

DESENROTLLAMENT D'INTERFÍCIES 2N DAM

SA7: Proves del programari.



Professor: Iván Martos



PROVES UNITARIES AMB VISUAL STUDIO 2022 I C#



Les pràctiques es lliuraran seguint la següent pauta: El nom de l'activitat o pràctica a enviar serà el nombre de la pràctica (per exemple: A43, seguit de guió baix i del vostre nom i cognoms). Per exemple: **A43_AntonioGómez.**

L'activitat s'ha de lliurar en la data establerta i no s'acceptaran els lliuraments fora del termini establert.
Les pràctiques han de ser originals i no s'admeten còpies de companys/es, Internet..

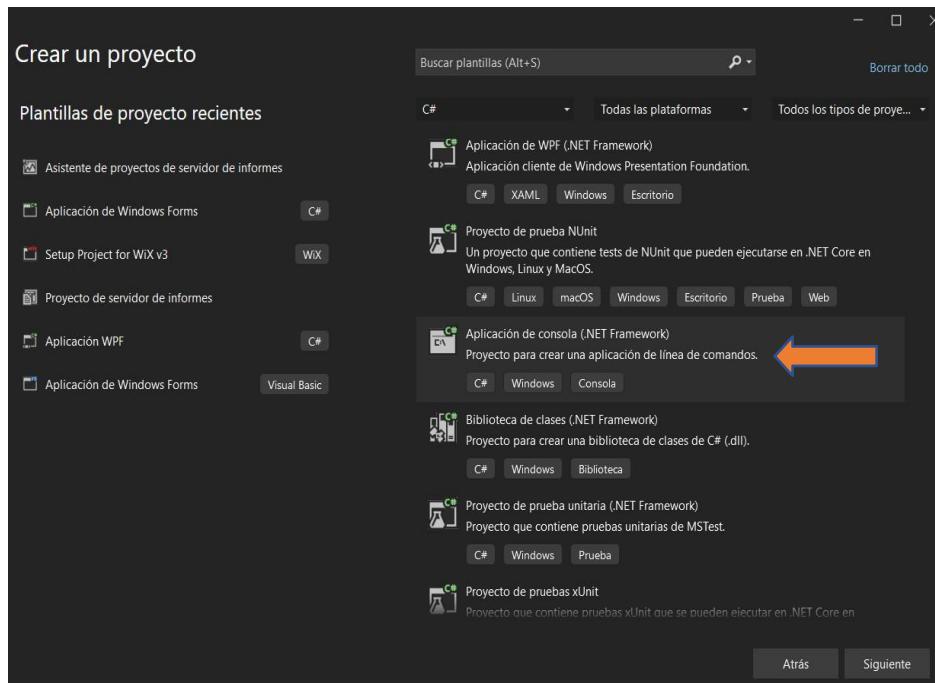
Objectius de l'activitat

1. Conèixer els diferents tipus de proves en el desenvolupament d'aplicacions multiplataforma.
2. Aprendre a considerar un protocol de proves en les fases de desenvolupament del programari.
3. Aprendre a fer proves unitàries en entorns C# amb Visual Studio.

Temporalització

L'activitat està prevista en una estimació de dues sessions lectives de 55 minuts cada sessió.

1. En aquesta pràctica guiada anem a veure un exemple senzill de creació de proves unitàries amb Visual Studio 2022 i el llenguatge C#. Obrirem el Visual Studio 2022 o 2026 i crearem un nou projecte però en aquest cas serà una **aplicació de consola** (.NET Framework) amb el llenguatge de programació C#.



A continuació veiem l'estructura d'una classe amb un **mètode main** que serà similar al que estem acostumats a veure en altres llenguatges de programació com Java.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ProvesTest
8 {
9     public class Program
10    {
11        static void Main(string[] args)
12        {
13        }
14    }
15 }
16

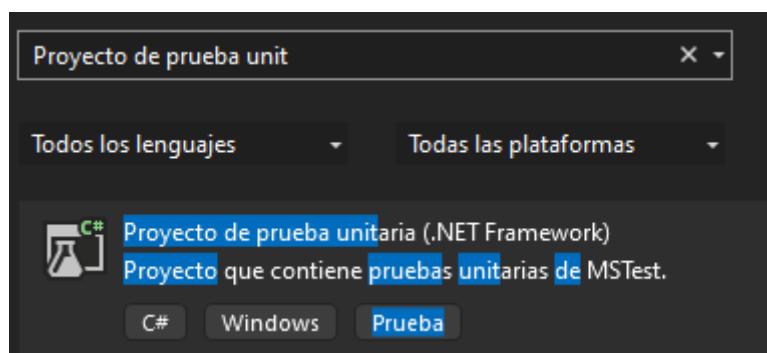
```

A continuació crearem un mètode suma que s'encarregue de sumar dos números enters.

