente, pero retorna un mensaje	Falso negativo
TestSales_003 BuyShoppingCar_Te st_Success	Se realiza la operación de compra con los elementos del carrito.	ID_User y Objeto Cloth válido.	El servidor procesa la operación exitosamente.	El servidor procesa la operación exitosamente.	Prueba pasada
TestSales_004 BuyShoppingCar_Te st_Fail_WrongUser	Se intenta realizar la operación con un carrito de compra inexiste nte.	ID_User y Cloth inválido	El servidor rechaza la operación.	El servidor rechaza la operación.	Prueba pasada
TestSales_005 BuyShoppingCar_Te st_Fail_Wrong_Qty	Se intenta realizar la operación con un carrito de compra con cantidad excesiva de productos.	ID_User y Cloth válidos, quota inválida.	El servidor rechaza la operación.	El servidor rechaza la operación.	Prueba pasada

ID de prueba	Descripción	Entradas	Resultado esperado	Resultado obtenido	Conclusión
TestCloths_001 Update_Articulo_Suc cess	Actualizar los valores de un artículo existente con datos válidos.	ID_Clothes valido, nombre, precio, o quota válidos.	Se actualiza la información del artículo de manera exitosa.	Se actualiza la información del artículo de manera exitosa.	Prueba pasada
TestCloths_002 Update_Articulo_Fail _NoFieldsToUpdate	Fallar al actualizar artículo sin campos a modificar.	ID_Clothes válido, sin campos válidos.	El sistema rechaza los cambios en el artículo.	El sistema rechaza los cambios en el artículo.	Prueba pasada
TestCloths_003 Update_Articulo_Fail _ItemNotFound	Fallar al actualizar artículo con id inválido.	ID_Clothes invalido.	El sistema rechaza los cambios en el artículo.	El sistema rechaza los cambios en el artículo.	Prueba pasada

TestCloths_004 Delete_Articulo_Success	Eliminar un artículo registrado en Cloth.	ID_Clothes. Valido.	El sistema procesa la solicitud y elimina el artículo.	El sistema rechaza la solicitud por autenticación.	Falso negativo
TestCloths_005 Delete_Articulo_Fail_NoID	Fallar al intentar eliminar artículo sin parametro.	N/A	Error: Falta de parámetros.	El sistema rechaza la solicitud.	Prueba pasada
TestCloths_006 Delete_Articulo_Fail_ItemNotFound	Fallar al intentar recuperar un artículo inexistente.	ID_Cloth invalido	Error: no se encuentra el artículo.	El sistema rechaza la solicitud, pero retorna mensaje.	Falso negativo
TestCloths_007 GetAllArticulos_Test_Success	Obtener lista de artículos completa.	Ninguno	Lista de artículos.	El sistema retorna la lista de artículos.	Prueba pasada
TestCloths_007 GetArticuloByName_Test_Success	Obtener artículo por nombre exitosamente.	Nombre de artículo válido.	Detalles del artículo.	El sistema retorna el artículo(s).	Prueba pasada
TestCloths_008 GetArticuloByName_Test_Fail	Fallar al obtener artículo por nombre inexistente.	Nombre de artículo inválido.	Error: 'Artículo no encontrado' .	El sistema no retorna ningún artículo	Prueba pasada
TestCloths_009 GetArticulosBySellin g_Test_Success	Obtener artículos por ventas exitosamente.	Ninguna entrada.	Lista de artículos ordenados por ventas.	El sistema retorna la lista de artículos relacionada	Prueba pasada
TestCloths_010 GetArticuloBySelling _Test_Fail	Fallar al obtener artículos por ventas debido a error a categoría inválida.	Cathegory inválida.	Error: 'No se pudieron obtener los datos'.	El sistema no retorna ningún artículo.	Prueba pasada
TestCloths_011 get_inventory_by_selling_Success	Obtener inventario por ventas exitosamente.	ID_User	Lista de inventario por ventas.	El sistema retorna la lista de artículos relacionados.	Prueba pasada
TestCloths_012 get_inventory_by_selling_Fail	Fallar al obtener inventario por ventas debido a usuario invalido.	ID_User inválido.	Error: 'No se pudieron obtener los datos'.	El sistema no retorna ningún artículo	Prueba pasada
TestCloths_013 SaveArticle_Success	Guardar artículo exitosamente.	Datos completos del artículo.	Mensaje: 'Artículo guardado exitosamente.'	El sistema registra el artículo exitosamente.	Prueba pasada
TestCloths_014 SaveArticle_Fail_Mis singFields	Fallar al guardar artículo debido a campos faltantes.	Datos incompletos del artículo.	Error: 'Campos obligatorios faltantes'.	El sistema no registra ningún artículo.	Prueba pasada

