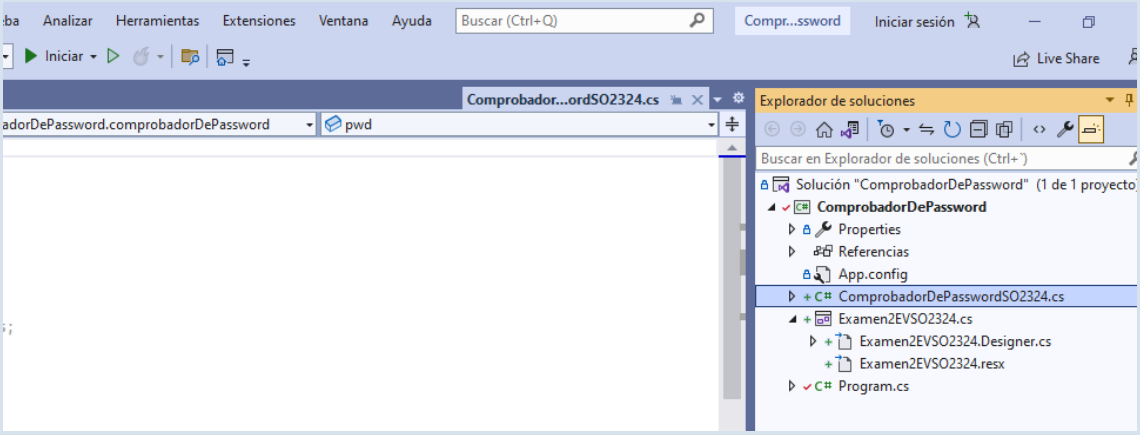
DOSSIER
EXAMEN SO2324

Cloneado el repositorio:



1. **(DOSSIER - 1 pt)** Encontrar cinco errores de normas de estilo en el fichero *comprobadorDePassword.cs,* indicando número de línea, error encontrado y solución. NO SE PUEDE REPETIR ERROR, <u>DEBEN SER DIFERENTES</u>.

Errores:

A.
```
public string pwd;

private bool mins;
private bool mays;
private bool nums;
private bool length;
```
:

nombres no descriptivos de las variables. Corrigiendo:

```
public string Password;

private bool Minusculas;
private bool Mayusculas;
private bool Numeros;
private bool Longitud;
```

B.
```
public comprobadorDePassword()
{
    minusculas = mays = numeros = length = false;
}
```

Asignación de los valores combinado:

Corrección:

```
//minusculas = mays = numeros = length = false;
Minusculas = false;
Mayusculas = false;
Numeros = false;
Longitud = false;
```

C.
```
if (password==null || password.Length<=0)
```

Falta de los espacios. Corregido:

```
if (password == null || password.Length <= 0)
```

D.
```
if (Password.Length > 12) length = true;
```

Faltan claves:

```
if (Password.Length > 12)
{
    length = true;
}
```

E.
```
// Calculamos el nivel de fortaleza
// 4: muy fuerte
// 3: fuerte
// 2: normal
// 1: débil
int f=0;
if (mins) f++;
if (mays) f++;
if (nums) f++;
if (length) f++;

return f;
```

Las condiciones y el return no han puesto correctamente. Corrección:

```
// Calculamos el nivel de fortaleza
// 4: muy fuerte
// 3: fuerte
// 2: normal
// 1: débil
int fuerzaDeContrasenya = 0;
if (mins)
{
    fuerzaDeContrasenya++;
}
if (mays)
{
    fuerzaDeContrasenya++;
}
if (nums)
{
    fuerzaDeContrasenya++;
}
if (length)
{
    fuerzaDeContrasenya++;
}
return fuerzaDeContrasenya;
```

