



Instituto Tecnológico de las Américas

Estudiante:

Francisco Concepcion

Matricula:

2020-10682

Docente:

Kelyn Tejada Belliard

Materia:

Programacion 3

Tarea:

Tarea 4

## 1- Historias de usuarios:

### Historia no.1

Una ferretería popular es un negocio de venta de productos ferreteros que inició recientemente en el país. Llegó un punto donde se ve en la necesidad de buscar un sistema donde se pueda tener la información de los paquetes de sus clientes para trasladar su negocio a un local más grande. Al mudarse a un nuevo local Don Marcos tuvo que empezar a comprar más productos para poder satisfacer la demanda de los clientes. También Don Marcos tiene algunos trabajadores atendiendo la ferretería, pero estos nuevos trabajadores no le generan mucha confianza. Don Marcos se vio obligado a recurrir a la búsqueda de un programa que pueda gestionar los paquetes de sus clientes. Muchas veces estos empleados se quedan solos y Don Marcos desconoce si los empleados recibieron clientes.

Criterios de aceptación:

Como usuario quiero que se registren los datos de los usuarios y se guarden.

### Historia no.2

Una empresa en el país acaba de lanzarse, la empresa se encarga de seguros humanos dichos seguros le favorecen a las personas en cualquier tipo de medicamento es decir que a la hora de que estas personas o clientes de esta empresa de seguros van al médico y le dan una indicación de algún medicamento o recibir un daño grave en el cual tengan que recurrir a emergencias de algún hospital o alguna consulta esta seguro le asegura a sus clientes un descuento para que ellos tengan un beneficio en cualquier médico con la posibilidad de pagar menos, esta empresa de seguros busca tener una gestión de los paquetes de medicamentos que vienen del extranjero.

Criterios de aceptación:

Los Usuarios cuando reciban nuevos tenga una apartado donde pueda tener y guardar la información de los usuarios.

## Historia no.3

Una suplidora quiere una programa que cumpla con las necesidades de su nuevo emprendimiento, el cual se tomaron detalles de los requerimientos de este programa para saber que necesita, el cual se llevo a la conclusión de que necesita un sistema que registre los nuevos usuarios que tiene la suplidora( eso fue uno de los requisitos). Se realizara un trabajo de calidad y que cumpla con todas las necesidades de la suplidora con el cual se pueda sentir conforme. La suplidora con esta gestión de datos de clientes podrá tener un mejor manejo de la nueva información sobre los clientes y que estos clientes se sientan seguros con los términos de la privacidad de los datos.

Criterios de aceptación:

Comprobar que los Usuarios estarán en la interfaz donde se puedan visualizar los datos que se generen cuando guarde la información de mis clientes.

## Historia no.4

Un cliente esta iniciando un negocio de cosmético y tiene varias suplidoras que las cuales les va a suministrar los productos necesarios para poner a la venta, el cliente quiere tener una gestion de datos en donde guarde la información de las suplidoras como el correo, teléfono, etc. El cliente quiere que esta tenga un registro de login donde solo puedan entrar usuarios que estén registrado para administrar dichos dato, el cliente quiere dicho proyecto lo más rápido posible.

Criterios de aceptación:

Verificar que los datos que ingrese el usuario sean guardados correctamente y se pueda visualizar.

## Historia no.5

Una empresa de bajo recursos busca tener un programa donde se pueda guardar la información de usuarios, y que estos datos se guarden, la empresa no cuenta con altos recursos para el proyecto pero necesita tener esas informaciones guardadas, lo cual se le evaluara un trabajo en base a sus recursos.

Criterios de aceptación:

Verificar que la información del crud se guarde de manera exitosa y verificar que los datos se muestren en la pantalla correctamente.

## 2-Codigo del Crud que utilize para hacer la prueba automatizada:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYXxHfc+NcPb1dKGj7Sk"
crossorigin="anonymous">

    <title>Agendar Usuarios</title>
    <style>

      body{
        background:url("fondo.jpg") ;
        color:rgb(0, 0, 0)
      }
      thead{

        background-color: rgb(0, 0, 0);
      }

      tbody, tr, .jumbotron{
        background:url("Fj.png");
        color: aliceblue;
      }

      .jumbotron {
        padding: 1rem 1rem;
        margin-bottom: 0rem;
        background-color: #00f8cb;
        border-radius: .3rem;
```

