

DOCUMENTO DE PLAN DE PRUEBAS

Plantilla inspirada en el estándar IEEE 829-2008 y adaptada a las necesidades del curso de
Construcción de Software

Tabla de contenido

1. INTRODUCCIÓN	3
1.1. PROPÓSITO	3
1.2. OBJETIVOS	3
2. ALCANCE	3
2.1. ALCANCE DE PRUEBAS	3
2.2. LIMITACIONES	4
3. DEFINICIONES, SIGLAS, Y ABREVIATURAS	4
4. DOCUMENTOS DE REFERENCIA	5
5. ESTRATEGIA DE PRUEBAS	6
5.1. PRUEBAS FUNCIONALES	6
5.2. PRUEBAS DE INTEGRACIÓN	22
6. CRITERIOS DE ENTRADA Y DE SALIDA	23
6.1. CRITERIO DE ENTRADA DEL PLAN DE PRUEBAS	23
6.2. CRITERIO DE ACEPTACIÓN DEL PLAN DE PRUEBAS	23
6.3. CRITERIO DE SUSPENSIÓN Y REANUDACIÓN	23
7. ENTREGABLES	23
7.1. PROTOCOLO DE PRUEBAS	23
7.2. INFORME DE PRUEBAS	23
7.3. REPORTE DE OBSERVACIONES	24
8. AMBIENTE DE PRUEBAS	24
8.1. REQUERIMIENTOS BASE DE HARDWARE	24
8.2. REQUERIMIENTOS BASE DE SOFTWARE EN EL AMBIENTE DE PRUEBAS	24
8.3. HERRAMIENTAS DE APOYO PARA LA EJECUCIÓN DE PRUEBAS	24
9. CRONOGRAMA DE TRABAJO	25
10. RIESGOS, DEPENDENCIAS, SUPOSICIONES Y RESTRICCIONES	25
10.1. RIESGOS	25
10.2. DEPENDENCIAS	26
10.3. SUPOSICIONES	26
10.4. RESTRICCIONES	26
11. APROBACIÓN	27

1. Introducción

1.1. Propósito

El propósito de este documento es otorgar la información necesaria para con ello poder planear y llevar un control de los esfuerzos de pruebas definidas en el Proyecto Los Rápidos S.A. que se han realizado a lo largo del mismo, garantizando así, que los requerimientos y especificaciones funcionales se encuentren desarrolladas y cumplan con la funcionalidad y estándares de calidad esperados, minimizando los incidentes al momento en el que se llevan a cabo las pruebas de aceptación del usuario funcional.

1.2. Objetivos

- Identificar los requisitos necesarios para llevar a cabo las pruebas y así desarrollar estrategias que nos permitan identificar los recursos necesarios para realizar el test.
- Elaborar las pruebas a partir de la estimación de las actividades de Software Testing realizada por el equipo.
- Desarrollar las clases, componentes y datos de prueba.
- Establecer casos y condiciones de prueba basados en las especificaciones funcionales, que al ejecutarse simulen un ambiente de producción.
- Identificar errores o problemas al ejecutar los distintos casos de prueba
- Definir planes de respuesta, los cuales deben ser específicos para cada situación particular y riesgo.
- Dar un seguimiento oportuno a la corrección de incidentes reportados, minimizando el tiempo de reprobación.
- Determinar las diferencias existentes entre las especificaciones funcionales o requerimientos y el desarrollo del producto.
- Validar y comprobar la calidad del producto a ser entregado, con base a los estándares definidos.

2. Alcance

2.1. Alcance de pruebas

En el alcance de nuestro proyecto se especifican el tipo de pruebas que se realizarán. El siguiente cuadro contiene las pruebas que se aplicarán a nuestro proyecto.

Tipo de pruebas	Aplica según el proyecto (si / no)
Pruebas funcionales	Si
Pruebas de regresión	Si
Pruebas de interfaz de usuario	No
Prueba de rendimiento	Si
Pruebas de carga	No
Pruebas de seguridad y control de acceso	Si
Pruebas de instalación	Si
Pruebas de integración	No

Pruebas de caja blanca	No
Pruebas de control de bitácoras	No

Los módulos a los cuales se le aplicarán las distintas pruebas para comprobar su calidad son:

- **Módulo de Login**
- **Módulo de Ambulancia**
 - Registrar Ambulancia
 - Consultar Ambulancia
- **Módulo de Cliente**
 - Registrar Cliente
 - Consultar Cliente
- **Módulo de Conductor**
 - Registrar Conductor
 - Consultar Conductor
- **Módulo de Petición**
 - Registrar Petición
 - Consultar Petición
- **Módulo de Secretaria**
 - Registrar Secretaria
 - Consultar Secretaria

2.2. Limitaciones

Se busca optimizar de manera considerable las pruebas, por tal motivo sólo se realizará un conjunto de pruebas para los casos esperados, no se abarcarán todos los casos posibles. Además, no se van a realizar:

- Pruebas de interoperabilidad.
- Pruebas de interfaz con otros módulos.
- Test para comprobar que el sistema no tenga errores.

3. Definiciones, siglas, y abreviaturas

Dentro de los casos de prueba, hay una serie de abreviaciones y siglas que indican los tipos de pruebas que se van a ejecutar. A continuación, se definen cada uno de ellos:

Siglas	Definición
CP	Caso de Prueba
Descripción de Prueba	Detalla los parámetros a ingresar al momento de hacer el caso de prueba agregando los tipos de parámetros y el valor de este.
Requisito previo	Representa las acciones o datos que se necesitan antes de realizar las pruebas.

4. Documentos de referencia

DOCUMENTO VERSIÓN
Documentos de caso de uso
Especificación de requisitos

5. Estrategia de pruebas

Una de las herramientas utilizadas para la realización de las pruebas unitarias es MSTest (Visual Studio Unit Testing Framework), un potente framework de pruebas que nos ofrece este ecosistema de Microsoft. Estas pruebas unitarias nos permiten observar cierto tipo de funcionamiento singular en el sistema, por ejemplo, la autenticación de un usuario o en este caso nuestro módulo de ingreso de usuario el cual como requisito previo debe constar con cierta cantidad de caracteres en su contraseña y así validar estos campos. De esta manera, se realizó todos los casos de pruebas de cada uno de los módulos, en los casos de Registrar usuario, Registrar cliente, Registrar conductores y Registrar Ambulancia, se aplicaron en la modificación, consulta y eliminación de estos.

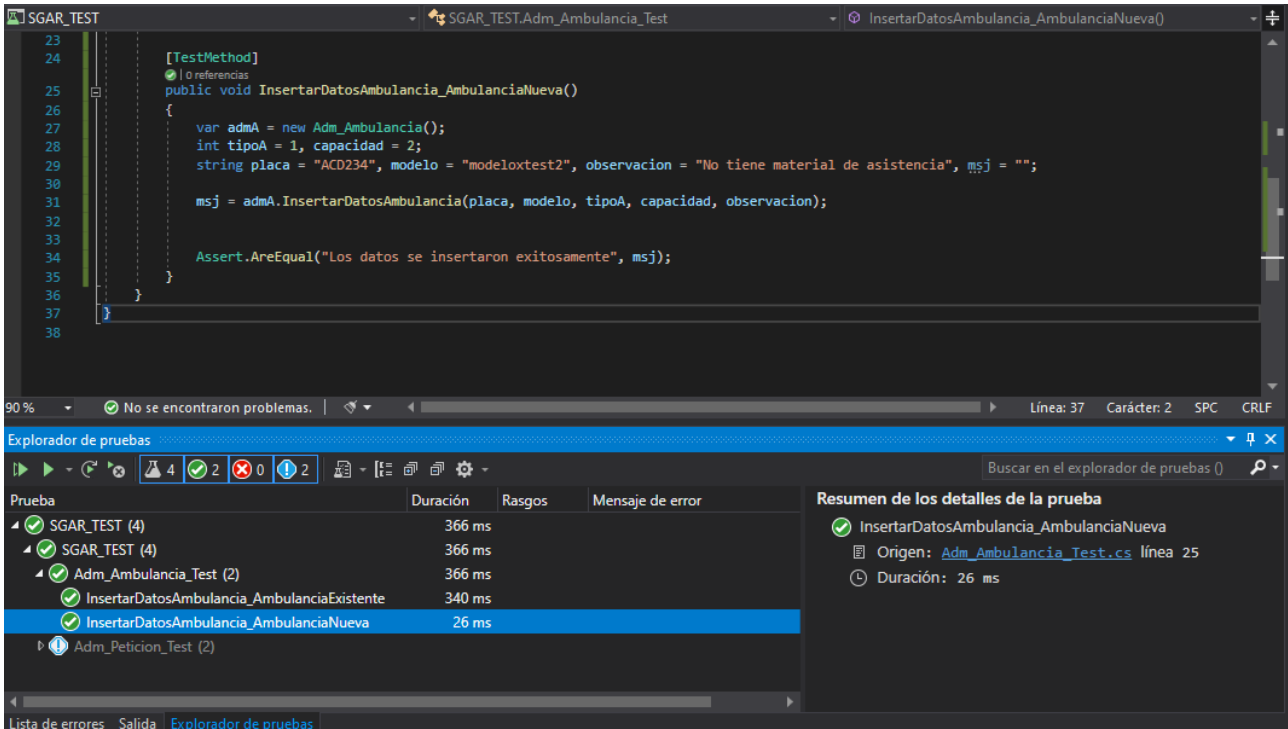
5.1. Pruebas funcionales

Las pruebas unitarias se realizan para velar que el producto esperado cumpla con las funciones específicas para los cuales fue creado. Este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, generalmente se requiere apoyo de los usuarios finales ya que ellos pueden aportar mucho en el desarrollo de casos de prueba complejos, enfocados al negocio.

