

PROYECTO DE ANALISIS Y MEJORAMIENTO DE SOFTWARE

Yaneth Virginia Aquino Huallp (2017059286), Jesus Alejandro Quenaya
Buiza (—), Daniel Angel Juarez Chumpe (—)

Tacna, Perú

Abstract

Con el presente documento se pretende presentar de forma concisa y clara las necesidades del cliente en el área de ventas en términos de software que se realizará. En esta documentación se plasmará los requerimientos que servirán de guía para desarrollar el software en sus distintas etapas, ayudándonos a validar e inspeccionar la construcción de este, aplicando la calidad de Software. Por ello se trabajará con Sonarqube para el análisis de código estático para analizar el código y encontrar errores de código, y vulnerabilidades de seguridad. El análisis C++ de SonarSource tiene una gran cobertura de estándares de calidad bien establecidos.

1. Antecedentes o Introducción

El sistema de "SysFerreteria" está desarrollado para poder automatizar los procesos que normalmente realiza de forma manual, esto para llevar un correcto control del inventario, compras a proveedores y ventas que se realizan a diario a los clientes; Además, implementar la emisión de Comprobantes de Pago para los clientes.

2. Sistema de Ferreteria

Propuesta de un sistema de facturación de compra y venta de artículos de ferretería.

3. Autores

- Yaneth Aquino Huallpa
- Jesus Alejandro Quenaya Buiza
- Daniel Angel Juarez Chumpe

4. Planteamiento del problema

En la actualidad, para las empresas resulta una obligación llevar un mejor control y resguardo de la información ya que de ésta depende, en su mayoría, el aumento o disminución de sus ganancias o, a su vez, permite que las gestiones realizadas por sus trabajadores sean lo más eficientes y óptimas para de ésta manera lograr los resultados esperados.

4.1. *Problema*

En la microempresa “sysferreteria” ningún proceso se encontraba automatizado, todas las actividades se realizaban manualmente y no se lograban concretar los procesos debido a la magnitud de los mismos; por tal razón se procederá a desarrollar el sistema que contará con módulos de usuarios, módulo de facturación, módulo de compras, gestión de inventario, módulo de consultas y la generación de reportes que cumplan con los requerimientos de los usuarios y términos.

4.2. *Justificación*

Con el presente proyecto se pretende desarrollar el sistema de facturación para que de ésta forma el propietario conozca exactamente las ganancias netas diarias o mensuales dependiendo de la necesidad que se tenga; además la falta de gestión del inventario impide al usuario conocer las cantidades exactas de los productos que mantiene, su costo real y si es necesario o no realizar pedidos en los tiempos determinados.

4.3. *Alcance*

El software a desarrollar se basará en el área de ventas de la empresa, usando también el stock de los productos por lo cual: El sistema permitirá registrar, actualizar, buscar y dar por “deteriorado” la información de los productos en stock. El sistema realizará ventas la cual nos permitirá agregar productos a la venta, restar con el stock, registrar o anular venta. El sistema

permitirá añadir pedido (datos del producto y del cliente), registrar o anular pedido y listar pedidos. El sistema permitirá generar reportes sobre: Ventas del día con el total ganado y Lista de Pedidos. El sistema permitirá realizar la autenticación de los usuarios.