```

    }
  </style>
</head>
<body>

  <div class="jumbotron jumbotron-fluid">
    <div class="container">
      <h1 class="display-8">Proyecto Final </h1>
    </div>
  </div>
  <div class="container">
    <div class="row">
      <div class="col-lg-6">
        <button id="btnNuevo" class="btn btn-primary" data-toggle="tooltip"
title="Nuevo Producto"><svg class="bi bi-plus-circle-fill" width="1em"
height="1em" viewBox="0 0 16 16" fill="currentColor"
xmlns="http://www.w3.org/2000/svg"><path fill-rule="evenodd" d="M16 8A8 8 0 1 1 0
8a8 8 0 0 1 16 0zM8.5 4a.5.5 0 0 0-1 0v3.5H4a.5.5 0 0 0 0 1h3.5V12a.5.5 0 0 0 1
0V8.5H12a.5.5 0 0 0 0-1H8.5V4z"/></svg></button>
        <!--Modal-->
        <div id="modalAltaEdicion" class="modal fade" tabindex="-1"
role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
          <div class="modal-dialog" role="document">
            <div class="modal-content">
              <div class="modal-header bg-primary text-light">
                <h5 class="modal-title"
id="exampleModalLabel">Alta / Edición</h5>
                <button type="button" class="close" data-
dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
              </div>
              <form>
                <div class="modal-body">
                  <input id="id" type="hidden">
                  <div class="form-
group">
                    <label>Nombre</label>
                    <input id="nombre" type="text" class="form-
control" required>
                  </div>
                  <div class="form-group">
                    <label>Apellido</label>
                    <input id="apellido" type="text"
class="form-control" required>

```

```

        </div>
        <div class="form-group">
            <label>Correo</label>
            <input id="correo" type="text" class="form-
control" required>

        </div>
        <div class="form-group">
            <label>Telefono</label>
            <input id="telefono" type="number"
class="form-control" required>

        </div>
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-
secondary" data-dismiss="modal" tabindex="2">Cancelar</button>
            <button type="submit" id="btnGuardar"
class="btn btn-primary" translate="1">Guardar</button>
        </div>
    </form>

</div>
</div>
</div>
<table id="tablaCrud" class="table table-bordered">
    <thead>
        <tr class="bg-red text-black">
            <th scope="col">NOMBRE</th>
            <th scope="col">APELLIDO</th>
            <th scope="col">CORREO</th>
            <th scope="col">TELEFONO</th>
            <th scope="col">ACCIONES</th>
        </tr>
    </thead>
    <tbody id="bodydelCrud">
    </tbody>
</table>

</div>
</div>
</div>
<div>
    <button onclick="deslogeo ()" class="">Deslogearse</button>
</div>

<!-- Optional JavaScript -->

```

```

    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
integrity="sha384-
OgVRvuATP1z7JjHLku0U7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
crossorigin="anonymous"></script>

    <script
src="https://cdn.jsdelivr.net/npm/sweetalert2@9.14.4/dist/sweetalert2.all.min.js"
></script>

    <!-- The core Firebase JS SDK is always required and must be listed first -->
    <script src="https://www.gstatic.com/firebasejs/7.15.0/firebase.js"></script>