A continuación, se describe cada uno de los módulos expuestos en el apartado del alcance de este documento, a los que se les aplicaron las pruebas:

Código	CP-001
Caso de prueba	Registro de ambulancia nueva
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <code>InsertarDatosAmbulancias_AmbulanciaNueva</code>. Se ingresan los siguientes parámetros: <ul style="list-style-type: none"> <code>id_ambulancia</code> 1 <code>placa</code> AAC123 <code>modelo</code> modeloxtest <code>tipoA</code> 2 <code>capacidad</code> 3 <code>observacion</code> NN <code>disponibilidad</code> 1 Comparar el mensaje de retorno del método de testeo con el método <code>InsertarDatosAmbulancia</code> de la clase <code>Adm_Ambulancia</code>.
Requisito previo	<p>Se debe cumplir con los siguientes requisitos:</p> <ul style="list-style-type: none"> Llenar todos los parámetros. La placa debe tener 3 letras y 3 o 4 dígitos.
Resultado esperado	La ambulancia será registrada en la aplicación.

Resultado obtenido	Los datos se insertaron exitosamente
Estado	Exitoso
Observaciones	No se puede registrar una placa ya existente



```

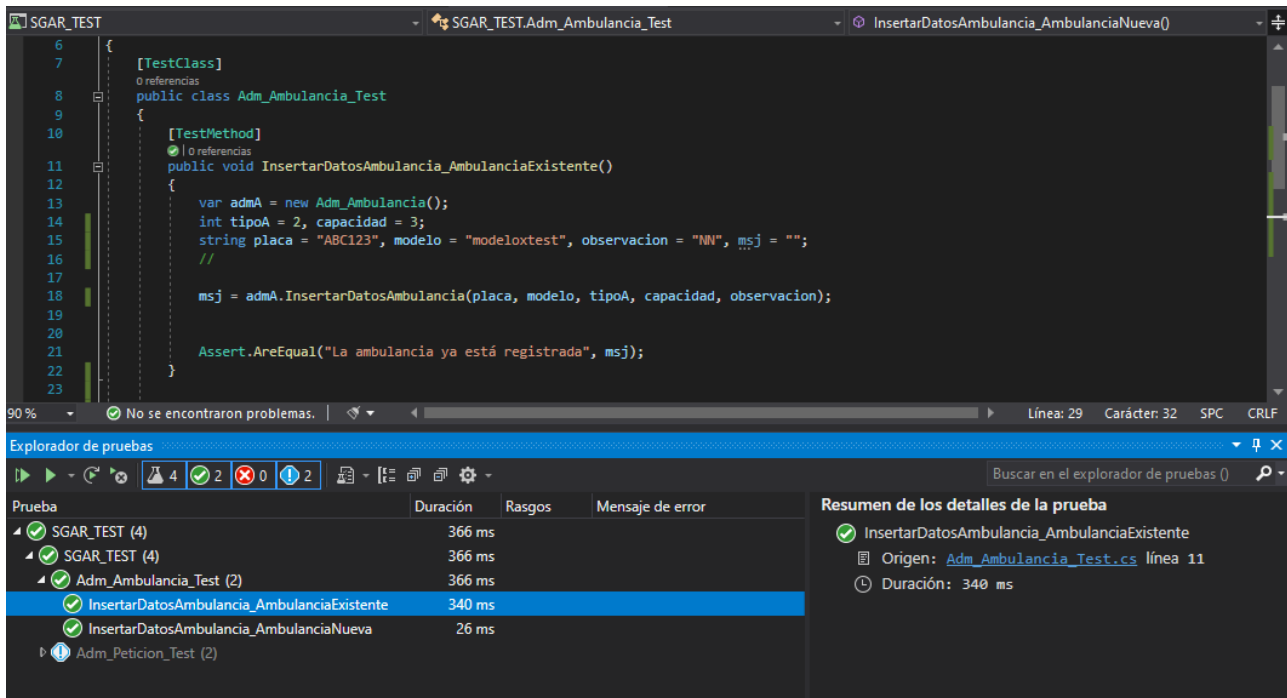
23
24 [TestMethod]
25 public void InsertarDatosAmbulancia_AmbulanciaNueva()
26 {
27     var admA = new Adm_Ambulancia();
28     int tipoA = 1, capacidad = 2;
29     string placa = "ACD234", modelo = "modeloxtest2", observacion = "No tiene material de asistencia", msj = "";
30
31     msj = admA.InsertarDatosAmbulancia(placa, modelo, tipoA, capacidad, observacion);
32
33     Assert.AreEqual("Los datos se insertaron exitosamente", msj);
34 }
35
36
37
38
    
```

Explorador de pruebas
 Prueba: SGAR_TEST (4), Adm_Ambulancia_Test (2), InsertarDatosAmbulancia_AmbulanciaExistente, InsertarDatosAmbulancia_AmbulanciaNueva, Adm_Peticion_Test (2)
 Duración: 366 ms, 366 ms, 366 ms, 340 ms, 26 ms
 Mensaje de error: No se encontraron problemas.

Resumen de los detalles de la prueba
 InsertarDatosAmbulancia_AmbulanciaNueva
 Origen: Adm_Ambulancia_Test.cs línea 25
 Duración: 26 ms

Código	CP-002														
Caso de prueba	Registro de ambulancia existente														
Responsable	Desarrolladores														
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <code>InsertarDatosAmbulancias_AmbulanciaExistente</code>. Se ingresan los siguientes parámetros: <table border="0"> <tr><td><code>id_ambulancia</code></td><td>1</td></tr> <tr><td><code>placa</code></td><td>WER456</td></tr> <tr><td><code>modelo</code></td><td>modeloxtest2</td></tr> <tr><td><code>tipoA</code></td><td>1</td></tr> <tr><td><code>capacidad</code></td><td>2</td></tr> <tr><td><code>observacion</code></td><td>NN</td></tr> <tr><td><code>disponibilidad</code></td><td>1</td></tr> </table> Comparar el mensaje de retorno del método de testeo con el método <code>InsertarDatosAmbulancia</code> de la clase <code>Adm_Ambulancia</code>. 	<code>id_ambulancia</code>	1	<code>placa</code>	WER456	<code>modelo</code>	modeloxtest2	<code>tipoA</code>	1	<code>capacidad</code>	2	<code>observacion</code>	NN	<code>disponibilidad</code>	1
<code>id_ambulancia</code>	1														
<code>placa</code>	WER456														
<code>modelo</code>	modeloxtest2														
<code>tipoA</code>	1														
<code>capacidad</code>	2														
<code>observacion</code>	NN														
<code>disponibilidad</code>	1														
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. 														

	<ul style="list-style-type: none"> La placa debe tener 3 letras y 3 o 4 dígitos.
Resultado esperado	La ambulancia NO será registrada en la aplicación.
Resultado obtenido	Los ambulancia ya está registrada
Estado	Fallido
Observaciones	No se puede registrar una placa ya existente



```

6  {
7  [TestClass]
8  0 referencias
9  public class Adm_Ambulancia_Test
10 {
11 [TestMethod]
12 0 referencias
13 public void InsertarDatosAmbulancia_AmbulanciaExistente()
14 {
15     var admA = new Adm_Ambulancia();
16     int tipoA = 2, capacidad = 3;
17     string placa = "ABC123", modelo = "modeloxtest", observacion = "NN", msj = "";
18     //
19     msj = admA.InsertarDatosAmbulancia(placa, modelo, tipoA, capacidad, observacion);
20
21     Assert.AreEqual("La ambulancia ya está registrada", msj);
22 }
23 }
    
```

Explorador de pruebas

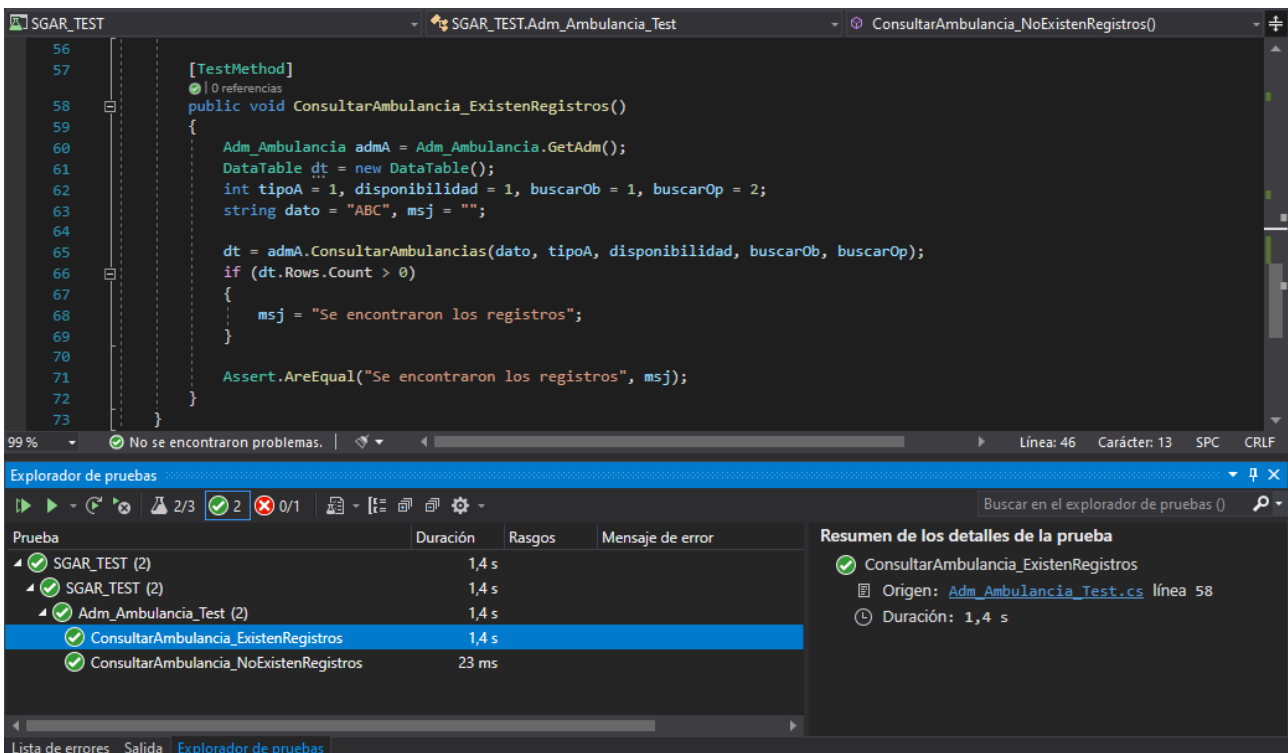
Prueba	Duración	Rasgos	Mensaje de error
SGAR_TEST (4)	366 ms		
SGAR_TEST (4)	366 ms		
Adm_Ambulancia_Test (2)	366 ms		
InsertarDatosAmbulancia_AmbulanciaExistente	340 ms		
InsertarDatosAmbulancia_AmbulanciaNueva	26 ms		
Adm_Peticion_Test (2)			

