

**APLICATIVO WEB PARA TIENDA ONLINE NICHE**

**AUTORES:**

**ANDRES DELGADO GARCIA**

**ANGELICA PACHECO ESCOBAR**

**DOCENTE:**

**MARIBEL MAESTRE**

**UNIVERSIDAD POPULAR DEL CESAR**

**VALLEDUPAR CESAR**

**2023**

## **TABLA DE CONTENIDO**

### **I. DESCRIPCION DEL SISTEMA**

- 1.1. DESCRIPCIÓN DETALLADA DEL SISTEMA O APLICACION
- 1.2. MODELO DE REQUERIMIENTOS
  - 1.2.1. Descripción Requisitos funcionales
  - 1.2.2. Descripción Requisitos no funcionales
- 1.3. MODELO DE CASOS DE USO
  - 1.3.1. Diagramas de caso de uso
  - 1.3.2. Descripción de caso de uso
- 1.4. MODELO DE DISEÑO DEL SISTEMA
  - 1.4.1. Diagrama de clases detallado
  - 1.4.2. Diagramas de secuencias
  - 1.4.3. Diagrama entidad relacion
  - 1.4.4. Diagrama de componentes
- 1.5. PRODUCTO DEL SOFTWARE

### **II. PRUEBAS DEL SOFTWARE**

- II.1. INTRODUCCIÓN
- II.2. PLANIFICACIÓN DE LAS PRUEBAS**
  - II.2.1. Objetivos de las pruebas
  - II.2.2. Alcance de las pruebas
  - II.2.3. Módulos del sistema a probar
  - II.2.4. Ambiente de pruebas
- II.3. PRUEBAS UNITARIAS**
  - II.3.1. Análisis de las pruebas
  - II.3.2. Diseño de casos de pruebas
  - II.3.3. Ejecución de las pruebas
  - II.3.4. Evaluación de las pruebas
- II.4. PRUEBAS DE INTEGRACIÓN**
  - 4.2.1. Estrategia de Pruebas incrementales
    - Diseño casos de pruebas
    - Ejecución pruebas de integración
    - Evaluación de las pruebas
  - 4.2.2. Estrategia de Pruebas basadas en hilos
    - Diseño casos de pruebas
    - Ejecución pruebas de integración
    - Evaluación de las pruebas
- II.5. PRUEBAS DE SISTEMAS**
  - II.5.1. Diseño de caso de pruebas
  - II.5.2. Ejecución de la prueba
  - II.5.3. Evaluación de la prueba
- II.6. PRUEBAS DE ACEPTACIÓN**
  - II.6.1. Diseño de caso de pruebas

- II.6.2.** Ejecución de la prueba
- II.6.3.** Evaluación de la prueba
- II.7. COCLUSIONES**
- II.8. REFERENCIAS**
- II.9.**

### **III. METRICAS DEL SOFTWARE**

#### **III.1. Introducción**

- III.1.1. Objetivos**
- III.1.2. Alcance**
- III.1.3. Tipos y herramientas de métricas**

#### **III.2. Métricas de Producto**

- III.2.1. Métricas de tamaño**
  - III.2.1.1. Métricas de tamaño de longitud**
  - III.2.1.2. Métricas de tamaño de funcionalidad**

- III.2.2. Métricas con el método GQM**
- III.2.3. Métricas de productos con atributos externos**
- III.2.4. Métricas de diseño**

#### **III.3. CONCLUSIONES**

### **IV. ESTIMACION DEL SOFTWARE**

#### **IV.1. Estimación del tamaño**

- IV.1.1. Tamaño por punto de caso de uso**
- IV.1.2. Tamaño por puntos de objetos – Composición de aplicación**

#### **IV.2. Estimación del esfuerzo**

- IV.2.1. Estimar el esfuerzo**
- IV.2.2. Análisis de los resultados**

#### **IV.3. Estimación del costo**

##### **IV.3.1. COCOMO II**

- IV.3.1.1. Composición de aplicación**
- IV.3.1.2. Diseño temprano - anticipado**
- IV.3.2. Otro método para determinar el costo del software**
- IV.3.3. Análisis de los resultados de Costos**

### **V. CALIDAD DE SOFTWARE**

- V.1. Aplicación herramienta SONARQUBE**
- V.2. Análisis de resultados**

### **VI. CONCLUSIONES**

### **VII. REFERENCIAS - BIBLIOGRAFICAS**

## 1. DESCRIPCION DETALLADA DEL SISTEMA

Este proyecto se enfoca en realizar un aplicativo web donde se implementa un sistema de gestión eficiente para una tienda online. El sistema efectuado fue diseñado con el propósito de sistematizar todos los procesos de la plataforma en línea, le permitirá al usuario realizar compras de productos, visualizar los detalles de cada producto que este disponible, consultar el estado de su pedido desde la facilidad de su casa, el sistema realizado amplia las ventas mediante el comercio electrónico de manera rápida y sencilla.

## 2. DEFINICION REQUISITOS

### 2.1. REQUERIMIENTOS FUNCIONALES

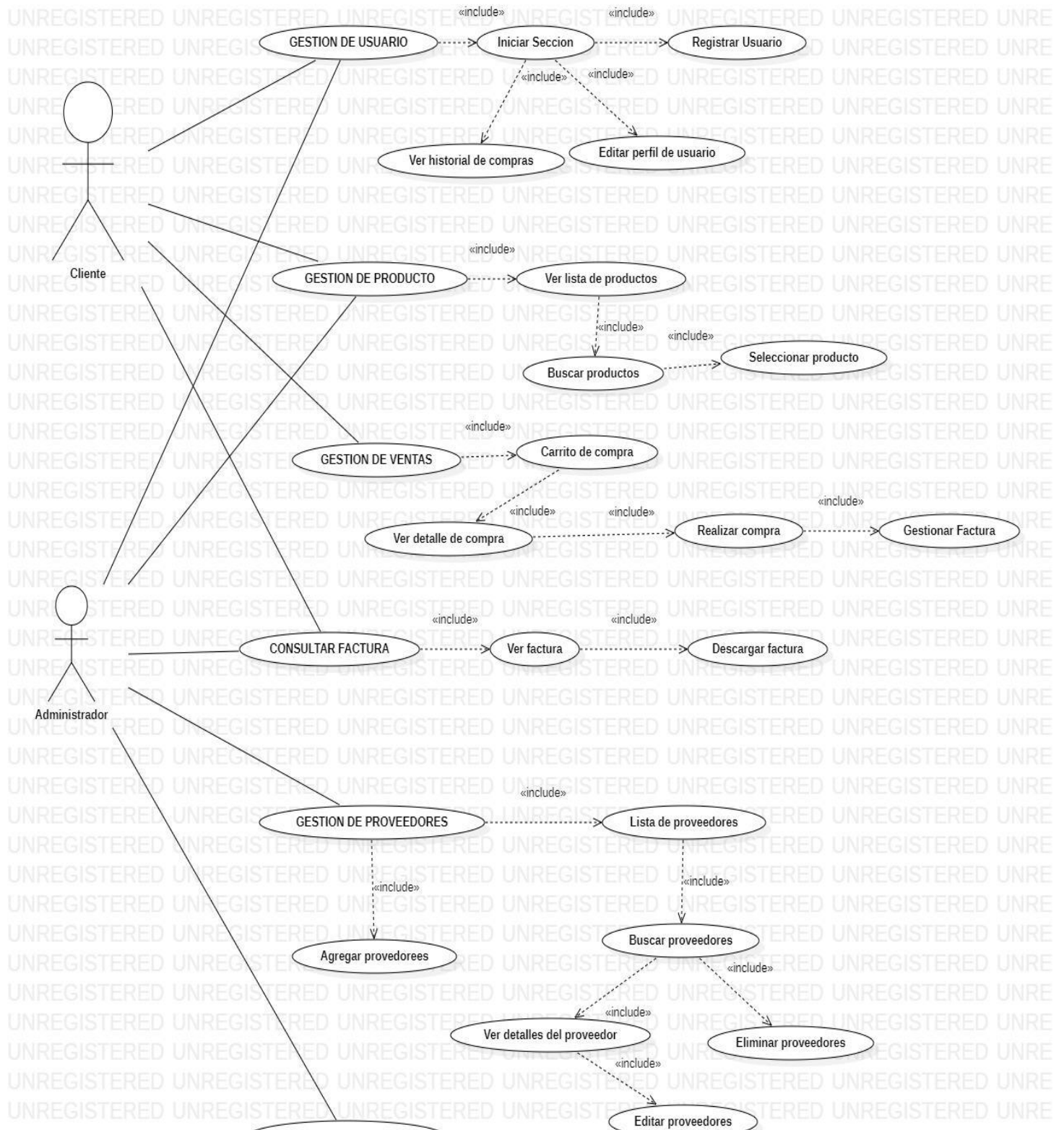
Nro.	Requerimiento	Descripción
RF1	Gestión de usuario	El sistema debe permitir iniciar sesion y registrar de usuarios nuevos
RF2	Gestión de productos	El sistema debe permitir al administrador registra y actualizar productos en la tienda, capturando información relevante como nombre, descripción, precio
RF3	Consultar Factura	El sistema debe permitir al usuario consultar su factura y descargarla.
RF5	Gestión de proveedor	El sistema de proveedores involucra el proceso de registrar proveedores que suministran la tienda
RF6	Gestión de venta	El sistema debe permitir a los usuarios realizar compras de los productos de la tienda de manera sencilla, eficiente y generar facturas.

## 2.2.REQUERIMIENTOS NO FUNCIONALES

Nro.	Requerimiento	Descripción
RF1	Usabilidad	El sistema debe ser fácil de usar para los usuarios, incluyendo una interfaz atractiva y una navegación intuitiva.
RF2	Seguridad	El sistema debe contar con medidas de seguridad adecuadas para proteger los datos de los usuarios y prevenir fraudes.
RF3	Disponibilidad	El sistema debe estar disponible en todo momento, con una alta disponibilidad para los usuarios.
RF4	Rendimiento	El sistema debe contar con un buen rendimiento, permitiendo una rápida carga de la página y un procesamiento rápido de las solicitudes.

### 3. MODELOS DE CASOS DE USO

#### 3.1. DIAGRAMA DE CASOS DE USO



### 3.2. DESCRIPCION DE CASOS DE USO

Documentación Casos de Uso	
Caso de uso	Gestión de usuario
Actores	Administrador, cliente
Tipo propósito	Asociación, incluye
Descripción	<p>Se permite al usuario iniciar sesión accediendo al sitio web, mantener actualizados los datos del usuario y ver su historial de compra</p> <p>Se permite al administrador iniciar sesión y tener acceso a todo el sitio web.</p>
Precondición	<ul style="list-style-type: none"><li>● El usuario ya este registrado para ver el historial de compra,</li><li>● el usuario debe haber iniciado sesión previamente y haber realizado al menos una compra en la tienda para tener acceso al historial usuario</li></ul>
Postcondiciones	El usuario está registrado satisfactoriamente en la base de datos del sistema.
Flujo normal	<ul style="list-style-type: none"><li>● ingresar al sitio web por medio de un navegador</li><li>● el usuario le da la opción de registrar usuario y se le muestra el formulario</li><li>● registro de usuario ingresando sus datos personales creando un usuario y contraseña</li><li>● Se crea la cuenta satisfactoriamente</li><li>● inicio de sesión de usuario ingresando el usuario y contraseña</li> <li>● editar datos personales del usuario</li> <li>● Cerrar sesión</li></ul>
Flujo alternativo	<ul style="list-style-type: none"><li>● En caso de que el usuario no puede crear una cuenta es porque la cuenta ya existe.</li><li>● En caso de que no se permita iniciar sesión deberá estar previamente registrado</li><li>● Si el sistema no puede verificar la integridad de los datos modificados por el usuario, muestra un mensaje de error y solicita al usuario que verifique los datos e intente modificarlos.</li></ul>

Excepciones	E-1 Si el usuario ingresa información incorrecta o incompleta en el formulario de registro, el sistema muestra un mensaje de error y solicita al usuario que corrija la información.
-------------	--

Documentación Casos de Uso	
Caso de uso	Gestión de producto
Actores	Administrador
Tipo propósito	Asociación, incluye
Descripción	El usuario puede buscar a cualquier producto disponible, acceder al listado de todos los productos y las especificaciones que tiene cada producto que el usuario seleccione
Precondición	<ul style="list-style-type: none"> <li>● El producto elegido debe existir en la base de datos del sistema.</li> </ul>
Postcondiciones	<ul style="list-style-type: none"> <li>● El usuario puede ver todo el listado de productos disponibles en el sistema y ver la información detallada de cada producto escogido por el usuario</li> </ul>
Flujo normal	<ul style="list-style-type: none"> <li>● El usuario administrador accede al sistema de gestión de la tienda virtual.</li> <li>● El administrador selecciona la opción de "Gestión de productos" en el panel de control.</li> <li>● El sistema muestra una lista de productos existentes en la tienda.</li> <li>● El administrador tiene la opción de buscar un producto específico o agregar uno nuevo.</li> <li>● El administrador puede agregar, eliminar y modificar un producto</li> <li>● El administrador solicita al proveedor ingresando los detalles del producto, como nombre, descripción, categoría, precio, etc.</li> <li>● Se agrega producto satisfactoriamente y se le notifica al administrador</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>● En caso no se puede seleccionar el producto que el usuario desea, se ofrecen ofertas similares</li> </ul>



Excepciones	<ul style="list-style-type: none"> <li>● Validar la existencia del producto y si hay las cantidades requeridas</li> </ul>
-------------	---

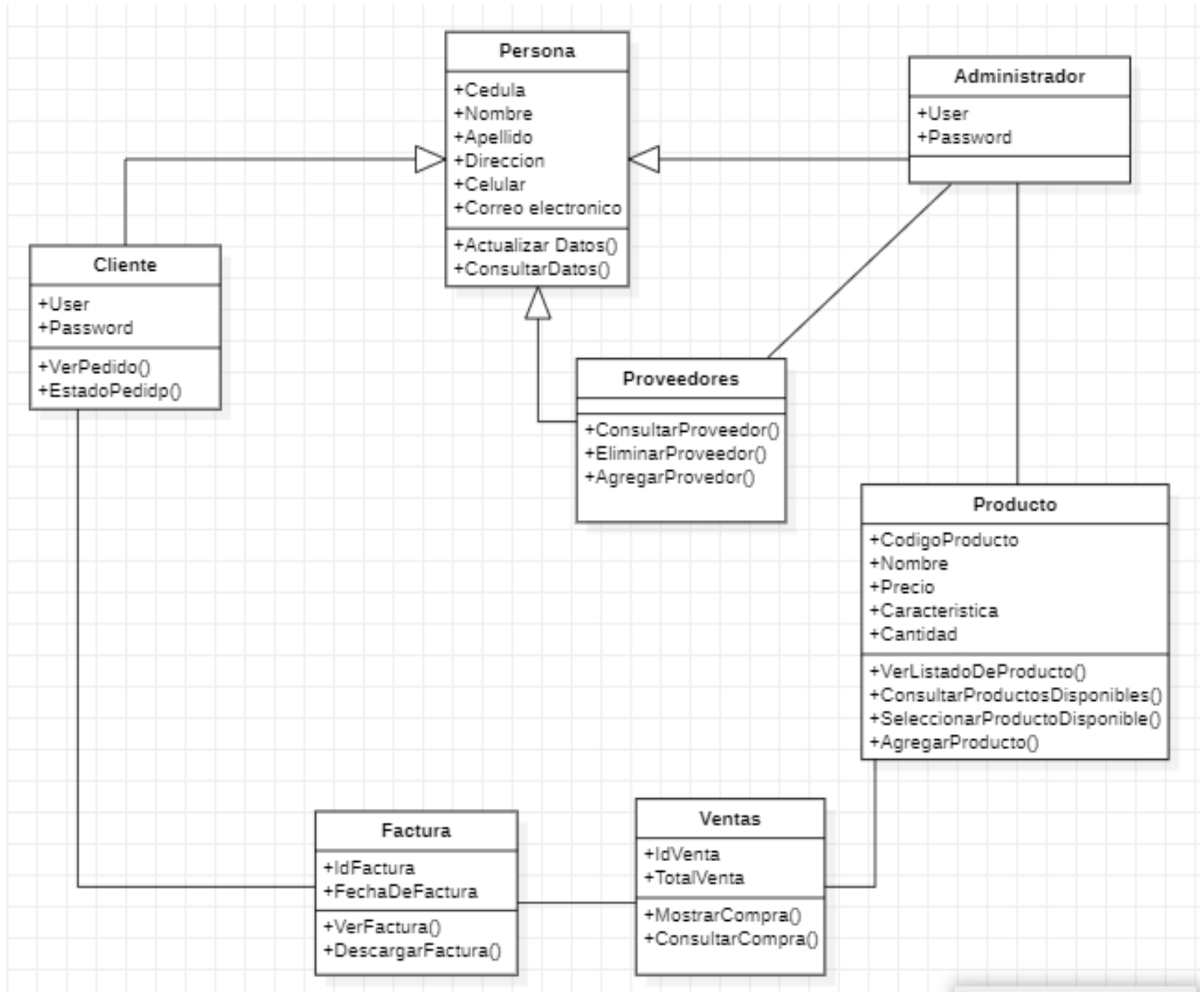
Documentación Casos de Uso	
Caso de uso	Gestión de venta
Actores	Cliente
Tipo propósito	Asociación, include
Descripción	<ul style="list-style-type: none"> <li>● Este caso de uso permite a un usuario realizar una compra de los productos que ha agregado a su carrito de compras y permite ver estado actual de su compra</li> </ul>
Precondición	<ul style="list-style-type: none"> <li>● El usuario debe haber iniciado sesión previamente y tener al menos un producto agregado al carrito de compra para realizar la compra</li> </ul>
Postcondiciones	<ul style="list-style-type: none"> <li>● Se registra la compra en el sistema y se muestra los detalles de la compra al usuario</li> </ul>
Flujo normal	<ul style="list-style-type: none"> <li>● El cliente inicia sesión y comienza a buscar productos.</li> <li>● El cliente al seleccionar el producto</li> <li>● En el sistema se verificará la disponibilidad del producto.</li> <li>● El usuario puede seguir agregando productos a su carrito hasta finalizar su compra</li> <li>● Hacer clic en realizar compra</li> <li>● El usuario elige un método de pago y realiza el pago</li> <li>● El usuario imprime la factura</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>● En caso de que no se pueda añadir un producto al carrito de compra se mostrara producto similares</li> <li>● En caso de que no se pueda realizar la compra se debe verificar los datos del pago</li> </ul>
Excepciones	<ul style="list-style-type: none"> <li>● Si el usuario pierde la conexión a Internet mientras usa el carrito de compras, el sistema debe asegurarse de que los datos del carrito se guarden y no se pierdan, para que el usuario pueda reanudar su compra más tarde.</li> </ul>

Documentación Casos de Uso	
Caso de uso	Consultar de Facturas
Actores	Cliente, administrador
Tipo propósito	Asociación, include
Descripción	<ul style="list-style-type: none"> <li>● El usuario puede acceder a todas las facturas de compra que haya hecho en la página web</li> </ul>
Precondición	<ul style="list-style-type: none"> <li>● El usuario debe iniciar sesión</li> </ul>
Postcondiciones	<ul style="list-style-type: none"> <li>● El usuario puede visualizar todas sus facturas y poder descargarla</li> </ul>
Flujo normal	<ul style="list-style-type: none"> <li>● El usuario ingresa a la página web con su usuario y contraseña, previamente debió realizado una compra</li> <li>● El usuario hace clic en el menú listado las facturas</li> <li>● El usuario puede seleccionar una factura en específico y ver la información detalladamente</li> <li>● Descargar factura</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>● En caso de que el usuario no puede visualizar sus facturas, debe haber realizado al menos una compra</li> <li>● En caso de que el usuario no pueda descargar su factura se le enviara por correo electrónico</li> </ul>
Excepciones	<ul style="list-style-type: none"> <li>● Si llegase haber un error de parte del sistema al ver o descargar la factura se le notificara por medio de correo electrónico al cliente.</li> </ul>

