



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

**Trabajo Final de Unidad II: Desarrollo de Proyecto
de Análisis y Mejoramiento de Software**

Curso: Calidad y Pruebas de Software

Docente: Ing. Patrick Cuadros Quiroga

Integrantes:

Anahua Huayhua, Jenny Karen	(2018062150)
Coloma Colquehuanca, Kiara Estefani	(2018062218)
Cuadros Napa, Raul Marcelo	(2017057851)
Limache Durand, Rodrigo Jeral	(2017059278)
Vilca Condori, Erlang Fernando	(2019064024)

**Tacna – Perú
2021**

PROYECTO DE MEJORAMIENTO Y DE ANALISIS DE LA APLICACIÓN

“Sistema de Reservas, Control de Hospedaje del Hotel Santa María”

Resumen

El presente describe el proyecto desarrollo de un sistema de reservas, control de hospedaje del Hotel Santa Maria en la ciudad de Tacna, el cual permitirá tener un mejor control de los usuarios de la empresa y para que las acciones principales se realicen de una manera más efectiva y eficaz.

Se trabajará con Visual Studio 2019 para realizar las pruebas unitarias y de comportamiento usando los paquetes nuget y las extensiones propias de visual Studio 2019.

Abstract

The present document describes the development Project of a reservation system, hotel control of the Hotel Santa Maria in the city of Tacna, which will allow to have a better control of the users of the company and so that the main actions are carried out in a more effective and effective.

It will work with Visual Studio 2019 to perform the unit and behavioral tests using the nuget packages and the extensions of visual Studio 2019.

1. Antecedentes o introducción

A continuación, se procederá a explicar con más profundidad el proyecto de Sistema de Reserva de Hotel para la Hotel Santa María. que le permitirá tener un mejor control de los usuarios de la empresa y para que las acciones principales se realicen de una manera más efectiva y eficaz. También se explicará más acerca del propósito, el alcance del proyecto y algunas definiciones o abreviaturas que darán a conocer el significado de los términos manejados por la empresa.

2. Título

Proyecto de mejoramiento de la aplicación: “Sistema web para la gestión de reservas, control de hospedaje Hotel Santa Maria de Tacna”

3. Autores

- Anahua Huayhua, Jenny Karen
- Coloma Colquehuanca, Kiara Estefani
- Cuadros Napa, Raul Marcelo
- Limache Durand, Rodrigo Jeral
- Vilca Condori Erlang Fernando

4. Planteamiento del problema

4.1. Problema

El presente proyecto es para solucionar una propuesta que trata de automatizar todos los procesos que se realizan en una hotelera santa maría en la Ciudad de Tacna.

El Hotel Santa María está ubicado en la Ciudad de Tacna en la Calle Augusto B Leguía 437, Tacna y se encuentra funcionando hace 10 años. Actualmente maneja una gran cantidad de turistas cuenta con 10 personas que se encargan en la organización de este distribuido en la parte administrativa y limpieza. Lastimosamente el Hotel no cuenta con un sistema que permita tener resultados óptimo en su rentabilidad en época de temporada.

Por la ubicación que se encuentra el hotel en alguna temporada presenta grandes cantidades de huéspedes, pero así también pierden una gran cantidad de turistas ya que muchas veces no hay control de checking del check out al no a ver eso no tenemos la certeza de poder ver el estado de las habitaciones y suelen quedar mal por no tener la información a la mano, por lo que los turistas optan de ir a su competencia.

4.2. Justificación

Debido al gran avance que se encuentra la tecnología se les ha propuesto una solución para sus problemas que luchan al día a día, algunos de ellos son el registro de huéspedes y el control de las reservaciones, mediante una propuesta que les permita visualizar el impacto favorable que ocasionaría la automatización de sus procesos.

4.3. Alcance

El sistema de Reserva de Hotel de la empresa Hotel Santa María. permitirá la Administración de Gestión de Hotel y el registro de nuevos clientes.

5. Objetivos

5.1. General

Ayudar a mejorar la gestión de las actividades del Hotel en su administración.

5.2. Específicos

- El sistema permitirá el control de inventarios
- El sistema facilitara el manejo de clientes
- El sistema permitirá agilizar procesos y mejorar el servicio a la ciudadanía.