5. Objetivos

5.1. *General*

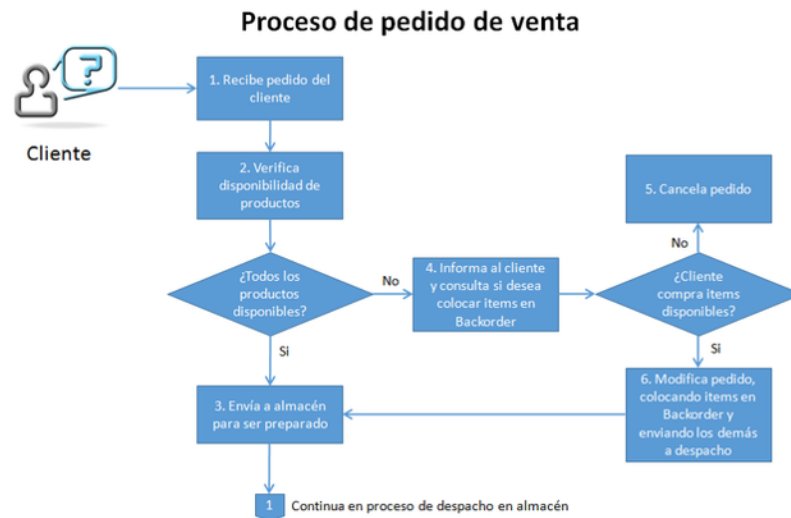
- Desarrollar un software enfocado en el área de compra y ventas de la microempresa.

5.2. *Específicos*

- El sistema permitirá registrar, actualizar, buscar y dar por “deteriorado” la información de los productos en stock.
- El sistema realizará ventas la cual nos permitirá agregar productos a la venta, restar con el stock, registrar o anular venta.
- El sistema permitirá añadir pedido (datos del producto y del cliente), registrar o anular pedido y listar pedidos.

6. Referentes teóricos

- Hemos realizado una investigación en empresas relacionadas con el rubro de la ferretería, en este caso elegimos el caso de la empresa HELEO situada en el parque industrial de Tacna, donde pudimos observar como trabaja su sistema de COMPRA Y VENTA de productos.



- Posee un anexo en la web de CATALOGO EN LINEA donde da una mayor promoción a sus productos antes de acudir a la tienda a comprarlos.



- Asimismo pudimos ver un detalle interesante que era un modulo de CONSULTA DE FACTURAS ELECTRONICAS VIA INTERNET para los clientes



7. Desarrollo de la propuesta

La calidad de código suele decirse que es un atributo interno de calidad, dado que no se hace visible al usuario. Pero llega un momento en el cual este atributo de calidad pasa de ser interno a externo, y esto se da cuando el hecho de tener modificar el código para hacer un cambio lleva mucho más tiempo del que debería. Con el fin de verificar la calidad interna de un sistema se suelen hacer análisis de código con SonarQube o herramientas similares. En este documento se muestra parte de de nuestro proyecto , donde básicamente cuenta cómo hacer una prueba de concepto rápidamente usando una imagen Docker de SonarQube, y ejecutando el análisis desde SonarQube Scanner. SonarQube, como tantas otras herramientas similares, permite realizar análisis estático de código fuente de manera automática, buscando patrones con errores, malas prácticas o incidentes. Además, realiza un cálculo de la deuda técnica. Dentro de las verificaciones que hacen herramientas como SonarQube, se encuentran las siguientes:

- Detección de código duplicado..
- Falta de pruebas unitarias, falta de comentarios.
- Código spaghetti, complejidad ciclomática, alto acoplamiento.
- Tamaño de archivos de código.
- Tamaño de métodos.

- No adecuación a estándares y convenciones de código.
- Vulnerabilidades conocidas de seguridad.

Una muestra en Nuestro proyecto:

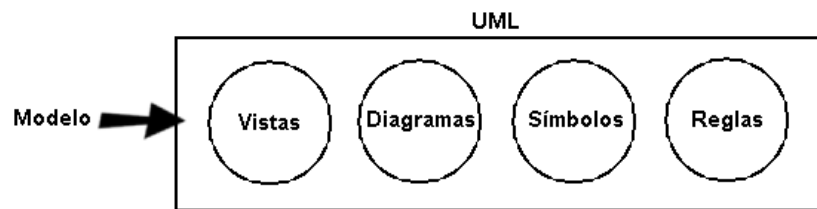
	Líneas de código	Loco	Vulnerabilidades	Olores de código	Hotspots de seguridad	Cobertura	Duplicaciones
ProyectoSysFerreteria							
Entidad	400	0.0	0.0	123	0.0	-	0.0%
Negocios	2,287	0.0	0.0	139	0.0	0.0%	32.3%
Presentacion	3,757	0.0	0.0	337	0.0	0.0%	13.0%
Propiedades	15	0.0	0.0	1	0.0	-	0.0%
Program.cs	19	0.0	0.0	3	0.0	0.0%	0.0%