Documentación Casos de Uso	
Caso de uso	Gestión de proveedores
Actores	Administrador
Tipo propósito	Asociación, include
Descripción	<ul style="list-style-type: none"> <li>Se permite ver la lista de los proveedores que suministran los productos a la tienda virtual y su información de contacto</li> </ul>
Precondición	
Postcondiciones	<ul style="list-style-type: none"> <li>Se visualiza todos los proveedores y los productos que suministrar</li> </ul>
Flujo normal	<ul style="list-style-type: none"> <li>El administrador ingresa al sistema</li> <li>se realiza una búsqueda de proveedores</li> <li>Se evalúan a los proveedores que ofrecen los productos o servicios que necesita la empresa.</li> <li>El administrador puede agregar, eliminar, o editar a los proveedores.</li> <li>Se seleccionan los proveedores para ver su información de forma detallada</li> <li>El administrador elige a los proveedores según las necesidades de la tienda</li> <li>El administrador agrega al proveedor al sistema</li> </ul> <p>Se hace la negociación</p>
Flujo alternativo	<ul style="list-style-type: none"> <li>si se reciben productos o servicios de baja calidad, se debe trabajar con el proveedor para resolver el problema y establecer medidas para prevenir futuros problemas.</li> <li>si un proveedor no cumple con los términos de entrega acordados se deberán tomar medidas para garantizar los productos</li> </ul>
Excepciones	<ul style="list-style-type: none"> <li>Si los precios de los productos o servicios ofrecidos por un proveedor cambian, puede ser necesario renegociar los términos de compra para garantizar que se obtengan los mejores precios</li> </ul>