Resumen de los detalles de la prueba

- InsertarDatosAmbulancia_AmbulanciaExistente
- Origen: [Adm_Ambulancia_Test.cs](#) línea 11
- Duración: 340 ms

Código	CP-003										
Caso de prueba	Consultar ambulancias existentes										
Responsable	Desarrolladores										
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <i>ConsultarAmbulancia_ExistenRegistros</i>. Se ingresan los siguientes parámetros: <table border="0"> <tr> <td>dato</td> <td>ABC</td> </tr> <tr> <td>tipoA</td> <td>1</td> </tr> <tr> <td>disponibilidad</td> <td>1</td> </tr> <tr> <td>buscarOb</td> <td>1</td> </tr> <tr> <td>buscarOp</td> <td>2</td> </tr> </table> Comparar el mensaje de retorno del método de testeo con el método <i>ConsultarAmbulancias</i> de la clase <i>Adm_Ambulancia</i>. 	dato	ABC	tipoA	1	disponibilidad	1	buscarOb	1	buscarOp	2
dato	ABC										
tipoA	1										
disponibilidad	1										
buscarOb	1										
buscarOp	2										
Requisito previo	Se debe cumplir con los siguientes requisitos:										

	<ul style="list-style-type: none"> Llenar el parámetro dato obligatoriamente con una placa o modelo según selección de radiobutton. Seleccionar checkbox de tipo ambulancia si se quiere buscar por ese filtro.
Resultado esperado	Existen registros de ambulancias.
Resultado obtenido	Se encontraron los registros.
Estado	Exitoso
Observaciones	Si no hay ambulancias registradas, no se muestran los registros.



```

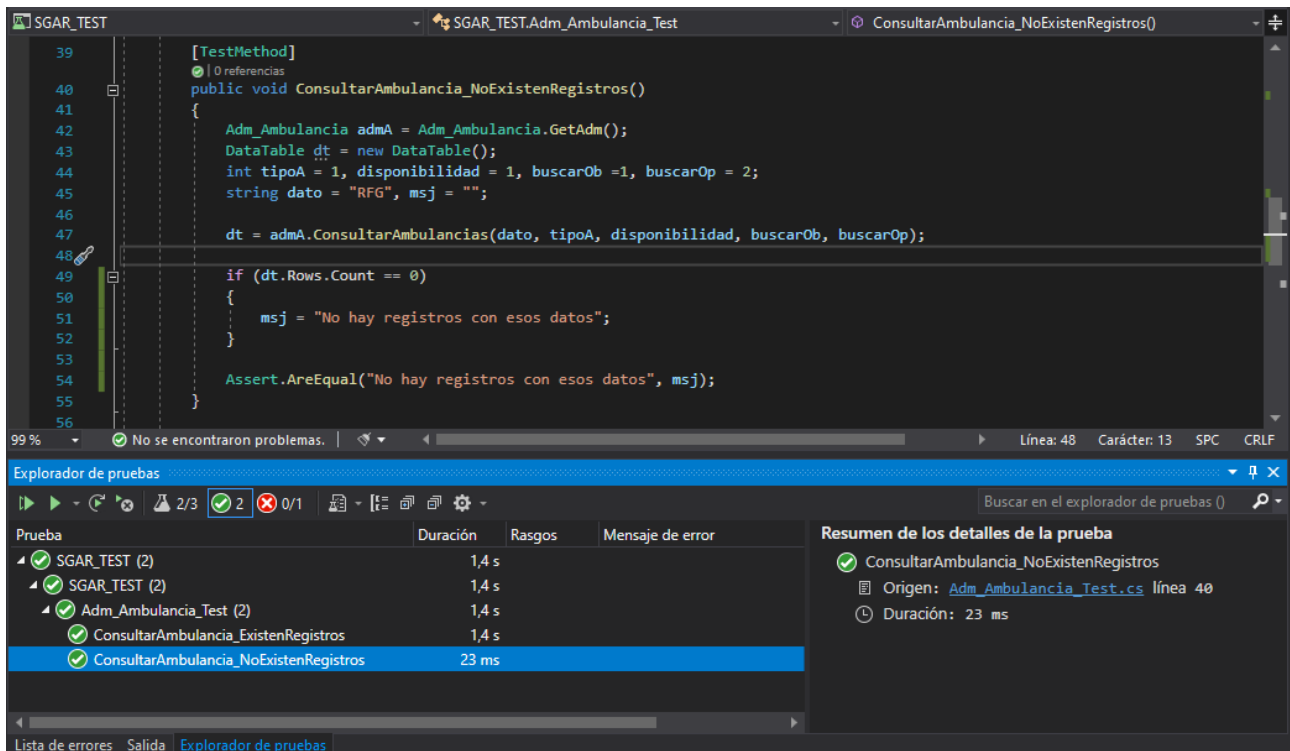
56
57
58 [TestMethod]
59 public void ConsultarAmbulancia_ExistenRegistros()
60 {
61     Adm_Ambulancia admA = Adm_Ambulancia.GetAdm();
62     DataTable dt = new DataTable();
63     int tipoA = 1, disponibilidad = 1, buscarOb = 1, buscarOp = 2;
64     string dato = "ABC", msj = "";
65
66     dt = admA.ConsultarAmbulancias(dato, tipoA, disponibilidad, buscarOb, buscarOp);
67     if (dt.Rows.Count > 0)
68     {
69         msj = "Se encontraron los registros";
70     }
71     Assert.AreEqual("Se encontraron los registros", msj);
72 }
73
  
```

Explorador de pruebas
 Prueba Duración Rasgos Mensaje de error
 SGAR_TEST (2) 1,4 s
 SGAR_TEST (2) 1,4 s
 Adm_Ambulancia_Test (2) 1,4 s
 ConsultarAmbulancia_ExistenRegistros 1,4 s
 ConsultarAmbulancia_NoExistenRegistros 23 ms

Resumen de los detalles de la prueba
 ConsultarAmbulancia_ExistenRegistros
 Origen: Adm_Ambulancia_Test.cs línea 58
 Duración: 1,4 s

Código	CP-004
Caso de prueba	Registro de ambulancias
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba ConsultarAmbulancia_NoExistenRegistros. Se ingresan los siguientes parámetros: id_ambulancia 1 placa WER456 modelo modeloxtest2 tipoA 1 capacidad 2

	<i>observacion</i> NN <i>disponibilidad</i> 1 <ul style="list-style-type: none"> Comparar el mensaje de retorno del método de testeo con el método <i>ConsultarAmbulancias</i> de la clase <i>Adm_Ambulancia</i>.
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. La placa debe tener 3 letras y 3 o 4 dígitos.
Resultado esperado	No existen registros de ambulancias.
Resultado obtenido	No hay registros con esos datos.
Estado	Fallido
Observaciones	Si no hay ambulancias registradas, no se muestran los registros.



