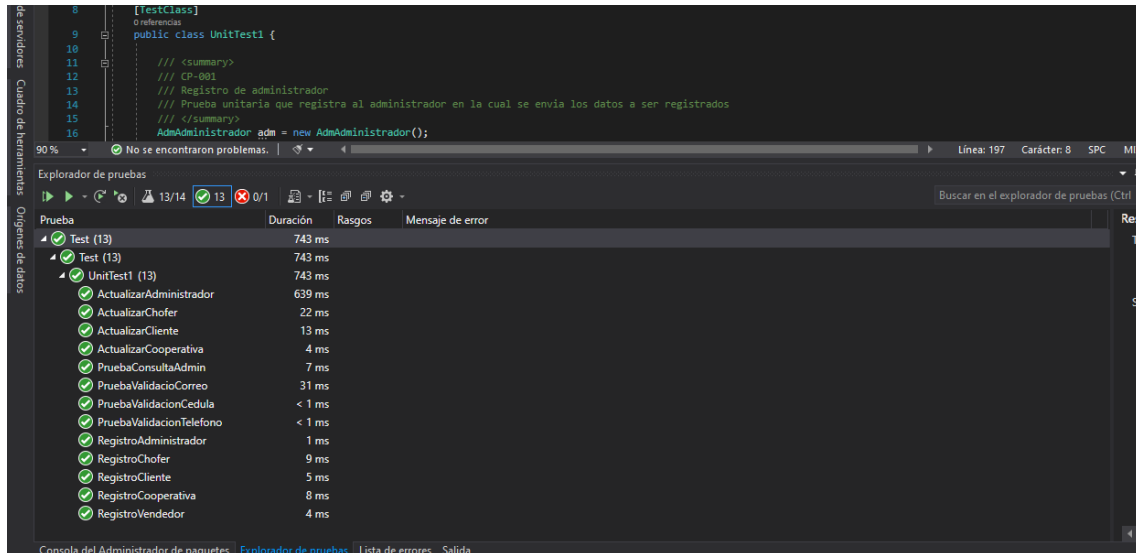


# Requerimientos del Sistema

## PRUEBAS UNITARIAS



## PRUEBA UNITARIA #1



## PRUEBAS UNITARIAS #2



### PRUEBAS UNITARIAS #3

```
/// <summary>
/// CP-003
/// Registro de cooperativa
/// Prueba unitaria que registra la cooperativa la cual envia los datos a ser registrados
/// </summary>
AdmCooperativa coop = new AdmCooperativa();
[TestMethod]
public void RegistroCooperativa()
{
    CredencialUsuario credencial = new CredencialUsuario("ecuatoriano@gmail.com", "ecuatori", 3);
    Cooperativa cooperativa = new Cooperativa("0923737737", "Manuel Reyes", "Ecuatoriano Pulman", "0938377636", credencial);
    var resultadoEsperado = coop.RegistrarCooperativa(cooperativa);
    var resultadoObtenido = true;
    Assert.AreEqual(resultadoEsperado, resultadoObtenido);
}
```

### PRUEBAS UNITARIAS #4

```
/// <summary>
/// CP-004
/// Prueba unitaria de registro de cliente en la cual se envia los datos a ser registrados
/// </summary>
AdmClienteProcedimiento cliente = new AdmClienteProcedimiento();
[TestMethod]
public void RegistroCliente()
{
    CredencialUsuario credencial = new CredencialUsuario("jose@gmail.com", "jose123456", 4);
    Cliente client = new Cliente("1263722998", "Jose Mendez", "Masculino", "0983646533", credencial);
    var resultadoEsperado = cliente.RegistrarCliente(client);
    var resultadoObtenido = true;
    Assert.AreEqual(resultadoEsperado, resultadoObtenido);
}
```

### PRUEBAS UNITARIAS #5

```
/// <summary>
/// CP-005
/// Registro de vendedor
/// Prueba unitaria que registra el vendedor en la cual se envia los datos a ser registrados
/// </summary>
AdmVendedor adv = new AdmVendedor();
[TestMethod]
public void RegistroVendedor()
{
    CredencialUsuario credencial = new CredencialUsuario("maria@hotmail.com", "maria12345", 5);
    Vendedor vendedor = new Vendedor("1938373374", "Maria Hurtado", "Femenino", "0984473847", 586.98, credencial);
    var resultadoEsperado = adv.RegistrarVendedor(vendedor);
    var resultadoObtenido = true;
    Assert.AreEqual(resultadoEsperado, resultadoObtenido);
}
```