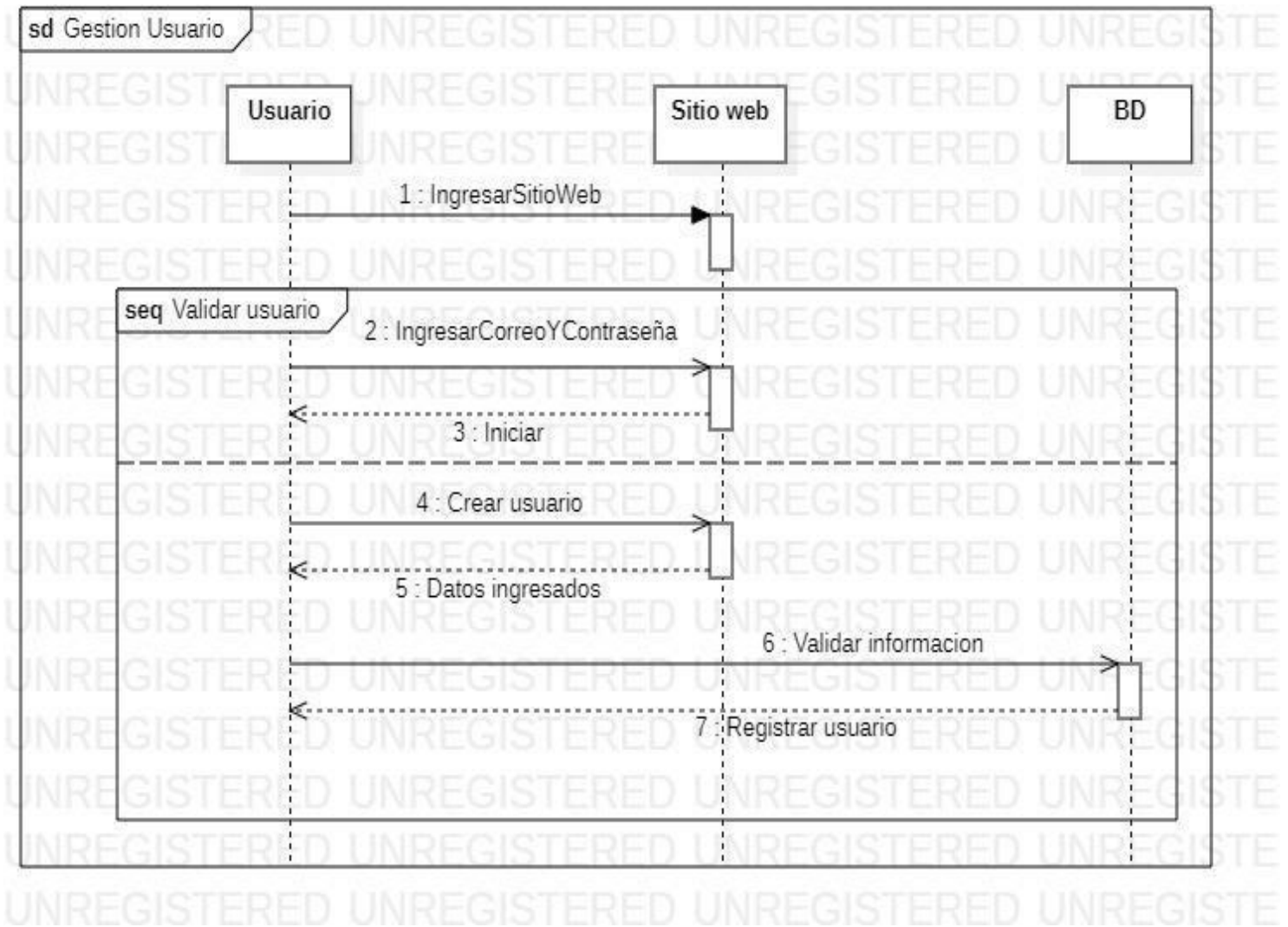
## 4. MODELOS DEL DISEÑO

### 4.1. Diagrama de clases

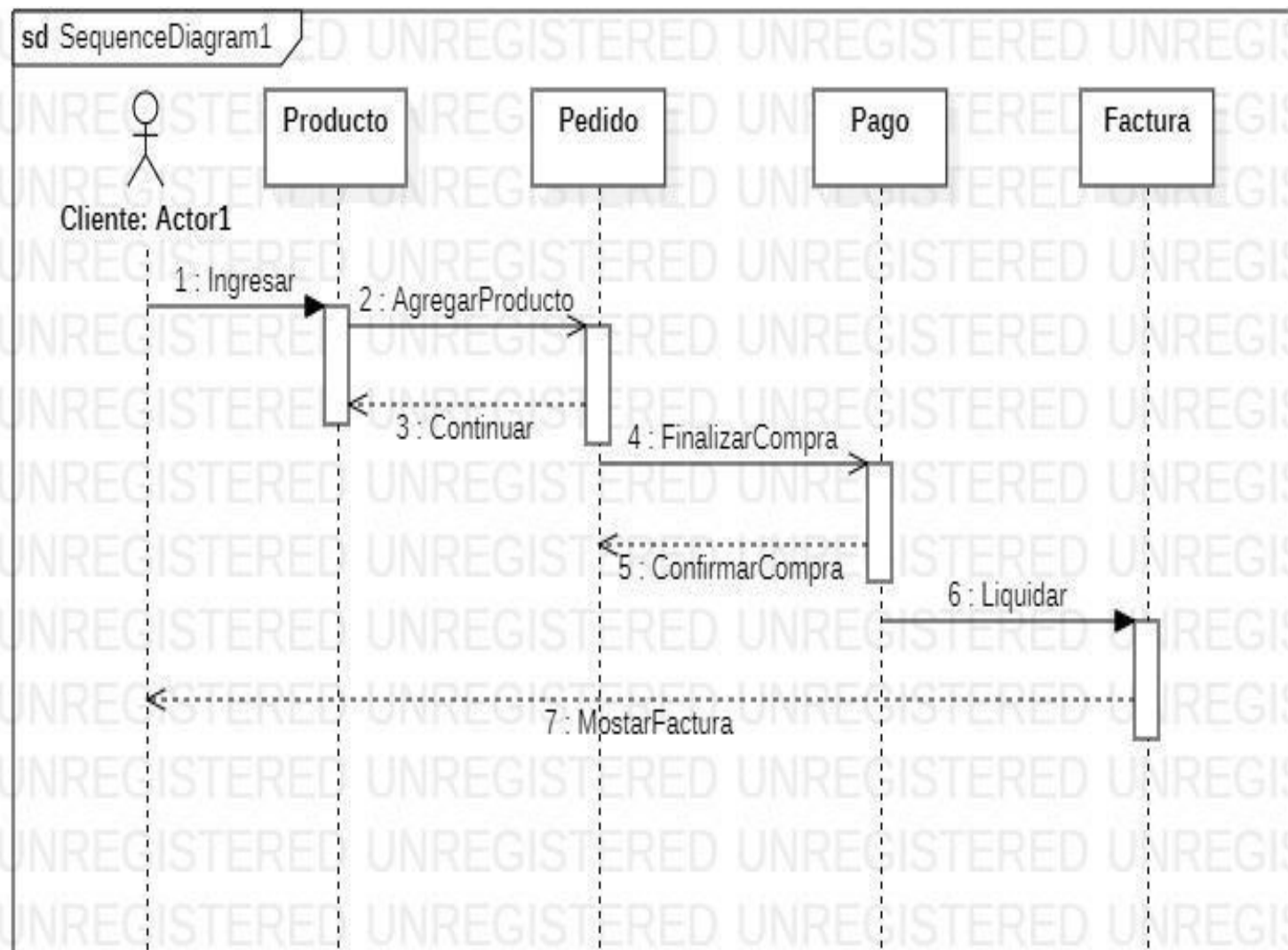


## 4.2.DIAGRAMAS DE SECUENCIA

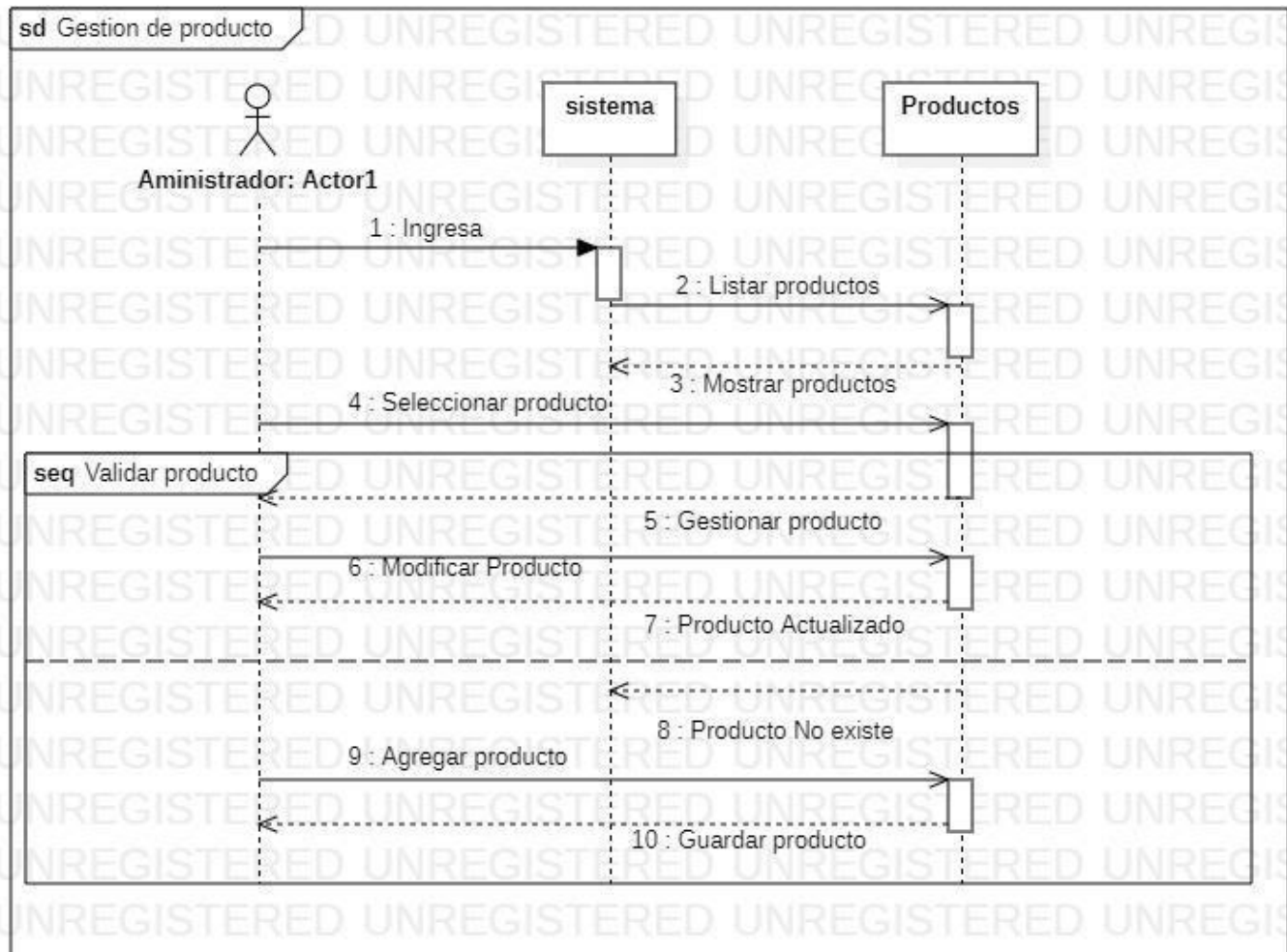
### Gestión de usuario



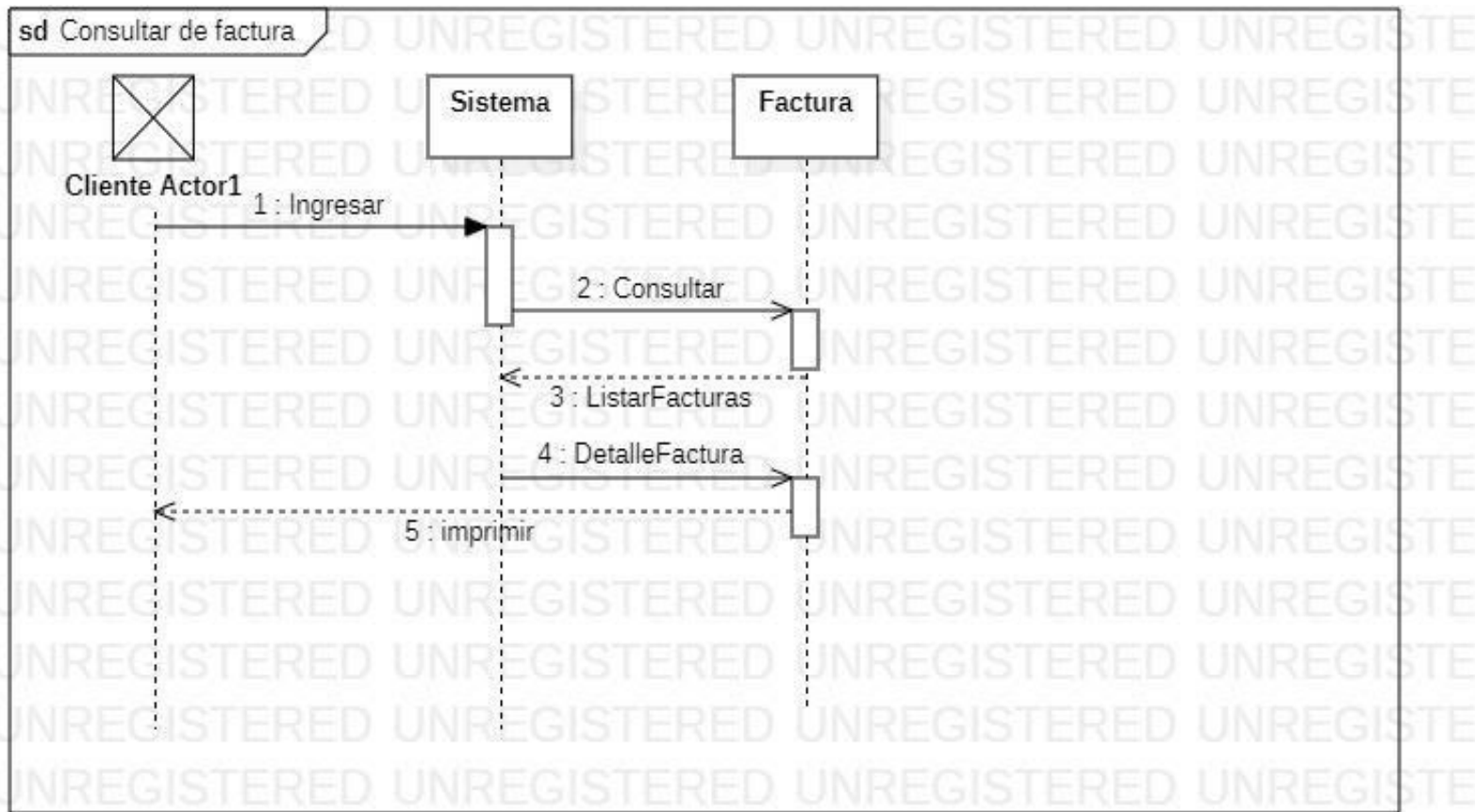
## Gestión de ventas



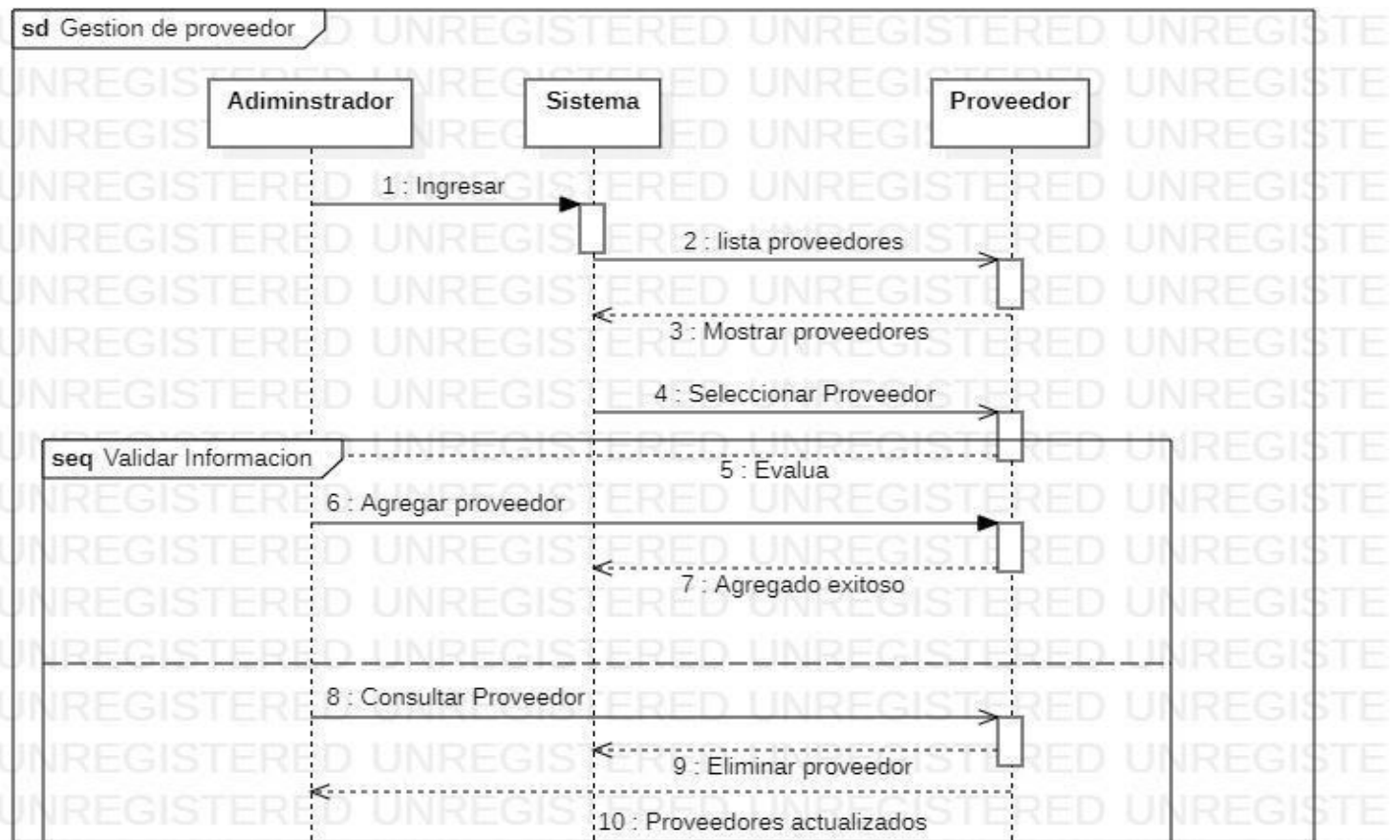
## Gestión de producto



## Consultar de factura

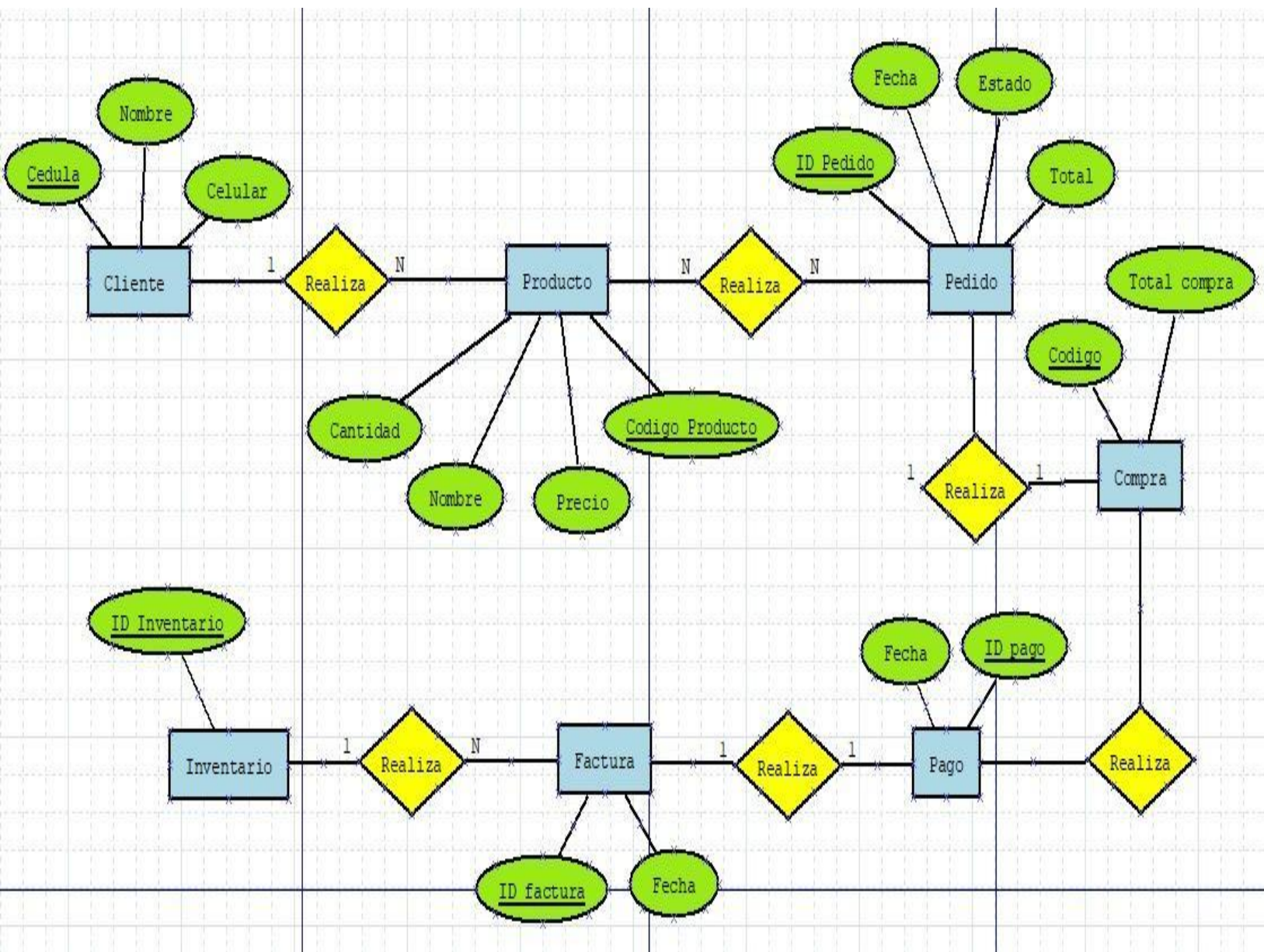


## Gestión de proveedores

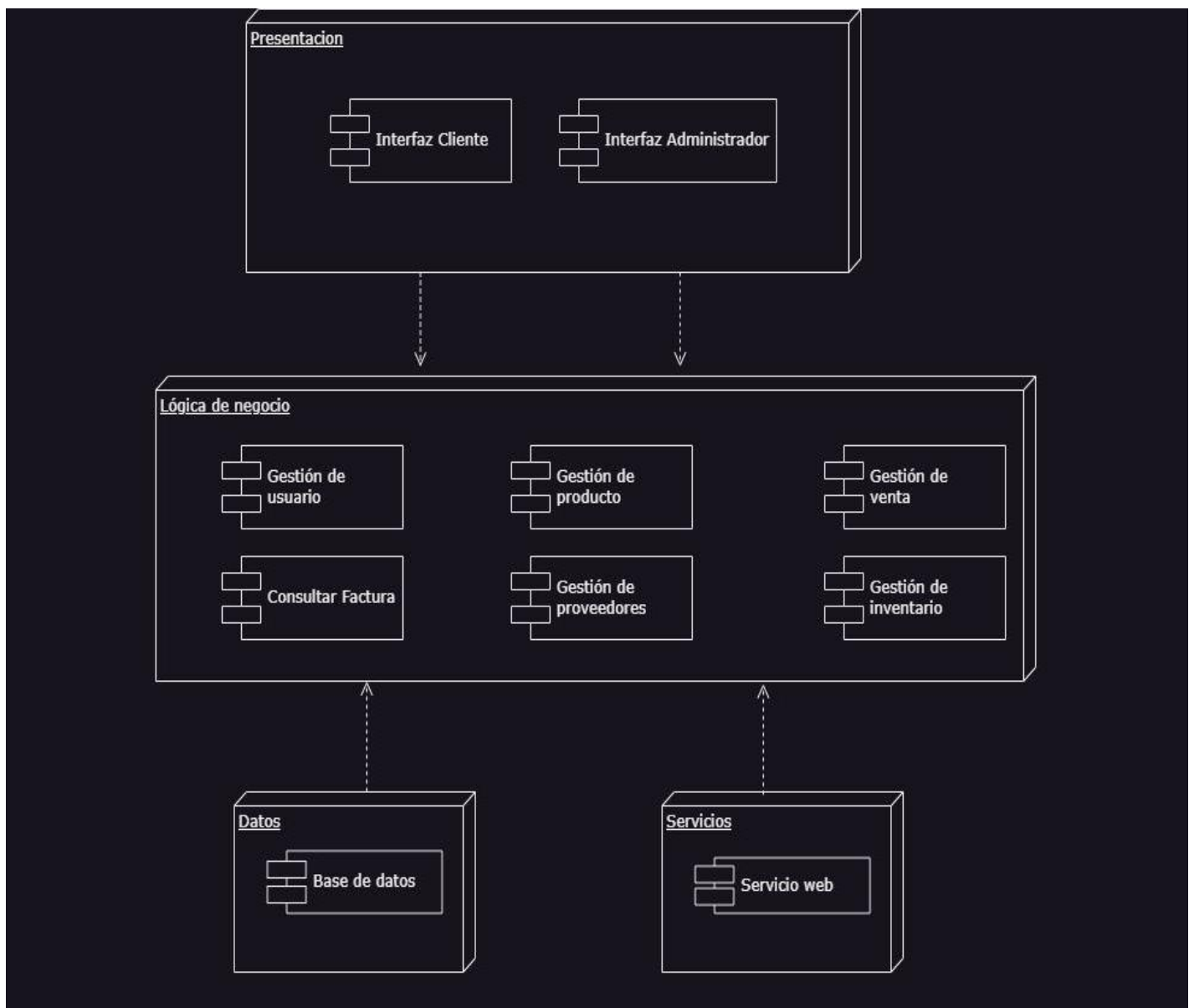




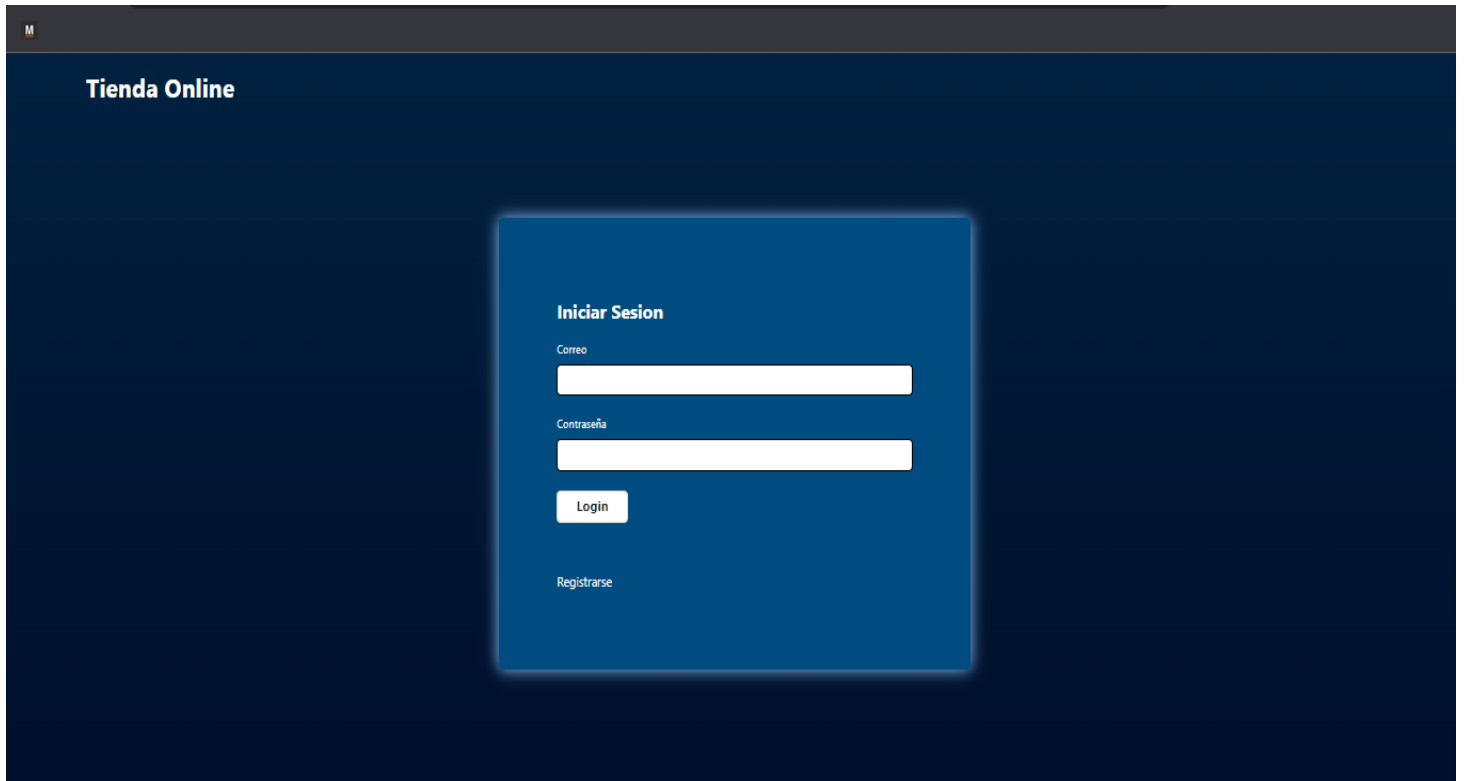
### 4.3. Modelo entidad relación



#### 4.4.DIAGRAMA DE COMPONENTES



## 1.5 Producto de la aplicación



## **2. PRUEBAS DEL SOFTWARE**

### **2.1 Introducción**

El desarrollo de un aplicativo requiere la implementación de un plan de pruebas para evaluar su puesta en marcha. Estas pruebas permiten identificar posibles fallos de funcionamiento, configuración o usabilidad que se hayan cometido durante las etapas de desarrollo. El objetivo es garantizar la confiabilidad y calidad del aplicativo, asegurando que cumpla con los requerimientos funcionales establecidos. Para lograrlo, se utilizan técnicas y herramientas especializadas que abarcan diferentes tipos de pruebas, como pruebas unitarias, de integración, de rendimiento y de aceptación. La realización de un plan de pruebas exhaustivo es esencial para minimizar riesgos y proporcionar una experiencia de uso satisfactoria a los usuarios.

### **2.2 PLANIFICACION DE PRUEBAS**

#### **2.2.1 OBJETIVOS**

##### **Objetivo general**

- Identificar y corregir eficientemente los defectos en el código del software antes de su entrega al cliente, a fin de garantizar un producto final de calidad y sin errores.

##### **Objetivos específicos**

- Identificar una amplia gama de datos válidos e inválidos para llevar a cabo pruebas exhaustivas, cubriendo diferentes escenarios y situaciones de uso del aplicativo.
- Diseñar casos de prueba utilizando diversas técnicas y enfoques, con el objetivo de detectar y reportar de manera eficiente los posibles fallos o errores en el aplicativo, optimizando el tiempo y esfuerzo empleados.
- Ejecutar los casos de prueba de forma rigurosa para evidenciar y registrar la presencia de bugs, asegurando que todos los aspectos del aplicativo sean evaluados minuciosamente.

- Validar los resultados de los casos de prueba diseñados y realizar un análisis exhaustivo de la calidad del aplicativo, utilizando los bugs encontrados como indicadores de posibles mejoras y áreas de corrección.

### 2.2.2 ALCANCE

Niveles, tipos y métodos de prueba		
Niveles, Métodos o Tipos	Nombre	Descripción
Nivel de pruebas	Pruebas unitarias	<b>Las pruebas unitarias se realizan individualmente en cada módulo del aplicativo, una vez obtenido, para garantizar su adecuado funcionamiento.</b>
	Pruebas de integración	<b>Las pruebas de integración se efectuarán después de las pruebas unitarias, considerando el funcionamiento adecuado de cada módulo del aplicativo. El propósito de estas pruebas es verificar la correcta interacción y ensamblaje de los componentes, garantizando que cumplan con la funcionalidad esperada y se ajusten a los requerimientos no funcionales del aplicativo.</b>
	Pruebas de aceptación	<b>Las pruebas de aceptación requieren que el aplicativo cumpla con todos los requisitos especificados por el cliente, siendo este último responsable de realizar las pruebas.</b>

<b>Técnicas de pruebas</b>	Técnica de caja negra	<b>El enfoque de pruebas basado en entradas y salidas se centra en probar el software evaluando sus entradas y salidas generadas, sin considerar su lógica interna o código subyacente</b>
	Técnica de caja blanca	<b>El enfoque de pruebas a nivel de código se basa en evaluar y probar las funciones internas del software, centrándose en la verificación de su lógica y comportamiento interno.</b>
<b>Métodos de prueba</b>	Clase de equivalencia	<b>Este método implica analizar y clasificar los datos de entrada en categorías de validez e invalidez, con el fin de realizar pruebas exhaustivas que abarquen diferentes escenarios y situaciones posibles.</b>
	Camino básico	<b>Este método implica la construcción de un grafo del software, donde cada nodo representa una sentencia y cada arco representa el flujo de ejecución. Mediante este grafo, se generan conjuntos de datos iniciales coherentes y relevantes para cada camino, lo que permite realizar pruebas exhaustivas y abarcar múltiples rutas posibles de ejecución.</b>
	Valores límites	<b>Este método se enfoca en probar la capacidad del programa para manejar datos que se encuentren en los límites aceptables, es decir, realizar pruebas con valores extremos, máximos o mínimos permitidos, así como situaciones límite que puedan poner a prueba</b>

		<b>la robustez y adecuado funcionamiento del software.</b>
--	--	--

### 2.2.3 MODULOS DEL SISTEMA A PROBAR

Módulos para probar	Registrar usuario Registrar producto Registrar proveedor
Objetivos de las pruebas	<p>estos módulos se realizan para validar:</p> <ul style="list-style-type: none"> <li>• Confiabilidad de la visualización de datos, Verificar que los datos ingresados o modificados se muestran correctamente en la interfaz, sin errores de visualización o información faltante.</li> <li>• Funcionamiento adecuado de los servicios, probar que los servicios respondan correctamente a las solicitudes y consultas, brindando resultados precisos y consistentes.</li> <li>• Coherencia en los estados y documentos generados, asegurar que los estados y documentos reflejan la secuencia lógica requerida, manteniendo la integridad de los datos.</li> </ul>
Detalle del orden de la ejecución de los módulos	Registrar usuario Registrar producto Registrar proveedor
Responsabilidad de la prueba	Las pruebas son responsabilidad del equipo de calidad o pruebas del proyecto, en colaboración con los usuarios, para seleccionar y realizar pruebas que garanticen la efectividad y calidad del sistema.

### 2.2.4 AMBIENTE DE PRUEBAS

- Hardware

Equipo	Procesador	DD	RAM	Aplicación a instalar
<b>Portátil</b>	AMD Ryzen I5	No	8GB	<b>Visual studio</b>
<b>Portátil</b>	<b>AMD Ryzen (5)</b>	<b>No</b>	<b>8GB</b>	<b>Visual studio</b>

· Software

· C#

## 2.3 PRUEBAS UNITARIAS

### 2.3.1 ANALISIS DE PRUEBAS

En esta prueba se debe:

- Todos los tipos de registro de entrada validos deben ser procesados.
- Todos los tipos de registro de entrada inválidos deben ser procesados.
- Todos los caminos lógicos deben ejecutarse correctamente en cada módulo de la aplicación.
- Verificar que todos los módulos del sistema estén libres de errores.

