SISTEMA DE CONTROL DE ACCESO PARA CONJUNTOS RESIDENCIALES

SAFEWARE COMPANY

David Andrés Hernández Triana

Michael Stick Mejia Perdomo

Luisa Fernanda Alarcón Castañeda

Harol Daniel Naranjo Yunez

ADSI 2067459

Servicio Nacional De Aprendizaje - SENA

Bogotá D.C.

2021

Tabla de contenido

1. Pruebas Unitarias

1.1. Vista - Persona

1.1.1. Obtención de la vista

Planteamiento de la prueba: Verificar la correcta obtención de la vista “Persona”

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void IndexViewNamePerson()
{
    PersonsController controller = new PersonsController();
    var result = controller.Index() as ViewResult;
    Assert.AreEqual("Index", result.ViewName);

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.1.2. Obtención del listado

Planteamiento de la prueba: Verificar si la obtención de los datos obtenidos en la vista “Persona” son iguales a los datos almacenados en el modelo en la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]

public void IndexViewNamePersona()
{
    PersonaController controller = new PersonaController();
    var result = controller.Index() as ViewResult;
    Assert.AreEqual("Index", result.ViewName);

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.1.3. Inserción de datos

Planteamiento de la prueba: Verificar la correcta inserción de datos en la base de datos.

Desarrollo de la prueba:

```
[TestMethod]
0 referencias
public void CreatePersona()
{
    PersonaController controller = new PersonaController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Create(new PersonaDTO
        {
            Nombres = "Luisa",
            Apellidos = "Alarcon",
            TI = 1,
            NumeroIdentificacion = "258896321"
        }));

    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.1.4. Actualización de datos

Planteamiento de la prueba: Verificar que la identificación del registro a editar exista previamente en la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void EditPersona()
{
    PersonaController controller = new PersonaController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Edit(2));
    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.1.5. Eliminación de registro

Planteamiento de la prueba: Verificar que el registro seleccionado se ha eliminado tanto en la vista como de la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void DeletePersona()
{
    SCContext db = new SCContext();
    PersonaDTO c = new PersonaDTO();
    try
    {
        PersonaController controller = new PersonaController();
        ActionResult result = controller.Delete(1) as ActionResult;
        db.Entry(c).State = EntityState.Deleted;
        db.SaveChanges();
        Assert.IsNotNull(result);
    }
    catch (Exception e)
    {
    }
```

Resultado de la prueba: La prueba se completó correctamente.

1.2. Vista - Registro

1.2.1. Obtención de la vista

Planteamiento de la prueba: Verificar la correcta obtención de la vista “Registro”.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void IndexViewNameRegistro()
{
    RegistroController controller = new RegistroController();
    var result = controller.Index() as ViewResult;
    Assert.AreEqual("Index", result.ViewName);

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.2.2. Obtención del listado

Planteamiento de la prueba: Verificar si la obtención de los datos obtenidos en la vista “Registro” son iguales a los datos almacenados en la base de datos.

Desarrollo de la prueba:

```
[TestMethod]
0 referencias
public void IndexListRegistro()
{
    RegistroController controller = new RegistroController();
    var result = controller.Index() as ViewResult;
    Assert.AreEqual(typeof(List<RegistroDto>), result.Model.GetType());

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.2.3. Inserción de datos

Planteamiento de la prueba: Verificar la correcta inserción de datos en el la base de datos.

Desarrollo de la prueba:

```
[TestMethod]
0 referencias
public void CreateRegistro()
{
    RegistroController controller = new RegistroController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Create(new RegistroDto
        {
            NombreTipoPersonaId = "Visitante",
            FechaRegistro = Convert.ToDateTime("01/05/2021"),
            NombreOperacionId = "Salida",
        }));

    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.2.4. Actualización de datos

Planteamiento de la prueba: Verificar que la identificación del registro a editar exista previamente en la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void EditRegistro()
{
    RegistroController controller = new RegistroController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Edit(2));
    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.2.5. Eliminación de registros

Planteamiento de la prueba: Verificar que el registro seleccionado se ha eliminado tanto en la vista como en la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void DeleteRegistro()
{
    SCContext db = new SCContext();
    RegistroDto c = new RegistroDto();
    try
    {
        RegistroController controller = new RegistroController();
        ActionResult result = controller.Delete(1) as ActionResult;
        db.Entry(c).State = EntityState.Deleted;
        db.SaveChanges();
        Assert.IsNotNull(result);
    }
    catch (Exception e)
    {
    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.3. Vista - Vehiculo

1.3.1. Obtención de la vista

Planteamiento de la prueba: Verificar la correcta obtención de la vista “Vehículo”.