6. Referentes teóricos

Diagrama de Casos de Uso

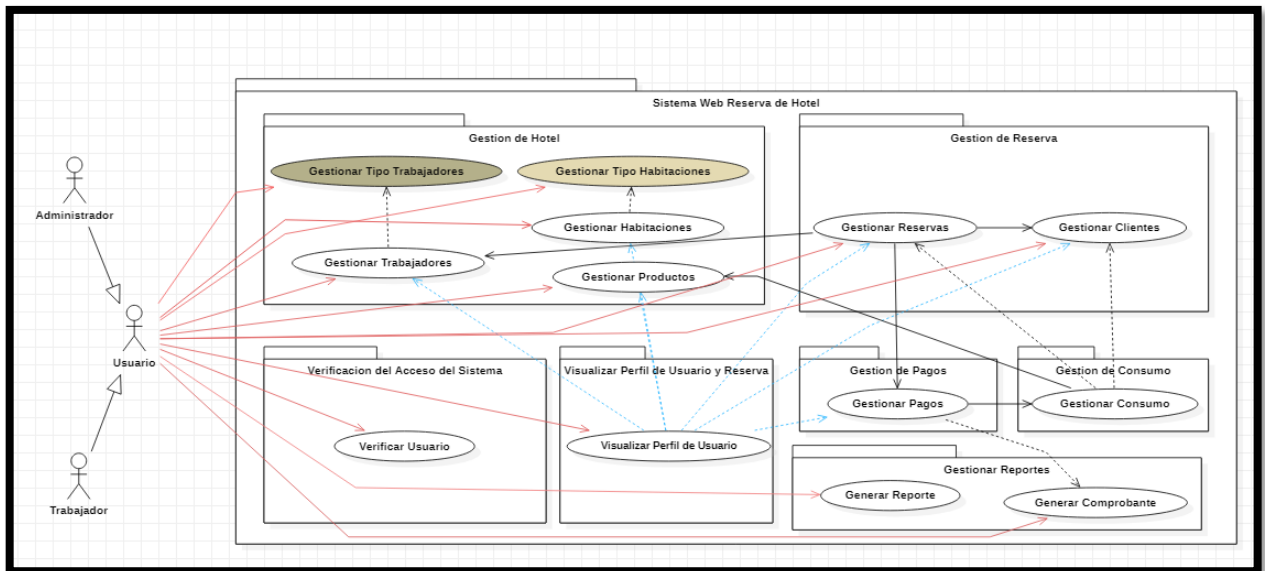


Diagrama de Clases

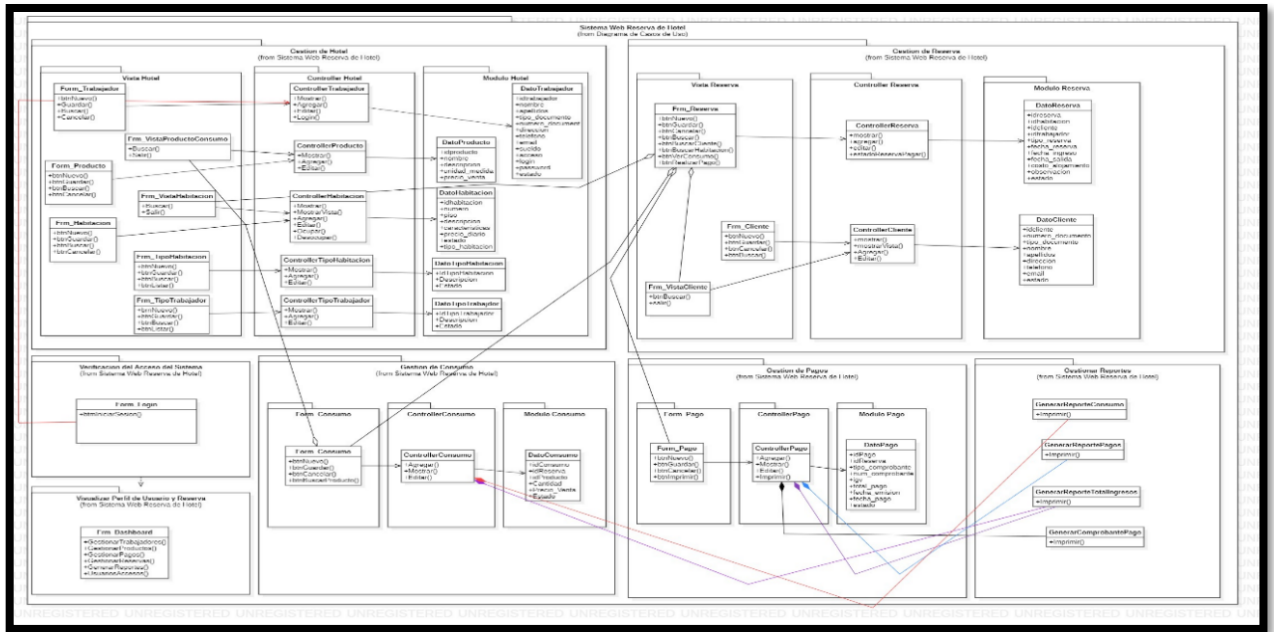
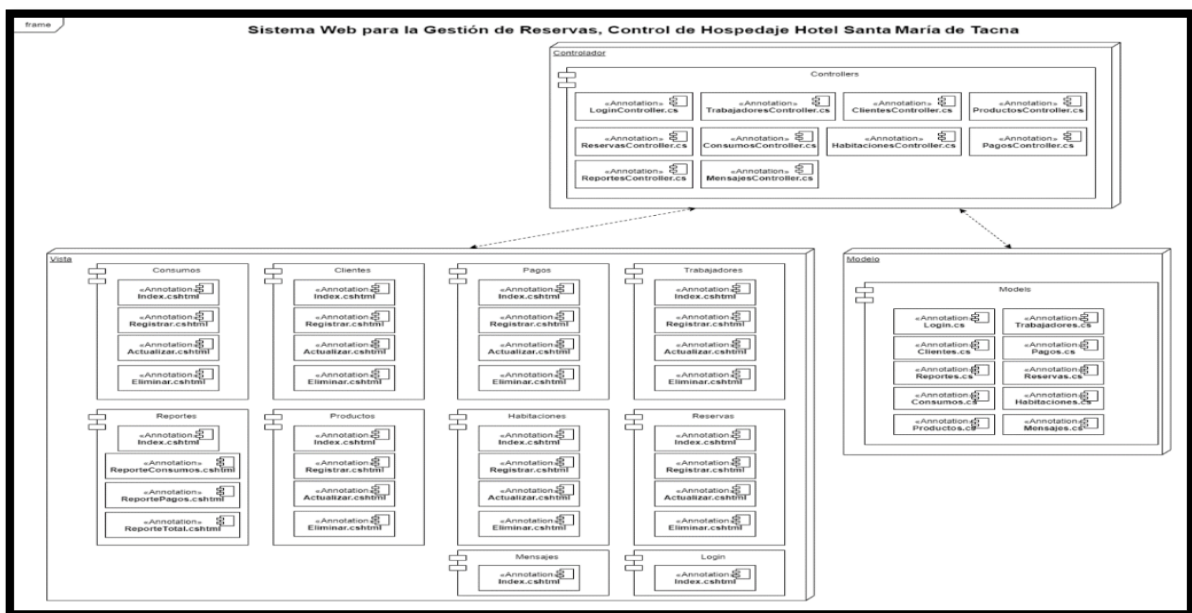
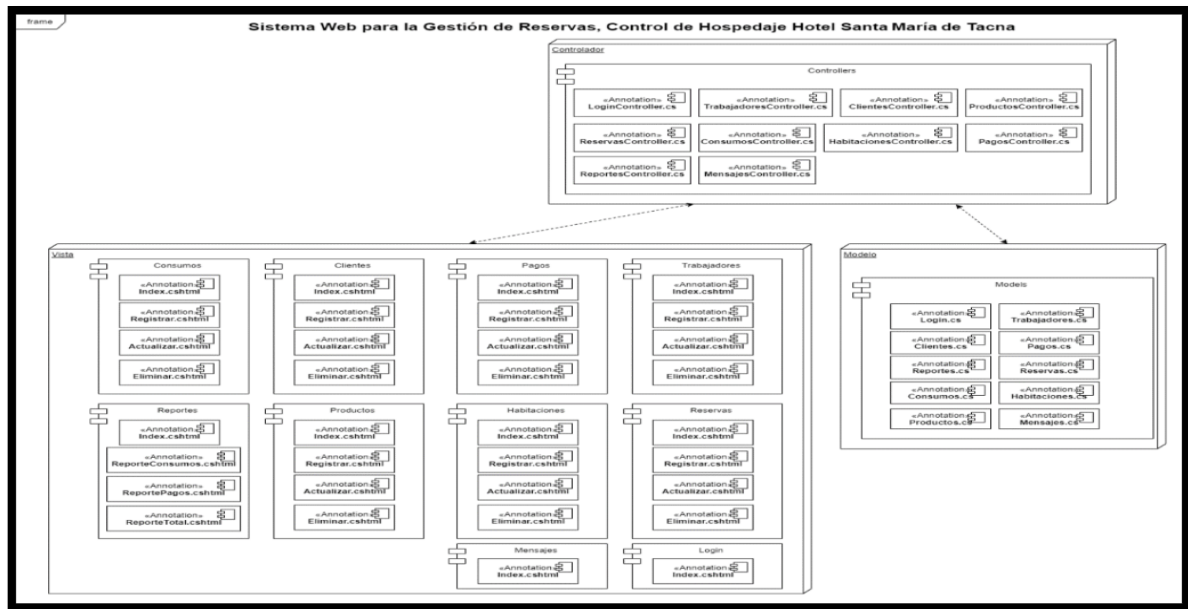


Diagrama de Componentes y Arquitectura.





7. Desarrollo de la propuesta

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 nuestro proyecto, donde se describe como hacer una prueba usando una imagen Docker de SonarQube, y ejecutando el análisis desde SonarQube Scanner.

7.1. Tecnología de información

- **Microsoft 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.

Microsoft SQL Server es la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos. Es un sistema de gestión de base de datos relacional desarrollado como un servidor que da servicio a otras aplicaciones de software que pueden funcionar ya sea en el mismo ordenador o en otro ordenador a través de una red (incluyendo Internet).

- **Visual Studio:** Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en

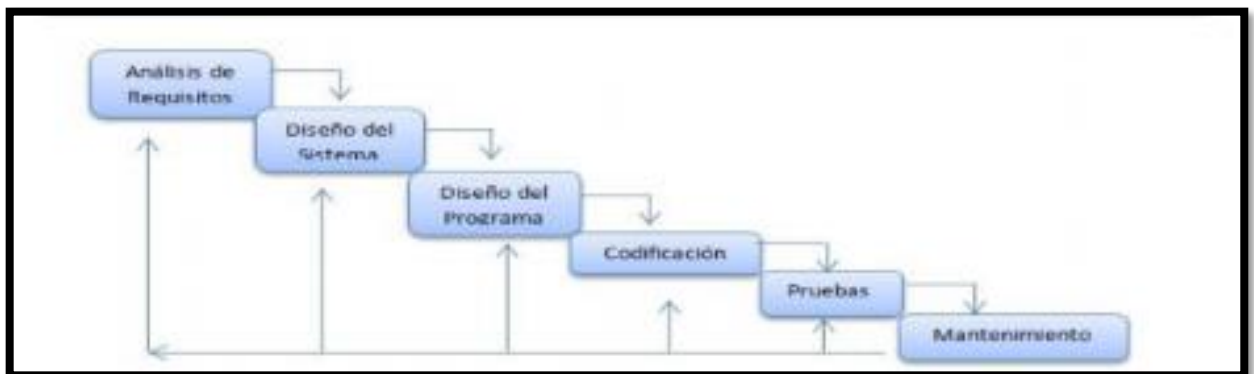
cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.