The screenshot displays the Visual Studio IDE with a unit test named `ConsultarAmbulancia_NoExistenRegistros()` in the `SGAR_TEST.Adm_Ambulancia_Test` class. The test method is as follows:

```

[TestMethod]
public void ConsultarAmbulancia_NoExistenRegistros()
{
    Adm_Ambulancia admA = Adm_Ambulancia.GetAdm();
    DataTable dt = new DataTable();
    int tipoA = 1, disponibilidad = 1, buscarOb = 1, buscarOp = 2;
    string dato = "RFG", msj = "";

    dt = admA.ConsultarAmbulancias(dato, tipoA, disponibilidad, buscarOb, buscarOp);

    if (dt.Rows.Count == 0)
    {
        msj = "No hay registros con esos datos";
    }

    Assert.AreEqual("No hay registros con esos datos", msj);
}
    
```

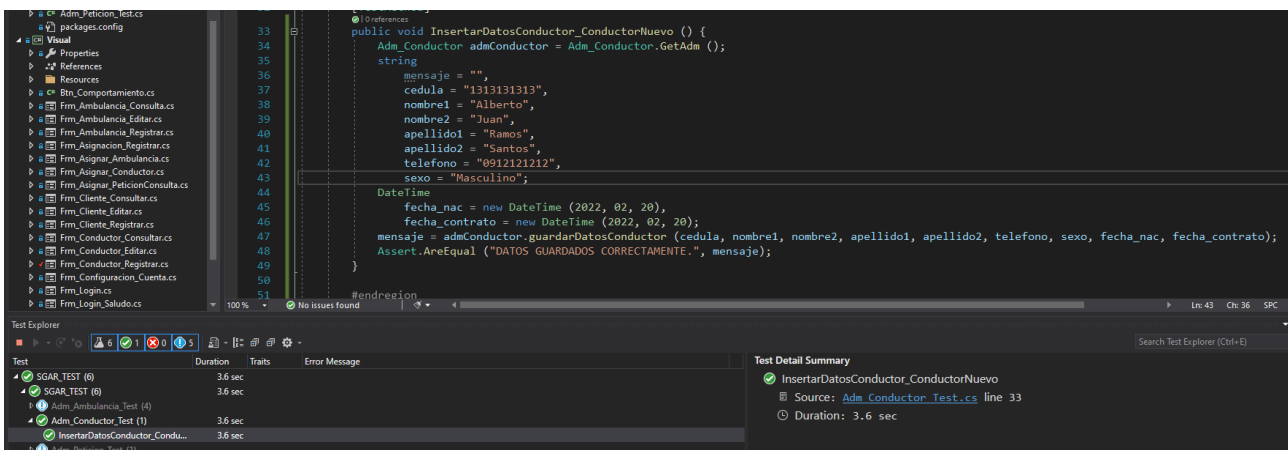
Below the code editor, the **Explorador de pruebas** (Test Explorer) shows the test results. The test `ConsultarAmbulancia_NoExistenRegistros` passed successfully. The summary of the test results is as follows:

Prueba	Duración	Rasgos	Mensaje de error
SGAR_TEST (2)	1,4 s		
SGAR_TEST (2)	1,4 s		
Adm_Ambulancia_Test (2)	1,4 s		
ConsultarAmbulancia_ExistenRegistros	1,4 s		
ConsultarAmbulancia_NoExistenRegistros	23 ms		

The **Resumen de los detalles de la prueba** (Test Details Summary) for `ConsultarAmbulancia_NoExistenRegistros` shows:

- Origen: `Adm_Ambulancia_Test.cs` línea 40
- Duración: 23 ms

Código	CP-005																		
Caso de prueba	Registro de conductor nuevo.																		
Responsable	Desarrolladores																		
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <code>InsertarDatosConductor_ConductorNuevo</code>. Se ingresan los siguientes parámetros: <table border="0"> <tr><td><i>cedual</i></td><td>1313131313</td></tr> <tr><td><i>nombre1</i></td><td>Alberto</td></tr> <tr><td><i>nombre2</i></td><td>Juan</td></tr> <tr><td><i>apellido1</i></td><td>Ramos</td></tr> <tr><td><i>apellido2</i></td><td>Santos</td></tr> <tr><td><i>telefono</i></td><td>0912121212</td></tr> <tr><td><i>sexo</i></td><td>Masculino</td></tr> <tr><td><i>fecha_nac</i></td><td>2022, 02, 20</td></tr> <tr><td><i>fecha_contrato</i></td><td>2022, 02, 20</td></tr> </table> Comparar el mensaje de retorno del método de testeo con el método <code>InsertarDatosConductor</code> de la clase <code>Adm_Conductor</code>. 	<i>cedual</i>	1313131313	<i>nombre1</i>	Alberto	<i>nombre2</i>	Juan	<i>apellido1</i>	Ramos	<i>apellido2</i>	Santos	<i>telefono</i>	0912121212	<i>sexo</i>	Masculino	<i>fecha_nac</i>	2022, 02, 20	<i>fecha_contrato</i>	2022, 02, 20
<i>cedual</i>	1313131313																		
<i>nombre1</i>	Alberto																		
<i>nombre2</i>	Juan																		
<i>apellido1</i>	Ramos																		
<i>apellido2</i>	Santos																		
<i>telefono</i>	0912121212																		
<i>sexo</i>	Masculino																		
<i>fecha_nac</i>	2022, 02, 20																		
<i>fecha_contrato</i>	2022, 02, 20																		
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. El número de cédula debe tener diez dígitos. 																		
Resultado esperado	El conductor será registrado en la aplicación.																		
Resultado obtenido	Los datos se insertaron exitosamente.																		
Estado	Exitoso.																		
Observaciones	No se puede registrar un conductor con cédula ya existente.																		



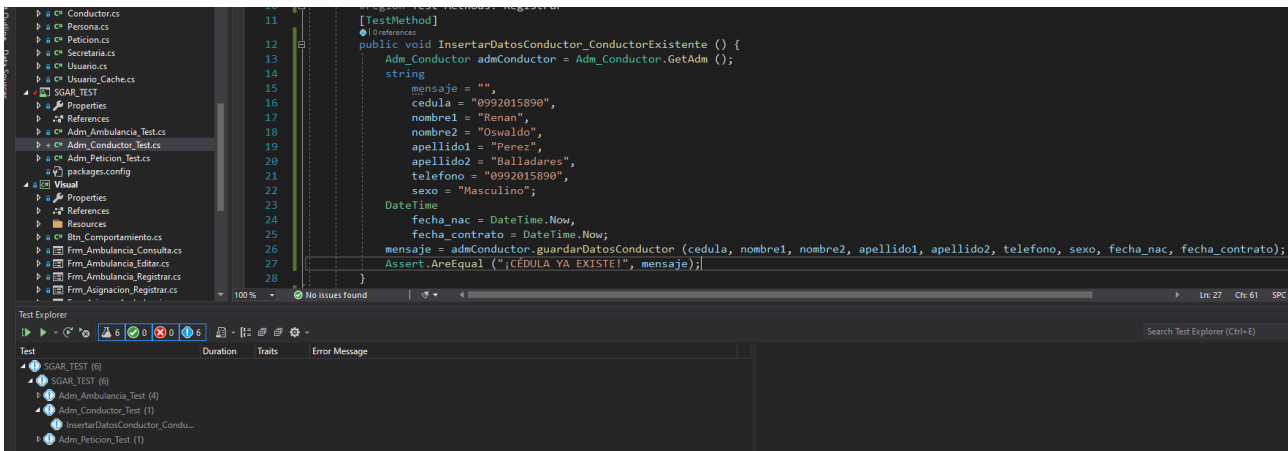
The screenshot shows the Visual Studio IDE. On the left, the Solution Explorer displays a project named 'Adm_Peticion_Test.cs'. The main editor shows the following C# code:

```

public void InsertarDatosConductor_ConductorNuevo () {
    Adm_Conductor admConductor = Adm_Conductor.GetAdm ();
    string
        mensaje = "",
        cedula = "1313131313",
        nombre1 = "Alberto",
        nombre2 = "Juan",
        apellido1 = "Ramos",
        apellido2 = "Santos",
        telefono = "0912121212",
        sexo = "Masculino";
    DateTime
        fecha_nac = new DateTime (2022, 02, 20),
        fecha_contrato = new DateTime (2022, 02, 20);
    mensaje = admConductor.guardarDatosConductor (cedula, nombre1, nombre2, apellido1, apellido2, telefono, sexo, fecha_nac, fecha_contrato);
    Assert.AreEqual ("DATOS GUARDADOS CORRECTAMENTE.", mensaje);
}
    
```

Below the code, the Test Explorer shows the execution results of the test. The test is named 'SGAR_TEST (6)' and has a duration of 3.6 seconds. The test is marked as 'Passed' (green checkmark). The Test Detail Summary shows the test name 'InsertarDatosConductor_ConductorNuevo' and the source file 'Adm_Conductor_Test.cs' at line 33. The duration is 3.6 seconds.

Código	CP-006																		
Caso de prueba	Registro de conductor existente.																		
Responsable	Desarrolladores.																		
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <i>InsertarDatosConductor_ConductorExistente</i>. Se ingresan los siguientes parámetros: <table> <tr><td>cedual</td><td>0992015890</td></tr> <tr><td>nombre1</td><td>Renan</td></tr> <tr><td>nombre2</td><td>Oswaldo</td></tr> <tr><td>apellido1</td><td>Perez</td></tr> <tr><td>apellido2</td><td>Balladares</td></tr> <tr><td>telefono</td><td>0992015890</td></tr> <tr><td>sexo</td><td>Masculino</td></tr> <tr><td>fecha_nac</td><td>2022, 02, 20</td></tr> <tr><td>fecha_contrato</td><td>2022, 02, 20</td></tr> </table> Comparar el mensaje de retorno del método de testeo con el método <i>guardarDatosConductor</i> de la clase <i>Adm_Conductor</i>. 	cedual	0992015890	nombre1	Renan	nombre2	Oswaldo	apellido1	Perez	apellido2	Balladares	telefono	0992015890	sexo	Masculino	fecha_nac	2022, 02, 20	fecha_contrato	2022, 02, 20
cedual	0992015890																		
nombre1	Renan																		
nombre2	Oswaldo																		
apellido1	Perez																		
apellido2	Balladares																		
telefono	0992015890																		
sexo	Masculino																		
fecha_nac	2022, 02, 20																		
fecha_contrato	2022, 02, 20																		
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. El número de cédula debe tener diez dígitos. 																		
Resultado esperado	El conductor NO será registrado en la aplicación.																		
Resultado obtenido	El conductor ya está registrad.																		
Estado	Fallido.																		
Observaciones	No se puede registrar un conductor con cédula ya existente.																		