#### A. Clases de equivalencia



# Registro de usuario

Registro Usuario

\* Cedula:

\* Nombre:

\* Apellido:

\* Direccion:

\* Celular:

\* Correo Electronico:

\* Contraseña:

Guardar

Ant Design ©2023 Created by Ant UED

## Clase de equivalencia

Condición de entrada	Clases validas	Clases invalidas
Cedula  String, mayor a 4 y menor a 11	1)1000<cedula<9999999999 99	2) cedula sea =< a 4 dígitos  3) cedula sea => a 11 dígitos  4) cedula no se string  5) sea null
Nombre  Carácter mayor a 0 y menor a 50	6)3=< nombre<50	7) nombre sea < a 3  8) nombre sea => a 50  9) nombre no sea carácter

		10) sea null
Apellido Carácter mayor a 0 y menor a 50	11) 3= <apellido<50	12) apellido sea < a 3 13) apellido sea => a 50 14) apellido no sea carácter
Dirección Caracter menor a 50 mayor a 0	15) 3=<dirección<50	16) dirección sea < a 3 17) dirección sea => a 50 18) dirección no sea alfanumérico 19) sea null
Celular String, 10 dígitos	20) celular=10	21) celular<>10 22) celular no sea string
Correo Carácter, mayor a 0 y menor a 50	23) 3=<<correo<50	24) correo sea < a 3 25) correo sea => a 50 26) correo no sea caracter 27) sea null
Contraseña caracter, un letra mayuscula, un numero, un carácter especial.	28) 6=< contraseña<20	29) contraseña sea < a 6 30) contraseña sea => a 20 31) no contenga letra mayúscula 32) no contenga numero 33) no contenga carácter especial 34) sea null

Casos de pruebas por clases de equivalencia

No clase	Dato de entrada	Valor	escenario
----------	-----------------	-------	-----------

1	Cedula	20000	Valido
2	Cedula	123	Invalido
6	Nombre	“Marta”	Valido
9	Nombre	8765	Invalido
11	Apellido	“Martínez”	Valido
12	Apellido	“ ”	Invalido
15	Dirección	“Calle20”	valido
16	Dirección	“ ”	invalido
20	Celular	3006758967	valido
21	Celular	7654	invalido
23	Correo	“andrea@hotmail.com”	valido
24	Correo	“ ”	invalido
28	Contraseña	“Camila12-”	Valido
29	Contraseña	“jahk”	invalido

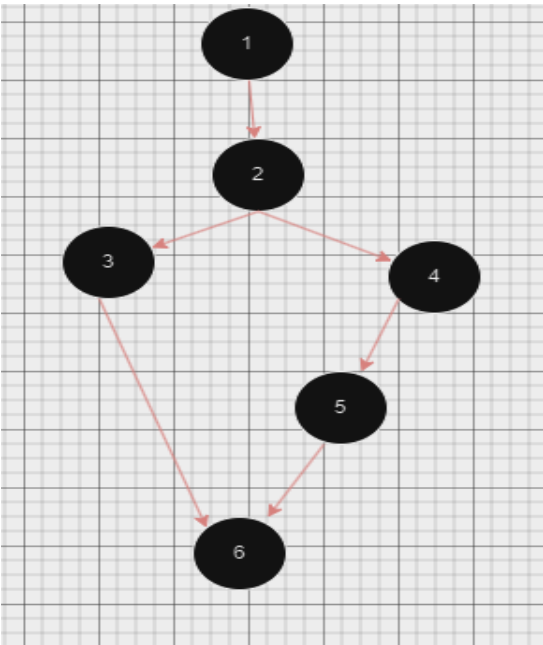
#### B. Análisis de valores limites

No clase	Datos de entrada	Valor	Escenario
1	Cedula	1167845378	Correcto
2	Cedula	12345	Correcto
3	Cedula	11678453789	Incorrecto

4	Cedula	1234	Incorrecto
5	Nombre	“maria...(49)”	Correcto
6	Nombre	“ang”	Correcto
7	Nombre	“maría...(50)”	Incorrecto
8	Nombre	“an”	Incorrecto
9	Apellido	“perez...(49)”	Correcto
10	Apellido	“vej”	Correcto
11	Apellido	“perez...(50)”	Incorrecto
12	Apellido	“ve ”	Incorrecto
13	Dirección	“calle15....(49)”	Correcto
14	Dirección	“abc”	Correcto
15	Dirección	“calle15....(50)”	Incorrecto
16	Dirección	“ ab ”	Incorrecto
17	Correo	“antonio@hotmail(49)”	Correcto
18	Correo	“mar”	Correcto
19	Correo	“antonio@hotmail(50)”	Incorrecto
20	Correo	“ ma”	Incorrecto
21	Contraseña	“Abc2-(19)”	Correcto
22	Contraseña	“Mari2-(6)”	Correcto
23	Contraseña	“bcd2-(20)”	Incorrecto
24	Contraseña	“Vmn1-(5)”	Incorrecto
25	Celular	3007658745	correcto
26	Celular	31478954879	incorrecto
27	Celular	319687458	incorrecto

C. Pruebas del camino básico

```
public Usuario CrearUsuario(Usuario usuario)
{
    try
    {
        var usuarioResponse = _context.Usuarios.Find(usuario.Cedula);
        if (usuarioResponse is not null) return null;
        _context.Usuarios.Add(usuario);
        _context.SaveChanges();
        return usuario;
    }
    catch (Exception ex)
    {
        return null;
    }
}
```



Arista- nodos +2	6-6+2=2
Regiones cerradas	2
nodos de condición+1	1+1=2

Números de caminos	Caminos
1	0,1,2,3,6
2	0,1,2,4,5,6

Caminos	Datos de entrada	Escenario
1,2,3,6	"1118273", camilo, torres	Correcto
1,2,4,5,6	"1182636", " null"	incorrecto

### Registro de producto

#### Tienda Online

### Registro Producto

\* Nombre:

\* Precio:

\* Característica:

\* Cantidad:

Guardar

Ant Design ©2023 Created by Ant UED

#### A. Clases de equivalencia

Condición de entrada	Clases validas	Clases invalidas
----------------------	----------------	------------------

Nombre Carácter, mayor = a 1 y menor a 50	1)1<=nombre<50	2) nombre sea < 1 3) nombre sea => 50 4)nombre no sea carácter 5) sea null
Precio #, mayor a 0	6)precio>0	7) precio sea <= 0 8) No sea numérico 9)sea null
Característica Alfanumérico, mayor a 0 y menor a 200	10)0<característica<=200	11)característica<=0 12)característica>200 13) sea null
Cantidad #mayor a 0	14)cantidad>0	15) precio sea <= 0 16) No sea numérico 17)sea null

#### Casos de pruebas por clases de equivalencia

No clase	Dato de entrada	Valor	escenario
1	Nombre	“blusa”	Valido
2	Nombre	“ ”	Invalido
6	Precio	12000	Valido
8	Precio	“abc”	Invalido
10	Característica	“poliester”	Valido
12	Característica	“algodón.....x300”	invalido
13	Cantidad	20	Valido
16	Cantidad	“dos”	invalido

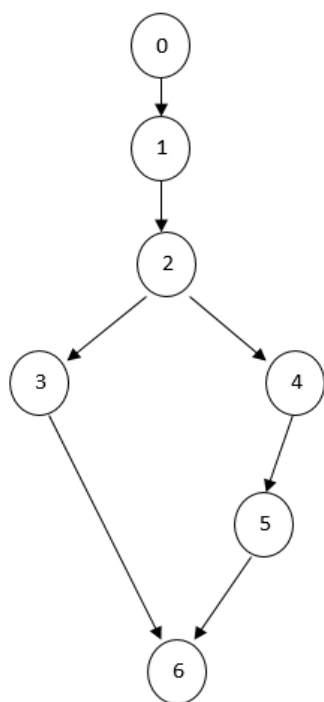
## B. Análisis de valores límites

N. Clase	Datos De Entrada	Valor	Escenario
1	Nombre	“Suéter.....(49)”	Correcto
2	Nombre	“A”	Correcto
3	Nombre	“Pantalon....(50)”	Incorrecto
4	Nombre	“ “	Incorrecto
5	Característica	“Rojo.....(200)”	Correcto
6	Característica	“A”	Correcto
7	Característica	“Largo.....(201)”	Incorrecto
8	Característica	“ “	incorrecto

## C. Pruebas del camino básico

```
@router.post("registrar_producto", status_code=status.HTTP_201_CREATED)
async def register_product(data: dict = Body(...)):
    try:
        product_create = ProductCreate(**data)
        search_producto = await buscarProducto(product_create.codigo)
        if (search_producto):
            return "Ya existe"
        response = guardarProductDatabase(product_create)
        return response
    except Exception as e:
        raise HTTPException(
            status_code=status.HTTP_500_INTERNAL_SERVER_ERROR,
            detail="Error: " + str(e),
        )
```





Arista- nodos +2	$6-6+2=2$
Regiones cerradas	2
nodos de condición+1	$1+1=2$

Números de caminos	Caminos
1	0,1,2,3,6
2	0,1,2,4,5,6

Caminos	Datos de entrada	Escenario
0,1,2,3,6	“ABC123”, Pantalón, 60.000	Correcto
0,1,2,4,5,6	"ABC123", “Sueter”, 20.000	Correcto

Registrar Proveedor

Tienda Online

Log Out

### Registro Proveedor

\* NIT:

\* Nombre:

\* Apellido:

\* Direccion:

Celular:

Guardar

#### A. Clases de equivalencia

Condición de entrada	Clases validas	Clases invalidas
Nit String mayor=1, menor a 10	1) 1 ≤ Nit < 9999999999	2) Nit < a 1 dígitos 3) Nit => a 10 digitos 4) No sea string 5) sea null
Nombre Carácter mayor a 0 y menor a 50	6) 3 ≤ nombre < 50	7) nombre sea < a 3 8) nombre sea => a 50 9) nombre no sea carácter 10) sea null
Apellido Carácter mayor a 0 y menor a 50	11) 3 ≤ <apellido < 50	12) apellido sea < a 3 13) apellido sea => a 50 14) apellido no sea carácter
Dirección Carácter menor a 50 mayor a 0	15) 3 ≤ <dirección < 50	16) dirección sea < a 3 17) dirección sea => a 50

		18) dirección no sea caracter 19) sea null
Celular string, 10 dígitos	20) celular=10	21) celular<>10 22) celular no sea string

#### Casos de pruebas por clases de equivalencias

N. Clases	Dato de entrada	Valor	Escenario
1	Nit	10031986	Correcto
3	Nit	1003198689	Incorrecto
6	Nombre	Antonio	Correcto
9	Nombre	“ ”	Incorrecto
10	Apellido	Cardona	Correcto
13	Apellido	“ ”	Incorrecto
14	Dirección	“Carrera 25”	Correcto
16	Dirección	“xxxxxxxxxx (51 caracteres)”	Incorrecto
18	Celular	3003471251	Correcto
19	Celular	30034712	Incorrecto

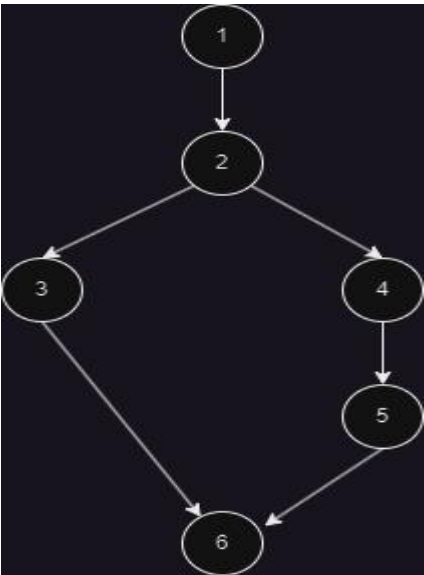
#### B. Análisis de valores límites

N. de caso	Dato de entrada	Valor	Escenario
1	Nit	9999999999	Correcto
2	Nit	1	Correcto

3	Nit	99999999991	Incorrecto
4	Nit	“ ”	Incorrecto
5	Nombre	Alejandro (49 caracteres)	Correcto
6	Nombre	Seb	Correcto
7	Nombre	Alald (50 caracteres)	Incorrecto
8	Nombre	Al	Incorrecto
9	Apellido	Garces (49 caracteres)	Correcto
10	Apellido	Mir	Correcto
11	Apellido	“xxxxxxx (50 caracteres)”	Incorrecto
12	Apellido	Mi	Incorrecto
13	Dirección	Calle 15 (49 caracteres)	Correcto
14	Dirección	Car	Correcto
15	Dirección	“xxxxxxx (50 caracteres)”	Incorrecto
16	Dirección	xx	Incorrecto
17	Celular	3022345916	Correcto
18	Celular	30034712511	Incorrecto
19	Celular	316501910	Incorrecto

### C. Pruebas del camino básico

```
1 referencia
public Proveedor RegistrarProveedor (Proveedor proveedor)
{
    try
    {
        Proveedor proveedorResponse = _context.Provedores.Find(proveedor.NIT);
        if (proveedorResponse is not null) return null;
        _context.Provedores.Add(proveedor);
        _context.SaveChanges();
        return proveedor;
    }
    catch(Exception ex)
    {
        return null;
    }
}
```



$V(G)= 6-6+2=2$

$V(G)=1+1=2$

$V(G)=2$

**CAMINOS**

- 1. 1,2,3,6
- 2. 1,2,4,5,6

Camino	datos de entrada	Escenario
1,2,3,6	new, "123456789"	Correcto
1,2,4,5,6	new, null	Incorrecto

### 2.3.2 Diseño casos de pruebas

Entrada Tipo	Objeto/ variable de estado(Valores)	Clases válidas	Clases inválidas
Registro de usuario Cedula string	Usuario Cedula Estado(Registrado/no registrado)	1. Formato valido (3-10 Digits) 2. no registrado	0. registrado 0. No cumpla con el formato(mayor a 10 dígitos y menor a 3)
Registro de Producto Código string	Producto Estado (Exista/no exista)	5)No existe	6)Registrado 7) No cumpla con el formato de dígitos 8) Campo Vacío
Registro de Proveedor NIt String	Proveedor Estado(Exista/No exista)	9) Formato válido(1-10 dígitos) 10) No registrado	11) registrado 12) No cumpla el formato

	Variable Entrada	Código Producto	tipo	String
#	Valido / No valido	Formato	Objeto/Variable Estado	
			Producto	Estado
1	Válido	8 Dígitos	No Existe	No Registrado
2	No válido	8 Dígitos	No existe	Registrado
3	No Válido	8 Dígitos	Existe	
4	No Válido	En Blanco NO digitos Más de 10 Dígitos Menos de 3 Dígitos		

	Variable Entrada	Nit Proveedor	tipo	String
#	Valido / No valido	Formato	Objeto/Variable Estado	
			Proveedor	Estado
1	Válido	10 Dígitos	NO existe	No existe
2	No válido	10 Dígitos	No existe	Registrado
3	No Válido	10 Dígitos	Existe	
4	No Válido	En Blanco NO digitos Más de 10 Dígitos Menos de 3 Dígitos		

	Variable Entrada	Codigo de usuarios	tipo	String
#	Valido / No valido	Formato	Objeto/Variable Estado	
			Usuario	Estado
1	Válido	8 Dígitos	No Existe	No Registrado
2	No válido	8 Dígitos	No Existe	Registrado
3	No Válido	8 Dígitos	No existe	
4	No Válido	En Blanco NO digitos Más de 10 Digitos Menos de 3 Dígitos		

### 2.3.3 Ejecución de casos de pruebas

#### Registrar usuario

```
        _usuarioService = new UsuarioService(_contextWrapperMock.Object);
    }

    [Test]
    public void CrearUsuario_ValidUsuario_ReturnsUsuario()
    {
        // Arrange (preparación)
        Usuario usuario = new Usuario
        {
            Cedula = "123456789",
            Nombre = "Usuario 1",
            Correo = "correo@example.com",
            HashPassword = "123456"
        };

        // Act (acción)
        Usuario result = _usuarioService.CrearUsuario(usuario);

        // Assert (verificación)
        Assert.NotNull(result); // Verifica que el resultado no sea nulo
        Assert.AreEqual(usuario, result); // Verifica que el objeto devuelto sea el mismo que se pasó como parámetro

        _contextWrapperMock.Verify(c => c.Usuarios.Find(usuario.Cedula), Times.Once);
        _contextWrapperMock.Verify(c => c.Usuarios.Add(usuario), Times.Once);
        _contextWrapperMock.Verify(c => c.SaveChanges(), Times.Once);
    }
}
```

#### Registrar proveedor



```

[Test]
0 referencias
public void RegistrarProveedor_ValidProveedor_ReturnsProveedor()
{
    // Arrange (preparación)
    Proveedor proveedor = new Proveedor
    {
        NIT = "123456789",
        Nombre = "Proveedor 1",
        Direccion = "Calle 123",
        Celular = "1234567890"
    };

    // Act (acción)
    Proveedor result = _proveedorService.RegistrarProveedor(proveedor);

    // Assert (verificación)
    Assert.NotNull(result); // Verifica que el resultado no sea nulo
    Assert.AreEqual(proveedor, result); // Verifica que el objeto devuelto sea el mismo que se pasó como parámetro

    _contextWrapperMock.Verify(c => c.Provedores.Find(proveedor.NIT), Times.Once);
    _contextWrapperMock.Verify(c => c.Provedores.Add(proveedor), Times.Once);
    _contextWrapperMock.Verify(c => c.SaveChanges(), Times.Once);
}

```

## Registrar producto

```

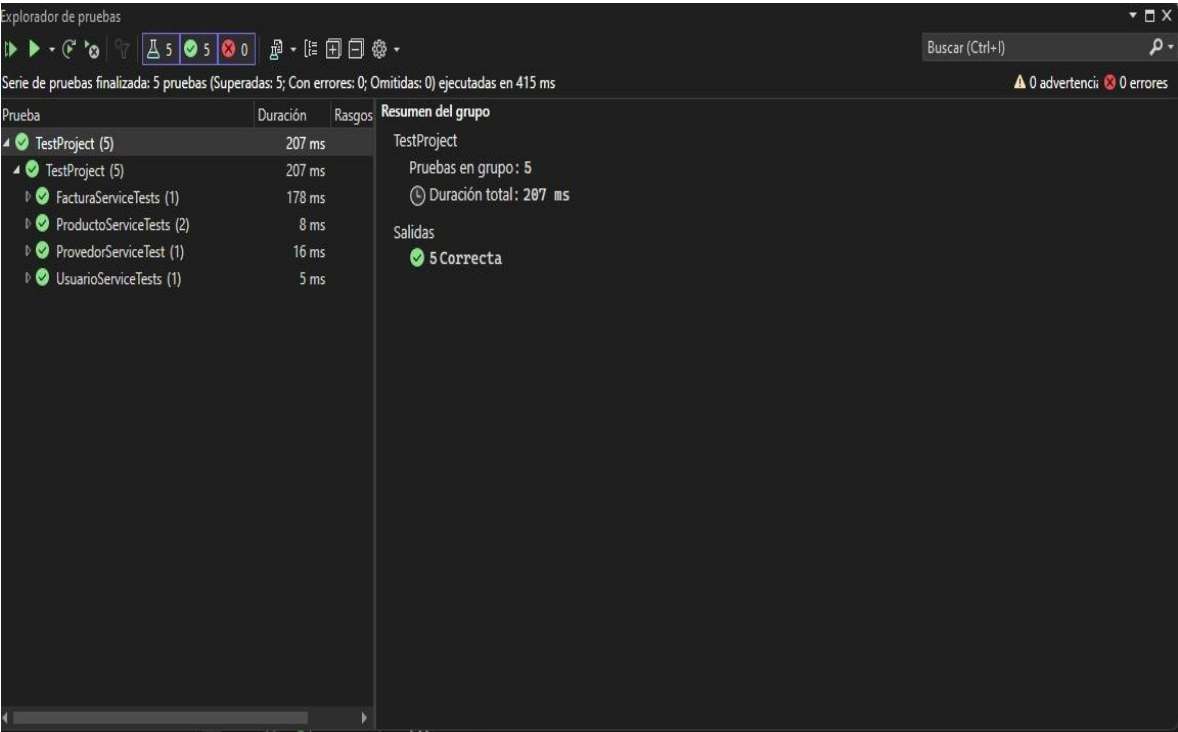
[Test]
0 referencias
public void GuardarProducto_ValidProducto_ReturnsProducto()
{
    // Arrange (preparación)
    Productos producto = new Productos
    {
        Codigo = "ABC123",
        Nombre = "Producto 1",
        Precio = 10,
        Cantidad = 5
    };

    // Act (acción)
    Productos result = _productoService.GuardarProducto(producto);

    // Assert (verificación)
    Assert.NotNull(result); // Verifica que el resultado no sea nulo
    Assert.AreEqual(producto, result); // Verifica que el resultado sea igual al producto ingresado

    _contextWrapperMock.Verify(c => c.Productos.Add(producto), Times.Once);
    _contextWrapperMock.Verify(c => c.SaveChanges(), Times.Once);
}