Desarrollo de la prueba:

```
[TestMethod()]
public void Index()
{
    // Arrange inicializa los objetos y establece los valores de los datos que vamos a utilizar en el Test que lo contiene.
    VehiculoController controller = new VehiculoController();

    // Act realiza la llamada al método a probar con los parámetros preparados para tal fin.
    var result = controller.Index() as ViewResult;

    // Assert comprueba que el método de pruebas ejecutado se comporta tal y como teníamos previsto que lo hiciera.
    Assert.AreEqual("Index", result.ViewName);

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.3.2. Obtención del listado

Planteamiento de la prueba: Verificar si la obtención de los datos obtenidos en la vista “Vehículo” son iguales a los datos almacenados en la base de datos.

Desarrollo de la prueba:

```
[TestMethod]
0 referencias
public void IndexListVehiculo()
{
    VehiculoController controller = new VehiculoController();
    var result = controller.Index() as ViewResult;
    Assert.AreEqual(typeof(List<VehiculoDTO>), result.Model.GetType());

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.3.3. Inserción de datos

Planteamiento de la prueba: Verificar la correcta inserción de datos en la base de datos.

Desarrollo de la prueba:

```
[TestMethod]
0 referencias
public void CreateVehiculo()
{
    VehiculoController controller = new VehiculoController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Create(new VehiculoDTO
        {
            TipoVehiculoId = 1,
            Placa = "FGT589",
            Descripcion = "BMW 2021",
            TipoVehiculo = "Automovil",
        }));

    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.3.4. Actualización de datos

Planteamiento de la prueba: Verificar que la identificación del registro a editar exista previamente en la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void EditVehiculo()
{
    VehiculoController controller = new VehiculoController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Edit(1));
    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.3.5. Eliminación de registros

Planteamiento de la prueba: Verificar que el registro seleccionado se ha eliminado tanto en la vista como en la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void DeleteVehiculo()
{
    SCContext db = new SCContext();
    VehiculoDTO c = new VehiculoDTO();
    try
    {
        VehiculoController controller = new VehiculoController();
        ActionResult result = controller.Delete(1) as ActionResult;
        db.Entry(c).State = EntityState.Deleted;
        db.SaveChanges();
        Assert.IsNotNull(result);
    }
    catch (Exception e)
    {
    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.4. Vista - Visita

1.4.1. Obtención de la vista

Planteamiento de la prueba: Verificar la correcta obtención de la vista “Visita”.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void IndexViewNameVisita()
{
    VisitaController controller = new VisitaController();
    var result = controller.Index() as ViewResult;
    Assert.AreEqual("Index", result.ViewName);

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.4.2. Obtención del listado

Planteamiento de la prueba: Verificar si la obtención de los datos obtenidos en la vista “Visita” son iguales a los datos almacenados en la base de datos.

Desarrollo de la prueba:

```
[TestMethod]
0 referencias
public void IndexListVisita()
{
    VisitaController controller = new VisitaController();
    var result = controller.Index() as ViewResult;
    Assert.AreEqual(typeof(List<VisitaDTO>), result.Model.GetType());

}
```

Resultado de la prueba: La prueba se completó correctamente.

1.4.3. Inserción de datos

Planteamiento de la prueba: Verificar la correcta inserción de datos en la base de datos.

Desarrollo de la prueba:

```
[TestMethod]
0 referencias
public void CreateVisita()
{
    VisitaController controller = new VisitaController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Create(new VisitaDTO
        {
            Torre = 1,
            Apartamento = 3,
            Telefono = "1050254788",
        }));

    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.4.4. Actualización de datos

Planteamiento de la prueba: Verificar que la identificación del registro a editar exista previamente en la base de datos.

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void EditVisita()
{
    VisitaController controller = new VisitaController();
    using (var ts = new SCContext())
    {
        Assert.IsNotNull(controller.Edit(2));
    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

1.4.5. Eliminación de registros

Planteamiento de la prueba: Verificar que el registro seleccionado se ha eliminado tanto en la vista como en el modelo relacional (base de datos).

Desarrollo de la prueba:

```
[TestMethod()]
0 referencias
public void DeleteVisita()
{
    SCContext db = new SCContext();
    VisitaDTO c = new VisitaDTO();
    try
    {
        VisitaController controller = new VisitaController();
        ActionResult result = controller.Delete(1) as ActionResult;
        db.Entry(c).State = EntityState.Deleted;
        db.SaveChanges();
        Assert.IsNotNull(result);
    }
    catch (Exception e)
    {
    }
}
```

Resultado de la prueba: La prueba se completó correctamente.

2. Pruebas Manuales

2.1. Vista - Persona

2.1.1. Obtención

Prueba: Obtención - La vista “Persona” presenta los registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Persona” de la barra lateral.
2. Comprobar que la Uri sea https://localhost:44384/Persona

Criterios de aceptación: La vista “Persona” retorna correctamente los valores almacenados en la base de datos.