```
2 referencias | 2/2 pasando
public static int Sumar(int num1, int num2)
{
    if (num1 < 0 || num2 < 0)
    {
        throw new ArgumentException("Ambos números deben ser positivos.");
    }

    return num1 + num2;
}
```

Ara el que farem és afegir un nou projecte dintre de la solució anterior. El nou projecte ens permetrà fer el testeig del mètode anterior de sumar. El nou projecte serà de tipus: **“Projecte de prova unitaria (.NET Framework) de MSTest”**.

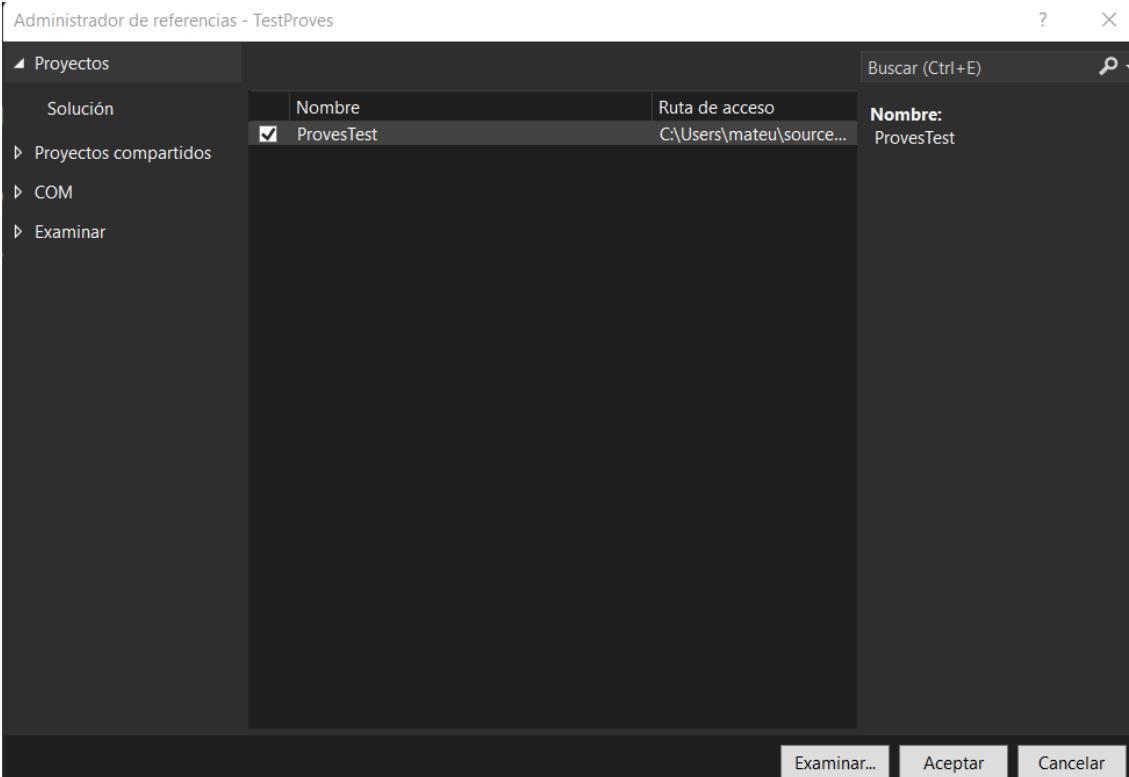


Ara veiem com serà l'aspecte del projecte que utilitzarem per a testejar. És important que el mètode que utilitzarem per a testejar tinga el **[TestMethod]**.

```
namespace TestProves
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
        }
    }
}
```

A continuació anem a afegir la referència a l'altre projecte per tindre visibilitat i poder fer ús dels mètodes que volem fer el test.

Afegim la referència a l'altre projecte.



Ara creem la prova com podem observar a la següent imatge. Destaquem l'ús d'**Assert** amb el mètode **AreEqual** per validar si el resultat obtingut és el que estem esperant.

```
using ProvesTest;

namespace TestProves
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
            int suma = ProvesTest.Program.suma(3, 6);

            Assert.AreEqual(9, suma);
        }
    }
}
```

Obrim **l'explorador de proves** i executem per comprovar si el resultat és correcte i dona el resultat esperat.

Evidentment si sumen 3+6 el resultat hauria de ser 9. Recordeu que és un exemple bàsic i no hem fet cap validació addicional i estem fent ús de valors enters.

```

using ProvesTest;
namespace TestProves
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
            int suma = ProvesTest.Program.suma(3, 6);
            Assert.AreEqual(9, suma);
        }
    }
}

```

No se encontraron problemas.

Prueba	Duración	Rasgos	Mensaje de error
TestProves (1)	2 s		
TestProves (1)	2 s		
UnitTest1 (1)	2 s		

L'exemple és molt bàsic però podem provar un resultat incorrecte i comprovar que no passa el test. En aquest cas, estem esperant un resultat de 12 per a la suma de 3 i 6, cosa que és incorrecte.

```

using ProvesTest;
namespace TestProves
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
            int suma = ProvesTest.Program.suma(3, 6);
            Assert.AreEqual(12, suma);
        }
    }
}

```

Veiem l'**explorar de proves** i comprovem que amb els valors indicats no és correcte el resultat i per tant no passa el test.