```



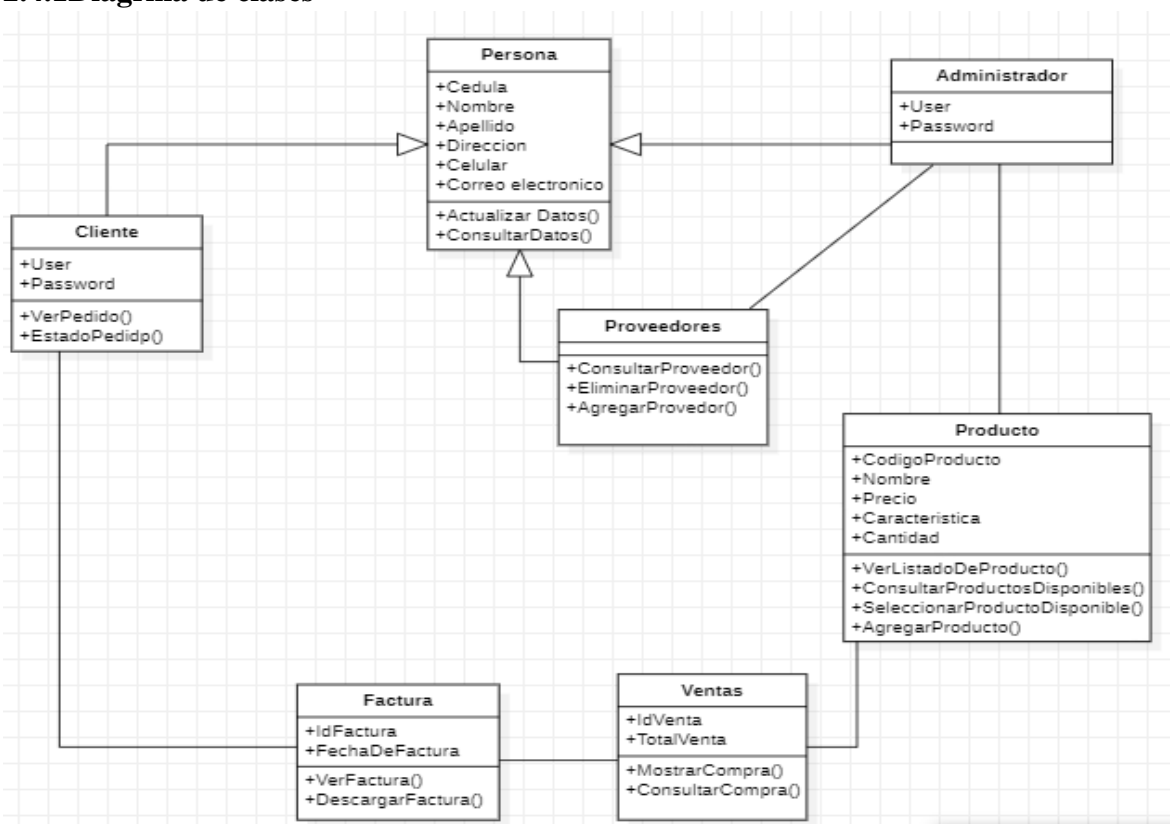
2.3.4 Evaluación de la prueba

Componente / Producto	Caso de prueba	Resultado	Seguimiento	Conclusión
Clase usuario	Registro usuario	Exitoso	Se llevarán a cabo las pruebas de integración	Se registró el usuario
Clase producto	Registro	Exitoso	Se llevarán a cabo las pruebas de integración	Se registró el producto

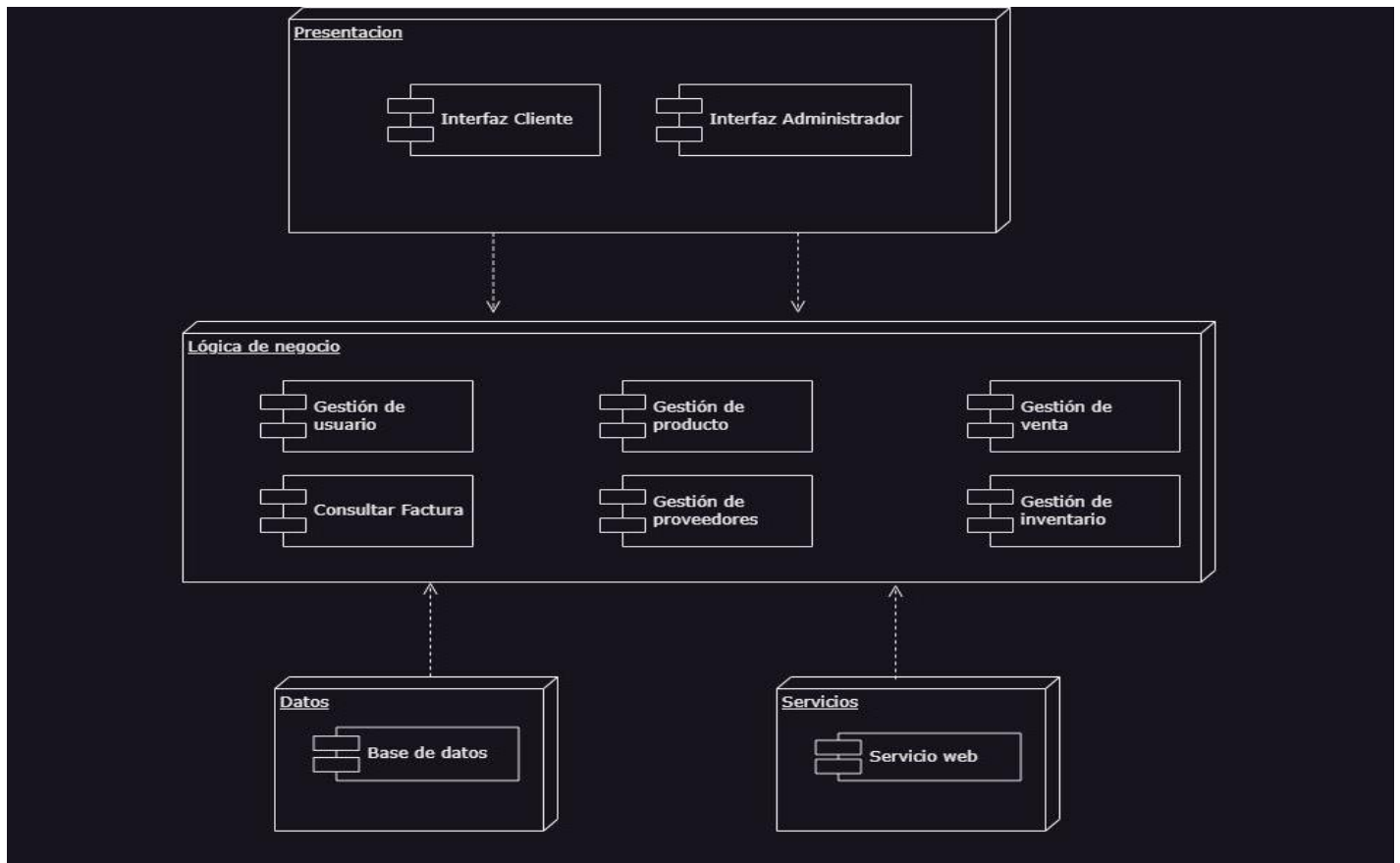
	producto			
<b>Clase proveedor</b>	Registro proveedor	Exitoso	Se llevará a cabo las pruebas de integración	se registró el proveedor

## 2.4 Pruebas de integración

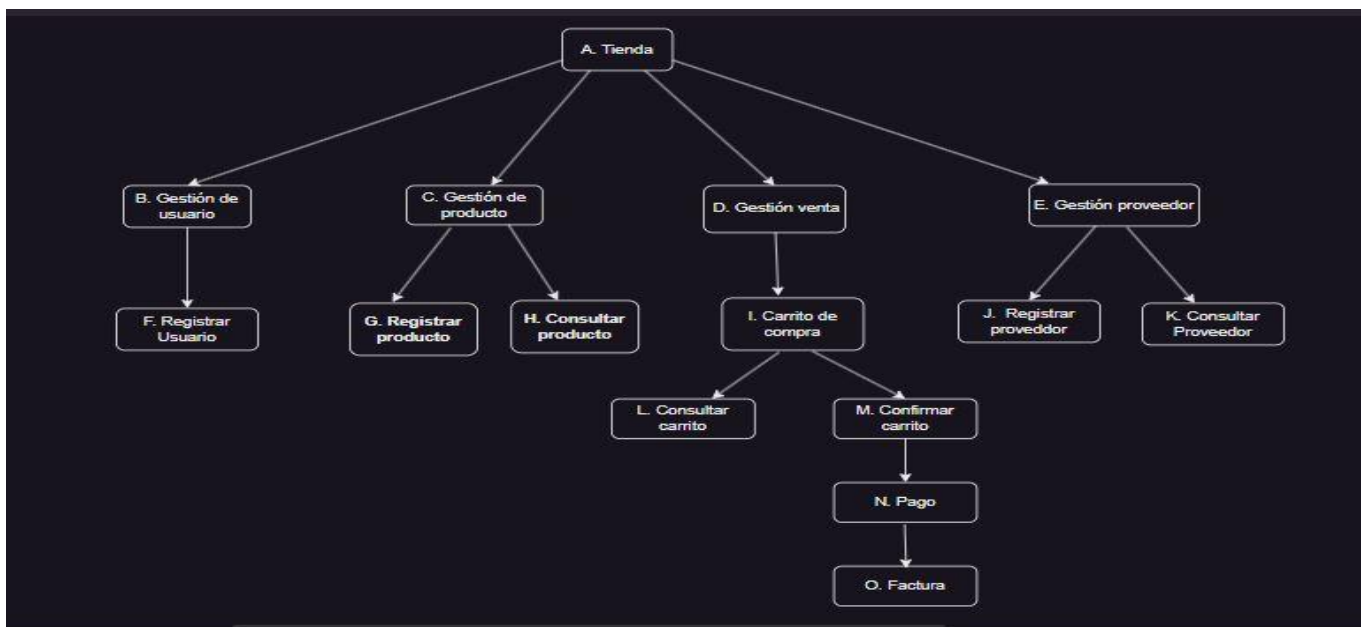
## 2.4.1 Diagrama de clases



## 2.4.2 Diagrama de componentes



## 2.4.3 Diseño estructural de la aplicación



## 2.4.3 Estrategia de pruebas incrementales

### 2.4.4.1 Incremental Ascendente

1. (F,G,H,L,Ñ,J,K)
2. (N-Ñ)
3. (M-N)
4. (I-L)(I-M)
5. (B-F)(C-G)(C-H)(D-I)(E-J)(E-K)
6. (A-B)(A-C)(A-D)(A-E)

### 2.4.4.2 Incremental descendente

- **POR PROFUNDIDAD**  
(A,B,F,C,G,H,D,I,L,M,N,Ñ,E,J,K)
- **POR ANCHURA**  
(A,B,C,D,E,F,G,H,I,J,K,L,M,N,Ñ)

### 2.4.4.3 Diseños de los casos de pruebas

NUMERO DE CASO DE PRUEBA	COMPONENTE	DESCRIPCIÓN DE LO QUE SE PROBARA	PRERREQUISITOS
1	B-F	Se probará el funcionamiento del caso registrar usuario para verificar que todos los métodos se realicen correctamente.	No estar registrado
2	C-F	Se probará el funcionamiento del caso de registrar producto para verificar los métodos correctamente	Haber registrado al usuario(administrador)
3	E-J	Se probará el funcionamiento del caso de uso registrar proveedor para verificar el correcto	Haber registrado al usuario(administrador)

		funcionamiento de los métodos.	
--	--	--------------------------------	--

Usuari o					
Paso	Descripción de paso a seguir	Datos de entrada	Salida esperada	¿Ok?	Observación
1	Ingresa a la página web		Mostrar página web	Ok	
2	Presionar el botón registrar		Formulario de registro	ok	
3	Rellenar formulario	Identificación: "10641", Nombre: "camilo" Apellido: "Gonzales" Dirección: "Calle 25#28-80 Cortijos" Celular: "3008765678", Correo electrónico: "camilo@gmail.com" Contraseña: "Camilo123-",		Ok	

PRODUCTO					
Paso	Descripción de paso a seguir	Datos de entrada	Salida esperada	¿Ok?	observación
1	Ingresar a la página web		Mostrar página web	Ok	
2	Iniciar sesión	Correo: <u>admin@gmail.com</u>  Contraseña: Hola123*	Sesión iniciada	ok	Ser administrador
3	Hacer click en productos		Desplegar Opciones de productos	Ok	
4	Hacer click en registrar productos	Nombre: “Blusa”  Precio: “29.999”  Característica: “Blusa roja talla s tela de algodón”  Cantidad: “100”	Formulario de registro de producto	ok	
5	Hacer click en guardar		Producto guardado	ok	



#### 2.4.4.4 Ejecución de las pruebas

Herramienta selenium para hacer las pruebas de integración

Registro de usuario

```
0 referencias
internal class Program
{
    0 referencias
    static void Main(string[] args)
    {
        IWebDriver driver = new ChromeDriver();
        driver.Navigate().GoToUrl("http://localhost:3000");
        SeleniumProducto(driver);
        driver.Quit();
    }

    0 referencias
    public static void SeleniumUsuario(IWebDriver driver)
    {
        IWebElement button = driver.FindElement(By.Id("singUpButton"));
        button.Click();
        Thread.Sleep(TimeSpan.FromSeconds(3));

        IWebElement cedula = driver.FindElement(By.Id("cedula"));
        cedula.SendKeys("2454545");

        IWebElement nombre = driver.FindElement(By.Id("nombre"));
        nombre.SendKeys("andres");

        IWebElement apellido = driver.FindElement(By.Id("apellido"));
        apellido.SendKeys("delgado");

        IWebElement direccion = driver.FindElement(By.Id("direccion"));
        direccion.SendKeys("valledupar");

        IWebElement celular = driver.FindElement(By.Id("celular"));
        celular.SendKeys("3103048901");

        IWebElement correo = driver.FindElement(By.Id("correo"));
        correo.SendKeys("hola@gmail.com");

        IWebElement password = driver.FindElement(By.Id("password"));
        password.SendKeys("Hola123*");

        IWebElement guardar = driver.FindElement(By.Id("guardar"));
        guardar.Click();
        Thread.Sleep(TimeSpan.FromSeconds(5));
    }
}
```

```
C:\Users\norma\source\repos >
Starting ChromeDriver 114.0.5735.90 [386bc09e8f4f2e025eddae123f36f6263095ae49-refs/branch-heads/57350[1052]] on port 56
268
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.

DevTools listening on ws://127.0.0.1:56266/devtools/browser/f3167401-1480-4278-8d2b-2914a16131bc
```

## Registro de producto

```
1 referencia
public static void SeleniumProducto (IWebDriver driver)
{
    IWebElement button = driver.FindElement(By.Id("singInButton"));
    button.Click();
    Thread.Sleep(TimeSpan.FromSeconds(3));

    IWebElement element = driver.FindElement(By.Id("correo"));
    element.SendKeys("admin@gmail.com");

    IWebElement element2 = driver.FindElement(By.Id("password"));
    element2.SendKeys("Hola123*");

    IWebElement buttonLogin = driver.FindElement(By.Id("loginBtn"));
    buttonLogin.Click();
    Thread.Sleep(TimeSpan.FromSeconds(3));

    IWebElement productosIcon = driver.FindElement(By.Id("productos"));
    productosIcon.Click();
    Thread.Sleep(TimeSpan.FromSeconds(1));

    IWebElement registroIcon = driver.FindElement(By.Id("registroProducto"));
    registroIcon.Click();
    Thread.Sleep(TimeSpan.FromSeconds(3));

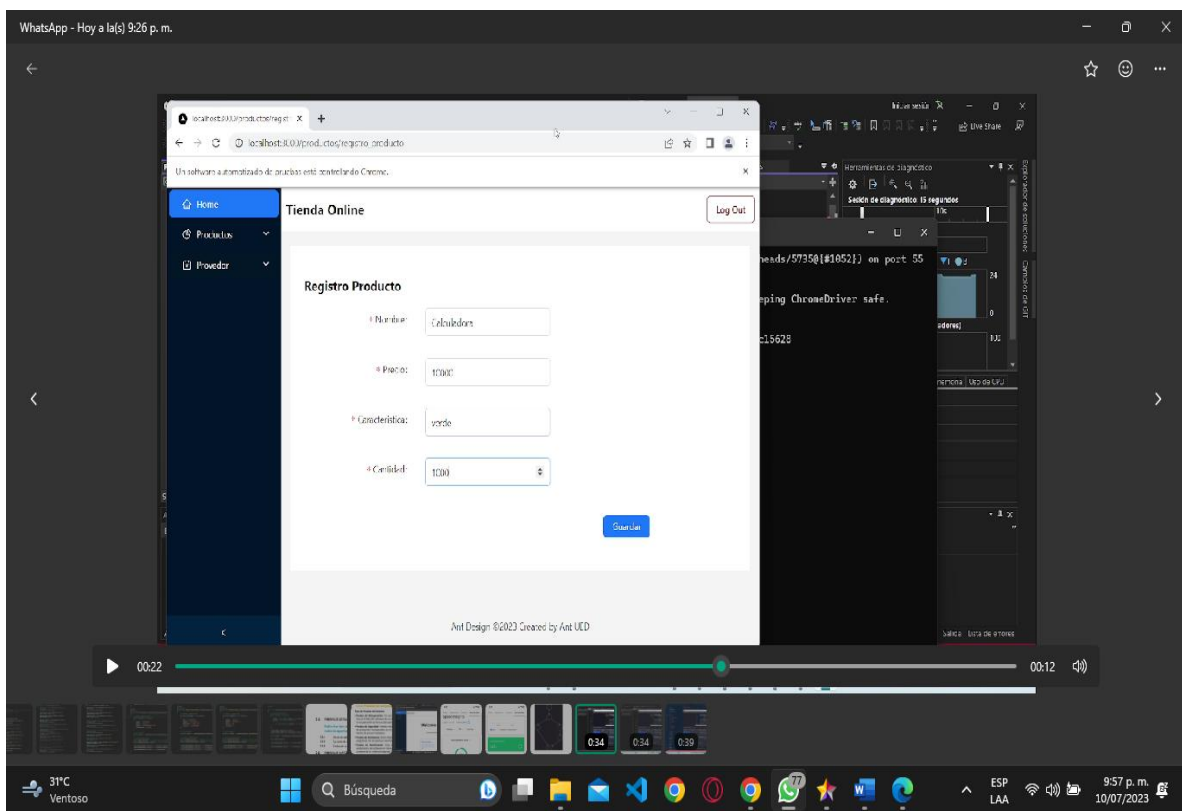
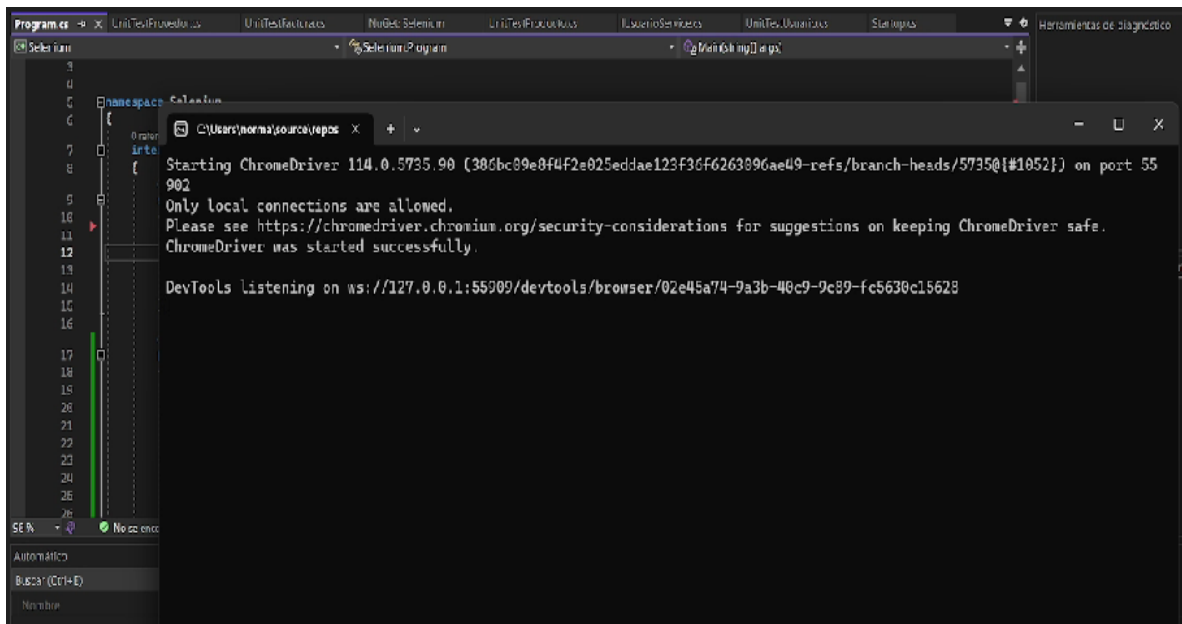
    IWebElement nombre = driver.FindElement(By.Id("nombre"));
    nombre.SendKeys("Calculadora");

    IWebElement precio = driver.FindElement(By.Id("precio"));
    precio.SendKeys("10000");

    IWebElement caracteristica = driver.FindElement(By.Id("caracteristica"));
    caracteristica.SendKeys("verde");

    IWebElement cantidad = driver.FindElement(By.Id("cantidad"));
    cantidad.SendKeys("1000");

    IWebElement guardar = driver.FindElement(By.Id("guardar"));
    guardar.Click();
    Thread.Sleep(TimeSpan.FromSeconds(5));
}
```