TestCloths_015 SaveArticle_Fail_DatabaseError	Fallar al guardar artículo debido a datos inválidos.	Datos de artículo incompletos.	Error: 'Error interno al guardar artículo'.	El sistema no registra ningún artículo.	Prueba pasada
--	--	--------------------------------	---	---	----------------------

d. Resultados

Se realizaron 37 pruebas sobre los servicios REST de la API, obteniendo 28 casos positivos de prueba y 9 falsos positivos.

i. Pruebas de integración

CU01 – Dar de alta cuenta

Paso	Entrada(s)	Acción resultante
1. El Vendedor o Cliente selecciona el botón Regístrate .	Ninguna	Se carga la ventana DarAltaUsuario .
2. El Vendedor o Cliente ingresa una imagen para su perfil.	C:\Users\danie\Pictures\gogeta.png	Se carga la ruta de la imagen a su campo y se muestra en la pantalla la previsualización de dicha imagen.
3. El Vendedor o Cliente llena los campos del formulario.	Nombre(s): Ares Judda Apellido(s): Rivera Soto Username: arecito Tipo de cuenta: Vendedor Correo electrónico: testing@gmail.com Contraseña: 123456	Se muestra los datos ingresados en su respectivo campo y se muestra el botón Crear Cuenta .
4. El Vendedor o Cliente da clic en el botón Crear Cuenta .	Usuario { 'Ares Judda', 'Rivera Soto', 'arecito', 2, 'testing@gmail.com', '123456' }	El sistema registra la información del usuario en la base de datos y muestra la ventana RegistroExitoso con el mensaje "El usuario fue agregado al sistema".
Flujos alternos		
a. El Vendedor o Cliente deja algún campo del formulario vacío.	Nombre(s): Ares Judda Apellido(s): Username: arecito Tipo de cuenta: Correo electrónico: testing@gmail.com Contraseña:	El sistema no muestra el botón Crear Cuenta .
b. El Vendedor o Cliente ingresa un correo ya registrado.	Nombre(s): Ares Judda Apellido(s): Rivera Soto Username: arecito Tipo de cuenta: Vendedor Correo electrónico: testing@gmail.com Contraseña: 123456	El sistema muestra la ventana CorreoRegistrado con el siguiente mensaje: "El correo con el que tratas registrarte ya se encuentra en el sistema".

CU02 – Modificar cuenta

Paso	Entrada(s)	Acción resultante
1. El Vendedor o Cliente selecciona el botón Actualizar .	Ninguna	Se carga la ventana ModificarCuenta con los datos

		del formulario con información del propio tipo de usuario.
2. El Vendedor o Cliente vuelve a ingresar una imagen para su perfil.	C:\Users\danie\Pictures\vegito.png	Se carga la ruta de la imagen a su campo y se muestra en la pantalla la previsualización de dicha imagen.
3. El Vendedor o Cliente vuelve a llenar los campos del formulario.	Nombre(s): Judda Ares Apellido(s): Soto Rivera Teléfono: 2288474739 Fecha de nacimiento: 2024-12-12 Dirección: Calle Esteban Colonia Quito #999 Código Postal: 66666	Se muestra los datos ingresados en su respectivo campo y se muestra el botón Actualizar perfil .
4. El Vendedor o Cliente da clic en el botón Actualizar perfil .	Usuario { 'Judda Ares', 'Soto Rivera', '2288474739', '2024-12-12', 'Calle Esteban Colonia Quito #999', '66666' }	El sistema actualiza la información del usuario previo en la base de datos y muestra la ventana <i>ProcesoExitoso</i> con el mensaje "Los datos fueron actualizados en el sistema"
Flujos alternos		
a. El Vendedor o Cliente deja algún campo del formulario vacío.	Nombre(s): Judda Ares Apellido(s): Teléfono: Fecha de nacimiento: 2024-12-12 Dirección: Código Postal: 66666	El sistema no muestra el botón Actualizar perfil .
b. El Vendedor o Cliente selecciona el botón Cancelar .	Ninguna	El sistema regresa a la ventana <i>MenuPrincipal</i> .