- C#: es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET.

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 que hacer, como hacerlo, cuando 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.

- Programación orientada a objetos (POO).
- Programación en 3 capas.
- Metodología ágil XP (Programación Extrema).
- Modelo cascada

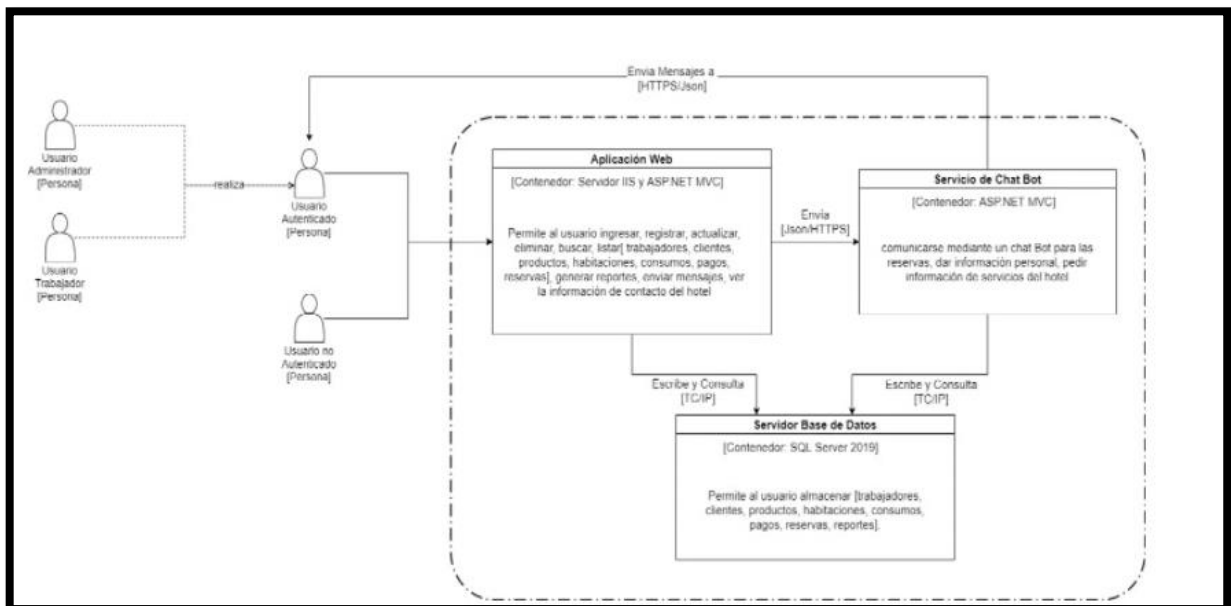


8. Cronograma

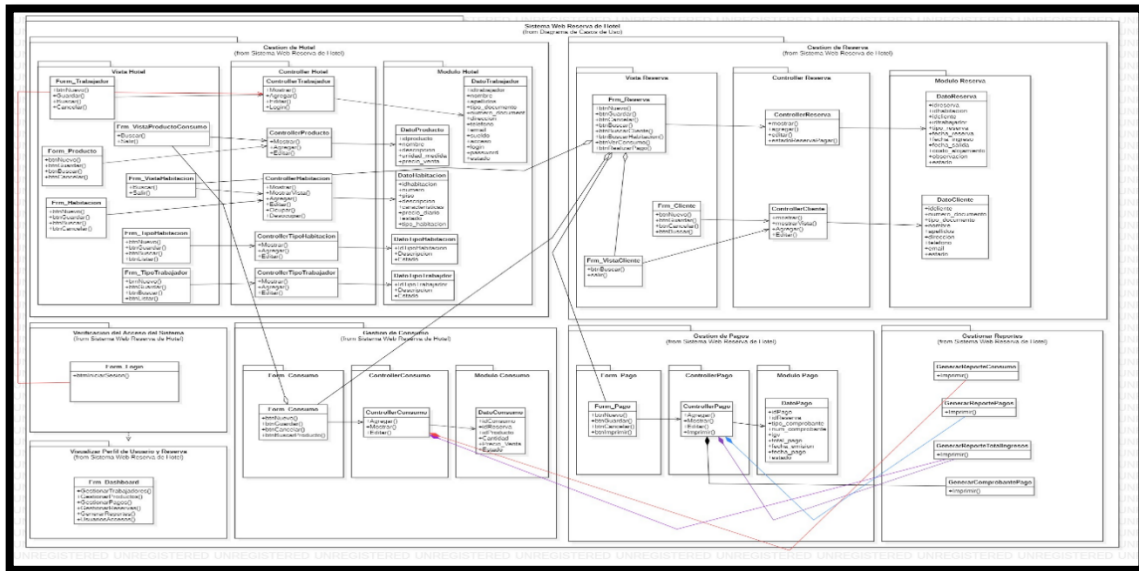
Item	Fases	Cronograma de Actividades											
		Agosto				Setiembre				Octubre			
1	I	X	X	X	X								
2	II		X	X	X	X							
3	III				X	X	X	X					
4	IV						X	X	X	X			
5	V								X	X	X	X	
6	VI										X	X	X