#### 2.4.4.5 Evaluación de las pruebas

Prueba				
Componente / Producto	Caso de prueba	Resultado	Seguimiento	Conclusión
Componente Usuario	Registro usuario	Exitoso	Se procederá a realizar las pruebas de aceptación	
Componente producto	registro producto	Exitoso	Se procederá a realizar las pruebas de aceptación	

#### 2.4.5 Pruebas basadas en hilos

##### 2.4.5.1. Descripción de casos de uso

#### 4.5.DESCRIPCION DE CASOS DE USO

Documentación Casos de Uso	
Caso de uso	Gestión de usuario
Actores	Administrador, cliente
Tipo propósito	Asociación, incluye
Descripción	Se permite al usuario iniciar sesión accediendo al sitio web, mantener actualizados los datos del usuario y ver su historial de compra  Se permite al administrador iniciar sesión y tener acceso a todo el sitio web.
Precondición	<ul style="list-style-type: none"><li>● El usuario ya este registrado para ver el historial de compra,</li><li>● el usuario debe haber iniciado sesión previamente y haber realizado al menos una compra en la tienda para tener acceso al historial usuario</li></ul>

Postcondiciones	El usuario está registrado satisfactoriamente en la base de datos del sistema.
Flujo normal	<ul style="list-style-type: none"> <li>● ingresar al sitio web por medio de un navegador</li> <li>● el usuario le da la opción de registrar usuario y se le muestra el formulario</li> <li>● registro de usuario ingresando sus datos personales creando un usuario y contraseña</li> <li>● Se crea la cuenta satisfactoriamente</li> <li>● inicio de sesión de usuario ingresando el usuario y contraseña</li> <li>● editar datos personales del usuario</li> <li>● Cerrar sesión</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>● En caso de que el usuario no puede crear una cuenta es porque la cuenta ya existe.</li> <li>● En caso de que no se permita iniciar sesión deberá estar previamente registrado</li> <li>● Si el sistema no puede verificar la integridad de los datos modificados por el usuario, muestra un mensaje de error y solicita al usuario que verifique los datos e intente modificarlos.</li> </ul>

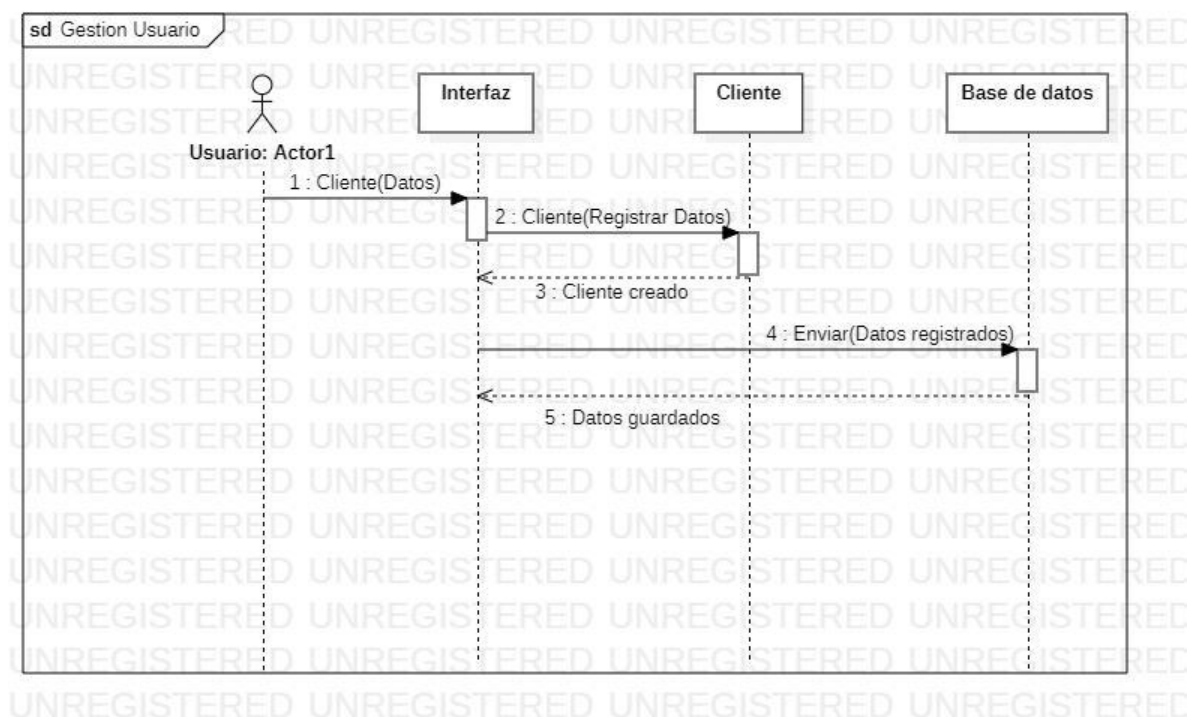
Documentación Casos de Uso	
Caso de uso	Gestión de producto
Actores	Administrador
Tipo propósito	Asociación, incluye
Descripción	El usuario puede buscar a cualquier producto disponible, acceder al listado de todos los productos y las especiaciones que tiene cada producto que el usuario seleccione
Precondición	<ul style="list-style-type: none"> <li>● El producto elegido debe existir en la base de datos del sistema.</li> </ul>
Postcondiciones	<ul style="list-style-type: none"> <li>● El usuario puede ver todo el listado de productos disponibles en el sistema y ver la información detalla de cada producto escogido por el usuario</li> </ul>
Flujo normal	<ul style="list-style-type: none"> <li>● El usuario administrador accede al sistema de gestión de la tienda virtual.</li> </ul>

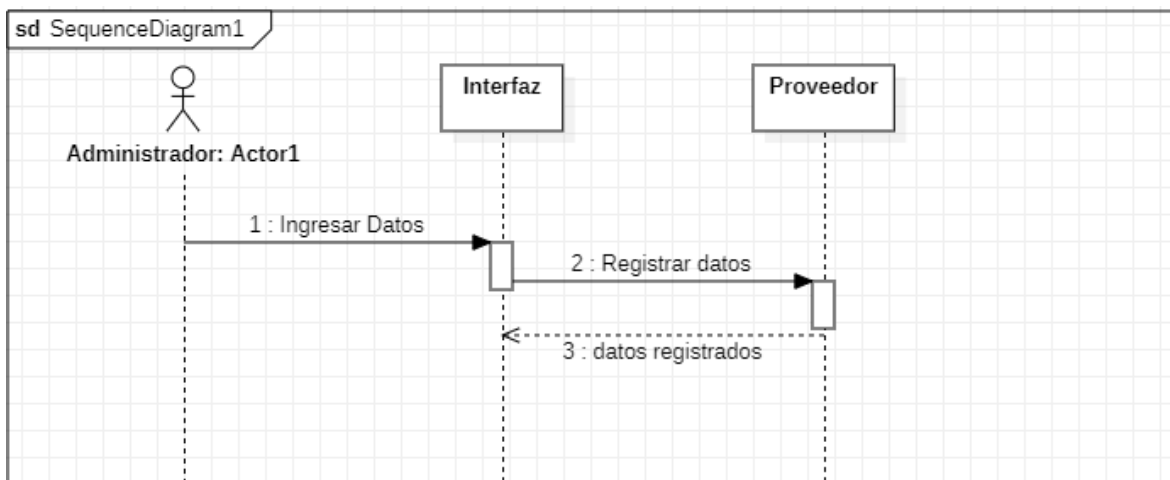
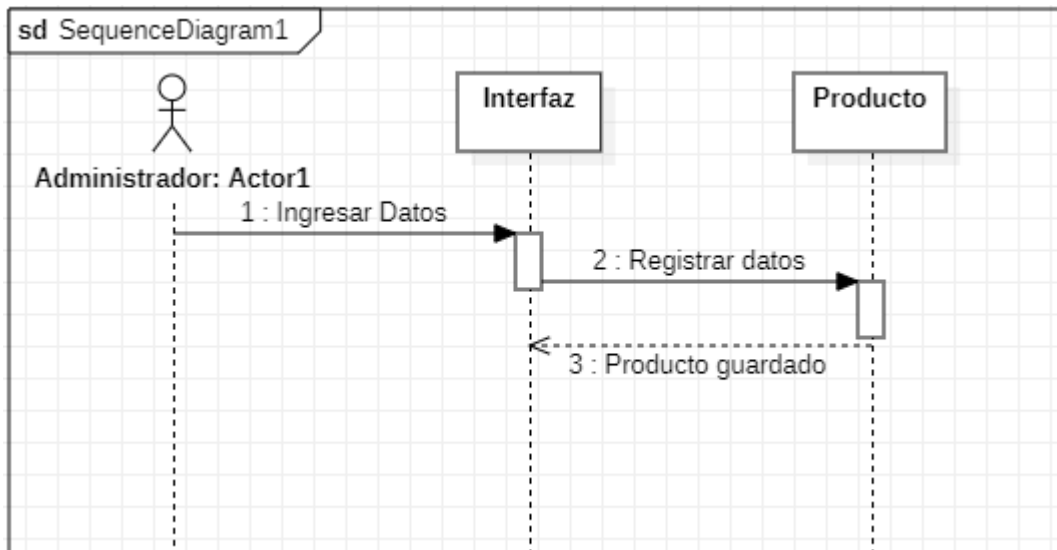
	<ul style="list-style-type: none"> <li>● El administrador selecciona la opción de "Gestión de productos" en el panel de control.</li> <li>● El sistema muestra una lista de productos existentes en la tienda.</li> <li>● El administrador tiene la opción de buscar un producto específico o agregar uno nuevo.</li> <li>● El administrador puede agregar, eliminar y modificar un producto</li> <li>● El administrador solicita al proveedor ingresando los detalles del producto, como nombre, descripción, categoría, precio, etc.</li> <li>● Se agrega producto satisfactoriamente y se le notifica al administrador</li> </ul>
Flujo alternativo	<ul style="list-style-type: none"> <li>● En caso no se puede seleccionar el producto que el usuario desea, se ofrecen ofertas similares</li> </ul>

Documentación Casos de Uso	
Caso de uso	Gestión de proveedores
Actores	Administrador
Tipo propósito	Asociación, incluye
Descripción	<ul style="list-style-type: none"> <li>● Se permite ver la lista de los proveedores que suministran los productos a la tienda virtual y su información de contacto</li> </ul>
Precondición	
Postcondiciones	<ul style="list-style-type: none"> <li>● Se visualiza todos los proveedores y los productos que suministrar</li> </ul>
Flujo normal	<ul style="list-style-type: none"> <li>● El administrador ingresa al sistema</li> <li>● se realiza una búsqueda de proveedores</li> <li>● El administrador puede agregar, eliminar, o editar a los proveedores.</li> <li>● Se seleccionan los proveedores para ver su información de forma detallada</li> <li>● El administrador elige a los proveedores según las necesidades de la tienda</li> </ul>

	Se hace la negociación
Flujo alternativo	<ul style="list-style-type: none"> <li>• si se reciben productos o servicios de baja calidad, se debe trabajar con el proveedor para resolver el problema y establecer medidas para prevenir futuros problemas.</li> <li>• si un proveedor no cumple con los términos de entrega acordados se deberán tomar medidas para garantizar los productos</li> </ul>

### 2.4.5.2 Diagramas de secuencia





### 2.4.5.3 Análisis de pruebas

Tabla de estado

Usuario

Elemento	Estado
Interfaz	Activo, no activo
Cliente	Registrado, no registrado



## Producto

Elemento	Estado
Interfaz	Activo, no activo
Producto	Existe, no existe

## Proveedor

Elemento	Estado
Interfaz	Activo, no activo
Proveedor	Registrado, no registrado

## Tabla de valores

### Datos

Usuario: cedula, nombre, apellido, dirección, celular, correo, contraseña

"119256721", "Camilo","Sánchez","Calle25","camilo@gmail.com","3008436578","Camilo1_"	Registrado
" 12","carlos","perez","calle30", " <u>carlos@gmail.com</u> ", "carlos1-"	Cedula incorrecta
"123454", "Angel", "Ruiz"," ", "3056785426","Angell."	Falto dirección
"11926758", "David", "Paez", " <u>davis@gmail.com</u> ", " ", "David1-"	Falto celular

### Datos

Producto: Nombre, precio, características, cantidad

"Blusa", "20000", "Blusa Talla s", "10",	Guardado
" ", "30000", "Short Talla 16", "20",	Falto nombre
"Jean", " ", "talla 8 rojo", "40",	Falto precio
"Blusa", "20000", "la", "10",	Característica incorrecta

## Datos

Proveedor: Nit, Nombre, Apellido, dirección, celular

"12345", "Alberto", "Araujo", "Calle 30", "3007685698"	Registrado
"678905 ", "Carlos", "Velez", " ", "3008765432"	Falto dirección
"73546", "al ", "Vega", "Calle 67", "3167895432"	Nombre incorrecto
" ", "Andres", "rojas", "carrera 23", "3124567654"	Falto nombre

### 2.4.5.4 Diseño de los casos de pruebas

Número del caso de prueba	Componente	Descripción de lo que se probará	Pre-requisitos
<b>Proveedor</b>	Registrar registrar	Cuando el usuario ingrese a esta interfaz, podrá registrar proveedor	Ser administrador

PROVEEDOR					
Paso	Descripción de paso a seguir	Datos de entrada	Salida esperada	¿Ok?	Observación
1	Ingresar a la página web		Mostrar página web	Ok	

2	Iniciar sesión	Correo: <u>admin@gmail.com</u>  Contraseña: Hola123*	Sesión iniciada	ok	Ser administrador
3	Hacer click en proveedor		Desplegar Opciones de proveedor	Ok	
4	Hacer click en registrar proveedor	Nit: "12345"  Nombre: "Luis"  Apellido: "Martinez"  Dirección: "Calle 28#23-56"  Celular "3008768970"	Formulario de registro de proveedor	ok	
5	Hacer click en guardar		Proveedor guardado	ok	

#### 2.4.5.5 Ejecución de prueba

Registro de proveedor

```

0 referencias
public static void SeleniumProvedor(IWebDriver driver)
{
    IWebElement button = driver.FindElement(By.Id("singInButton"));
    button.Click();
    Thread.Sleep(TimeSpan.FromSeconds(3));

    IWebElement element = driver.FindElement(By.Id("correo"));
    element.SendKeys("admin@gmail.com");

    IWebElement element2 = driver.FindElement(By.Id("password"));
    element2.SendKeys("Hola123*");

    IWebElement buttonLogin = driver.FindElement(By.Id("loginBtn"));
    buttonLogin.Click();
    Thread.Sleep(TimeSpan.FromSeconds(3));

    IWebElement productosIcon = driver.FindElement(By.Id("proveedor"));
    productosIcon.Click();
    Thread.Sleep(TimeSpan.FromSeconds(1));

    IWebElement registroIcon = driver.FindElement(By.Id("registroProvedor"));
    registroIcon.Click();
    Thread.Sleep(TimeSpan.FromSeconds(3));

    IWebElement nit = driver.FindElement(By.Id("nit"));
    nit.SendKeys("24545-5");

    IWebElement nombre = driver.FindElement(By.Id("nombre"));
    nombre.SendKeys("andres");

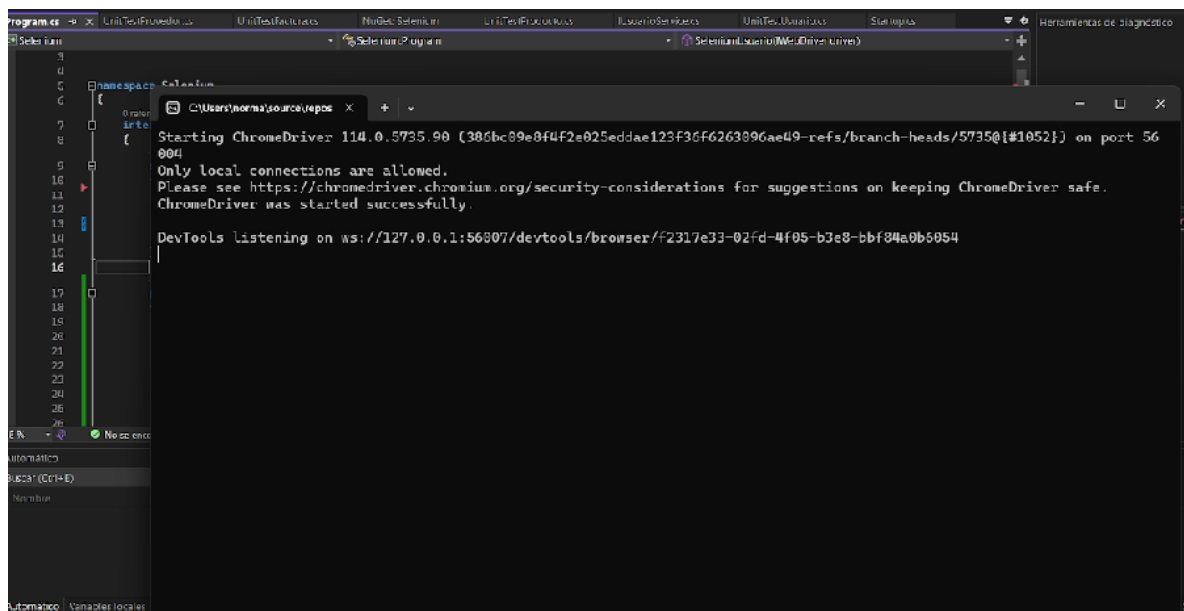
    IWebElement apellido = driver.FindElement(By.Id("apellido"));
    apellido.SendKeys("delgado");

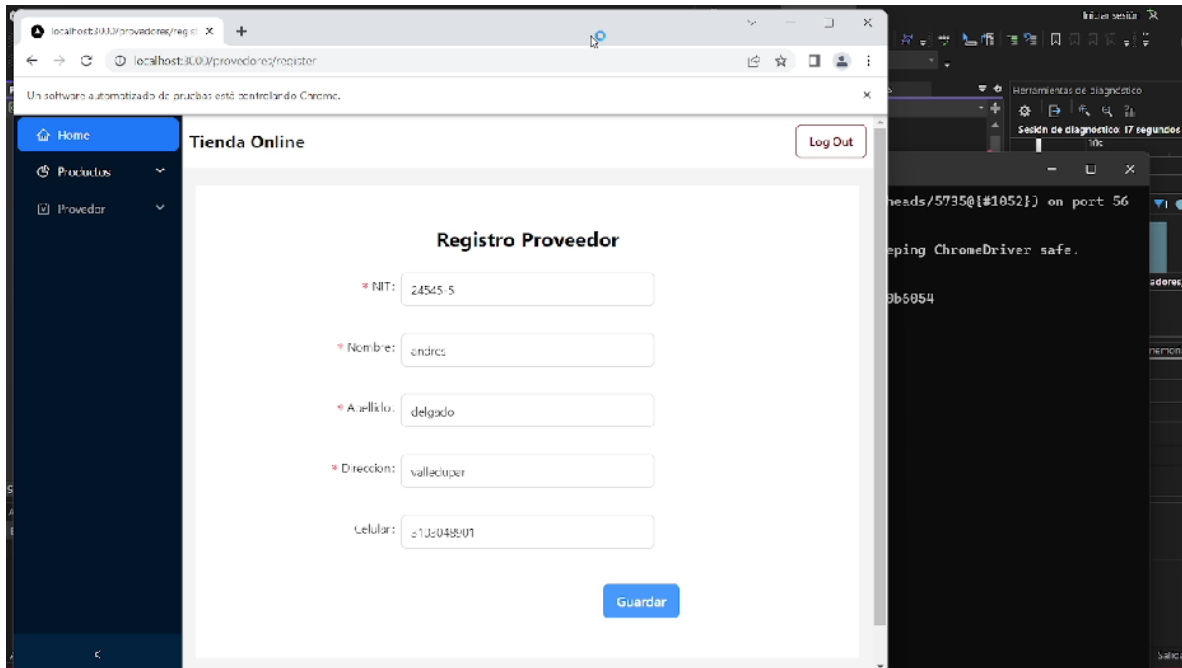
    IWebElement direccion = driver.FindElement(By.Id("direccion"));
    direccion.SendKeys("valledupar");

    IWebElement celular = driver.FindElement(By.Id("celular"));
    celular.SendKeys("3103048901");

    IWebElement guardar = driver.FindElement(By.Id("guardar"));
    guardar.Click();
    Thread.Sleep(TimeSpan.FromSeconds(5));
}