Respuestas:

1. (+) Se retornan correctamente los valores y se muestran en pantalla.
2. (-) Los valores no se retornan correctamente y la vista se queda en blanco.

Resultado: La prueba se cumplió correctamente.

2.1.2. Inserción

Prueba: Inserción - Crear nuevos registros en la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Persona” de la barra lateral.
2. Dar click en el botón “Agregar Persona”.
3. Comprobar que la Uri sea https://localhost:44384/Persona/Create
4. Diligenciar los datos.
5. Dar click en el botón “Crear Persona”.

Criterios de aceptación:

1. Los datos se ingresan de manera correcta cumpliendo con cada una de las validaciones predefinidas.
2. Se crea el nuevo registro.

Respuestas:

1. (+) Se almacenan los datos y se guarda el registro correctamente.
2. (-) Los datos están mal diligenciados y no se crea el registro.

Resultado: La prueba se cumplió correctamente

2.1.3. Actualización

Prueba: Actualización - Actualizar registros en la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Persona” de la barra lateral.
2. Dar click en el botón con el icono de edición del registro que se desee editar.
3. Comprobar que la Uri sea https://localhost:44384/Persona /Edit/IdentificadorRegistro
4. Diligenciar los nuevos datos.
5. Dar click en el botón “Editar”.

Criterios de aceptación:

1. Los datos se ingresan de manera correcta.
2. Se actualiza el registro existente.

Respuestas:

1. (+) Se almacenan los datos y se guarda la actualización correctamente.
2. (-) Los datos están mal diligenciados y no se actualiza el registro.

Resultado: La prueba se cumplió correctamente

2.1.4. Eliminación

Prueba: Eliminación - Eliminar registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Persona” de la barra lateral.
2. Dar click en el botón con el icono de eliminación del registro que se desee eliminar.

Criterios de aceptación: Se elimina el registro de la base de datos y se retorna la vista con los registros preexistentes.

Respuestas:

1. (+) Se elimina de manera correcta el registro.
2. (-) No se elimina el registro.

Resultado: La prueba se cumplió correctamente.

2.2. Vista - Registro

2.2.1. Obtención

Prueba: Obtención - La vista “Registro” presenta los registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Registro” de la barra lateral.
2. Comprobar que la Uri sea https://localhost:44384/Registro

Criterios de aceptación: La vista retorna correctamente los valores almacenados en la base de datos

Respuestas:

1. (+) Se retornan correctamente los valores y se muestran en pantalla.
2. (-) Los valores no se retornan correctamente y la vista se queda en blanco.

Resultado: La prueba se cumplió correctamente.

2.2.2. Inserción

Prueba: Inserción - Crear nuevos registros en la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Registro” de la barra lateral
2. Dar click en el botón “Agregar Registro”
3. Comprobar que la Uri sea https://localhost:44384/Registro/Create
4. Diligenciar los datos.
5. Dar click en el botón “Crear Registro”

Criterios de aceptación:

1. Los datos se ingresan de manera correcta cumpliendo con cada una de las validaciones predefinidas.
2. Se crea el nuevo registro.

Respuestas:

1. (+) Se almacenan los datos y se guarda el registro correctamente.
2. (-) Los datos están mal diligenciados y no se crea el registro.

Resultado: La prueba se cumplió correctamente.

2.2.3. Actualización

Prueba: Actualización - Actualizar registros en la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Registro” de la barra lateral.
2. Dar click en el botón con el icono de edición del registro que se desee editar.
3. Comprobar que la Uri sea https://localhost:44384/Registro /Edit/IdentificadorRegistro
4. Diligenciar los nuevos datos.
5. Dar click en el botón “Editar”.

Criterios de aceptación:

1. Los datos se ingresan de manera correcta.
2. Se actualiza el registro existente.

Respuestas:

1. (+) Se almacenan los datos y se guarda la actualización correctamente.
2. (-) Los datos están mal diligenciados y no se actualiza el registro.

Resultado: La prueba se cumplió correctamente.

2.2.4. Eliminación

Prueba: Eliminación - Eliminar registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Registro” de la barra lateral.
2. Dar click en el botón con el icono de eliminación del registro que se desee eliminar.

Criterios de aceptación: Se elimina el registro de la base de datos y se retorna la vista con los registros preexistentes.

Respuestas:

1. (+) Se elimina de manera correcta el registro.
2. (-) No se elimina el registro.

Resultado: La prueba se cumplió correctamente.