The screenshot shows the Visual Studio IDE. The main window displays a C# test method named `InsertarDatosConductor_ConductorExistente` within the `SGAR_TEST` namespace. The method uses `Assert.AreEqual` to compare the return message of `Adm_Conductor.guardarDatosConductor` with the expected message `"¡CÉDULA YA EXISTE!"`. The Test Explorer window at the bottom shows a list of tests, including `SGAR_TEST (6)`, `Adm_Ambulancia_Test (4)`, `Adm_Conductor_Test (1)`, and `InsertarDatosConductor_ConductorExistente (1)`.

Código	CP-007
Caso de prueba	Registro de petición nueva
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba guardarPetición_PeticiónNueva. Se ingresan los siguientes parámetros: cantAmb 2 tipo_ambulancia 2 punto_Origen Clínica Kennedy punto_Destino Hospital del niño Comparar el mensaje de retorno del método de testeo con el método guardarPetición de la clase Adm_Petición.
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros.
Resultado esperado	Su petición fue ingresada correctamente.
Resultado obtenido	Su petición fue ingresada correctamente.
Estado	Exitoso
Observaciones	No se puede registrar una petición ya existente.

```

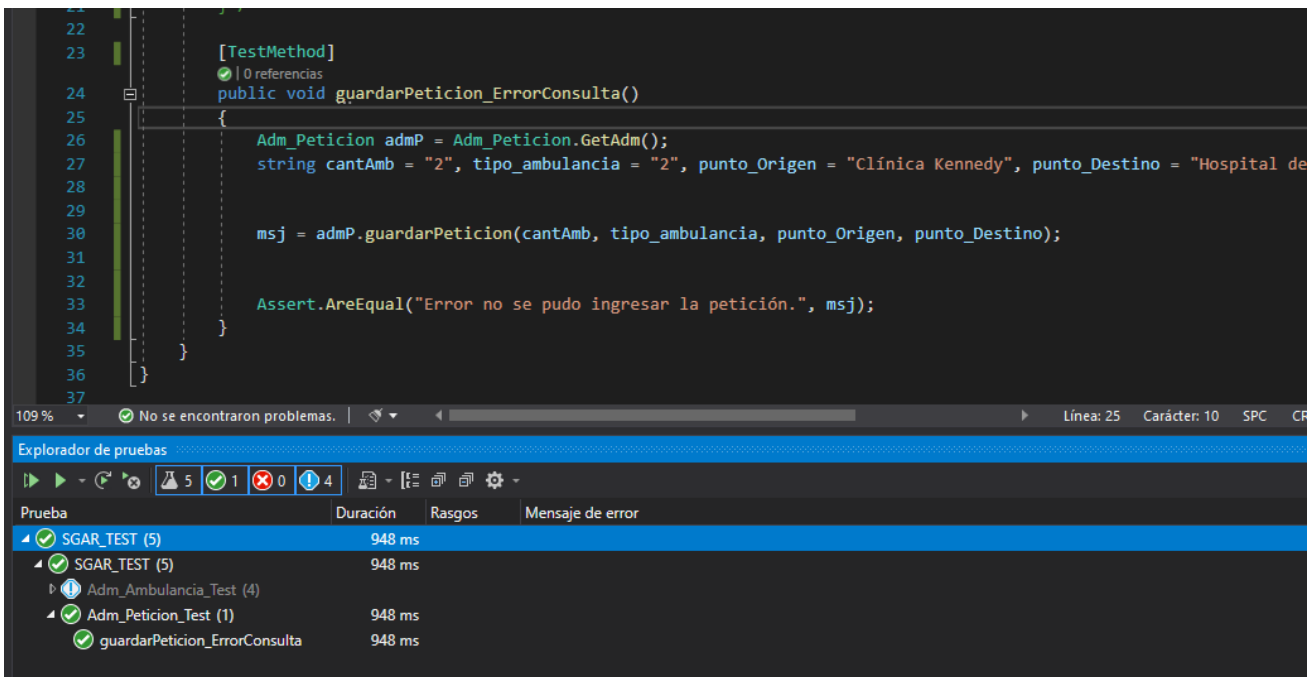
8      public class Adm_Peticion_Test
9      {
10         [TestMethod]
11         public void guardarPeticion_PeticionNueva()
12         {
13             Adm_Peticion admP = Adm_Peticion.GetAdm();
14             string cantAmb = "2", tipo_ambulancia = "2", punto_Origen = "Clínica Kennedy", punto_Destino = "Hospital del
15
16
17             msj = admP.guardarPeticion(cantAmb, tipo_ambulancia, punto_Origen, punto_Destino);
18
19
20             Assert.AreEqual("Su petición fue ingresada correctamente.", msj);
21         }
22     }
23     /*[TestMethod]
    
```

109 % No se encontraron problemas. Línea: 21 Carácter: 10 SPC CRLF

Explorador de pruebas

Prueba	Duración	Rasgos	Mensaje de error
SGAR_TEST (5)	992 ms		
SGAR_TEST (5)	992 ms		
Adm_Ambulancia_Test (4)			
Adm_Peticion_Test (1)	992 ms		
guardarPeticion_PeticionNueva	992 ms		

Código	CP-008
Caso de prueba	Registro de petición fallido
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba guardarPetición_ErrorConsulta. Se ingresan los siguientes parámetros: cantAmb 2 tipo_ambulancia 2 punto_Origen Clínica Kennedy punto_Destino Hospital del niño Comparar el mensaje de retorno del método de testeo con el método guardarPetición de la clase Adm_Petición.
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> No obtener el id del tipo de usuario dentro del método.
Resultado esperado	Error no se pudo ingresar la petición.
Resultado obtenido	Error no se pudo ingresar la petición.
Estado	Exitoso
Observaciones	No se puede registrar una petición ya existente.



```

22
23 [TestMethod]
24 public void guardarPetición_ErrorConsulta()
25 {
26     Adm_Petición admP = Adm_Petición.GetAdm();
27     string cantAmb = "2", tipo_ambulancia = "2", punto_Origen = "Clínica Kennedy", punto_Destino = "Hospital de
28
29     msj = admP.guardarPetición(cantAmb, tipo_ambulancia, punto_Origen, punto_Destino);
30
31     Assert.AreEqual("Error no se pudo ingresar la petición.", msj);
32 }
33
34
35
36
37
    
```

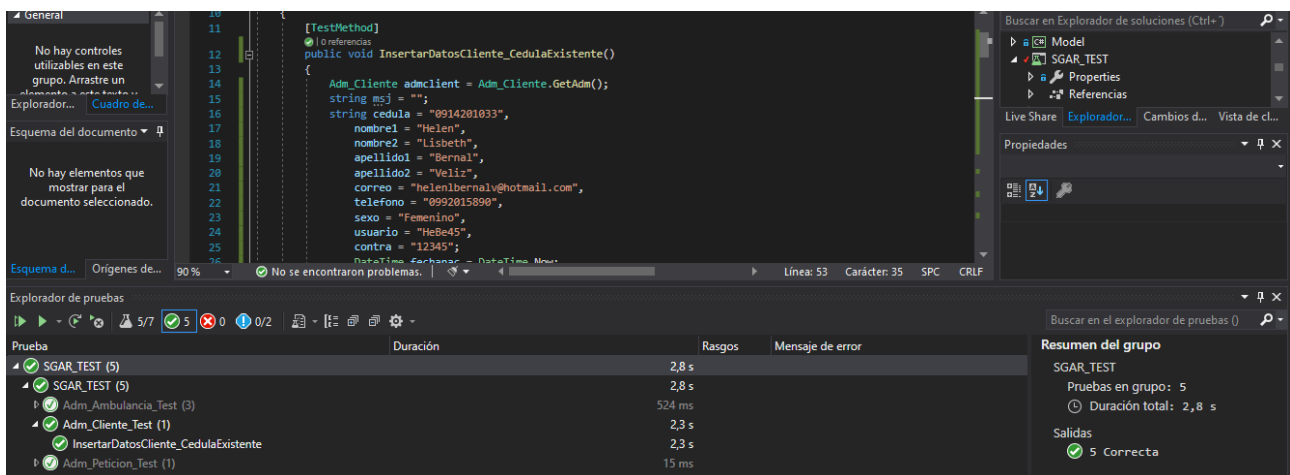
109 % No se encontraron problemas. Línea: 25 Carácter: 10 SPC

Explorador de pruebas

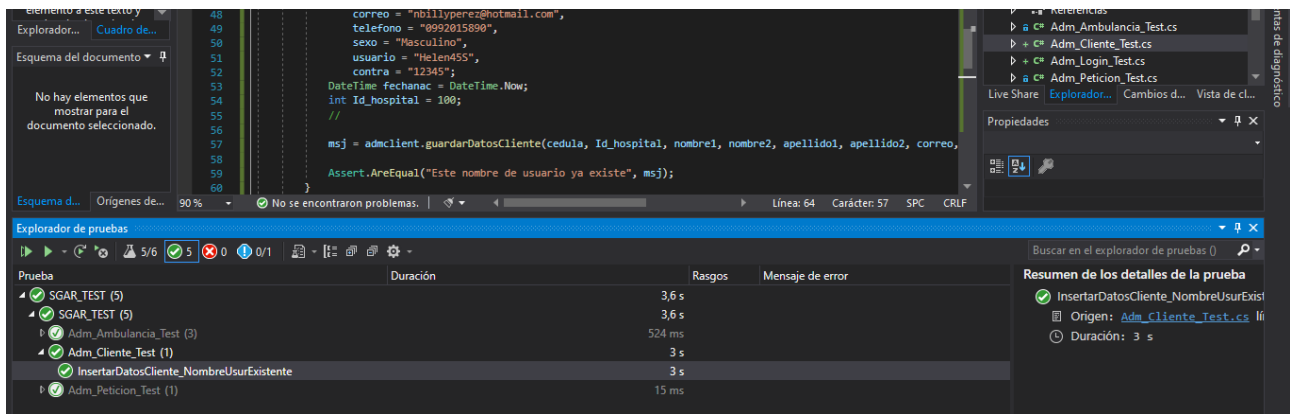
Prueba	Duración	Rasgos	Mensaje de error
SGAR_TEST (5)	948 ms		
SGAR_TEST (5)	948 ms		
Adm_Ambulancia_Test (4)			
Adm_Petición_Test (1)	948 ms		
guardarPetición_ErrorConsulta	948 ms		