```





#### 2.4.5.6 Evaluación de las pruebas

Prueba				
Componente / Producto	Caso de prueba	Resultado	Seguimiento	Conclusión
Componente proveedor	Registro proveedor	Exitoso	Se procederá a realizar las pruebas de aceptación	

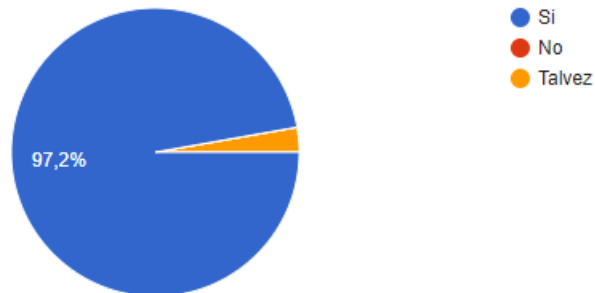
## 2.5 PRUEBAS DEL SISTEMA

Se utilizó la herramienta de encuesta de Google donde 36 usuarios respondieron dicha encuesta respecto a la experiencia que han tenido con el aplicativo web

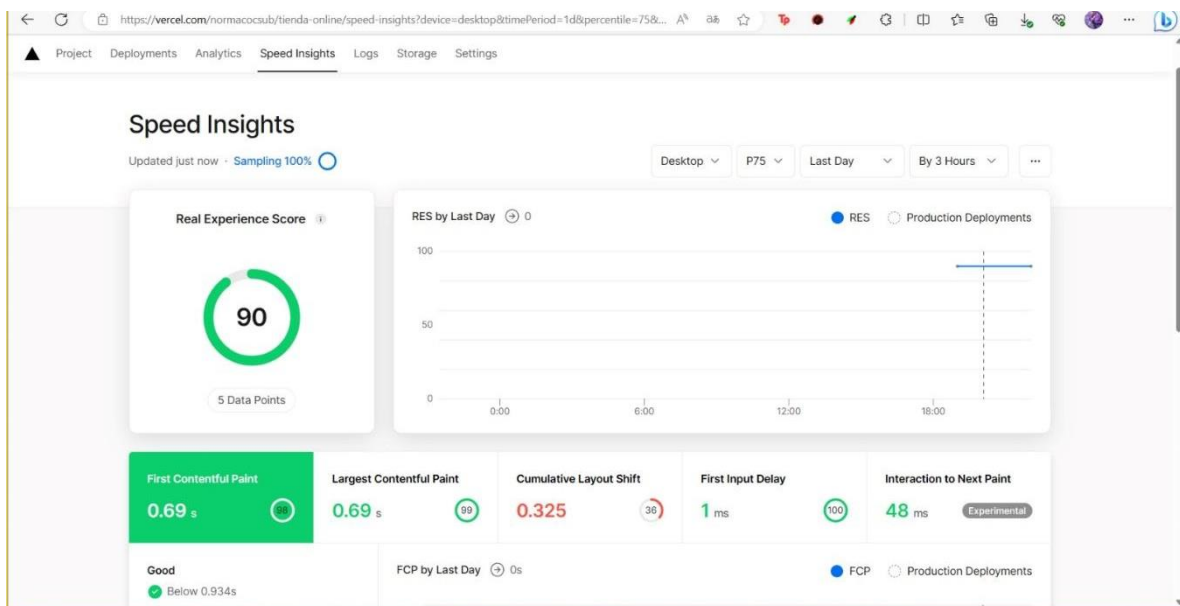
Teniendo en cuenta tu experiencia con nuestra tienda, ¿nos recomendarías?

 Copiar

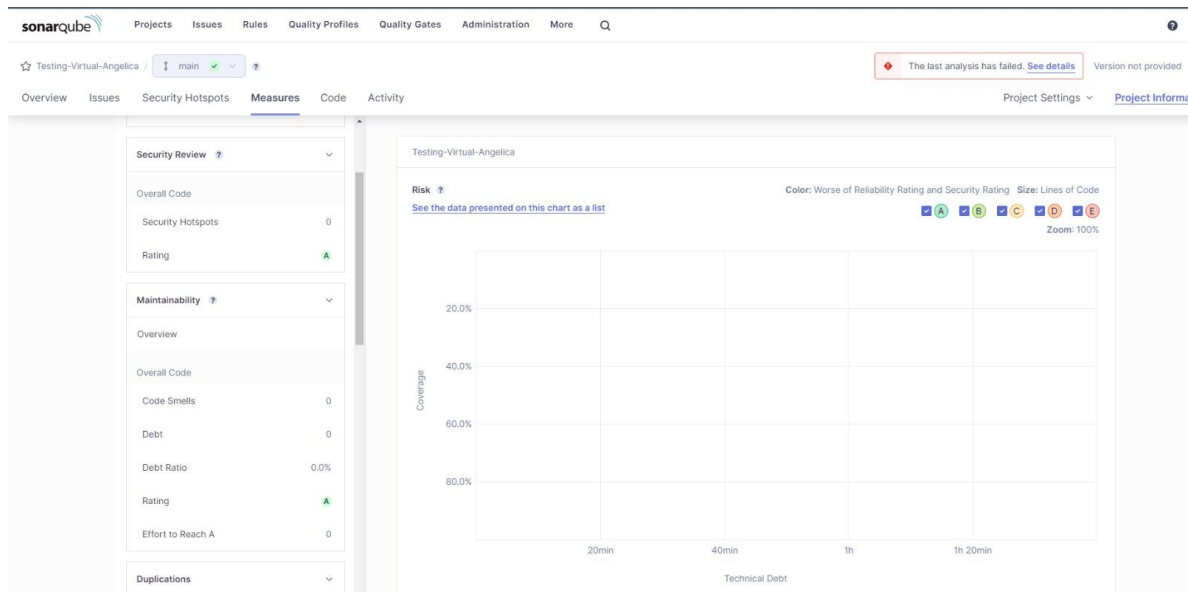
36 respuestas



**VERCEL** sitio donde está desplegado el aplicativo proporciona estadísticas en tiempo real del rendimiento del aplicativo, se puede observar que el software tiene un rendimiento del 90%, lo que significa que es esta funcionando de manera exitosa



Con la herramienta de sonarqube se hizo una prueba donde la mantenibilidad y la seguridad se da de manera exitosa ya que las dos cumple con un nivel alto de seguridad y mantenibilidad.



## 2.6 PRUEBAS DE ACEPTACION

### 2.6.1 Diseño de casos de pruebas

Caso de prueba modulo usuario

Descripción:

Al ingresar al módulo de usuario, se debe ingresar toda la información solicitada en el formato , incluyendo correo y contraseña.

Prerrequisitos

Pasos:

1. Ingresar al aplicativo
2. Dar clic en la opción registrar
3. ingresar datos en los campos solicitados en el formato

<p>4. Dar clic en el botón registrar, si algún campo del formato fue diligenciado de manera errónea, se le mostrar al usuario en que campo esta el error</p> <p>5. Hacer click en el botón guardar</p>
<p>Resultado esperado</p> <p>Registrar el usuario</p>
<p>Resultado obtenido</p> <p>Exitoso</p>

Caso de prueba módulo producto
<p>Descripción:</p> <p>Al ingresar al módulo de producto, se debe ingresar toda la información solicitada en el formato.</p>
Prerrequisitos: Ser usuario administrador
<p>Pasos:</p> <ol style="list-style-type: none"> <li>1. Ingresar al aplicativo web, iniciar sesión</li> <li>2. ingresar al módulo de producto</li> <li>3. hacer click en registrar</li> <li>4. ingresar datos en los campos solicitados en el formulario, si algún campo del formato fue diligenciado de manera errónea, se mostrará el campo erróneo</li> <li>5. hacer clic en el botón guardar</li> </ol>
<p>Resultado esperado</p> <p>Registrar producto</p>
<p>Resultado obtenido</p> <p>Exitoso</p>



Caso de prueba módulo proveedor
<p>Descripción:</p> <p>Al ingresar al módulo de proveedor, se debe ingresar toda la información solicitada en el formato.</p>
Prerrequisitos: Ser usuario administrador
<p>Pasos:</p> <ol style="list-style-type: none"> <li>0. Ingresar al aplicativo web, iniciar sesión</li> <li>0. ingresar al módulo de proveedor</li> <li>0. hacer click en registrar</li> <li>0. ingresar datos en los campos solicitados en el formulario, si algún campo del formato fue diligenciado de manera errónea, se mostrará el campo erróneo</li> <li>0. hacer clic en el botón guardar</li> </ol>
<p>Resultado esperado</p> <p>Registrar proveedor</p>
<p>Resultado obtenido</p> <p>Exitoso</p>

## 2.6.2 Evaluación de las pruebas

<b>Evaluación de las pruebas de aceptación</b>	
<b>Participantes:</b> Usuarios, desarrolladores	<b>Fecha:</b> 10/07/2023
<b>Tipo de pruebas:</b> Aceptación	Área (a realizar la prueba): interfaces
	Líder: Andres Delgado, Angelica Pacheco
<b>Porcentaje de satisfacción obtenido:</b>	95%

<b>Análisis de resultados:</b> los resultados de las pruebas fueron satisfactorios, se logró alcanzar la mayoría de los requerimientos establecidos. El cliente quedó satisfecho.		
Número de pruebas	Aprobadas	No aprobadas
3	3	0

## 2.7 CONCLUSIONES

Al trabajar en un proyecto de software, cada etapa del ciclo La vida hay que hacerla correctamente para que acabe siendo un proyecto Éxito, calidad.

Teniendo en cuenta que los integrantes del equipo para desarrollar un software son seres humanos, cabe resaltar que cometen errores, aclarado esto se puede decir que ningún software es perfecto y la mayoría tienen fallos.

Hay algunos procesos que pueden ayudan a lograr a obtener un software de calidad, como lo son las pruebas, por medio de las cuales se prueba que el software cumple con los requerimientos del cliente. Entre las pruebas se encuentra la prueba de caja negra que se centra en la funcionalidad del software, también existen las pruebas de caja blanca esta prueba ayuda a revisar las buenas prácticas del desarrollador, por medio del análisis del código fuente.

## 2.8 REFERENCIAS

Romero. M (2020) Pruebas de software generalidades. Universidad Popular del Cesar. Programa de Ingeniería de sistemas. Tomado de:

<https://aulaweb.unicesar.edu.co/course/view.php?id=4222>

Romero. M (2020) Introducción a los casos de prueba. Universidad Popular del Cesar. Programa de Ingeniería de sistemas. Tomado de:

<https://aulaweb.unicesar.edu.co/course/view.php?id=4222>

Plantilla Plan de Pruebas de Integración (2020) Tomado de:

<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/461>

Pruebas de software (2020) Tomado de:

[https://www.ecured.cu/Pruebas\\_de\\_software](https://www.ecured.cu/Pruebas_de_software)

### **III.METRICAS DEL SOFTWARE**

#### **3.1 Introducción**

En el campo de la ingeniería de software una métrica es cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software, un sistema de información, generalmente para realizar comparativas o para la planificación de proyectos de desarrollo.

En este contexto, siendo las métricas una medida estática se puede decir que ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. Es por esto, que las métricas son un agente clave al momento de desarrollar un software, ya que estas están presentes en todo ciclo de vida del desarrollo, desde la iniciación, cuando se estiman los costos, al seguimiento y control de la fiabilidad de los productos finales, y a la forma en que los productos cambian a través del tiempo debido a la aplicación de mejoras.

##### **3.1.1Objetivos**

El objetivo principal es fabricar un producto de alta calidad. Para llegar a este objetivo, se emplean métodos efectivos junto con herramientas como la medición ya que es una herramienta eficaz que permite destacar las tendencias y hacer mejores estimaciones en las

pruebas realizadas al producto software, es por esto, que se puede decir que la fase de métricas, es la base para detectar las desviaciones del rendimiento aceptable en los procesos y producto de software, asimismo, como las oportunidades de mejora e identificación de errores, los cuales se deben priorizar para dar seguimiento a la solución y mejorar la calidad del producto.

- Identificar las medidas mediante los métodos de medición y definir dichas medidas.
- Definir cuáles son las métricas referentes para el producto.
- Analizar los resultados de las métricas.
- Medir los errores y defectos encontrados para asegurar la efectividad de las actividades de control y de calidad.
- Evaluar los procesos para comprobar la calidad del producto software.

### 3.1.2 Alcance

Es importante resaltar que para el desarrollo de un producto software es muy importante, tener en cuenta la fase de las métricas ya que, cualquier medida que se obtenga está destinada a conocer o calcular el tamaño u otra característica de un software, esto permitirá visualizar una visión de la eficacia del proceso que se ha desarrollado y el marco de trabajo que se ha llevado a cabo en la realización de este, produciendo un impacto para la planificación o desarrollo de cualquier proyecto.

### 3.1.3 Tipos y herramientas de métricas

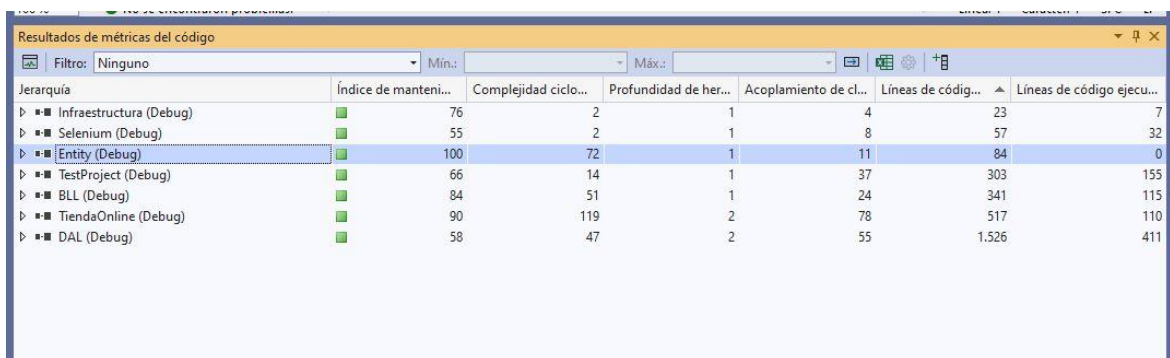
Métrica	Descripción
<b>MODELO GQM</b>	GQM (Goal-Question-Metric) es un paradigma para desarrollar y mantener un significativo programa de métricas, este proporciona una manera útil para definir mediciones tanto del proceso como de los resultados de un proyecto.
<b>Técnica de puntos de función (PF)</b>	Los puntos de función son un número positivo con escala arbitraria que refleja el tamaño de la funcionalidad del sistema (a mayor tamaño mayor número de puntos de función).
<b>COCOMO</b>	permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de coste y de escala
<b>SNARQUBE</b>	Esta herramienta permite realizar un análisis completo del código, para verificar la calidad del proyecto.

## 3.2 Métricas de producto

### 3.2.1 Métricas de tamaño

#### 3.2.1.1 Métricas de tamaño de longitud

- Líneas de código dadas por el lenguaje de programación



The screenshot shows a window titled "Resultados de métricas del código". It contains a table with columns for hierarchy, maintenance index, cyclomatic complexity, hereditary depth, coupling, lines of code, and lines of executed code. The "Entity (Debug)" project is highlighted in blue.

Jerarquía	Índice de manteni...	Complejidad ciclo...	Profundidad de her...	Acoplamiento de cl...	Líneas de código...	Líneas de código ejecu...
▸ ■■ Infraestructura (Debug)	76	2	1	4	23	7
▸ ■■ Selenium (Debug)	55	2	1	8	57	32
▸ ■■ Entity (Debug)	100	72	1	11	84	0
▸ ■■ TestProject (Debug)	66	14	1	37	303	155
▸ ■■ BLL (Debug)	84	51	1	24	341	115
▸ ■■ TiendaOnline (Debug)	90	119	2	78	517	110
▸ ■■ DAL (Debug)	58	47	2	55	1.526	411

- Se utilizó la herramienta VSCodeCounter para el tamaño de líneas de código en el lenguaje c#.

language	files	code	comment	blank	total
JSON	35	55,542	0	7	55,549
C#	88	2,828	225	659	3,712

## Análisis

Según las estimaciones presentadas, las líneas de código en el lenguaje de programación son mas bajas que las de la herramienta, esto se debe a que en el lenguaje de programación solo tiene en cuenta los procedimientos, a diferencia de la herramienta va a tener los procedimientos, archivos de compilación y archivos de configuración y demás para la funcionalidad del aplicativo.

### 3.2.1.2. Métricas de tamaño por funcionalidad

N	Entradas
1	Login
2	Registrar usuario
3	Registrar producto
4	Actualizar producto
5	Registrar proveedor
6	Actualizar proveedor
7	Registrar venta

N	Salidas
1	Factura

N	Consultas
1	Consultar producto
2	Consultar proveedor

N	Archivo Lógico
1	Entidad usuario
2	Entidad producto
3	Entidad proveedor
4	Entidad factura

Parámetros de medición	Cuenta	Simple	Medio	Complejo	Total
Número de entradas de usuario	7	7x3=21	0	0	21
Número de salida de usuario	1	1x4=4	0	0	4
Número de peticiones de usuario	2	2x3=6	0	0	6
Número de archivo	4	4x7=28	0	0	28
Cuenta total (PFS)					<b>59</b>

#### PUNTOS DE CONFIABILIDAD DE PROCESAMIENTO

Factor de ajuste		Peso
1	Comunicación de datos	4
2	Procesamiento distribuido de los datos	3
3	Rendimiento	4
4	Configuraciones fuertemente utilizadas	2
5	Tasas de transacción	3
6	Entrada de datos en línea	4
7	Diseño para la eficiencia del usuario final	4
8	Actualización en línea	3
9	Procesamiento complejo	1
10	Reusabilidad	3
11	Facilidad de instalación	4
12	Facilidad de operación	3

13	Puestos múltiples	2
14	Facilidad de cambio	3
Grado total de influencia (TDI)		43

#### **Puntos de función sin ajuste**

$$\text{PFS} = \sum ((\text{número de ítems de la clase } i) * \text{peso})$$

$$\text{PFS} = 59$$

#### **Puntos de complejidad de procesamiento**

$$\text{FCP} = 0.65 + (0.01 \times \text{Puntos de Complejidad de Procesamiento})$$

$$\text{FCP} = 0.65 + (0.01 \times 43)$$

$$\text{FCP} = 1.08$$

#### **Punto de función**

$$\text{PF} = \text{PFS} * \text{FCP}$$

$$\text{PF} = 59 * 1.08$$

$$\text{PF} = 63.72$$

#### **Tamaño líneas de código**

**C# (58)**

$$\text{KLOC} = (\text{PF} * \text{Líneas de código por cada PF}) / 1000$$

$$\text{KLOC} = (63.72 * 58) / 1000$$

$$\text{KLOC} = 3.695$$

#### **3.2.1.3. Análisis comparativo de tamaño del software**

<b>Fuente</b>	<b>LOC</b> Lenguaje de programación C#	<b>LOC</b> Herramienta VSCodeCounter	<b>LOC</b> Punto funcion
<b>Tamaño</b>	<b>1,526</b>	<b>3,712</b>	<b>3.695</b>

En el cuadro comparativo se puede observar que las estimaciones del LOC con la herramienta y el LOC de punto de función, los resultados son similares puesto que en ambas métricas se tomó el tamaño en LOC, éste resulta ser de las líneas de código funcionales del aplicativo, por lo tanto, existe una cercanía entre los resultados de ambas métricas.