7.1. *Tecnología de información*

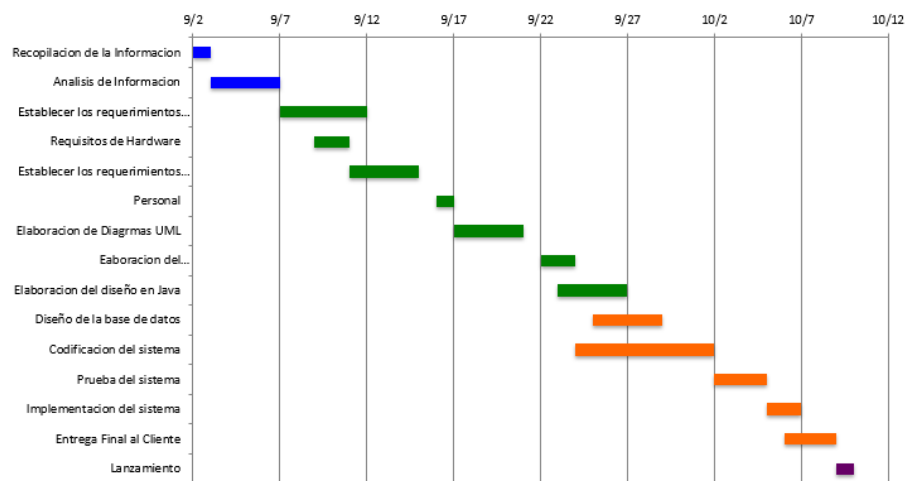
- SQL SERVER: Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado es Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas..
- C char : lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft. Es un lenguaje de programación creado para diseñar aplicaciones en la plataforma.NET.
- Visual Studio: es un entorno de desarrollo en diferentes sistemas operativos y compatibles con múltiples lenguajes de programación al igual que entornos de desarrollo web.

7.2. *Metodología, técnicas usadas*

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un método es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.