9. Desarrollo de Solución de Mejora

9.1 Diagrama de Arquitectura de la aplicación

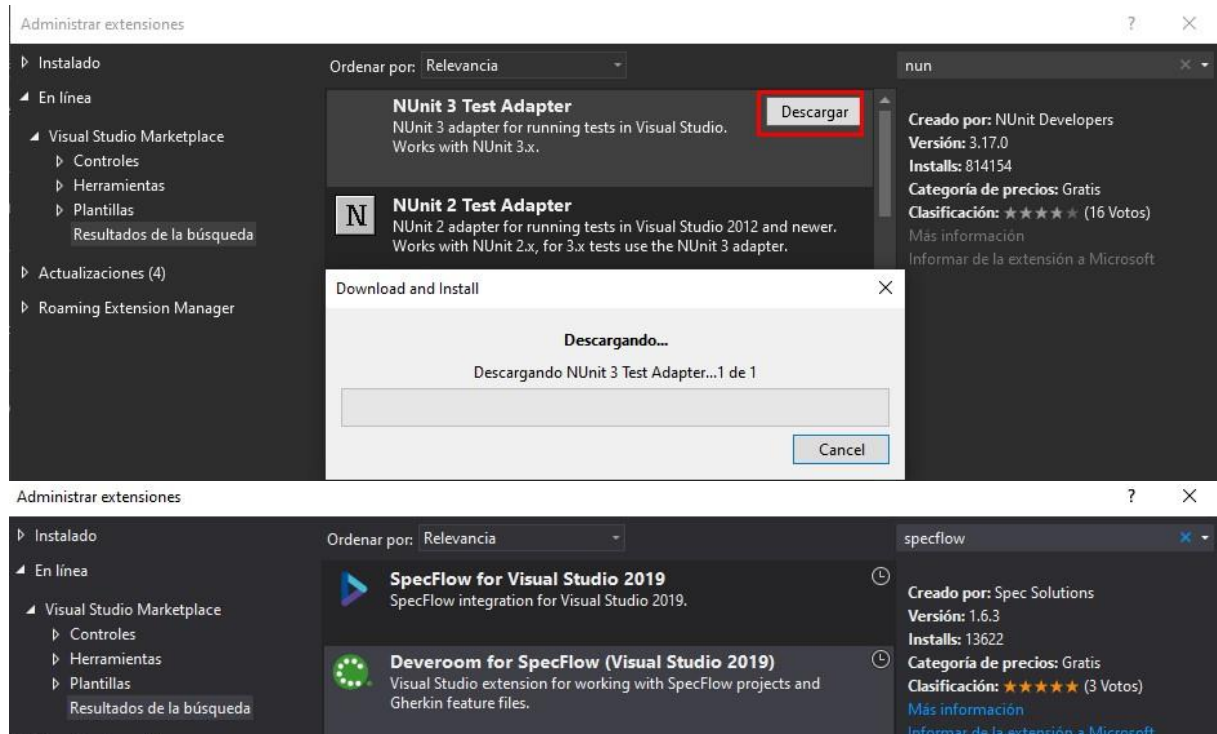


9.2 Diagrama de Clases de la aplicación

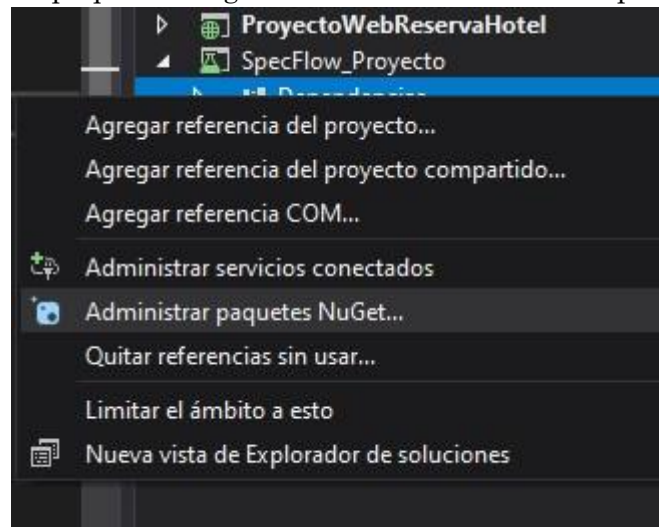


9.3 Metodos de pruebas implementados para coberturar la aplicación

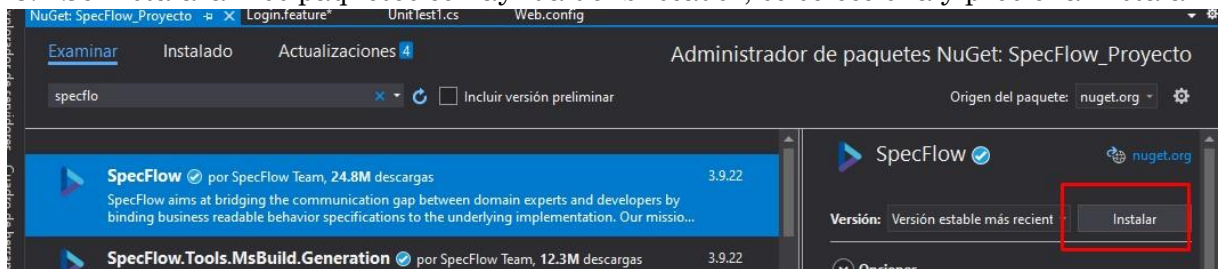
Paso 1: Instalando extensiones para usar Specflow con Selenium en c# (extensiones Nunit 3 Test Adapter, Specflow for Visual Studio 2019) en proyecto de prueba agregado a la solución.



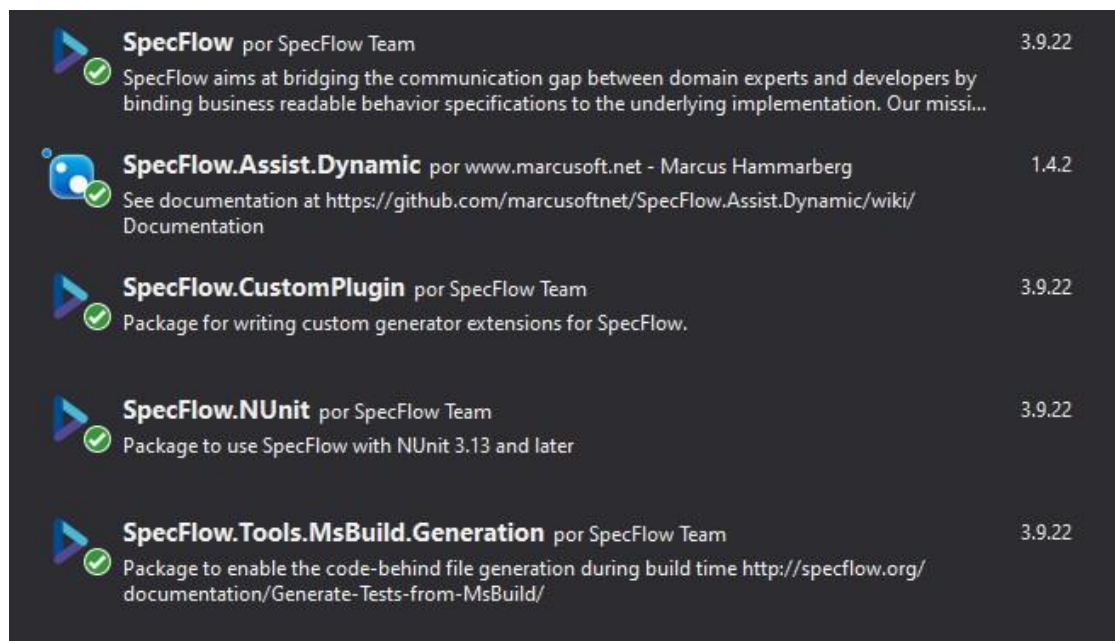
2. Se descargarán los paquetes Nuget al dar click derecho en dependencias del proyecto



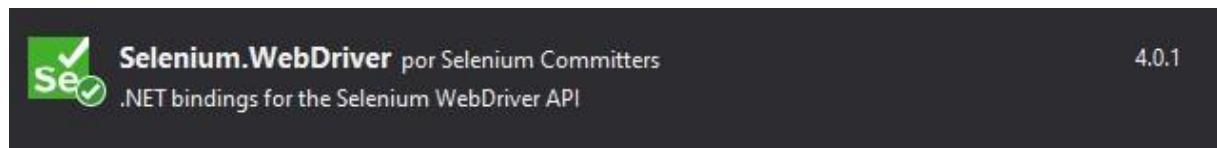
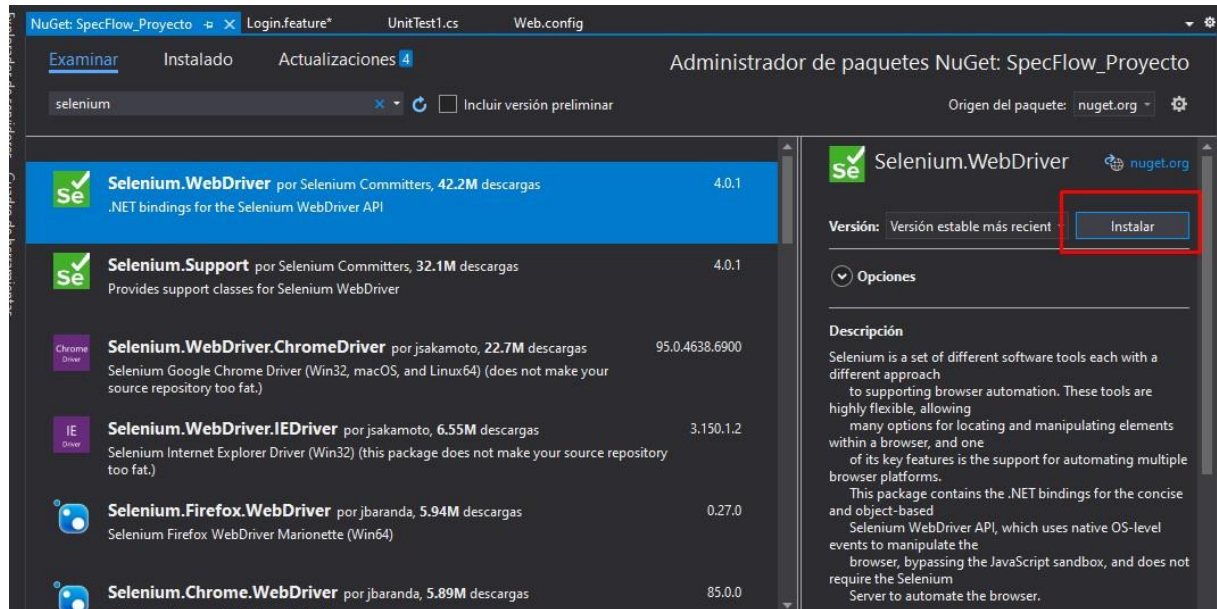
3. Se instalarán los paquetes con ayuda del buscador, se selecciona y presiona Instalar