### **3.3. Métricas de producto – Atributos externos**

Métrica: Factor de Impacto de Riesgo (Risk Impact Factor)

Autor: Dan Shoemaker y Anne Kohnke

Referencia: "Principles of Information Security", 6ta edición (2015)

**Objetivo 1:** Garantizar la seguridad del sistema

Pregunta 1: ¿Cómo calcular la seguridad del sistema?

Métricas

#A: Número de amenazas

#AS: Número de amenazas solucionadas

%Integridad= ((#A/#AS) x100)

**Objetivo 2:** Eficacia y usabilidad del sistema

Pregunta 2: ¿Cómo calcular la tasa de conversión actual del sistema?

Métricas

NCR: Numero de compras realizadas

NTV: Número total de visitas

Tasa de conversión = ((NCR/NTV) X 100)

**Objetivo 3:** Tiempo de respuesta promedio

Pregunta 3: ¿Cuál es el tiempo de respuesta actual del sistema?

Métricas

STR: suma de tiempos de respuesta

NTS: Numero total de solicitudes

%Tiempo de respuesta = STR / NTS

**Objetivo 4:** Aumentar el MTBF para mejorar la disponibilidad del sistema y reducir las interrupciones del servicio.

Pregunta 4: ¿Cuál es el valor actual del MTBF?

Métricas

TTO: Tiempo Total de Operación

NF: Número de Fallas

MTBF = TTO/NF

**Objetivo 5:** Garantizar la Mantenibilidad del software

Pregunta 5 ¿Cuál es el número de fallos en el software?

Métricas

NF=Número de fallas

**NE**=Numero de examinadas

**NFallos**=(NF/NE)

**Objetivo 6** Identificar los diferentes ambientes que sean compatibles con el sistema

**Pregunta 6:** ¿Cómo calcular la compatibilidad del software con diferentes ambientes?

**Métricas**

**NAS:** Numero de ambientes compatibles por el software

**NAE:** Numero de ambientes existentes

**%Compatibilidad**=(NAS/NAE) X100)

**Objetivo 7:** Evaluar el funcionamiento del software ante fallos

**Pregunta 7:** ¿Cómo evaluar el funcionamiento del software?

**Métricas**

**NFP:** Numero de fallos presentados

**NFS:** Numero de fallos solucionados

**%Funcionamiento**= (NFP / NFS) X100)

**Objetivo 8:** Garantizar que el software opere en diferentes ambientes

**Pregunta 8** ¿Qué recursos son necesarios para hacer una migración de un entorno otro?

**Métricas**

**RSE:** Recursos para crear el sistema en el entorno

**RNM:** Recursos necesarios para mover la aplicación

**Instalación** = (RSE-RNM)

Medida

<b>Característica: Usabilidad</b>		
<b>Subcaracterística: Tarifa de éxito al realizar una tarea</b>		
<b>Nombre Métricas</b>	<b>Descripción de la métrica</b>	<b>Medición o formula de calculo</b>
<b>Tarifa de éxito</b>	Al realizar una tarea, se busca determinar el porcentaje de usuarios de la prueba capaces de lograr lo que se pidió.	$E = (TC + TM) * \frac{100}{TT}$ <ul style="list-style-type: none"> <li>• E = éxito de la tarea</li> <li>• TC = tarea completada totalmente</li> <li>• TM = tarea completada hasta la mitad</li> <li>• TT = total de tareas</li> </ul>
<b>Aplicación de la métrica:</b> $E = (15 + 3) * \frac{100}{22}$ $E = 81$		
<b>Interpretación del resultado:</b>  <b>Por medio de la aplicación de la métrica de usabilidad, obtuvimos una tasa de éxito del 81 por ciento, lo cual no sería una medida totalmente satisfactoria pero aceptable.</b>		

### III. ESTIMACIÓN DEL SOFTWARE

#### 4.1. Estimación del tamaño

##### 4.1.1. Tamaño por punto de caso de uso

Actores: Administrador, cliente=2

Clases: 6

Casos de usos=16

Categoría de casos de uso	Descripción	Peso (Factor)
Complejo	Transacciones = 16	15

	Clases = 6	
--	------------	--

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF} * \text{PF}$$

- Calcular UUCP

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCW} = (\text{Cantidad casos de uso} * \text{factor})$$

$$\text{UUCW} = (16 * 15) = 240$$

Categoría de caso de uso	Descripción	Peso(factor)
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI)	3

$$\text{UAW} = (\text{Cantidad de un tipo de actor} * \text{Factor})$$

$$\text{UAW} = (2 * 3) = 6$$

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

$$\text{UUCP} = 6 + 240 = 246$$

- Calcular TCF

Factor técnico	Descripción	Peso	Impacto Percibido	Factor calculado
T1	Sistema distribuido	2	1	2
T2	Rendimiento o tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4

T4	Procesamiento interno complejo	1	3	3
T5	Código reutilizable	1	3	3
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	2	2
T11	Características especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	2	2
T13	Se requiere facilidades especiales de entrenamiento a usuario	1	2	2
Factor total técnico				<b>41.5</b>

$$\text{TCF} = 0.6 + (0.01 * \text{factor total técnico})$$

$$\text{TCF} = 0.6 + (0.01 * 41.5) = 1.015$$

#### - CALCULAR ECF

Factor Ambiental	Descripción	Peso	Impacto percibido	Factor calculado
E1	Familiaridad con el modelo del proyecto utilizado. Familiaridad con UML	1.5	5	7.5

E2	Personal tiempo parcial	-1	0	0
E3	Capacidad de analista líder	0.5	4	2
E4	Experiencia en la aplicación	0.5	4	2
E5	Experiencia en orientación a objetos	1	4	4
E6	Motivación	1	5	5
E7	Dificultad del lenguaje de la programación	-1	2	-2
E8	Estabilidad de los requerimientos	2	4	8
Factor Ambiental Total				26,5

$$ECF = 1.4 + (-0.03 * \text{Factor Ambiental Total})$$

$$ECF = 1.4 + (-0.03 * 26,5) = 0.605$$

$$UCP = UUCP * TCF * ECF * PF$$

$$UCP = 246 * 1.015 * 0.605 * 1$$

$$UCP = 151.062$$

#### 4.1.2. Tamaño por puntos de objetos

Ventanas	7
Informe	3
Componente	16
Total OP	26

## 4.2. Estimación del esfuerzo

Estimación del esfuerzo

Utilice las medidas referentes al tamaño para encontrar el esfuerzo y realice un análisis

Método	Tamaño	Esfuerzo
LOC	3,712	$E = a(KLOC)^b$ $E = 2.49(3,712)^{1.1}$ $E=21.026$
Puntos de función	63.72	$63.72 * 12 = 764.64$
Puntos de casos usos	151.062	$151.062 * 20 = 3.021$
Puntos de objeto	189	$189 * 10.17 = 1,922$

## 4.3.1. ESTIMACIÓN COCOMO II

### 4.3.1.1 Modelo de composición de aplicación

**Clasificación de las instancias de objeto dentro de niveles de complejidad**

Entrada=7 Medio

Informes=3 difícil

Archivo lógico =4

Componentes= 16

Tipo de objeto	Complejidad-Peso			
	Simple	Media	Difícil	Subtotal
Ventana	X1	7X2	X3	14
Informe	X2	3X5	X8	15
Componente de 3GL			16X10	160
Total OP				189

**Porcentaje de reutilización (30%)**

**Nuevos Puntos Objeto para desarrollar.**

$NOP = ((OP) * (100 - \%reuso) / 100)$

$NOP = ((276) * (100 - 30\%)) / 100$

$NOP = 132.3$

**Determinar la razón de productividad a partir de la siguiente tabla**

Experiencia y capacidad de	Muy baja	Baja	Nominal	Alta	Muy alta
----------------------------	----------	------	---------	------	----------

los desarrolladores					
Madurez y capacidad de ICASE	<b>Muy baja</b>	<b>Baja</b>	<b>Nominal</b>	<b>Alta</b>	<b>Muy alta</b>
PROD	4	7	13	25	50

### **Esfuerzo mes-persona**

PM=NPO/PROD

PM=132.3/13

PM=10.17

### **4.3.1.2 Modelo de diseño temprano-Anticipado**

#### **Factor de escala**

	Muy bajo	Bajo	Normal	Alto	Muy alto	Extra alto	Sub Total
Precedentes (PREC)	6.2	4.96	3.72	2.48	1.24	0	4.96
Flexibilidad (FLEX)	5.07	4.05	3.04	2.03	1.01	0	2.03
Arquitectura / resolución del riesgo (RESL)	7.07	5.65	4.24	2.83	1.41	0	5.65
Cohesión del equipo ()	5.48	4.38	3.29	2.19	1.10	0	1.10
Madurez del proceso (PMAT)	7.80	6.24	4.68	3.12	1.56	0	4.68
Total							18.42

FE=18.42

PM(Nominal)=Ax (Tamaño) ^B

A= Constante de calibración (2,45 o 2,94)

Tamaño= KLOC

B= 0.91 + 0.01 x FE



$$B=0.91+0.01*(18.42)$$

$$B=1.1$$

$$KLOC=3.695$$

$$PM(Nominal)=Ax \text{ (Tamaño) } ^B$$

$$PM(Nominal)= (2.94(3.695) ^ (1.1))$$

$$PM(Nominal)=12.38$$

### Factor de ajuste

	Extra bajo	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto	Subtotal
Flexibilidad y complejidad del producto (RCPX)	0.73	0.81	0.98	1	1.3	1.74	2.38	1.3
Requerimientos de reusabilidad (RUSE)	-	-	0.95	1	1.07	1.15	1.24	1
Dificultad de la plataforma (PDIF)	-	-	0.87	1	1.29	1.81	2.61	1
Capacidad del personal (PERS)	2.12	1.62	1.26	1	0.83	0.63	0.5	0.83
Experiencia del personal (PREX)	1.59	1.33	1.12	1	0.87	0.71	0.62	0.87
Facilidad para el desarrollo (FCIL)	1.43	1.30	1.10	1	0.87	0.73	0.62	1
Esfuerzo de calendario (SCED)	-	1.43	1.14	1	1	1	-	1
TOTAL								0.94

$$FA=0.94$$

### Calculo pm Ajustado

$$PMAjustado = PMNominal * FA$$

$$PM_{ajustado} = 12.38 * 0.94$$

$$PM_{ajustado} = 11.6372$$

### Calculo tiempo de desarrollo

$$TDES = [c \times (PM)^d] \times SCED\% / 100$$

Siendo

PM = esfuerzo expresado en persona-mes

$$c = 3$$

$$d = 0.33 + 0.2 \times [B - 1.01]$$

$$D = 0.33 + 0.2 \times (1.1 - 1.01)$$

$$D = 0.348$$

**SCED% es el porcentaje de compresión/expansión del tiempo.**

	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
SCED	75%	85%	100%	130%	160%	

$$SCED\% = 85\%$$

$$TDES = [c \times (PM)^d] \times SCED\% / 100 \text{ Siendo}$$

$$TDES = [3 \times (11.6372)^{(0.348)}] \times 85 / 100$$

$$TDES = 5.99$$

$$TDES = 5$$

El personal a tiempo completo necesario para el desarrollo (PDTC) será:

$$PM = 10.17$$

$$PDTC = PM / TDES$$

$$PDTC = 10.17 / 5$$

$$PDTC = 2.0$$

$$PDTC = 2$$

$$\text{Costo} = 3.000.000 \times 5 \times 2$$

$$\text{Costo} = 30.000.000$$

$PM=11.63$

$PDTC = 11.63/5$

$PDTC = 2,3$

$PDTC = 2$

$\text{Costo} = 3.000.000 * 5 * 2$

Costo 30.000.000

Análisis

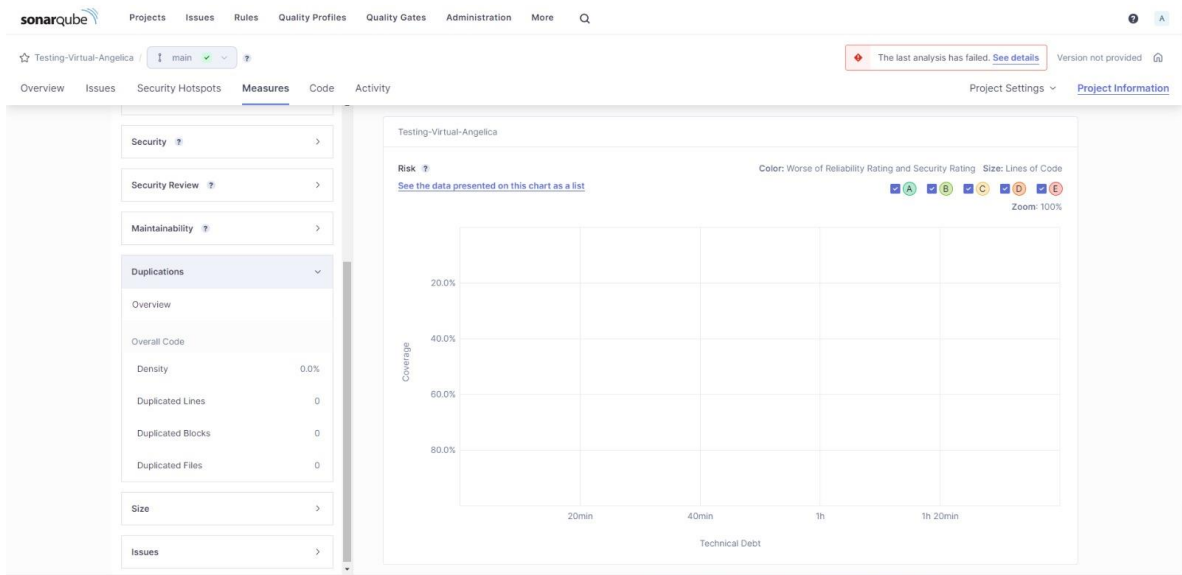
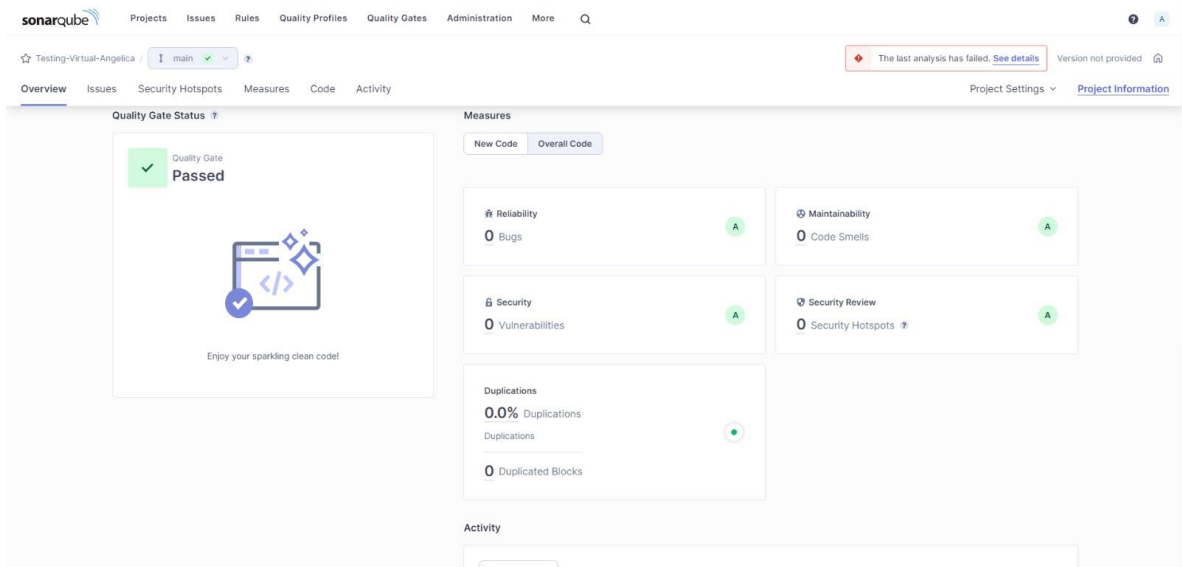
En este análisis se puede que el Pm ajustado y PM esfuerzo mes-persona, son bastante similares, por lo tanto el costo de proyecto es el mismo

### **III. CALIDAD DE SOFTWARE**

#### **5.1. Aplicación herramienta SONARQUBE**

La calidad se puede definir como los atributos y características de un producto. Para medir la calidad, es posible comparar las diferencias entre la producción de dos productos idénticos. El objetivo principal es minimizar las variaciones entre las unidades producidas. Para lograr esto, se deben corregir diferentes problemas según la naturaleza del producto. Brindar calidad es fundamental para cualquier negocio que fabrique productos destinados al uso de otras personas.

En cuanto a la evaluación de la calidad, se utilizó la herramienta SONARQUBE, que se enfoca en criterios clave como la finalidad, la seguridad y la mantenibilidad. Estos criterios son de gran importancia en la actualidad. El análisis realizado arrojó los siguientes resultados.



	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
Testing-Virtual-Angelica	—	—	—	—	—	—	—
BLL	—	0	0	0	0	—	0.0%
DAL	—	0	0	0	0	—	0.0%
Entity	—	0	0	0	0	—	0.0%
Infraestructura	—	0	0	0	0	—	0.0%
Selenium	—	0	0	0	0	—	0.0%
TestProject	—	0	0	0	0	—	0.0%
TiendaOnline	—	0	0	0	0	—	0.0%

7 of 7 shown

Como se puede observar en las imágenes, la herramienta muestra el índice de fiabilidad en nuestro software, como resultado A, esto quiere decir que el software es fiable. la seguridad tiene un índice alto lo cual es bueno. También arroja un índice de mantenibilidad alto con color verde lo cual se traduce a que es un software mantenible, una revisión de seguridad alta y por último cuenta con cero niveles de duplicidad y cero deudas.

## **VI. CONCLUSIONES**

En conclusión, el enfoque en las estimaciones y la calidad del software es esencial para el éxito de cualquier proyecto. Las estimaciones precisas proporcionan una base sólida para la planificación y gestión adecuada de recursos, tiempo y presupuesto. Al utilizar técnicas y herramientas adecuadas, se puede evaluar de manera realista la complejidad del proyecto y anticipar los posibles desafíos.

La calidad del software es fundamental para garantizar que cumpla con los requisitos, sea confiable y proporcione una experiencia satisfactoria al usuario, las métricas implementadas, las de pruebas y la aplicación de buenas prácticas de desarrollo son elementos clave para lograr un software de alta calidad. Al utilizar técnicas y herramientas adecuadas, se puede evaluar de manera real la complejidad del proyecto.

## **VII BIBLIOGRAFIA**

Romero. M (2020) Pruebas de software generalidades. Universidad Popular del Cesar. Programa de Ingeniería de sistemas. Tomado de:

<https://aulaweb.unicesar.edu.co/course/view.php?id=4222>

Romero. M (2020) Introducción a los casos de prueba. Universidad Popular del Cesar. Programa de Ingeniería de sistemas. Tomado de:

<https://aulaweb.unicesar.edu.co/course/view.php?id=4222>

Plantilla Plan de Pruebas de Integración (2020) Tomado de:

<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/461>

Pruebas de software (2020) Tomado de:

[https://www.ecured.cu/Pruebas\\_de\\_software](https://www.ecured.cu/Pruebas_de_software)