Código	CP-009
---------------	--------

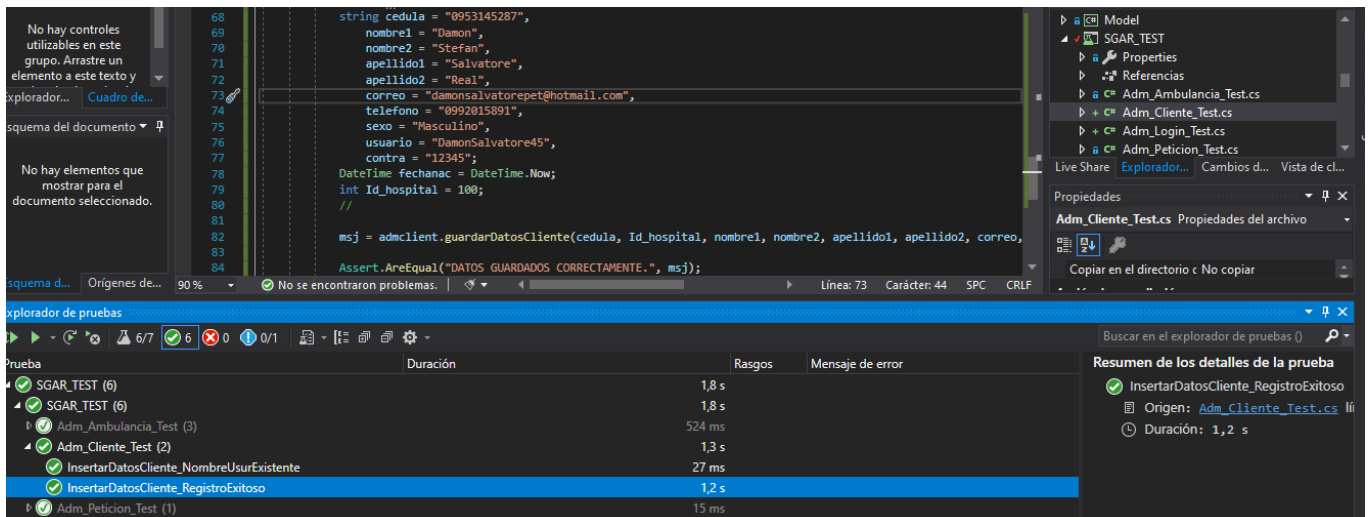
Caso de prueba	<i>Registro de Cliente Fallido: Cédula Existente</i>
Responsable	<i>Desarrolladores</i>
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <i>In InsertarDatosCliente_CedulaExistente</i>. Se ingresan los siguientes parámetros: cedula 0914201033 nombre1 Helen nombre2 Lisbeth apellido1 Bernal apellido2 Veliz correo helenlbernalv@hotmail.com telefono 0992015890 sexo Femenino fecha_nac 2022, 02, 20 id_hospital 100 usuario HeBe45 contraseña 12345 Comparar el mensaje de retorno del método de testeo con el método <i>guardarDatosCliente</i> de la clase <i>Adm_Cliente</i>.
Requisito previo	<i>Se debe cumplir con los siguientes requisitos:</i> <ul style="list-style-type: none"> Llenar todos los parámetros. La cédula debe tener 10 dígitos numéricos.
Resultado esperado	<i>El Cliente NO será registrado en la aplicación.</i>
Resultado obtenido	<i>Este número de cedula ya está registrado</i>
Estado	<i>Fallido</i>
Observaciones	<i>No se puede registrar un número de cedula ya existente en la base de datos.</i>



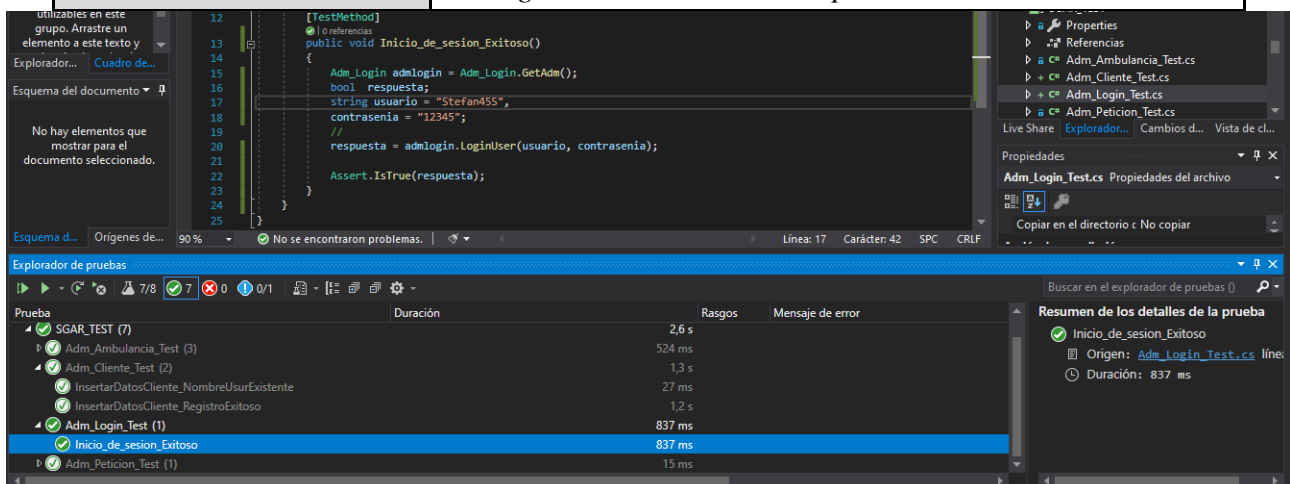
Código	CP-010
Caso de prueba	Registro de Cliente Fallido : Nombre de Usuario ya reservado
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba In InsertarDatosCliente_RegistroExitoso. Se ingresan los siguientes parámetros: cedula 0921537642 nombre1 Billy nombre2 Juan apellido1 Alvear apellido2 Perez correo nbillyperez@hotmail.com telefono 0992015890 sexo Masculino fecha_nac 2022, 02, 20 id_hospital 100 usuario Helen45S contraseña 12345 Comparar el mensaje de retorno del método de testeo con el método guardarDatosCliente de la clase Adm_Cliente.
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. La cédula debe tener 10 dígitos numéricos.
Resultado esperado	El Cliente NO será registrado en la aplicación.
Resultado obtenido	Este número de cedula ya está registrado
Estado	Fallido
Observaciones	No se puede registrar un número de cedula ya existente en la base de datos.



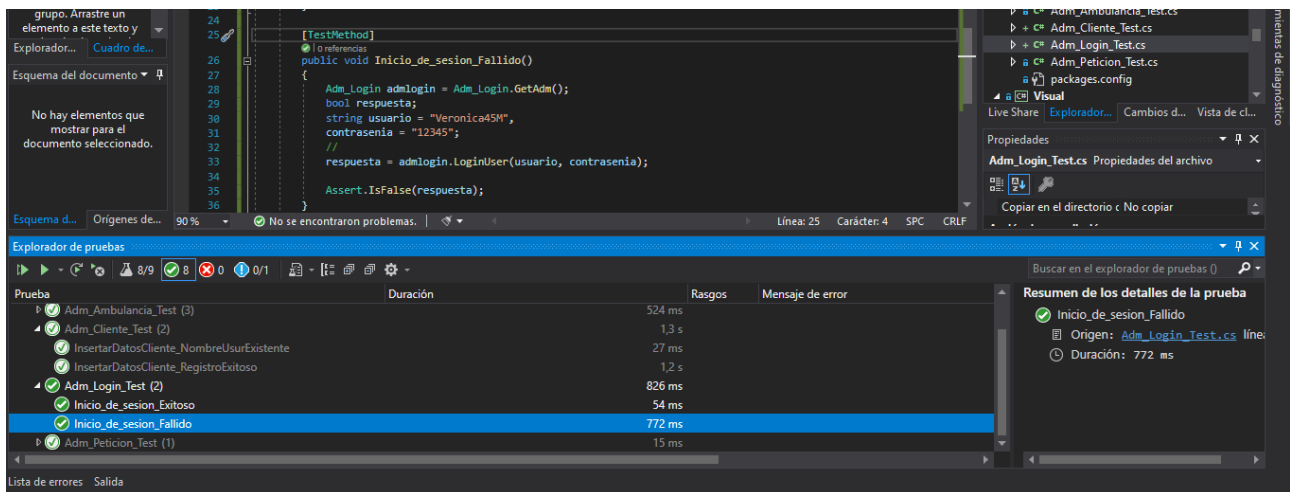
Código	CP-011
Caso de prueba	Registro de Cliente Exitoso
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <code>InInsertarDatosCliente_RegistroExitoso</code>. Se ingresan los siguientes parámetros: <ul style="list-style-type: none"> <code>cedula</code> 0953145287 <code>nombre1</code> Damon <code>nombre2</code> Stefan <code>apellido1</code> Salvatore <code>apellido2</code> Real <code>correo</code> damonsalvatorepet@hotmail.com <code>telefono</code> 0992015891 <code>sexo</code> Masculino <code>fecha_nac</code> 2022, 02, 20 <code>id_hospital</code> 100 <code>usuario</code> DamonSalvatore45 <code>contraseña</code> 12345 Comparar el mensaje de retorno del método de testeo con el método <code>guardarDatosCliente</code> de la clase <code>Adm_Cliente</code>.
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. La cédula debe tener 10 dígitos numéricos.
Resultado esperado	El Cliente SÍ será registrado en la aplicación.
Resultado obtenido	Datos Guardado Correctamente
Estado	Exitoso
Observaciones	Se pudo insertar el registro a la base de datos sin problemas.