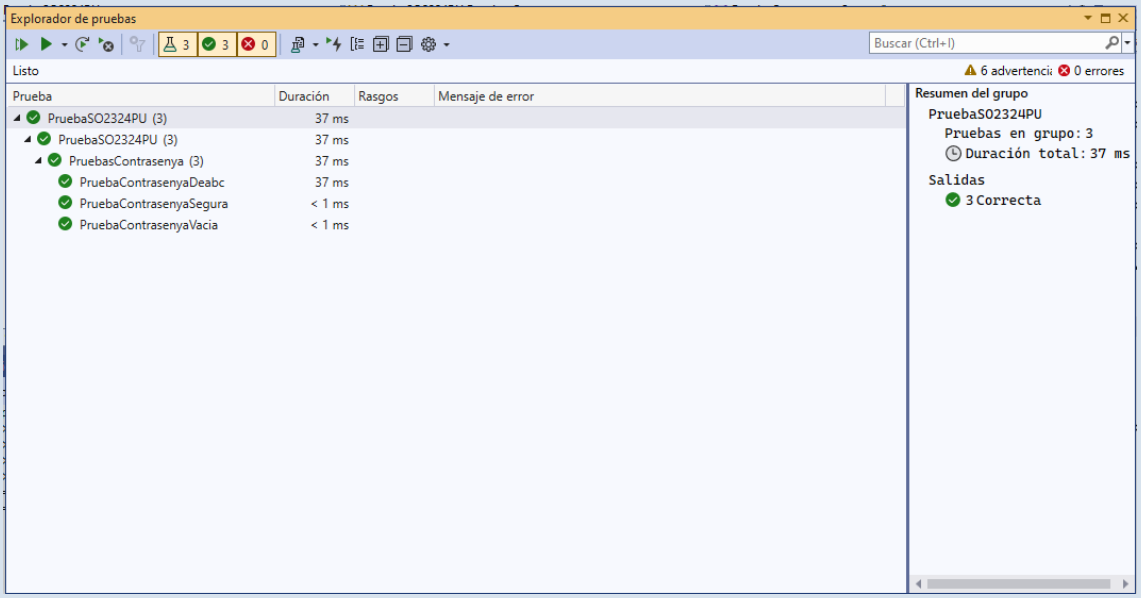
2. **(DOSSIER - 1 pt)** Realizar el diseño de pruebas (caja negra) para el método *comprobadorDePassword.test()* .

Clases de equivalencia para el diseño de pruebas (caja negra):

I. Contraseña tiene valor null o longitud no positiva: error “La contraseña está vacía”. Return -1
II. Contraseña tiene menos de 6 símbolos: error “Longitud insuficiente”. Return 0
III. Contraseña tiene 6 o más símbolos y tiene fuerzaDeContrasenya de 1. Return 1
IV. Contraseña tiene 6 o más símbolos y tiene fuerzaDeContrasenya de 2. Return 2
V. Contraseña tiene 6 o más símbolos y tiene fuerzaDeContrasenya de 3. Return 3
VI. Contraseña tiene 6 o más símbolos y tiene fuerzaDeContrasenya de 4: Return 4

3. **(COMMIT - 1,5 pt)** Crear los métdos de prueba que correspondan a los siguientes valores de prueba: "", "abc", "C0ntr@s3ñ@S3gur@"

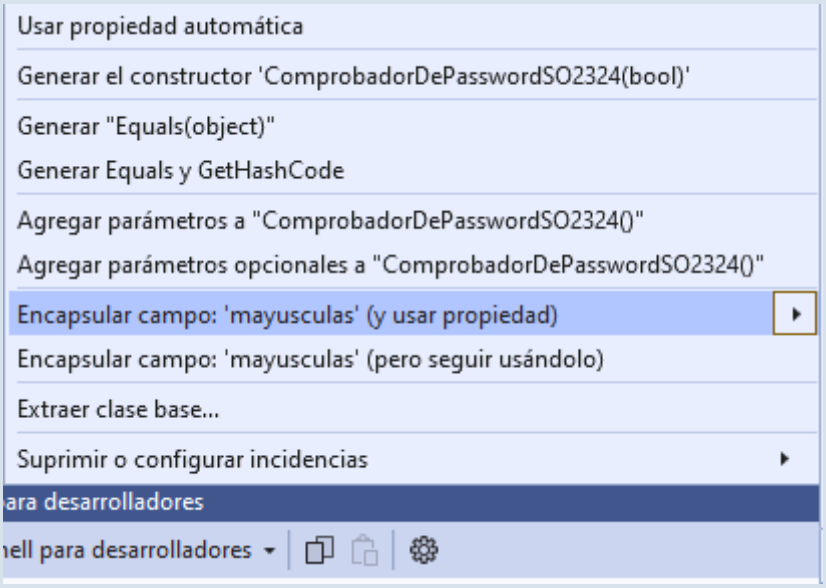
Ejecutando:



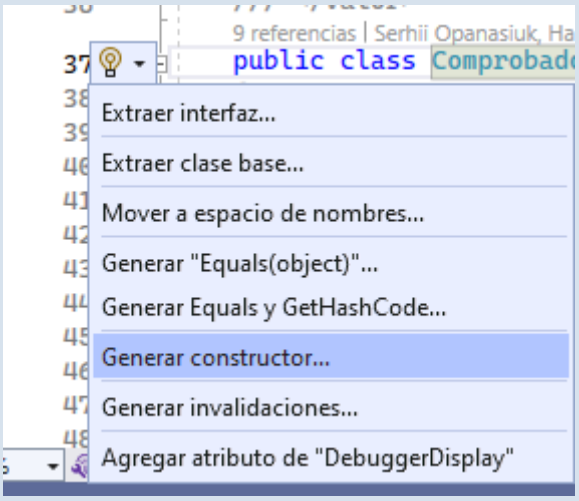
4. **(DOSSIER+COMMIT - 1,5 pt)** Si existen, detectar y aplicar al menos tres patrones de refactorización DISTINTOS en el fichero *comprobadorDePassword*.cs, indicando el patrón que se aplica y, si es posible aplicarlo con Visual Studio, la opción que se usa.

Patrónes:

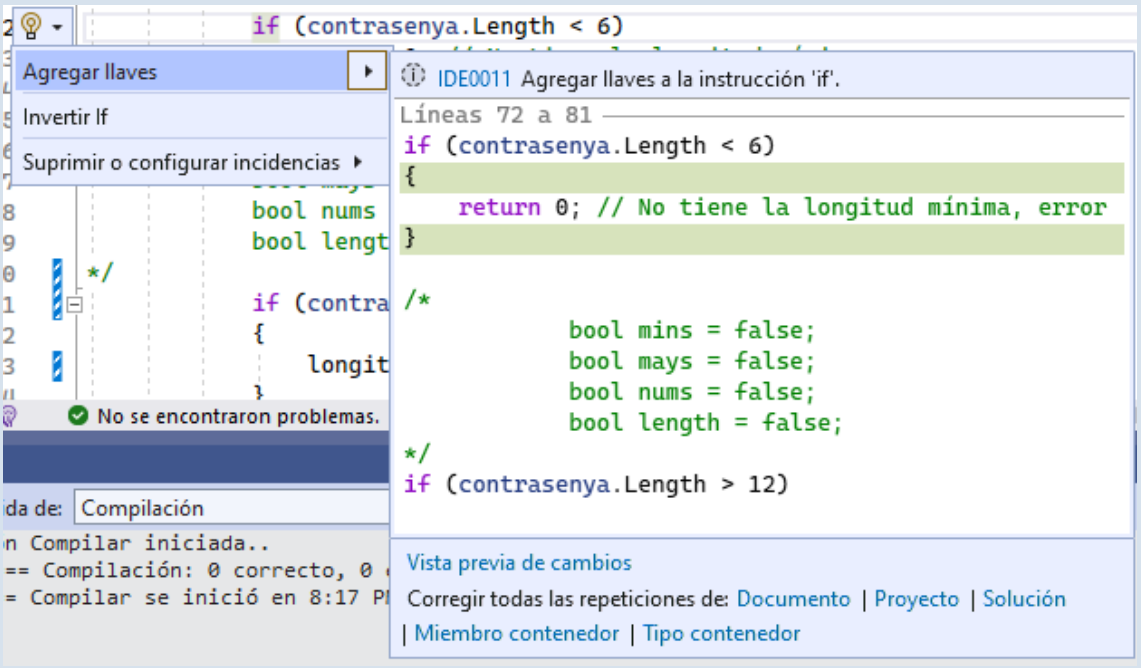
He encapsulado los campos de las propiedades de la clase:



He generado la función-constructora para crear la instancia de la clase:



He agregado las llaves a la instrucción If:



5. **(DOSSIER + COMMIT - 1,5 pt)** En base a los cambios realizados en el punto (4), modificar los métodos de prueba creados en el punto (3).

Hecho (me faltó tiempo para notar detalles)

6. **(COMMIT - 1 pt)** Documentar el fichero *comprobadorDePassword.cs*. Sólo se debe documentar los constructores y los métodos públicos.

Hecho