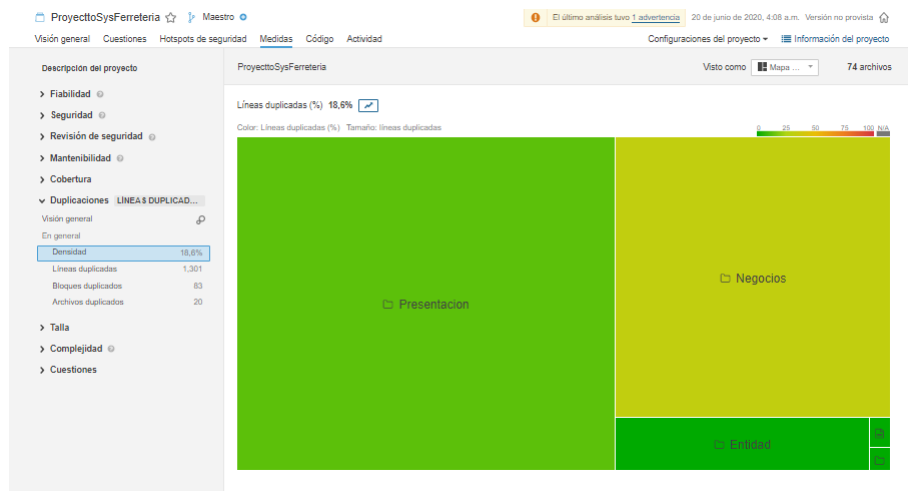
8. Cronograma



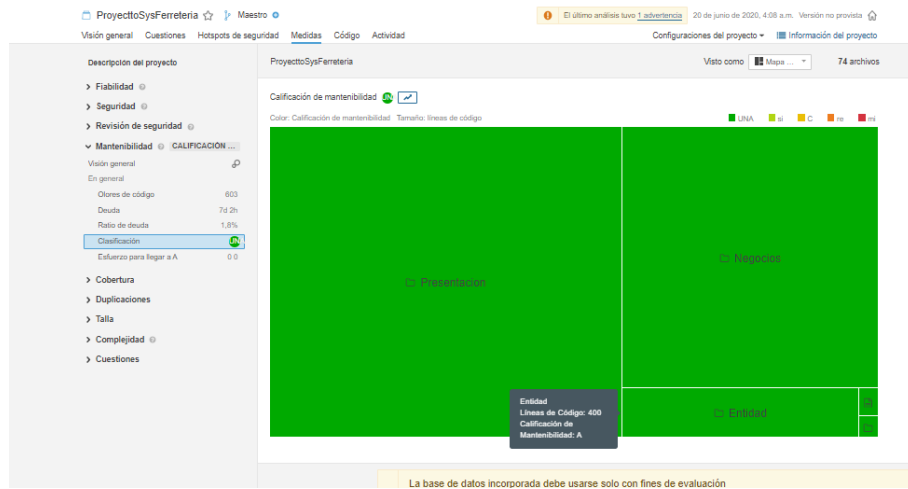
9. Resultado Sonarqube

Definición de métricas

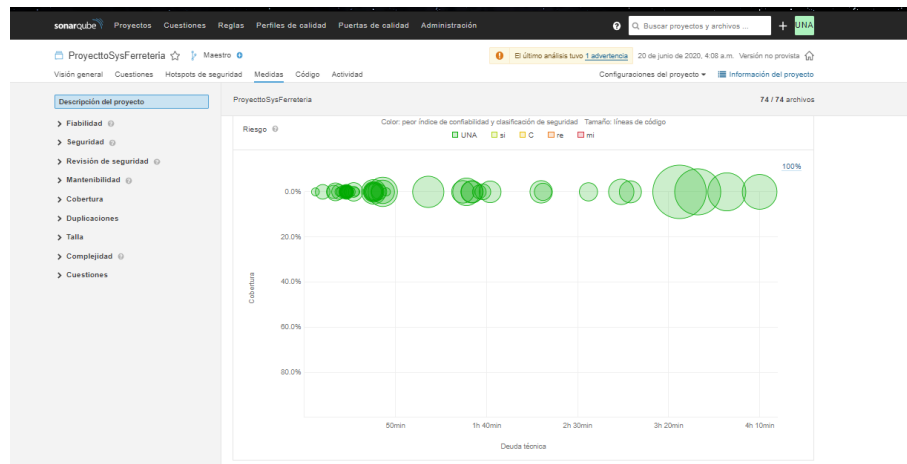
- Código duplicado



■ Mantenibilidad