```

using ProvesTest;
namespace TestProves
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestMethod1()
        {
            int suma = ProvesTest.Program.suma(3, 6);
            Assert.AreEqual(12, suma);
        }
    }
}

No se encontraron problemas.
Explorador de pruebas
Serie de pruebas finalizada: 1 pruebas (Superadas: 0; Con errores: 1; Omitidas: 0) ejecutadas en 130 ms
Prueba | Duración | Rasgos | Mensaje de error
-----|-----|-----|-----
TestProves (1) | 59 ms | | 
TestProves (1) | 59 ms | | 
UnitTest1 (1) | 59 ms | | 

```

2. A continuació heu de omplir i millorar aquesta pràctica:

- Heu de crear un mètode resta, un mètode multiplicar i un mètode dividir. Tots quatre mètodes han de treballar amb números enters.
- Heu de crear proves unitàries pera el mètode suma que al menys comproven:
 - Suma de dos números enters positius.
 - Realitza una prova unitària d'excepció de suma número negatiu.
- Heu de crear proves unitàries pera el mètode resta.
 - Resta de dos números enters positius.
 - Resta de de dos números enters, un positiu i un altre negatiu.
 - Resta de de dos números enters, els dos negatius.
- Heu de crear proves unitàries pera el mètode multiplicar que al menys comproven:
 - Multiplicació de dos números enters positius.
 - Multiplicació de de dos números enters, un positiu i un altre negatiu.
 - Multiplicació de de dos números enters, els dos negatius.
- Heu de crear proves unitàries pera el mètode dividir que al menys comproven:
 - Divisió de dos números enters positius.
 - Divisió de de dos números enters, un positiu i un altre negatiu.
 - Divisió de de dos números enters, els dos negatius.
 - Realitzar prova unitària d'excepció de divisió per 0.

Puntuacions del exercici:

Apartat	Puntuació
Creació dels mètodes resta, multiplicar i dividir.	0,5 punts
Realitza proves unitàries del mètode suma.	2 punts
Realitza proves unitàries del mètode resta.	2 punts
Realitza proves unitàries del mètode multiplicar.	2 punts
Realitza proves unitàries del mètode dividir.	3 punts
Claredat del codi, documentació del codi, llegibilitat, bon funcionament, execució sense errors, fidelitat en l'aparença, correcta estructuració, modularitat...	0,5 punts

PROVES DE SEGURETAT



Les pràctiques es lliuraran seguint la següent pauta: El nom de l'activitat o pràctica a enviar serà el nombre de la pràctica (per exemple: A43, seguit de guió baix i del vostre nom i cognoms). Per exemple: **A73_AntonioGómez**.

L'activitat s'ha de lliurar en la data establerta i no s'acceptaran els lliuraments fora del termini establert. Les pràctiques han de ser originals i no s'admeten còpies de companys/es, Internet..

Objectius de l'activitat

1. Conèixer eines per a analitzar aplicacions mòbils.
2. Conèixer errors més comuns a l'hora de desenrotllar aplicacions mòbils.
3. Analitzar possibles vulnerabilitats en aplicacions mòbils IOS i Android.

Anàlisis de Vulnerabilitats con MobSF

En el món del desenvolupament d'aplicacions mòbils, la seguretat és un aspecte crític que no es pot passar per alt. Una aplicació vulnerable pot exposar dades sensibles, comprometre la privacitat dels usuaris o fins i tot ser utilitzada com a punt d'entrada per a atacs més sofisticats.

En aquesta activitat, explorarem **Mobile Security Framework (MobSF)**, una eina automatitzada que permet analitzar aplicacions Android i iOS a la recerca de vulnerabilitats de seguretat. MobSF facilita l'auditoria estàtica i dinàmica d'aplicacions, proporcionant informes detallats sobre permisos, configuracions insegures, riscos de privacitat i vulnerabilitats en el codi.

ACTIVITAT

1. Ajudant-te del [tutorial que hem vist de MOBSF](#), instal·la l'eina MOBSF en línia des de la següent url https://kodekloud.com/studio/labs/docker/docker_run
2. Realitza una anàlisi de la [següent aplicació](#) i comenta si trobes alguna anomalia. Si no realitza l'anàlisi amb l'id, hauràs de descarregar l'apk, existeixen moltes pàgines en línia per fer-ho.
3. Realitza una anàlisi de la [següent aplicació](#) i comenta si trobes alguna anomalia.
4. Realitza una cerca a Google Play d'alguna altra aplicació que consideris "sospitosa" i analitza-la. Adjunta la seu url i comenta per què la veus anòmala.

Puntuacions del exercici:

Apartat	Puntuació
Realitza una anàlisi de llanterna brillant i comenta anomalies.	3 punts
Realitza una anàlisi de lector QR i comenta anomalies.	3 punt
Realitza una cerca a Google Play d'alguna altra aplicació que consideris "sospitosa". Adjunta la seu url i comenta per què la veus anòmala.	4 punts