CU03 – Publicar artículo

Paso	Entrada(s)	Acción resultante
1. El Vendedor selecciona el botón Vender .	Ninguna	Se carga la ventana <i>PublicarArticulo</i> .
2. El Vendedor ingresa una imagen para su artículo.	C:\Users\danie\Pictures\golf_tshirt_01.png	Se carga la ruta de la imagen a su campo y se muestra en la pantalla la previsualización de dicha imagen.
3. El Vendedor llena los campos del formulario.	Nombre: Golf Tshirt 01 Tipo de artículo: Playera Precio: 499.00 Talla: Mediana Cantidad: 10	Se muestra los datos ingresados en su respectivo campo y se muestra el botón Publicar .
4. El Vendedor da clic en el botón Crear Cuenta .	Articulo { 'Golf Tshirt 01', 1, 499.00, 2, 10 }	El sistema registra la información del nuevo artículo en la base de datos y muestra la ventana <i>ProcesoExitoso</i> con el mensaje "El artículo fue publicado en el sistema"

Flujos alternos		
a. El Vendedor deja algún campo del formulario vacío.	Nombre: Tipo de artículo: Playera Precio: Talla: Mediana Cantidad: 10	El sistema no muestra el botón Publicar .
b. El Vendedor selecciona el botón Cancelar .	Ninguna	El sistema regresa a la ventana <i>MenuPrincipal</i> .

CU04 – Modificar artículo

Paso	Entrada(s)	Acción resultante
1. El Vendedor selecciona el botón Modificar .	Ninguna	Se carga la ventana <i>ListaArticulos</i> con una lista de artículos actualmente publicados por el usuario.
2. El Vendedor selecciona un artículo.	Articulo { 1 }	Se carga la ventana <i>ModificarArticulo</i> con los datos del formulario con información del propio artículo.
3. El Vendedor vuelve a ingresar una imagen para su artículo.	C:\Users\danie\Pictures\golf_tshirt_01.png	Se carga la ruta de la imagen a su campo y se muestra en la pantalla la previsualización de dicha imagen.
4. El Vendedor vuelve a llenar los campos del formulario.	Nombre: Golf Tshirt 01 modelo 2 Tipo de artículo: Playera Precio: 699.00 Talla: Mediana Cantidad: 15	Se muestra los datos ingresados en su respectivo campo y se muestra el botón Modificar .
5. El Vendedor da clic en el botón Modificar .	Articulo { 'Golf Tshirt 01 modelo 2', 1, 699.00, 2, 15 }	El sistema actualiza la información del artículo previo en la base de datos y muestra la ventana <i>ProcesoExitoso</i> con el mensaje “El artículo fue actualizado en el sistema”.
Flujos alternos		
a. El Vendedor deja algún campo del formulario vacío.	Nombre: Tipo de artículo: Playera Precio: Talla: Mediana Cantidad:	El sistema no muestra el botón Modificar .
b. El Vendedor da clic en el botón Eliminar .	Ninguna	El sistema muestra la ventana <i>ProcesoExitoso</i> con el mensaje “El artículo fue eliminado del sistema”.

CU05 – Consultar inventario

Paso	Entrada(s)	Acción resultante
1. El Vendedor selecciona el botón Inventario .	Ninguna	Se carga la ventana <i>ListaArticulos</i> con una lista de artículos actualmente publicados por el usuario.

2. El Vendedor ingresa una palabra clave en el campo de búsqueda.	Buscar Artículos: Golf Tshirt	El sistema muestra los artículos cuyo nombre empieza con el introducido en el campo de texto.
3. El Vendedor selecciona el tipo de artículo que desea buscar.	Filtrar por: Playera	El sistema muestra los artículos que sean de tipo "Playera".
Flujos alternos		
a. El Vendedor selecciona el ícono de Exportar a PDF .	Ninguna	El sistema abre la ventana de <i>ExploradorArchivos</i> del Vendedor, genera el reporte de inventario en formato PDF con los siguientes datos: nombre, código, categoría, talla y lo guarda en la ubicación definida; además, muestra la ventana <i>ProcesoExitoso</i> con el siguiente mensaje: "PDF del inventario fue generado correctamente"

CU06 – Consultar ventas realizadas

Paso	Entrada(s)	Acción resultante
1. El Vendedor selecciona el botón Ganancias .	Ninguna	Se carga la ventana <i>ListaVentas</i> con una lista de ventas referentes a artículos actualmente publicados por el usuario.
2. El Vendedor selecciona una fecha de inicio y corte en específico.	Fecha de inicio: 2024-12-12 Fecha de corte: 2024-12-13	El sistema muestra las estadísticas de los artículos que coincidan con dichas fechas.