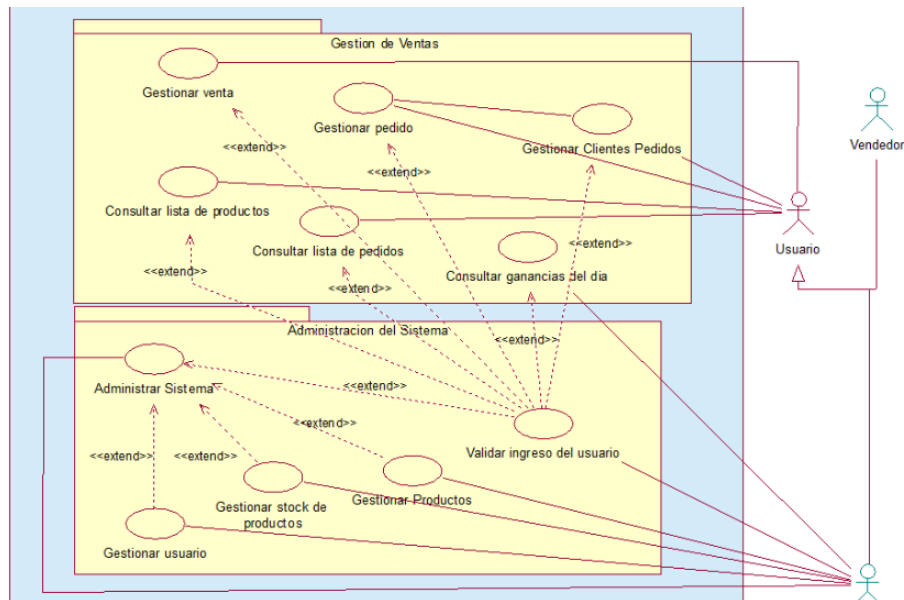


■ Medida



10. Desarrollo de Solución de Mejora

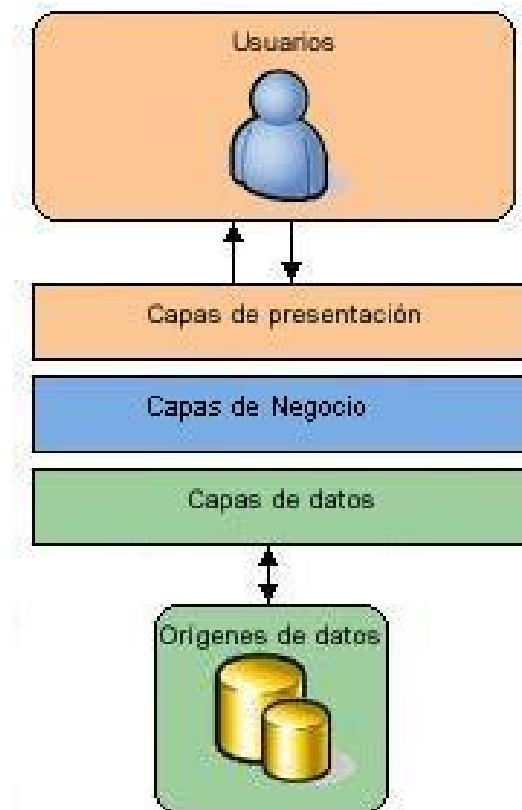
10.1. Casos de Uso de la aplicación

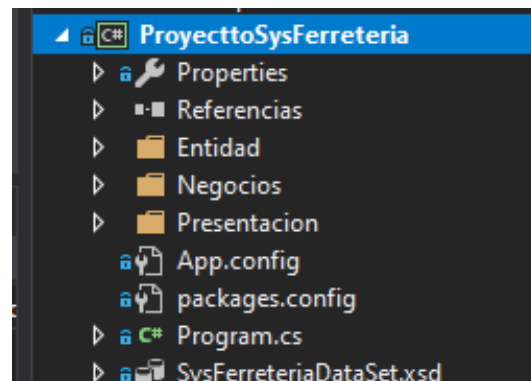
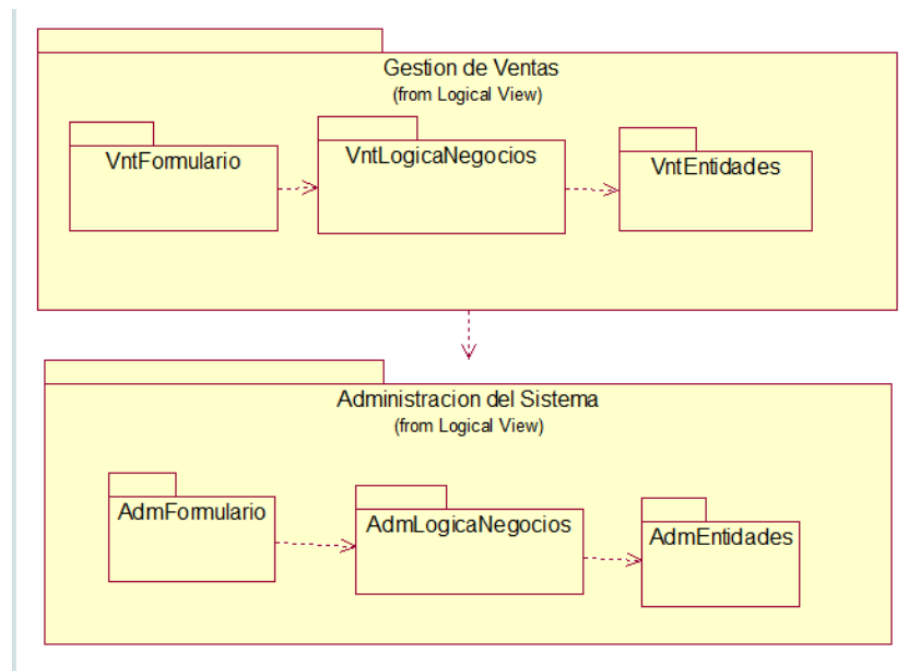


