

Alumna: Cathaysa Navarro Benítez.

## 1.- Errores encontrados:

1 error: todos los miembros tienen un nombre poco descriptivo (líneas 16, 17 y 18) renombramiento de los miembros.

```
11
12 3 referencias | 0 cambios | 0 autores, 0 cambios
13 public class comprobadorDePasswordCNB2324
14 {
15     public string pwd;
16     private bool mins;
17     private bool mays;
18     private bool nums;
19     private bool length;
```

2 error: Una espacio entre las declaraciones (línea 15) supresión del espacio.

```
11
12 3 referencias | 0 cambios | 0 autores, 0 cambios
13 public class comprobadorDePasswordCNB2324
14 {
15     public string pwd;
16     private bool mins;
17     private bool mays;
18     private bool nums;
19     private bool length;
```

3 error: Nombre del parámetro inconcluso P y el nombre del método debe empezar por mayúsculas (línea 26) renombramiento del parámetro por la palabra “palabra” y reescritura del nombre del método

```
26 1 referencia | 0 cambios | 0 autores, 0 cambios
27 public int test(string p)
28 {
29     pwd = p;
```

4 error: Código sin llaves en los if (línea 31 y 33) introducción de las llaves de apertura y cierre.

```
29
30 if (pwd==null || pwd.Length==0)
31     return -1; // Si la contraseña es nula o vacía, devolvemos un código de error
32
33 if (pwd.Length < 6)
34     return 0; // No tiene la longitud mínima, error
35
```

5 error: las variables tienen nombres poco descriptivos (líneas 37, 38 y 70) renombramiento de las variables.

```
36
37 bool mins = false;
38 bool mays = false;
39 bool nums = false;
40 bool length = false;
41
```

```
70 int f=0;
71 if (Minusculas) f++;
72 if (Mayusculas) f++;
73 if (Numeros) f++;
74 if (Longitud) f++;
75
76 return f;
77 }
```

6 error: la declaración de las variables está en mitad del método (líneas 27 hasta 30) ordenación del método y colocación en la parte superior del método.

```
25
26
27     bool mins = false;
28     bool mays = false;
29     bool nums = false;
30     bool length = false;
31
```

7 error : parte del código con mal espaciado. (línea 41 hasta la 55) a los elementos foreach se les hace una sangría y se introduce dentro de unas llaves de apertura y cierre del if.

```

37 if (password.Length > 12) Longitud = true;
38
39 // Recorremos la cadena buscando minúsculas, mayúsculas y números
40 //
41 foreach (char c in password)
42 {
43     if (char.IsLower(c))
44     {
45         Minusculas=true;
46     }
47 }
48 foreach (char c in password)
49 {
50     if (char.IsUpper(c))
51     {
52         Mayusculas=true;
53     }
54 }
55 foreach (char c in password)
56 {
57     if (char.IsDigit(c))
58     {
59         Numeros=true;
60     }
61 }
62

```

## 2 diseño de pruebas.

Agregamos el archivo de pruebas.

Pruebas a realizar.

1er test:

Si se introduce una contraseña vacía, dará un error de nulo.

```
[TestMethod]
public void ValidarContrasenya1()
{
    int PasswordEsperado = -1;
    string PasswordIntroducido = "";

    comprobadorDePasswordCNB2324 contrasenya = new
comprobadorDePasswordCNB2324();

    int ResultadoEsperado = contrasenya.Test(PasswordIntroducido);

    Assert.AreEqual(PasswordEsperado, ResultadoEsperado);
}
```

TRAS LA REFACTORIZACIÓN QUEDA ASÍ:

```
public void ValidarContrasenya1()
{
    try
    {
        string PasswordIntroducido = "";
        ComprobadorDePasswordCNB2324 contrasenya = new
ComprobadorDePasswordCNB2324();

        int ResultadoEsperado = contrasenya.Test(PasswordIntroducido);
    }
    catch (ArgumentOutOfRangeException exception)
    {
        // assert
        StringAssert.Contains(exception.Message,
ComprobadorDePasswordCNB2324.ERROR_CONTRASENIA_NULA);
        return;
    }
    Assert.Fail("Error. Se debía haber producido una excepción.");
}
```

2º test:

Si se introduce una contraseña con una longitud minima, error.

[TestMethod]

```
public void ValidarContrasenia2()
{
    int PasswordEsperado = 0;
    string PasswordIntroducido = "abc";

    comprobadorDePasswordCNB2324 contrasenia = new
comprobadorDePasswordCNB2324();

    int ResultadoEsperado = contrasenia.Test(PasswordIntroducido);

    Assert.AreEqual(PasswordEsperado, ResultadoEsperado);
}
```

TRAS LA REFACTORIZACIÓN QUEDA ASÍ:

```
public void ValidarContrasenia2()
{
    try
    {
        string PasswordIntroducido = "abc";
        ComprobadorDePasswordCNB2324 contrasenia = new
ComprobadorDePasswordCNB2324();

        int ResultadoEsperado = contrasenia.Test(PasswordIntroducido);
    }
    catch (ArgumentOutOfRangeException exception)
    {
        StringAssert.Contains(exception.Message,
ComprobadorDePasswordCNB2324.ERROR_CONTRASENYA_INVALIDA);
        return;
    }
    Assert.Fail("Error. Se debía haber producido una excepción.");
}
```

### 3er test:

Si la contraseña tiene la longitud mínima y tiene minúsculas, mayúsculas, números y un carácter, la contraseña es válida.

```
[TestMethod]
public void ValidarContrasenya3()
{
    int PasswordEsperado = 4;
    string PasswordIntroducido = "C0ntr@s3ñ@S3gur@";

    comprobadorDePasswordCNB2324 contrasenya = new
comprobadorDePasswordCNB2324();

    int ResultadoEsperado = contrasenya.Test(PasswordIntroducido);

    Assert.AreEqual(PasswordEsperado, ResultadoEsperado);
}
```

## 3 Refactorización

Primera refactorización:

Crear un get y set para el miembro password. (línea 5) a través del propio visual studio, clicando sobre la palabra con el botón derecho, “Acciones rápidas y refactorización→ encapsular campo y usar la propiedad.

```
4      {
5          public string password;
```

Segunda refactorización (de manera manual) creación de constantes para devolver los mensajes de error.

```
public const string ERROR_CONTRASENYA_NULA = "Contraseña nula";
public const string ERROR_CONTRASENYA_INVALIDA = "No tiene la longitud mínima";
```

Y posterior cambio en los mensajes de error

Tercera refactorización (de manera manual) ordenación y colocación de los miembros del constructor.

```
4 referencias | 0 cambios | 0 autores, 0 cambios
public ComprobadorDePasswordCNB2324()
{
    minusculas = mayusculas = numeros = longitud = false;
}
```

```
4 referencias | 0 cambios | 0 autores, 0 cambios
23 public ComprobadorDePasswordCNB2324()
24 {
25     this.minusculas = false;
26     this.mayusculas = false;
27     this.numeros = false;
28     this.longitud = false;
29
30 }
```

Cuarta refactorización creación de los get y set de los miembros a través del propio visual studio, clicando sobre la palabra con el botón derecho, “Acciones rápidas y refactorización→ encapsular campo y usar la propiedad.

```
private bool minusculas;  
private bool mayusculas;  
private bool numeros;  
private bool longitud;
```