CU08 – Añadir producto al carrito de compras

Paso	Entrada(s)	Acción resultante
1. El Cliente selecciona un artículo.	Articulo { 1 }	Se carga la ventana <i>DetallesArticulo</i> .
2. El Cliente da clic en el botón Agregar al carrito .	CarritoCompras { 1 }	El sistema añade el artículo a la cola del carrito de compras y vuelve a la ventana <i>ListaArticulos</i>
Flujos alternos		
a. El Cliente selecciona el botón Cancelar .	Ninguna	El sistema regresa a la ventana <i>ListaArticulos</i> .

CU09 – Dar de alta pedido

Paso	Entrada(s)	Acción resultante
1. El Cliente selecciona el ícono de Carrito de Compras .	Ninguno	Se carga la ventana <i>CarritoCompras</i> con una lista de artículos agregados a la cola.
2. El Cliente da clic en el botón Realizar Pedido .	CarritoCompras { 1, 1 }	El sistema registra la información del pedido previo en la base de datos; además, muestra la ventana <i>ProcesoExitoso</i> con el

		siguiente mensaje: "El pedido fue realizado correctamente".
Flujos alternos		
a. El Vendedor selecciona el botón Cancelar .	Ninguna	El sistema regresa a la ventana <i>ListaArticulos</i> .

CU10 – Consultar historial de pedidos

Paso	Entrada(s)	Acción resultante
1. El Cliente selecciona el botón Historial de pedidos .	Ninguna	Se carga la ventana <i>ListaPedidos</i> con una lista de pedidos realizados por él mismo.
2. El Cliente selecciona una fecha de compra en específico.	Fecha de compra: 2024-12-12	El sistema muestra las ventas que coincidan con dicha fecha.
3. El Cliente selecciona el botón Detalles .	Pedidos { 1 }	Se muestra la tabla <i>DetallesPedido</i> con la información de dicho pedido.
Flujos alternos		
a. El Cliente selecciona el ícono Regresar .	Ninguna	El sistema regresa a la ventana <i>ListaPedidos</i> .

VI. Estrategia de Despliegue

- **Servidores:** Los servidores que se van a utilizar para el despliegue del sistema Golf Clothing Shop van a ser 2:
 - Un servidor con el sistema operativo Windows 10/11 para hostear contenedores de **Docker** personalizados con la API, hosteandolos en los puertos 10001 y 10002.
 - Una base de datos en la nube creada con **Aiven**, que es una plataforma para desplegar tecnologías open-source.
- **Contenedores de Docker:** Los contenedores que se vayan creando a base de imágenes personalizadas para el proyecto, se van a hostear en una computadora enfocada únicamente para eso, con el sistema operativo Windows 10/11. Para el contenedor creado con la imagen para la API de **Node.js/Express.js**, de igual manera, se va a ejecutar en el servidor dedicado, pero exponiendo el puerto 10002, y el contenedor va a exponer el puerto 10002 para escuchar y recibir peticiones por medio de los servicios de la API.
- **Cliente:** Simplemente va a desplegarse con un archivo **.jar**, para que pueda ser ejecutado en cualquier dispositivo que contenga los sistemas operativos Windows, Linux o MacOS.

VII. Conclusiones

a. Luis Ángel Elizalde Arroyo

¿Cómo me sentí?

Durante el desarrollo del proyecto, experimenté una mezcla de emociones. Me sentí emocionado por enfrentar nuevos retos y aprender tecnologías desconocidas, pero también hubo momentos de frustración al encontrarme con problemas inesperados. Al final, la satisfacción de superar cada obstáculo fue mayor que las dificultades, y eso me motiva a seguir aprendiendo.

¿Qué aprendí al realizar el proyecto?

Aprendí a gestionar mejor mi tiempo y recursos al trabajar en equipo. También adquirí habilidades técnicas relacionadas con la implementación de diagramas UML, el uso de Git para control de versiones y el diseño de pruebas automatizadas. Este proyecto me ayudó a comprender la importancia de la planificación y la comunicación efectiva para el éxito del desarrollo.

¿Qué se me dificultó?

Una de las principales dificultades que enfrenté fue adaptarme a las herramientas de control de versiones. Aunque tenía conocimientos previos, fue un desafío integrarlos con las nuevas prácticas del equipo. Además, comprender los flujos de integración continua tomó más tiempo del esperado, pero con paciencia y apoyo logré superar estos problemas.

b. Daniel Mongeote Tlachy

¿Cómo me sentí?