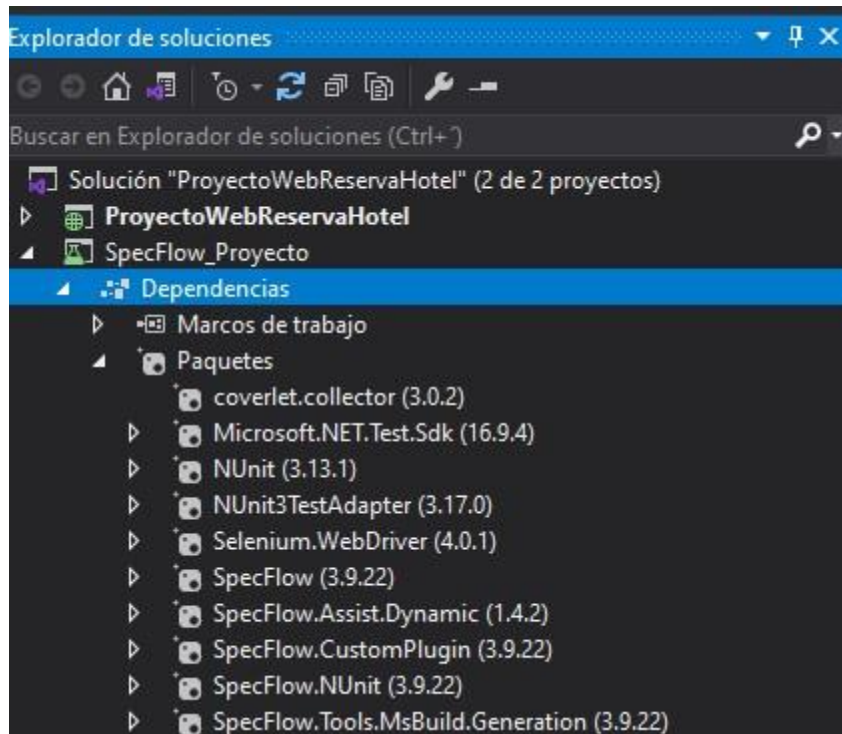
Paquetes instalados



Selenium webdriver









4. Como se observa todos los paquetes se ubican en dependencias/paquetes:





5. Para usar el chromedriver sin errores de compatibilidad por la versión, se descargo la versión del navegador.

Index of /95.0.4638.69/

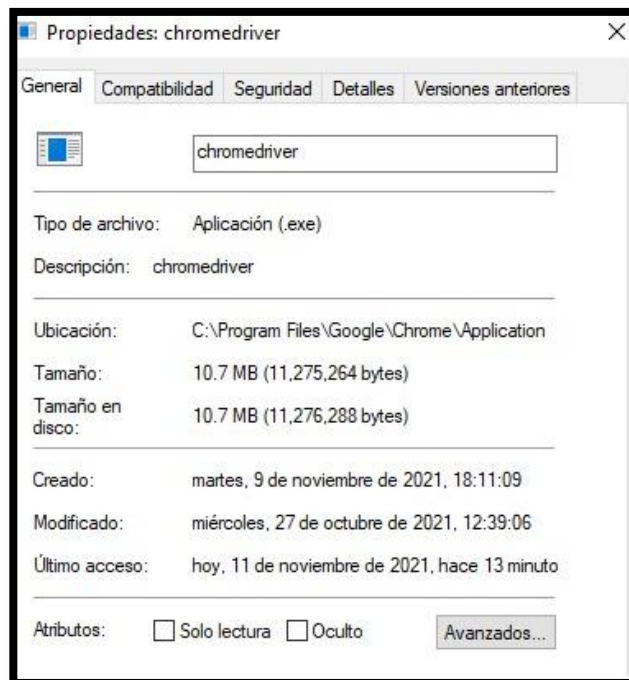
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>ETag</u>
 Parent Directory		-	
 chromedriver_linux64.zip	2021-11-08 09:31:24	9.52MB	a1fdedf1bdec0e140458bef35006cdc4
 chromedriver_mac64.zip	2021-11-08 09:31:26	7.81MB	f97559e10cbe2bd4ef29a019958af514
 chromedriver_mac64_m1.zip	2021-11-08 09:31:28	7.39MB	5d7f30a75623bc63d2d3705928a82102
 chromedriver_win32.zip	2021-11-08 09:31:30	5.74MB	82a40345336b35b9c3f1779f5089cfd3
 notes.txt	2021-11-08 09:31:35	0.00MB	0fe40272d622c011b7523dcca7a7397

Acerca de Chrome

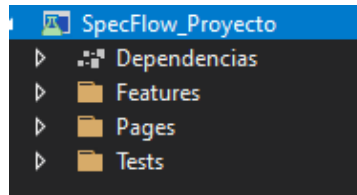
 **Google Chrome**

 Chrome está actualizado
Versión 95.0.4638.69 (Build oficial) (64 bits)

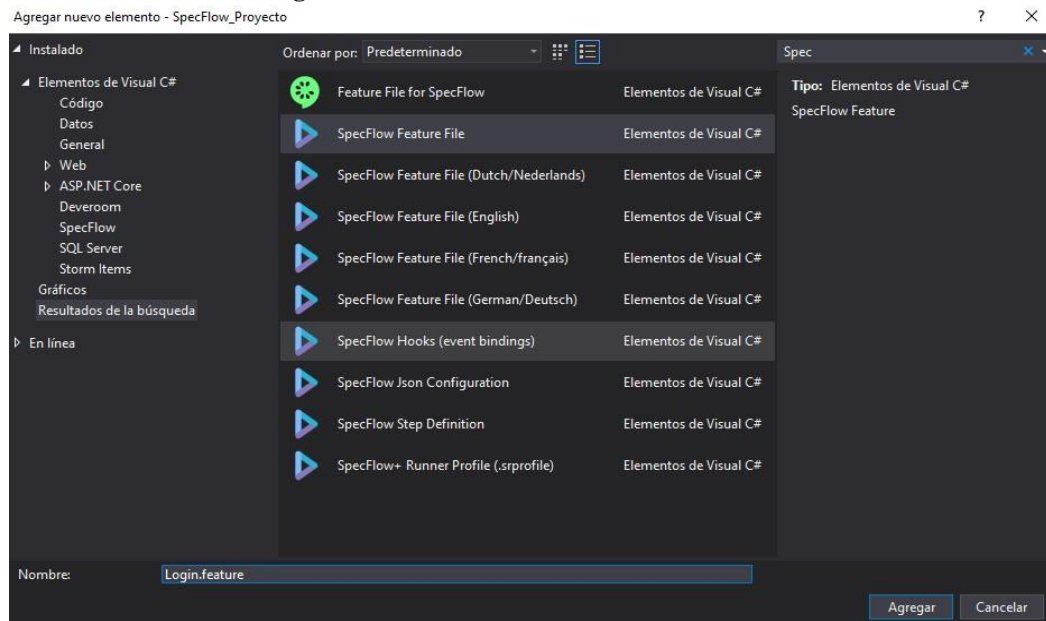
6. Además se uso la ruta donde se encuentra el archivo al instanciar IWebDriver.



