

Proyecto IBAN en C#

- **INTRODUCCIÓN AL PROYECTO:**

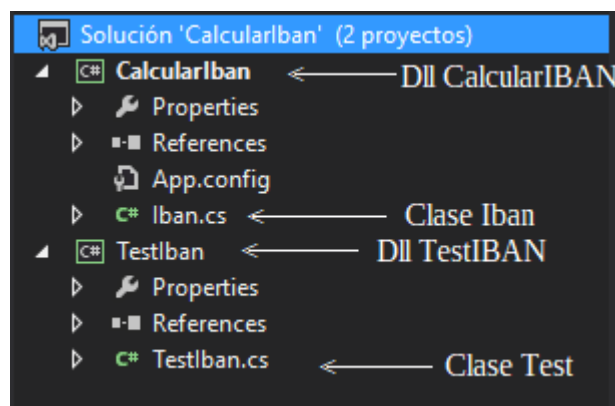
En este proyecto lo que haremos es implementar el cálculo del IBAN y su posterior validación. También le haremos sus posteriores pruebas unitarias. Finalmente subiremos poco a poco, por cada cambio que hagamos en el proyecto, a Git los cambios y commits que definamos en el momento y su posterior inyección a GitHub.

- **CREACIÓN DEL PROYECTO:**

Creamos el **ProyectoIBAN** con nuestro entorno gráfico Visual Studio 2012.

Agregamos a nuestro proyecto dos bibliotecas de clases, “**CalcularIBAN**” y “**TestIBAN**”. CalcularIBAN será la biblioteca de clase que contendrá nuestra codificación del cálculo, y TestIBAN será la biblioteca de clase que testeará la dll CalcularIBAN.

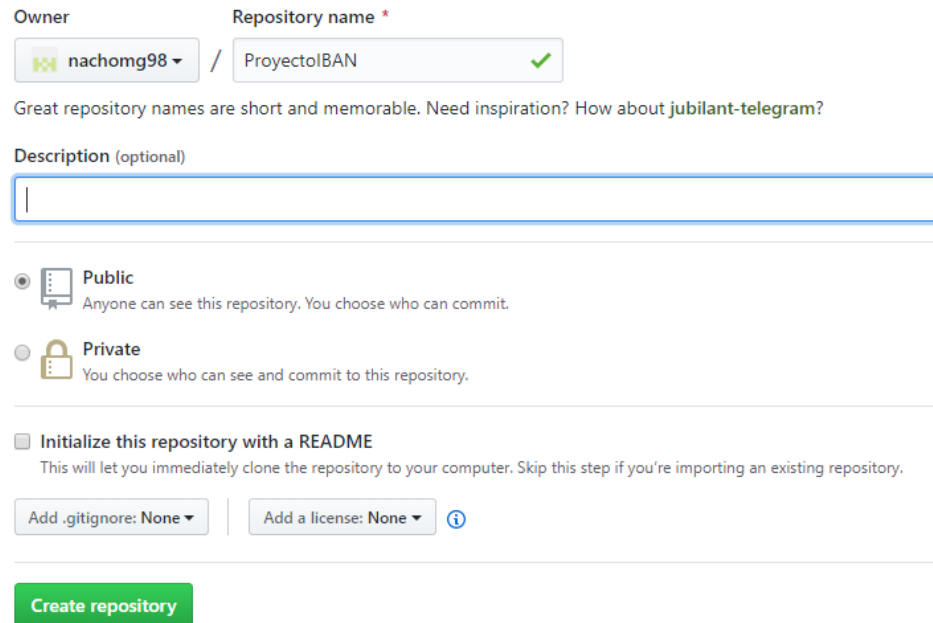
Cada Dll tendrá una clase que se encargará de lo explicado anteriormente. En la imagen siguiente se muestra como está realizado nuestro árbol de nuestro proyecto.