Código	CP-012
Caso de prueba	Inicio de Sesión Exitoso
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba <i>Inicio_de_sesion_Exitoso</i>. Se ingresan los siguientes parámetros: <div style="display: flex; justify-content: space-between;"> usuario Stefan45S </div> <div style="display: flex; justify-content: space-between;"> contrasenia 12345 </div> Comparar el valor de retorno del método de testeo con el método <i>LoginUser</i> de la clase <i>Adm_Login</i>.
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros.
Resultado esperado	Que pueda ingresar al sistema con sus credenciales.
Resultado obtenido	True; Significa que tiene paso libre para ingresar al sistema
Estado	Exitoso
Observaciones	Se ingreso al sistema sin complicaciones



Código	CP-012
Caso de prueba	Inicio de Sesión Fallido
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba Inicio_de_sesion_Fallido. Se ingresan los siguientes parámetros: usuario Veronica45M contrasenia 12345 Comparar el mensaje de retorno del método de testeo con el método LoginUser de la clase Adm_Login.
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros.
Resultado esperado	Que no pueda ingresar al sistema.
Resultado obtenido	False; Significa que tiene paso libre para ingresar al sistema
Estado	Fallido
Observaciones	No se puede ingresar al sistema



Código	CP-013
Caso de prueba	Registro de secretaria existente
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba InsertarDatosSecretaria_SecretariaExistente. Se ingresan los siguientes parámetros: cedual 9876543210 nombre1 María nombre2 Alejandra

	<p> <i>apellido1</i> <i>García</i> <i>apellido2</i> <i>Mero</i> <i>correo</i> <i>maria.garcia@ug.edu.ec</i> <i>telefono</i> <i>0987542133</i> <i>sexo</i> <i>Femenino</i> <i>nombre_usuario</i> <i>MaAl</i> <i>contrasenia1</i> <i>123</i> <i>fecha_nac</i> <i>1999, 11, 16</i> <i>fecha_contrato</i> <i>2022, 01, 05</i> </p> <ul style="list-style-type: none"> • Comparar el mensaje de retorno del método de testeo con el método guardarDatosSecretaria de la clase Adm_Secretaria.
Requisito previo	<p>Se debe cumplir con los siguientes requisitos:</p> <ul style="list-style-type: none"> • Llenar todos los parámetros. • El número de cédula debe tener diez dígitos.
Resultado esperado	La secretaria NO será registrada en la aplicación.
Resultado obtenido	La secretaria ya está registrada.
Estado	Fallido
Observaciones	No se puede registrar un con cédula y nombre de usuario ya existente.

```

13 public void InsertarDatosSecretaria_SecretariaExistente () {
14     Adm_Secretaria admSecretaria = Adm_Secretaria.GetAdm ();
15     string
16         mensaje = "",
17         cedula = "9876543210",
18         nombre1 = "Maria",
19         nombre2 = "Alejandra",
20         apellido1 = "Garcia",
21         apellido2 = "Mero",
22         correo = "maria.garcia@ug.edu.ec",
23         telefono = "0987542133",
24         sexo = "Femenino",
25         nombre_usuario = "MaAl",
26         contrasenia1 = "123";
27     DateTime
28         fecha_nac = new DateTime (1999, 11, 16),
29         fecha_contrato = new DateTime (2022, 01, 05);
30     mensaje = admSecretaria.guardarDatosSecretaria (cedula, nombre1, nombre2, apellido1, apellido2, correo, telefono, sexo, fecha_nac, fecha_contrato);
31     Assert.AreEqual ("¡DATOS NO GUARDADOS!", mensaje);
32 }
33

```

Test Explorer

Test	Duration	Traits	Error Message
SGAR_TEST (8)	2.3 sec		
Adm_Ambulancia_Test (4)	2.3 sec		
Adm_Conductor_Test (2)	2.3 sec		
Adm_Peticion_Test (1)	2.3 sec		
Adm_Secretaria_Test (1)	2.3 sec		
InsertarDatosSecretaria_SecretariaExistente	2.3 sec		

Group Summary

SGAR_TEST

Tests in group: 8

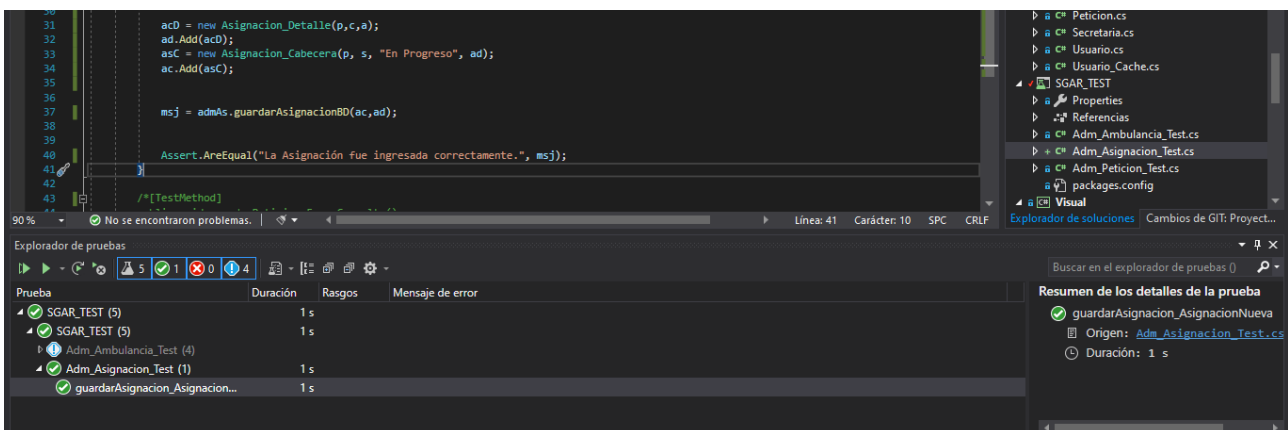
Total Duration: 2.3 sec

Outcomes

7 Not Run

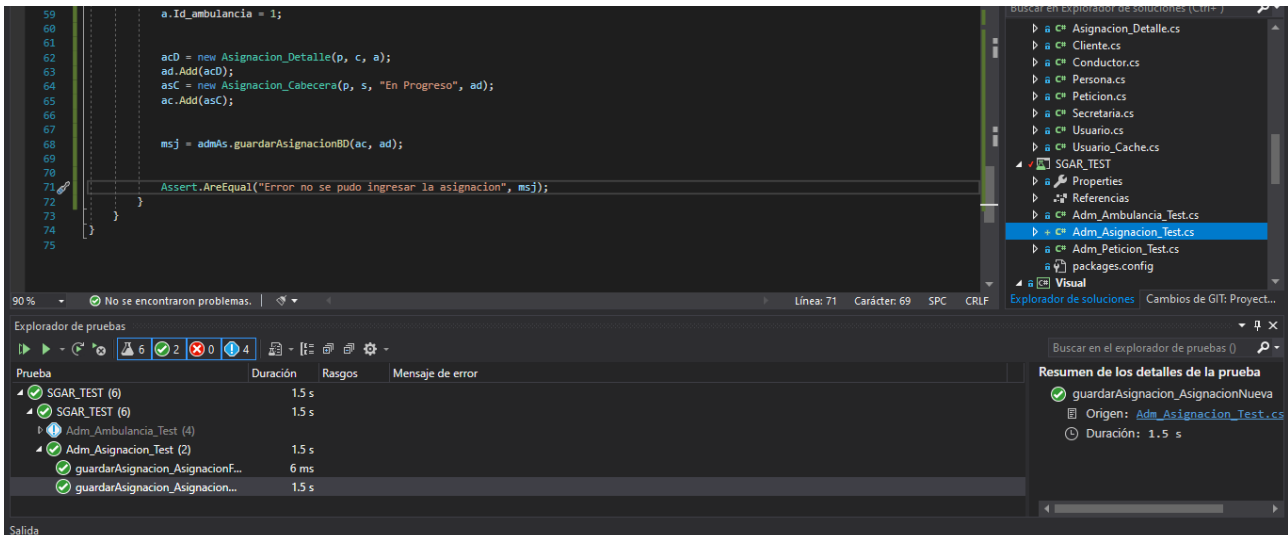
1 Passed

Código	CP-014
Caso de prueba	Registro de Asignacion nueva
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba guardarAsignacion_AsignacionNueva. Se ingresan los siguientes parámetros: $p.Id_peticion = 100;$ $s.Id_secretaria = 100;$ $c.Id_conductor = 100;$ $a.Id_ambulancia = 101;$ Comparar el mensaje de retorno del método de testeo con el método guardarAsignacionBD de la clase Adm_Asignacion
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. Los parámetro ingresados deben existir en la base de datos.
Resultado esperado	La Asignación fue ingresada correctamente.
Resultado obtenido	La Asignación fue ingresada correctamente.
Estado	Exitoso
Observaciones	No se puede registrar una asignación ya existente.