2.3. Vista - Vehiculo

2.3.1. Obtención

Prueba: Obtención – La vista "Vehículo" presenta los registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado "Vehículo" de la barra lateral.
2. Comprobar que la Uri sea https://localhost:44384/Vehículo

Criterios de aceptación: La vista retorna correctamente los valores almacenados en la base de datos

Respuestas:

1. (+) Se retornan correctamente los valores y se muestran en pantalla
2. (-) Los valores no se retornan correctamente y la vista se queda en blanco

Resultado: La prueba se cumplió correctamente

2.3.2. Inserción

Prueba: Inserción - Crear nuevos registros en la base de datos.

Pasos a seguir:

1. Dar click en el apartado Vehículo de la barra lateral.
2. Dar click en el botón "Agregar Vehículo".
3. Comprobar que la Uri sea https://localhost:44384/Vehículo/Create
4. Diligenciar los datos.
5. Dar click en el botón "Crear Vehiculo".

Criterios de aceptación:

1. Los datos se ingresan de manera correcta cumpliendo con cada una de las validaciones predefinidas.
2. Se crea el nuevo registro.

Respuestas:

1. (+) Se almacenan los datos y se guarda el registro correctamente.
2. (-) Los datos están mal diligenciados y no se crea el registro.

Resultado: La prueba se cumplió correctamente.

2.3.3. Actualización

Prueba: Actualización - Actualizar registros en la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Vehículo” de la barra lateral.
2. Dar click en el botón con el icono de edición del registro que se desee editar.
3. Comprobar que la Uri sea https://localhost:44384/ Vehículo /Edit/IdentificadorRegistro
4. Diligenciar los nuevos datos.
5. Dar click en el botón “Editar”.

Criterios de aceptación:

1. Los datos se ingresan de manera correcta.
2. Se actualiza el registro existente.

Respuestas:

1. (+) Se almacenan los datos y se guarda la actualización correctamente
2. (-) Los datos están mal diligenciados y no se actualiza el registro

Resultado: La prueba se cumplió correctamente

2.3.4. Eliminación

Prueba: Eliminación - Eliminar registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Vehículo” de la barra lateral.
2. Dar click en el botón con el icono de eliminación del registro que se desee eliminar.

Criterios de aceptación: Se elimina el registro de la base de datos y se retorna la vista con los registros preexistentes.

Respuestas:

1. (+) Se elimina de manera correcta el registro.
2. (-) No se elimina el registro.

Resultado: La prueba se cumplió correctamente.

2.4. Vista - Visita

2.4.1. Obtención

Prueba: Obtención - La vista “Visita” presenta los registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Visita” de la barra lateral.
2. Comprobar que la Uri sea https://localhost:44384/Visita

Criterios de aceptación: La vista retorna correctamente los valores almacenados en la base de datos.

Respuestas:

1. (+) Se retornan correctamente los valores y se muestran en pantalla.
2. (-) Los valores no se retornan correctamente y la vista se queda en blanco.

Resultado: La prueba se cumplió correctamente.

2.4.2. Inserción

Prueba: Inserción - Crear nuevos registros en la base de datos

Pasos a seguir:

1. Dar click en el apartado “Visita” de la barra lateral
2. Dar click en el botón “Agregar Visita”
3. Comprobar que la Uri sea https://localhost:44384/Visita/Create
4. Diligenciar los datos.
5. Dar click en el botón “Crear Visita”.

Criterios de aceptación:

1. Los datos se ingresan de manera correcta cumpliendo con cada una de las validaciones predefinidas
2. Se crea el nuevo registro.

Respuestas:

1. (+) Se almacenan los datos y se guarda el registro correctamente.
2. (-) Los datos están mal diligenciados y no se crea el registro.

Resultado: La prueba se cumplió correctamente.

2.4.3. Actualización

Prueba: Actualización - Actualizar registros en la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Visita” de la barra lateral.
2. Dar click en el botón con el icono de edición del registro que se desee editar.
3. Comprobar que la Uri sea https://localhost:44384/Visita/Edit/IdentificadorRegistro
4. Diligenciar los nuevos datos
5. Dar click en el botón “Editar”.

Criterios de aceptación:

1. Los datos se ingresan de manera correcta.
2. Se actualiza el registro existente.

Respuestas:

1. (+) Se almacenan los datos y se guarda la actualización correctamente.
2. (-) Los datos están mal diligenciados y no se actualiza el registro.

Resultado: La prueba se cumplió correctamente**.**

2.4.4. Eliminación

Prueba: Eliminación - Eliminar registros de la base de datos.

Pasos a seguir:

1. Dar click en el apartado “Visita” de la barra lateral.
2. Dar click en el botón con el icono de eliminación del registro que se desee eliminar.

Criterios de aceptación: Se elimina el registro de la base de datos y se retorna la vista con los registros preexistentes.

Respuestas:

1. (+) Se elimina de manera correcta el registro.
2. (-) No se elimina el registro.

Resultado: La prueba se cumplió correctamente.