Durante el desarrollo del proyecto, me sentí motivado por aprender y dominar tecnologías como RESTful y la creación de APIs. Aunque hubo momentos de desmotivación, mantuve el compromiso de continuar, sabiendo que al final todo el esfuerzo sería recompensado. El mundo de las APIs siempre me ha parecido fascinante, y este semestre me permitió no solo entender su funcionamiento, sino también aprender a desarrollarlas desde cero.

¿Qué aprendí al realizar el proyecto?

A lo largo del proyecto, adquirí conocimientos clave como el desarrollo de APIs RESTful, la creación de servicios, la conexión de métodos RESTful con un cliente y la implementación de sistemas en red. También aprendí a elaborar diagramas de componentes que me ayudaron a estructurar mejor el proyecto.

¿Qué se me dificultó?

Uno de los mayores retos fue trabajar con las rutas de la API, que inicialmente resultaron confusas. Además, integrar solicitudes con tokens JWT representó un desafío importante que logré superar con dedicación. La curva de aprendizaje de la tecnología fue compleja al principio, pero con el tiempo logré adaptarme y aplicar lo aprendido de manera efectiva. Por último, aunque la documentación y el diseño no son mi punto fuerte, logré cumplir con estas tareas, mejorando en áreas que usualmente me resultan complicadas.

c. Ares Judda Rivera Soto

¿Cómo me sentí?

Al principio me sentí abrumado por la cantidad de tareas que debíamos realizar, pero conforme avancé en el proyecto, encontré satisfacción al completar cada objetivo. Colaborar con mis compañeros fue una experiencia enriquecedora, ya que compartimos ideas y aprendimos juntos.

¿Qué aprendí al realizar el proyecto?

Descubrí la importancia del trabajo en equipo y cómo cada integrante aporta habilidades únicas al proyecto. Además, mejoré mi capacidad para documentar procesos y aprendí a estructurar sistemas utilizando la arquitectura de microservicios, lo que incrementó mi interés en la creación de la API.

¿Qué se me dificultó?

Tuve dificultades con la implementación de medidas de seguridad, especialmente al manejar tokens JWT y proteger los datos sensibles del sistema. Este proceso requirió estudiar más sobre encriptación y autenticación, pero valió la pena, ya que me permitió crecer profesionalmente y asegurar la calidad del sistema.

d. Alesia de Jesús Torres Osorio

¿Cómo me sentí?

Me sentí retado en cada etapa del proyecto, especialmente al trabajar con tecnologías y metodologías que no dominaba del todo. Sin embargo, esta experiencia también fue gratificante, ya que cada logro, por pequeño que fuera, me hacía sentir más confiado y motivado para continuar.

¿Qué aprendí al realizar el proyecto?

Aprendí a gestionar de manera eficiente los recursos disponibles y a priorizar tareas según su impacto en el sistema. También profundicé en el diseño de interfaces de usuario, mejorando la usabilidad y accesibilidad del sistema, lo cual fue un área que anteriormente no consideraba tan relevante.

¿Qué se me dificultó?

Se me complicó el ajuste de las interfaces para que fueran compatibles en diferentes dispositivos y resoluciones. Entender cómo optimizar el diseño responsive y adaptarlo correctamente me llevó tiempo, pero finalmente logré implementar una solución funcional con el apoyo de la documentación y las pruebas realizadas.

VIII. Referencias

- Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2009). Non-functional requirements in software engineering. Springer Science & Business Media.
- Cockburn, A. (2000). Writing effective use cases. Addison-Wesley Professional.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns: Elements of reusable object-oriented software. Addison-Wesley.
- Newman, S. (2015). Building microservices: Designing fine-grained systems. O'Reilly Media.
- Sommerville, I. (2011). Software engineering (9th ed.). Addison-Wesley.
- Wiegers, K., & Beatty, J. (2013). Software requirements (3rd ed.). Microsoft Press.
- Elizalde, A., Mongeote, D., Rivera, A. & Torres, A. (2025). Servidor para el proyecto GOLF. Recuperado de: https://github.com/Ares-Judda/Golf_Proyect.git
- Elizalde, A., Mongeote, D., Rivera, A. & Torres, A. (2025). Cliente de Escritorio para el proyecto GOLF. Recuperado de: https://github.com/Ares-Judda/GOLF_DESKTOP_API.git