The screenshot shows the Visual Studio IDE with a unit test file open. The code in the test file includes creating objects for 'Asignacion_Detalle', 'Asignacion_Cabecera', and 'Asignacion', then calling 'guardarAsignacionBD' and asserting the result. The test runner at the bottom shows the test 'guardarAsignacion_AsignacionNueva' passing successfully. The right sidebar shows the 'Resumen de los detalles de la prueba' (Test Details Summary) for the same test.

Código	CP-015
Caso de prueba	Registro de Asignacion fallida
Responsable	Desarrolladores
Descripción de la prueba	<ul style="list-style-type: none"> Se utiliza el método de prueba guardarAsignacion_AsignacionNueva. Se ingresan los siguientes parámetros: <code>p.Id_peticion = 1;</code> <code>s.Id_secretaria = 1;</code> <code>c.Id_conductor = 1;</code> <code>a.Id_ambulancia = 1;</code> Comparar el mensaje de retorno del método de testeo con el método guardarAsignacionBD de la clase Adm_Asignacion
Requisito previo	Se debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> Llenar todos los parámetros. Los parámetro ingresados no deben existir en la base de datos.
Resultado esperado	Error no se pudo ingresar la asignación.
Resultado obtenido	Error no se pudo ingresar la asignación.
Estado	Fallido
Observaciones	No se puede registrar una asignación ya existente.



5.2. Pruebas de integración

<La prueba de integración busca verificar que los módulos o componentes que lo conforman funcionan correctamente en conjunto o integrados. Las pruebas de integración en los proyectos de desarrollo de software, no solo se presentan en la integración entre módulos de una misma aplicación de software sino entre varias aplicaciones de software.>

6. Criterios de entrada y de salida

Es de suma importancia tener definidos apropiadamente estos criterios y comunicarlos a toda la organización para que no haya ninguna duda sobre los mismos y que todos deben cumplir.

6.1. Criterio de entrada del plan de pruebas

Todo el paquete de código fuente está completo y ha sido revisado informalmente por el equipo de trabajo.

Tener un entorno de pruebas adecuado.

Todas las herramientas están preparadas para las pruebas.

6.2. Criterio de aceptación del plan de pruebas

100% de la ejecución de las pruebas especificadas al inicio de la iteración son completadas exitosamente.

100% de los defectos de prioridad crítica resueltos.

6.3. Criterio de suspensión y reanudación

Si se encuentra algún defecto que impacte seriamente en el progreso de las tareas de prueba, el desarrollador podrá elegir suspender las pruebas.

Esto aplica sólo para aquellos métodos que aún no estén disponibles o en su defecto, se encontraron defectos en la prueba.

7. Entregables

Dentro del proceso de pruebas se contemplan los siguientes entregables.

7.1. Protocolo de pruebas

El documento de protocolo de pruebas contiene los casos de pruebas que se realizarán y los resultados.

7.2. Informe de pruebas

El informe de pruebas contendrá la evaluación de resultados con los siguientes puntos:

- *Resumen de pruebas, el cual incluye la identificación del caso de prueba y el resultado de la ejecución.*
- *Acciones sugeridas.*
- *Análisis de Resultados.*

7.3. Reporte de observaciones

Los reportes de observaciones serán elaborados para todos los bugs que se encuentren durante las pruebas, se incluirá su severidad y su priorización para la solución a través del líder de proyecto o por parte de ingeniería de requerimientos si el documento de caso de uso debe ser actualizado. Cualquier modificación a los requerimientos debe ser hecha a través de solicitudes de cambios, las cuales aseguran que el cambio propuesto sea revisado antes de ser incorporado en el producto.

8. Ambiente de pruebas

Las versiones a testear de la aplicación se instalarán en el la maquina local de cada tester. Los requerimientos para las estaciones de trabajo del equipo de testing se detallan seguidamente.

8.1. Requerimientos base de hardware

Recurso	Cantidad	Nombre y tipo
Procesador	1	Pentium(R)Dual-Core de 2.50GHz
RAM	1	2GB

8.2. Requerimientos base de software en el ambiente de pruebas

Elemento de software	Versión	Tipo
.NET Framework	4.0.0	Framework
Microsoft Visual Studio 2019	16.11.3	IDE
Microsoft SQL Server Express Edition	15.0.2000.5	Server
Microsoft SQL Server Management Studio	18.10	SMBD

8.3. Herramientas de apoyo para la ejecución de pruebas

La siguiente tabla define las herramientas utilizadas para soportar el proceso de pruebas de este plan.

Elemento de software	Versión	Tipo
Word	3.0	Documento de Diseño Detallado de Software
Word	2.0	Documento de Alcance

9. Cronograma de trabajo

El cronograma se elaboró en Microsoft Excel.

	A	B	C	D
1	Nombre de tarea	Duración	Comienzo	Fin
2	Sistema de Gestión de Ambulancias Los Rápidos	60 días	17/1/2022	25/3/2022
3	Inicio del proyecto	1 día	17/1/2022	17/1/2022
4	Requerimientos			
5	Documento de Diseño Detallado del Software	24 días	18/1/2022	14/2/2022
6	Documento de Alcance	2 días	18/1/2022	20/1/2022
7	Codificación			
8	Módulo de Login	7 días	18/1/2022	24/1/2022
9	Módulo de Ambulancias: Insertar y Consultar	8 días	14/2/2022	21/2/2022
10	Módulo de Asignación Insertar	8 días	14/2/2022	21/2/2022
11	Módulo de Cliente: Insertar	8 días	14/2/2022	21/2/2022
12	Módulo de Conductor: Insertar	8 días	14/2/2022	21/2/2022
13	Módulo de Petición: Insertar	8 días	14/2/2022	21/2/2022
14	Módulo de Secretaria: Insertar	8 días	14/2/2022	21/2/2022
15	Pruebas			
16	Plan de Pruebas	4 días	18/2/2022	21/2/2022
17	Prueba unitaria: Adm_Login	4 días	18/2/2022	21/2/2022
18	Prueba unitaria: Adm_Ambulancias	4 días	18/2/2022	21/2/2022
19	Prueba unitaria: Adm_Asignación	4 días	18/2/2022	21/2/2022
20	Prueba unitaria: Adm_Cliente	4 días	18/2/2022	21/2/2022
21	Prueba unitaria: Adm_Conductor	4 días	18/2/2022	21/2/2022
22	Prueba unitaria: Adm_Petición	4 días	18/2/2022	21/2/2022
23	Prueba unitaria: Adm_Secretaria	4 días	18/2/2022	21/2/2022

10. Riesgos, dependencias, suposiciones y restricciones

10.1. Riesgos

Riesgo	Estrategia de mitigación	Contingencia
--------	--------------------------	--------------

Falta de tiempo	Aplicar al cronograma planteado inicialmente una adición de tiempo equivalente al 10% sobre el tiempo inicialmente pactado	Contratar más recursos
Atrasos en corrección de errores	Dar prioridad a errores funcionales y bloqueantes que impidan la continuación de las pruebas	Contratar más personal en desarrollo
Plan de Pruebas deficiente	Para el caso en que algún probador ejecute mal un plan de pruebas, se debe contar con un segundo probador que valide la correcta aplicación de las estrategias de pruebas, en puntos aleatorios.	Negociar previamente una adición de un 10% más de tiempo a la fase de pruebas inicialmente planeada

10.2. Dependencias

Dependencia con otras áreas	Impacto de la dependencia
Interdependencia entre proyectos.	Medio
Accesos (Permisos) a otros sistemas.	Medio

10.3. Suposiciones

Describe las suposiciones y su impacto.

Suposición a ser probada	Impacto de suposición incorrecta
El ambiente de pruebas debe contar con las especificaciones mínimas de hardware y software.	<ul style="list-style-type: none"> - Pruebas deficientes. - Caídas frecuentes. - Atrasos en el cronograma.
Requerimientos funcionales depurados y consistentes.	<ul style="list-style-type: none"> - Pruebas inconsistentes. - Atrasos en el cronograma. - Redefinición de requerimientos. - Cambios en otros módulos del sistema.
Datos de pruebas suficientes.	<ul style="list-style-type: none"> - Inversión de tiempo en la generación de datos de prueba. - Atraso en el cronograma. - Inconsistencia en las pruebas.


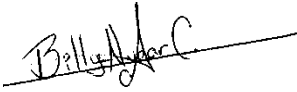
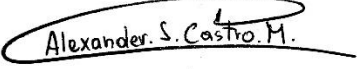
10.4. Restricciones

Restricciones	Impacto de la restricción
La fecha límite para la finalización del plan de pruebas es la segunda semana del mes de Marzo	<ul style="list-style-type: none"> - Iniciar a tiempo el paso a producción de la solución - Incumplimiento en la entrega del proyecto.
El máximo número de recursos disponibles para pruebas es de .	<ul style="list-style-type: none"> - Atrasos en el cronograma.

Utilización de herramientas libres y/o licenciadas.	<ul style="list-style-type: none"> - Incurrir en sanciones por efectos de utilización de software pirata - Herramientas con funcionalidades limitadas o nulas - Herramientas con fecha de caducidad
---	--

11.Aprobación

21/2/2022

<p>Elaborado por:</p>  <p>Nombre Líder de Testeo</p>	<p>Revisado por:</p> <p>Nombre Docente de la materia Construcción de Software</p>
<p>Aprobado por:</p>  <p>Nombre Líder del Proyecto</p>	<p>Aprobado por:</p>  <p>Nombre Director de la Unidad de Tecnologías de la Información y Comunicaciones de la Empresa Los Rápidos S.A.</p>