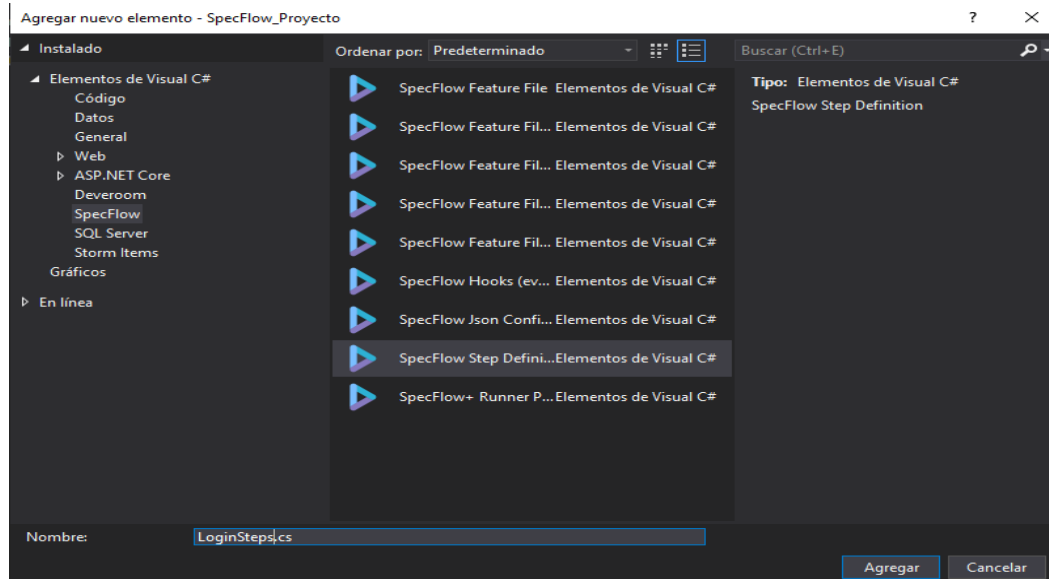
7. El proyecto de prueba de Specflow necesita tres partes importantes, guardadas en las carpetas de sus respectivos nombres: Features, Pages(hooks) y Steps:



8. Se creara un archivo login.feature:



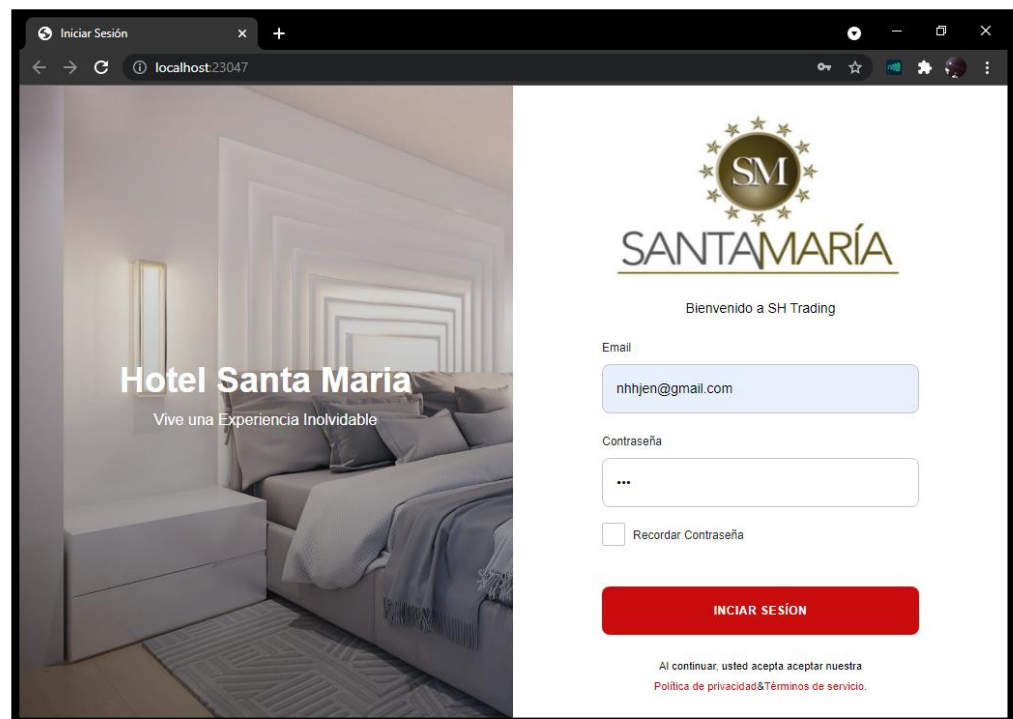
9. Archivo de pasos loginSteps.cs



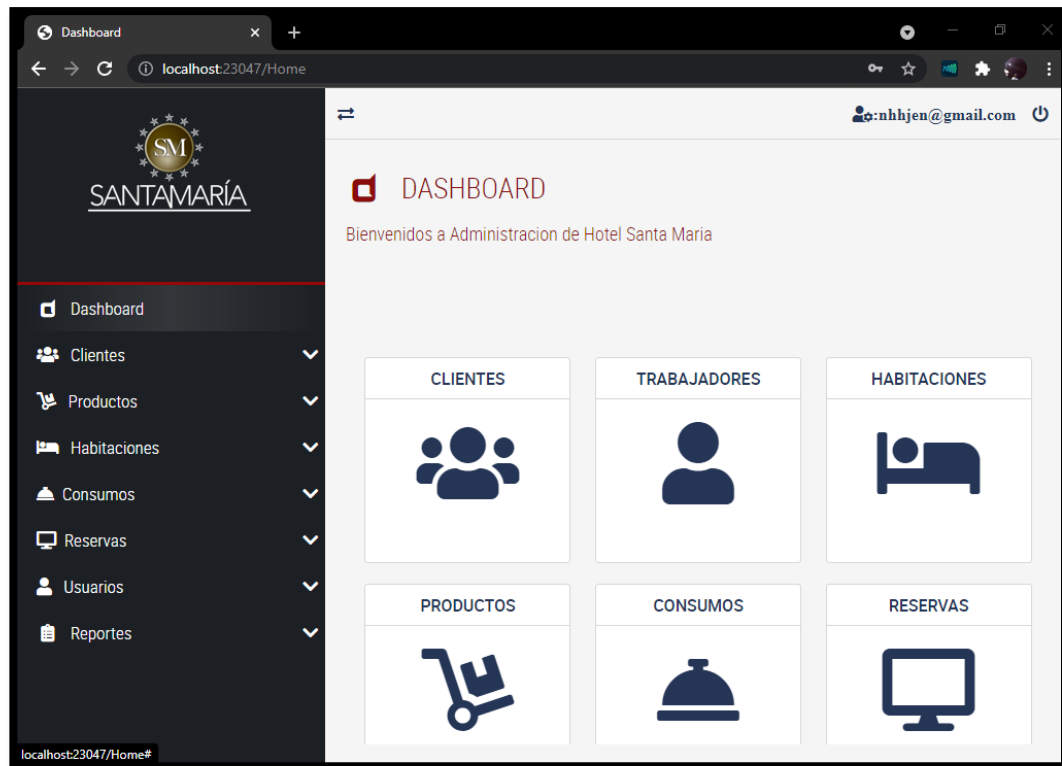
10. y la clase LoginPage.cs también se creara.
11. Feature Login, definirá el comportamiento a realizar al ingresar a la aplicación e ingresar los datos como email y contraseña para luego presionar el botón iniciar sesión y redirigirnos al Dashboard de la aplicación , en este ejemplo se muestra que los valores son admin admin como ejemplo de prueba. El feature muestra el comportamiento a realizar aplicando BDD.

```
w_Proyecto.csproj  LoginPage.cs  LoginSteps.cs  login.feature  X
1  Feature: Login
2      Iniciar sesion en el sistema de gestion de Hotel
3
4  @smoke
5  Scenario: Login del sitio web
6      #pasos
7      Given Inicio la Aplicacion
8      And Ingreso email y contraseña
9          | email | password |
10         | admin | admin |
11      When Presiono el boton Iniciar Sesion
12      Then Se muestra el Dashboard
```