10.2. Diagrama de Arquitectura de la aplicación

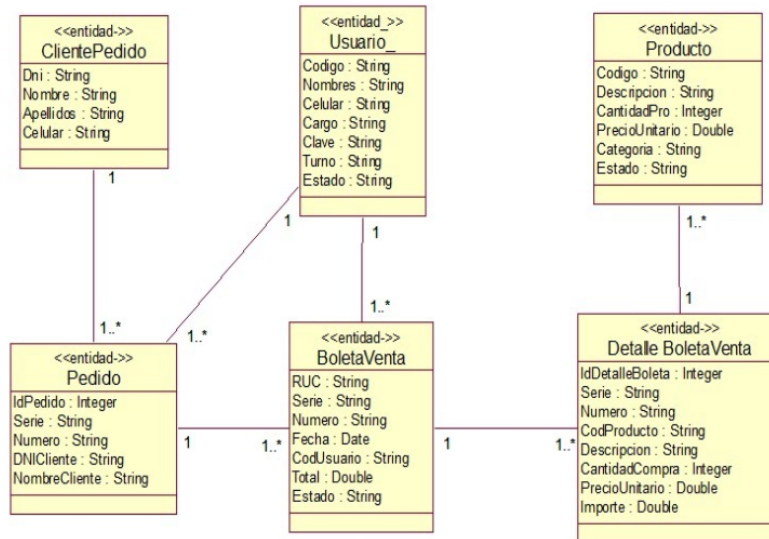
- Existen diferentes tipos de arquitectura o patrones a seguir para desarrollar un software, en este caso voy a explicar en que consiste la arquitectura de 3 Capas, que a mi parecer es la más general o la más básica para desarrollar.

- La Capa de Presentación : Donde se encuentran los formularios y la parte visual de la aplicación.
- La Capa de Negocios : Donde se encuentra toda la logica del negocio y clases que las componete es decir, Entidades y controladoras)
- La Capa de Acceso a Datos: Donde se encuentra las conexiones y las transacciones que se utilizan para comunicarse con la base de datos.