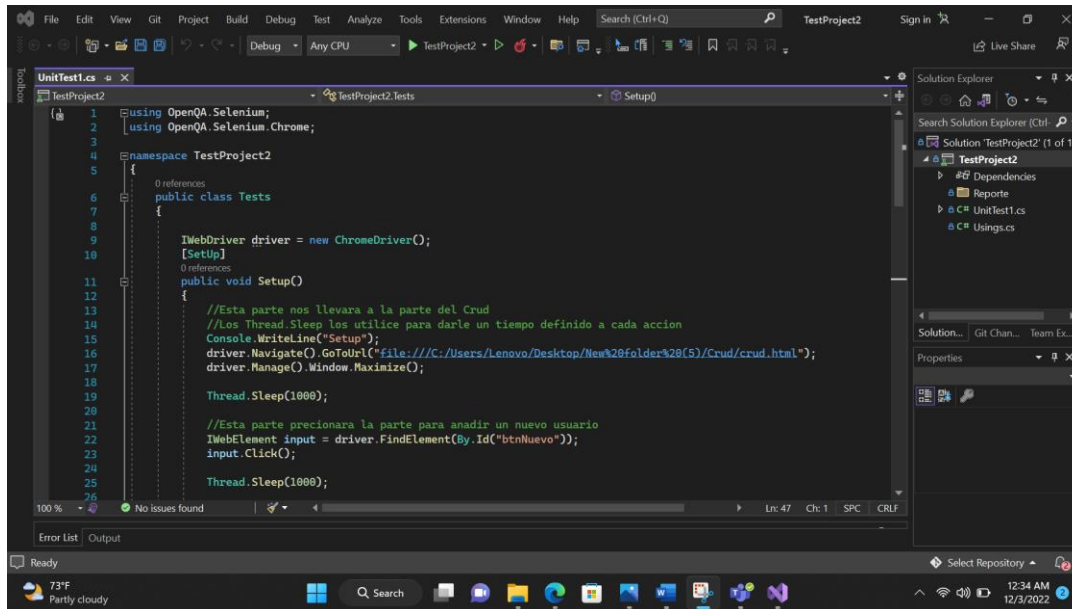
    <script src="app.js"></script>
    <script>
        function deslogueo ()
        {
            Swal.fire({
            position: 'top-center',
            icon: 'success',
            title: 'Se ha deslogueado correctamente',
            showConfirmButton: true,
            timer: 7500
        })
        location.href="index.html";
        }
    </script>
</body>
</html>

```



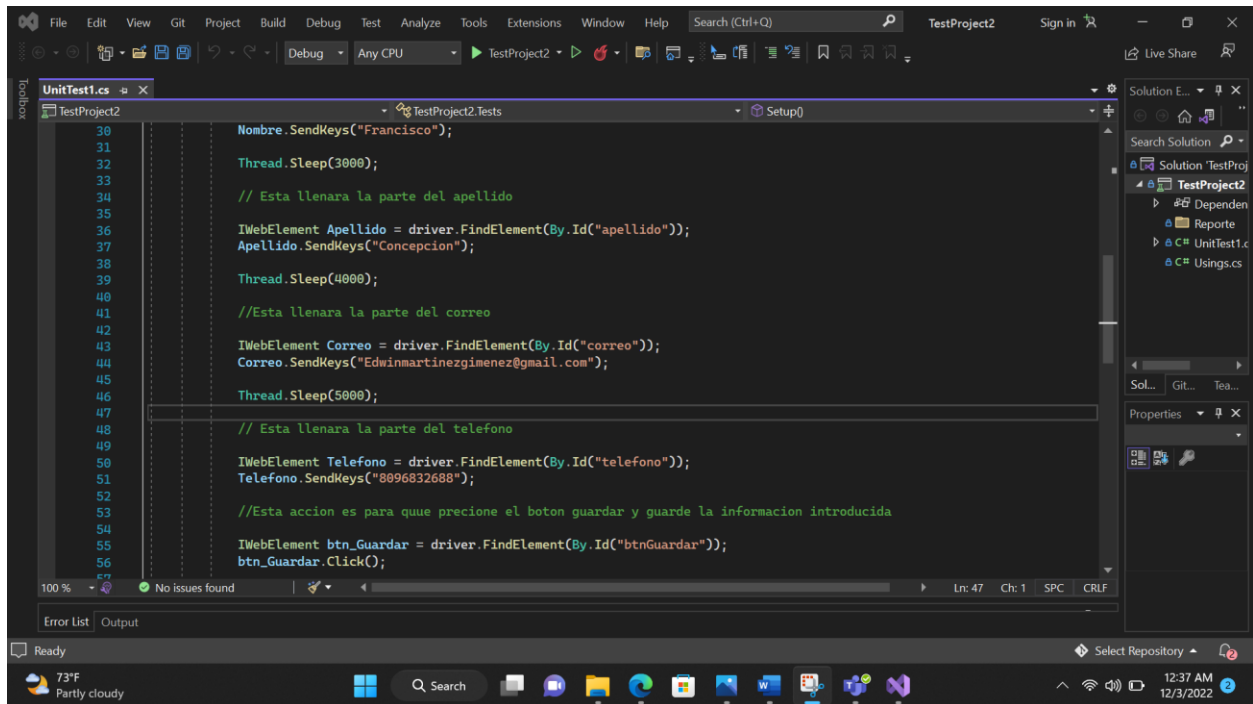
### 3-Prueba del código de la librería hecho en C#:

Primera parte del código de la librería de selenium:



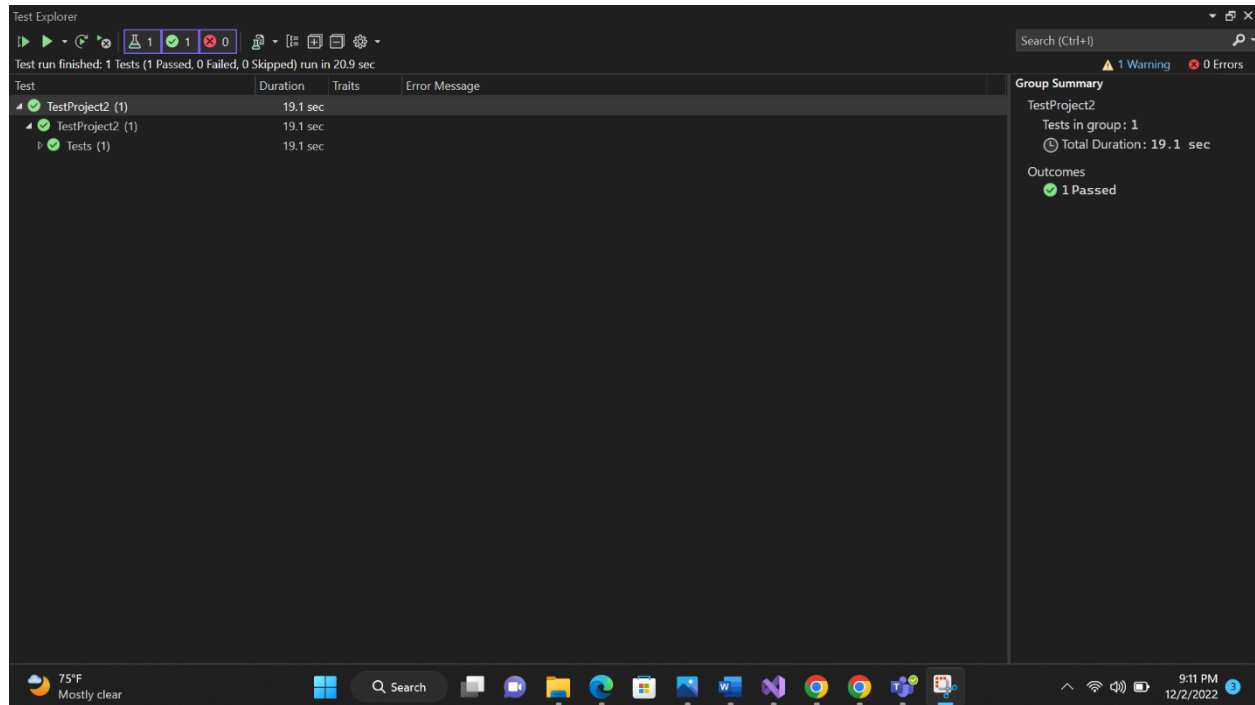
```
1 using OpenQA.Selenium;
2 using OpenQA.Selenium.Chrome;
3
4 namespace TestProject2
5 {
6     public class Tests
7     {
8
9         IWebDriver driver = new ChromeDriver();
10        [SetUp]
11        public void Setup()
12        {
13            //Esta parte nos llevara a la parte del Crud
14            //Los Thread.Sleep los utilice para darle un tiempo definido a cada accion
15            Console.WriteLine("Setup");
16            driver.Navigate().GoToUrl("file:///C:/Users/Lenovo/Desktop/New%20folder%20(5)/Crud/crud.html");
17            driver.Manage().Window.Maximize();
18
19            Thread.Sleep(1000);
20
21            //Esta parte precionara la parte para anadir un nuevo usuario
22            IWebElement input = driver.FindElement(By.Id("btnNuevo"));
23            input.Click();
24
25            Thread.Sleep(1000);
26        }
27    }
28 }
```

Segunda parte del código:



```
30 Nombre.SendKeys("Francisco");
31
32 Thread.Sleep(3000);
33
34 // Esta llenara la parte del apellido
35
36 IWebElement Apellido = driver.FindElement(By.Id("apellido"));
37 Apellido.SendKeys("Concepcion");
38
39 Thread.Sleep(4000);
40
41 //Esta llenara la parte del correo
42
43 IWebElement Correo = driver.FindElement(By.Id("correo"));
44 Correo.SendKeys("Edwinmartinezgimenez@gmail.com");
45
46 Thread.Sleep(5000);
47
48 // Esta llenara la parte del telefono
49
50 IWebElement Telefono = driver.FindElement(By.Id("telefono"));
51 Telefono.SendKeys("8096832688");
52
53 //Esta accion es para que precione el boton guardar y guarde la informacion introducida
54
55 IWebElement btn_Guardar = driver.FindElement(By.Id("btnGuardar"));
56 btn_Guardar.Click();
```

#### 4-Reporte de que el proyecto paso la prueba de automatización en C#:



Aportes de la tarea :

## 1-¿Qué es Selenium?

Selenium es un conjunto de utilidades que facilita la labor de obtener juegos de pruebas para aplicaciones web. Para ello nos permite grabar, editar y depurar casos de prueba, que podrán ser ejecutados de forma automática e iterativa posteriormente.

Además de ser una herramienta para registrar acciones, permite editarlas manualmente o crearlas desde cero. Las acciones se basan en el uso de diferentes API's en diferentes lenguajes (PHP, Ruby, JAVA, Javascript, etc). Entre su principales características podemos nombrar:

- 1-Facilidad de registro y ejecución de los test.
- 2-Referencia a objetos DOM en base al ID, nombre o a través de XPath.
- 3-Auto-completado para todos los comandos.
- 4-Las acciones pueden ser ejecutadas paso a paso.
- 5-Herramientas de depuración y puntos de ruptura (breakpoints).
- 6-Los test pueden ser almacenados en diferentes formatos.

El potencial de esta herramienta puede ser utilizado para la grabación de las pruebas funcionales durante la Generación de pruebas de regresión. Con este servicio se consigue obtener una batería de pruebas automatizadas que podrán ser utilizadas cuando sea necesario repetir las pruebas.

## 2-¿Cuales son los Recursos necesarios de selenium?

Los navegadores mas conocidos compatibles con la utilización de Selenium son:

- 1-Explorer
- 2-Mozilla Firefox
- 3-Google Chrome
- 4-Safari

Las componentes de Selenium que son necesarias para la grabación y ejecución de las pruebas son:

Selenium client v1.0.1: cliente de Selenium, necesario para crear pruebas Junit con Selenium.

Selenium IDE v1.0.2: plugin de Firefox para la grabación de las pruebas, paso a paso.

Selenium Server v1.0.1: servidor de Selenium, que es el que realiza las pruebas.

Para mas detalles sobre esto [pincha aqui](#).

Otras consideraciones

La versión de Java necesaria es la JDK 1.6.0\_16.

Es recomendable tener instalado un entorno de desarrollo, por ejemplo Eclipse 3.4.2.

Otra herramienta recomendable ya que nos va a ser muy útil para el reciclaje de las pruebas es Firebug. Se trata de un plugin de Firefox que permite inspeccionar el código HTML de una página y, entre otras opciones, obtener el Xpath de los diferentes elementos.

Para que las pruebas puedan ser extendidas para y puedan ser ejecutadas en varios navegadores, necesitaremos tener instalada la herramienta JUnit. Para mas detalles sobre esto pincha aquí.

### 3-Diferentes tipos de entorno de selenium.

Selenium Core: el módulo del núcleo contiene la funcionalidad básica del marco de trabajo, incluyendo el JavaScriptRunner y la API de comandos de prueba básicos.

Selenium IDE: Selenium IDE es el entorno de desarrollo del marco de pruebas, sobre el que se basa la extensión IDE para Chrome y Firefox. Se utiliza para grabar y reproducir pruebas.

Selenium WebDriver: WebDriver es la interfaz básica para simular las interacciones del usuario con cualquier navegador, ya sea Firefox, Chrome, Edge, Safari o Internet Explorer. Desde 2018, la API es un estándar W3C oficial.

Selenium Grid: Selenium Grid es una extensión de WebDriver y de su predecesor Selenium Remote Control (RC), que permite la ejecución de pruebas en paralelo en múltiples servidores. De este modo, el tiempo de pruebas se reduce considerablemente.

### 4-¿Para qué se utiliza Selenium WebDriver?

Selenium es una herramienta popular en entornos de pruebas y se utiliza sobre todo como base para una amplia gama de programas en este espectro. Entre los ejemplos más conocidos está el marco de trabajo de extremo a extremo Protractor, que fue desarrollado especialmente para las pruebas de aplicaciones de Angular y Angular JS y que utiliza la API WebDriver para simular las interacciones con el usuario. El software de automatización de pruebas Appium, diseñado para aplicaciones móviles nativas e híbridas, usa la interfaz estandarizada para realizar las pruebas de forma sencilla y cómoda en el lenguaje de programación deseado.

El popular servicio web basado en la nube BrowserStack también utiliza Selenium: este servicio, desarrollado en la India, ofrece distintas suscripciones de pago después de un período de prueba gratuito y emplea este marco para sus pruebas automatizadas de aplicaciones móviles y de escritorio.