### PRUEBAS UNITARIAS #6

```
/// <summary>
/// CP-006
/// Actualización de administrador
/// Prueba unitaria que se actualizo al administrador en la cual se envia los datos a ser registrados
/// </summary>
[TestMethod]
public void ActualizarAdministrador()
{
    CredencialUsuario credencial = new CredencialUsuario("deiber@gmail.com", "Deiber12", 1);
    Administrador admin = new Administrador("0968387363", "William Encalada", "Masculino", "0987466344", "12345@88. @", 920.45, credencial);
    var resultadoEsperado = adm.RegistrarAdministrador(admin);
    var resultadoObtenido = true;
    Assert.AreEqual(resultadoEsperado, resultadoObtenido);
}
```

## PRUEBAS UNITARIAS #7

```
/// <summary>
/// CP-007
/// Actualizacion de chofer
/// Prueba unitaria que se actualizo el chofer en la cual se envia los datos a ser registrados
/// </summary>
[TestMethod]
public void ActualizarChofer()
{
    CredencialUsuario credencial = new CredencialUsuario("manuel@gmail.com", "manuel1", 2);
    Chofer chofer = new Chofer("1292938774", "Manuel Mendieta", "Masculino", "0976453763", "1274738840", 845.50, credencial);
    var resultadoEsperado = adf.RegistrarChofer(chofer);
    var resultadoObtenido = true;
    Assert.AreEqual(resultadoEsperado, resultadoObtenido);
}
```

## PRUEBAS UNITARIAS #8

```
/// <summary>
/// CP-008
/// Actualizacion de cooperativa
/// Prueba unitaria que se actualizo la cooperativa la cual envia los datos a ser registrados
/// </summary>
[TestMethod]
public void ActualizarCooperativa()
{
    CredencialUsuario credencial = new CredencialUsuario("ecuatoriano@gmail.com", "ecuatoriano12", 3);
    Cooperativa cooperativa = new Cooperativa("0923737737", "Manuel Reyes", "Ecuatoriano Pulman", "0938377636", credencial);
    var resultadoEsperado = coop.RegistrarCooperativa(cooperativa);
    var resultadoObtenido = true;
    Assert.AreEqual(resultadoEsperado, resultadoObtenido);
}
```

## PRUEBAS UNITARIAS #9

```
/// <summary>
/// CP-009
/// </summary>
[TestMethod]
public void ActualizarCliente()
{
    CredencialUsuario credencial = new CredencialUsuario("jose@gmail.com", "jose12", 4);
    Cliente client = new Cliente("1263722998", "Jose Mendez", "Masculino", "0983646533", credencial);
    var resultadoEsperado = cliente.RegistrarCliente(client);
    var resultadoObtenido = true;
    Assert.AreEqual(resultadoEsperado, resultadoObtenido);
}
```

## PRUEBAS UNITARIAS #10

```
/// <summary>
/// CP-010
/// PruebaValidacionCedula
/// </summary>
[TestMethod]
public void PruebaValidacionCedula()
{
    Validacion validacion = new Validacion();
    bool result = validacion.ValidarCedula("0302527742");
    Assert.AreEqual(true, result);
}
```

## PRUEBAS UNITARIAS #11

```
/// <summary>
/// CP-011
/// Validacion de telefono
/// </summary>
[TestMethod]
0 | 0 referencias
public void PruebaValidacionTelefono()
{
    Validacion validacion = new Validacion();
    bool result = validacion.ValidarTelefono("0991887965");
    Assert.AreEqual(true, result);
}
```

## PRUEBAS UNITARIAS #12

```
/// <summary>
/// CP-012
/// Validacion de correo
/// </summary>
[TestMethod]
0 | 0 referencias
public void PruebaValidacioCorreo()
{
    Validacion validacion = new Validacion();
    bool result = validacion.validarEmail("Kevin7520@hotmail.com");
    Assert.AreEqual(true, result);
}
/// </summary>
```

## PRUEBAS UNITARIAS #13

```
/// <summary>
/// CP-013
/// Actualizacion de cooperativa
/// Prueba unitaria consultar admin
/// </summary>
[TestMethod]
0 | 0 referencias
public void PruebaConsultaAdmin()
{
    int id = 1;
    AdmAdministrador servidor = new AdmAdministrador();
    List<Administrador> result = servidor.ConsultaAdmin(id);
    Assert.AreEqual(result, result);
}
```