Al ingresar nuestro correo y contraseña (paso línea 8 y 11)



Se muestra el Dashboard (línea 12 del feature)



12. Por cada paso hay un método generado:

Inicio la aplicación : (se instancio usando la ruta de chromedriver para evitar errores de version y se uso link de pruebo debido a que nuestra aplicación no esta subida a algún servidor)

```
[Given(@"Inicio la Aplicacion")]
0 referencias
public void GivenInicioLaAplicacion()
{
    IWebDriver webdriver = new ChromeDriver(@"C:\Program Files\Google\Chrome
        \Application");
    //webdriver.Navigate().GoToUrl("localhost:23047");
    webdriver.Navigate().GoToUrl("https://aula.upt.edu.pe/login/index.php");
    loginp = new LoginPage(webdriver);
}
```

Al ingresar email y contraseña

```
[Given(@"Ingreso email y contraseña")]
0 referencias
public void GivenIngresoEmailYContraseña(Table table)
{
    dynamic data = table.CreateDynamicInstance();
    loginp.Login((string)data.email, (string)data.password);
}
```

Al presionar botón de inicio de sesión

```
[When(@"Presiono el boton Iniciar Sesion")]
0 referencias
public void WhenPresionoElBotonIniciarSesion()
{
    loginp.ClickLoginButton();
}
```

Al ingresar a la dirección esperada

```
[Then(@"Se muestra el Dashboard")]
0 referencias
public void ThenSeMuestraElDashboard()
{
    Assert.That(loginp.IsEmployeeDetailsExist(), Is.True);
}
```

13. En el loginPage están todos los elementos que se espera probar (la parte verde corresponde a nuestro proyecto y la parte mostrada es en relación a la pagina del aula virtual para realizar la demostración)

Se usa el constructor para instanciar el IWebDriver

```
6 referencias
public IWebDriver WebDriver { get; }
1 referencia
public LoginPage(IWebDriver driver)
{
    WebDriver = driver;
}
```

Los elementos UI

```
//elementos UI
1referencia
public IWebElement lnkLogin => WebDriver.FindElement(By.LinkText("login"));
//public IWebElement txtuser => WebDriver.FindElement(By.Name("email"));
1 referencia
public IWebElement txtuser => WebDriver.FindElement(By.Name("username"));
1 referencia
public IWebElement txtpassword => WebDriver.FindElement(By.Id("password"));
//public IWebElement txtpassword => WebDriver.FindElement(By.Name("password"));
1 referencia
public IWebElement btnlogin => WebDriver.FindElement(By.Id("login"));
//public IWebElement btnlogin => WebDriver.FindElement(By.Name("login"));
1 referencia
public IWebElement lnkHome => WebDriver.FindElement(By.LinkText("aula"));
```

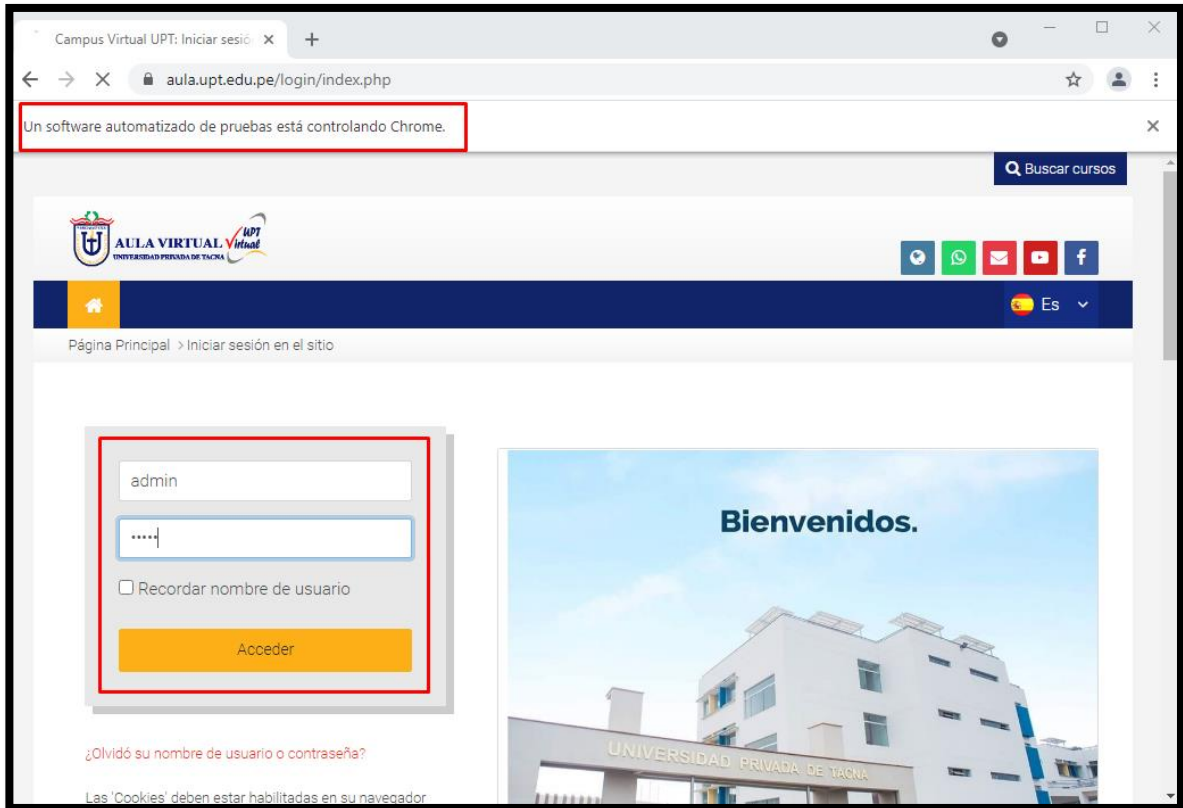

Los métodos para realizar la prueba

```
0 referencias
public void ClickLogin() => lnkLogin.Click();

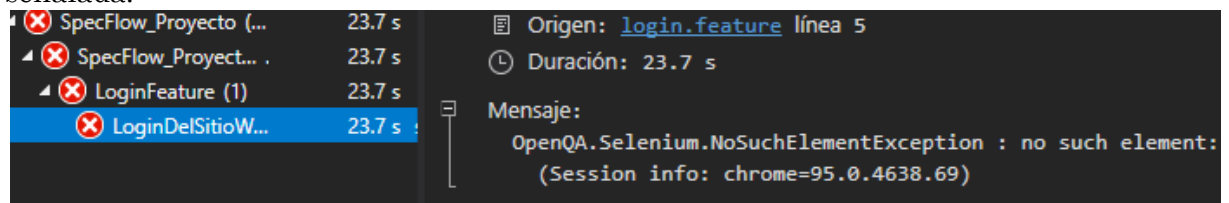
1 referencia
public void Login(string user,string pass)
{
    txtuser.SendKeys(user);
    txtpassword.SendKeys(pass);
}

1 referencia
public void ClickLoginButton() => btnlogin.Submit();
1 referencia
public bool IsEmployeeDetailsExist() => lnkHome.Displayed;
```