10.3. Diagrama de Clases de la aplicación



10.4. Metodos de pruebas implementados para cubrir la aplicación

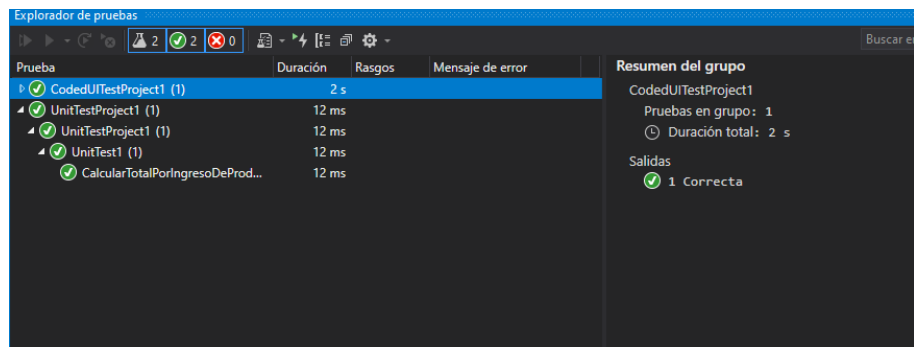
1) Pruebas Unitarias

```
public class UnitTest1
{
    [TestMethod]
    0 referencias | Virginiayjd7, Hace 1 día | 1 autor, 1 cambio
    public void CalcularTotalPorIngresoDeProducto()
    {
        EDetalleVenta detalleProducto = new EDetalleVenta();
        List<EDetalleVenta> listaDetalleProducto = new List<EDetalleVenta>();
        EVenta lineaDeVenta = new EVenta();

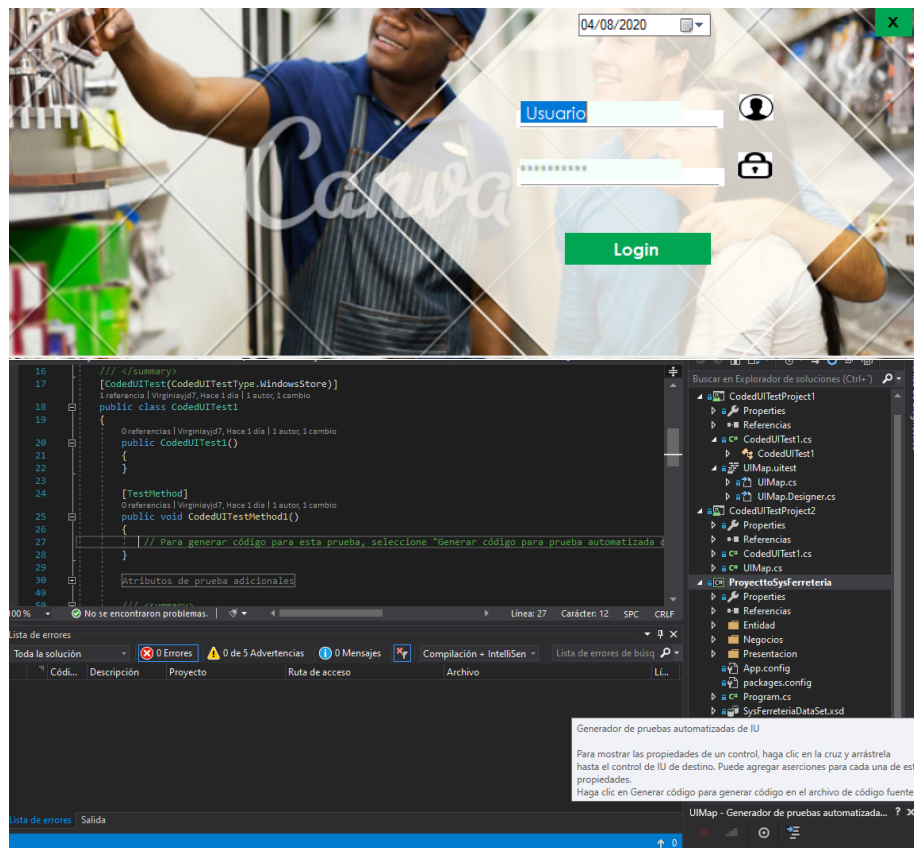
        detalleProducto.Precio = 20;
        detalleProducto.Cantidad = 5;
        double valorEsperado = 100;
        listaDetalleProducto.Add(detalleProducto);
        lineaDeVenta.ListaDetalleProducto = listaDetalleProducto;

        double total = lineaDeVenta.CalcularTotalPorIngresoDeProducto();

        Assert.AreEqual(valorEsperado, total);
    }
}
```



2) Pruebas de Interfaz



3) Pruebas basadas en Desarrollo Guiado por el Comportamiento