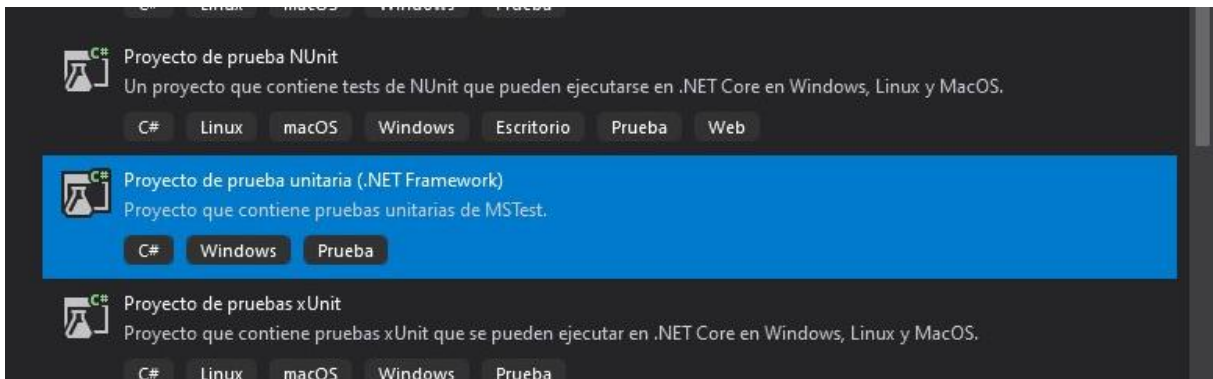
Al realizar la prueba



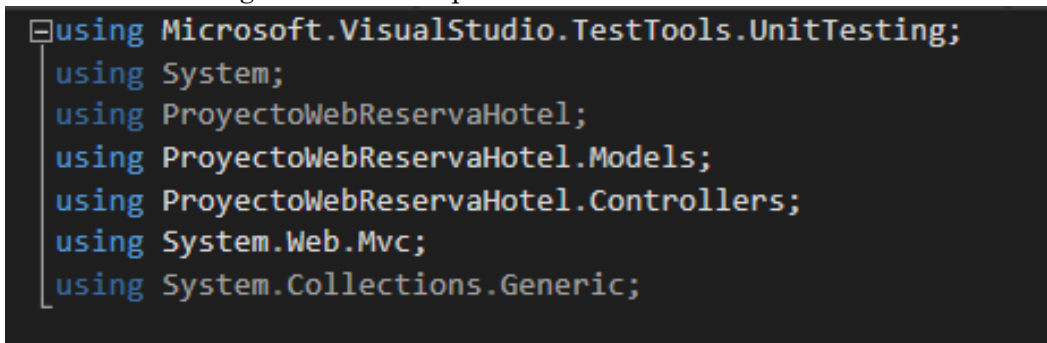
No se ingreso con los datos correctos por lo que muestra error al no ingresar a la pagina señalada.



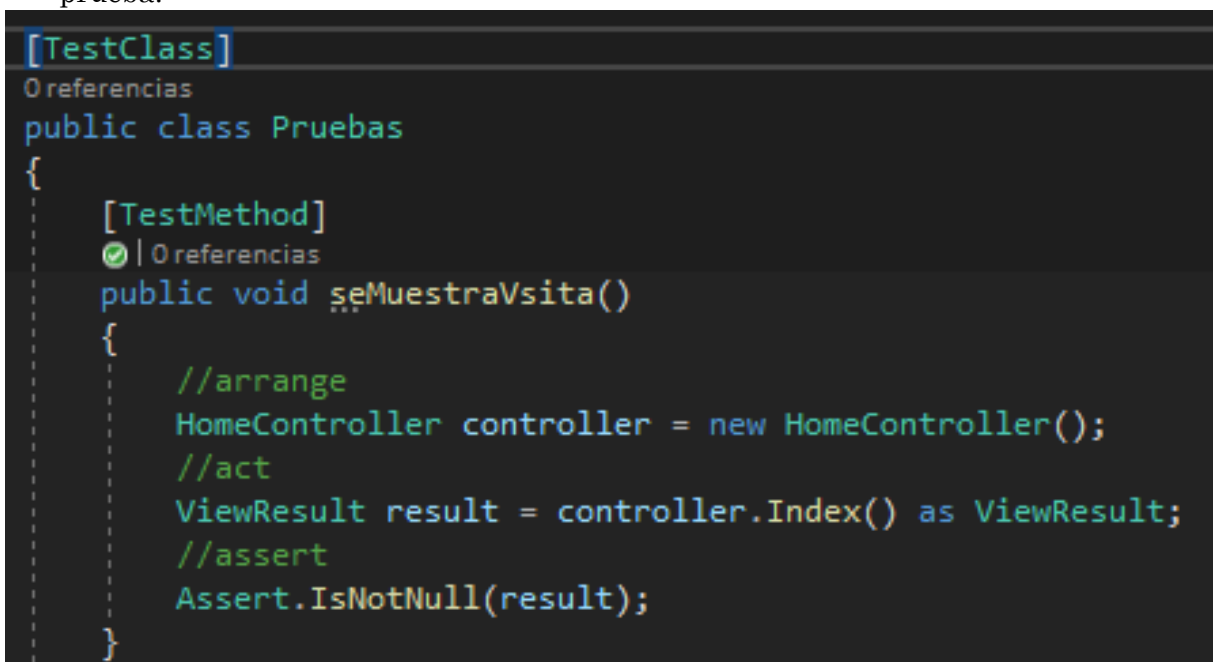
14. Pruebas unitarias: se realizaron usando el proyecto de prueba unitaria para Net Framework



15. Usando los siguientes namespaces



16. Para realizar pruebas dentro de la clase se coloca [TestClass] y para los metodos [TestMethod] de este modo Visual Studio reconoce a Clases y métodos que son de prueba.



```
[TestMethod]
✓ | 0 referencias
public void TrabajadorNoesNulo()
{
    //arrange
    trabajador objtrab = new trabajador("pepe", "perez");
    //act
    var worker = objtrab.nombres;
    //assert
    Assert.IsNotNull(worker);
}
```

```
[TestMethod]
✓ | 0 referencias
public void clienteNulo()
{
    //arrange
    cliente objcliente = new cliente();
    //act
    var cli = objcliente.nombres;
    //assert
    Assert.IsNull(cli);
}
```

```
[TestMethod]
✓ | 0 referencias
public void clienteNoesnulo()
{
    //arrange
    cliente objcliente = new cliente("edgar", "lopez");
    //act
    string nom= objcliente.nombres;
    //assert
    Assert.IsNotNull(nom);
}
```

```
[TestMethod]
✓ | 0 referencias
public void productonulo()
{
    //arrange
    producto objprod = new producto();
    //act
    var prod = objprod.idproducto;
    //assert
    Assert.IsNotNull(prod);
}
```

```
[TestMethod]
✓ | 0 referencias
public void CompararNombreCliente()
{
    //arrange
    cliente objcliente = new cliente("pepe","perez");
    //act
    var nombcompleto = objcliente.nombres + " " + objcliente.apellidos;
    //assert
    Assert.AreEqual("pepe perez", nombcompleto);
}
```

```
[TestMethod]
✓ | 0 referencias
public void reportesnulo()
{
    //arrange
    reportes objrepo = new reportes();
    //act
    var repo = objrepo.idreportes;
    //assert
    Assert.IsNotNull(repo);
}
```

Los resultados de las pruebas:

Explorador de pruebas

8 7 1

Prueba	Duración	Rasgos
▶ PruebasUnitarias (7)	4.4 s	
▶ PruebasUnitarias (7)	4.4 s	
▶ Pruebas (7)	4.4 s	
clienteNoesnulo	< 1 ms	
clientenulo	4 ms	
CompararNom...	109 ms	
productonulo	< 1 ms	
reportesnulo	< 1 ms	
seMuestraVsita	146 ms	
TrabajadorNoe...	4.2 s	

Resumen del grupo

Pruebas

Pruebas en grupo: 7

⌚ Duración total: 4.4 s

Salidas

7 Correcta

6. Conclusiones

al realizar las pruebas se entendio mejor el comportamiento de los métodos probados, pudiendo constatar los errores que podría tener nuestro código, pero además se mostro el comportamiento de la aplicación usando BDD mediante los features y los pasos para que el programa funcione correctamente.