- **CREACION DE LOS REPOSITORIOS:**

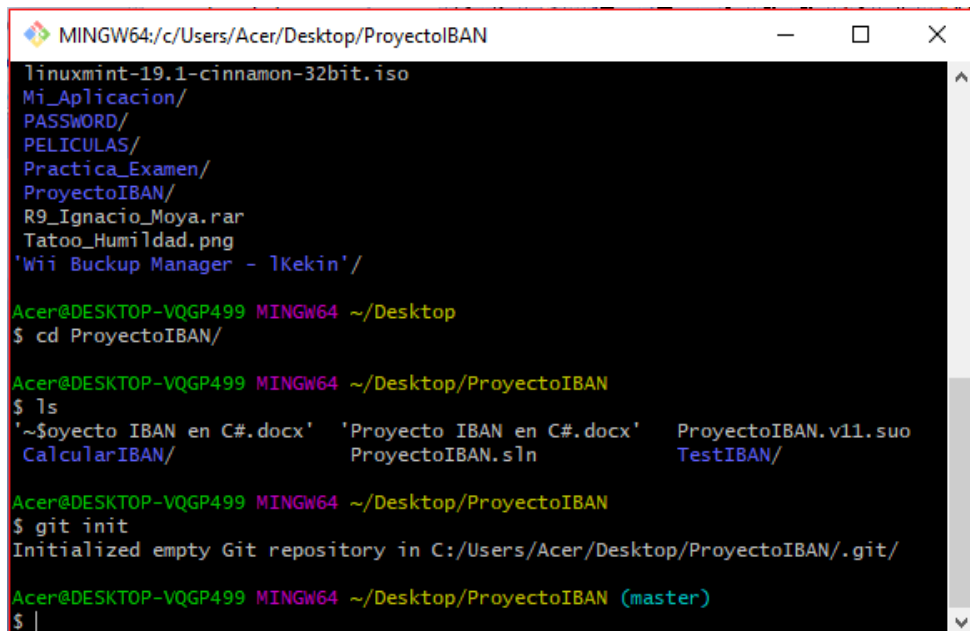
Creamos nuestro repositorio de GitHub y Git. Van a ser nuestros repositorios donde subiremos los cambios que vayamos haciendo en nuestro proyecto y así tener un control de cambios y versiones del proyecto a realizar:

Creamos un nuevo repositorio en GitHub:



The screenshot shows the GitHub 'Create repository' form. At the top, the 'Owner' is 'nachomg98' and the 'Repository name' is 'ProyectoIBAN', which has a green checkmark next to it. Below this, a message says: 'Great repository names are short and memorable. Need inspiration? How about [jubilant-telegram](#)?'. There is a 'Description (optional)' text area. Under 'Visibility', the 'Public' option is selected with a radio button, and a description says 'Anyone can see this repository. You choose who can commit.' The 'Private' option is also visible with a description 'You choose who can see and commit to this repository.' There is a checkbox for 'Initialize this repository with a README' with a description: 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.' At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by an information icon. A green 'Create repository' button is at the bottom.

Ahora creamos un repositorio en GitBash:



```
MINGW64:/c:/Users/Acer/Desktop/ProyectoIBAN
linuxmint-19.1-cinnamon-32bit.iso
Mi_Aplicacion/
PASSWORD/
PELICULAS/
Practica_Examen/
ProyectoIBAN/
R9_Ignacio_Moya.rar
Tattoo_Humildad.png
'Wii Backup Manager - 1Kekin'/'

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop
$ cd ProyectoIBAN/

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN
$ ls
'~$oyecto IBAN en C#.docx'  'Proyecto IBAN en C#.docx'  ProyectoIBAN.v11.suo
CalcularIBAN/              ProyectoIBAN.sln              TestIBAN/

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN
$ git init
Initialized empty Git repository in C:/Users/Acer/Desktop/ProyectoIBAN/.git/

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$
```

Ahora vinculamos nuestro repositorio Local con nuestro repositorio Remoto, porque los cambios lo realizaremos en nuestro repositorio local y luego lo subiremos al remoto.

```
Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git remote add ProyectoIban https://github.com/nachomg98/ProyectoIBAN.git

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git remote -v
ProyectoIban    https://github.com/nachomg98/ProyectoIBAN.git (fetch)
ProyectoIban    https://github.com/nachomg98/ProyectoIBAN.git (push)

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ |
```

- **IMPLEMENTACIÓN DEL CÓDIGO:**

Implementamos el código C# para el cálculo del iban en nuestra clase Iban.cs:

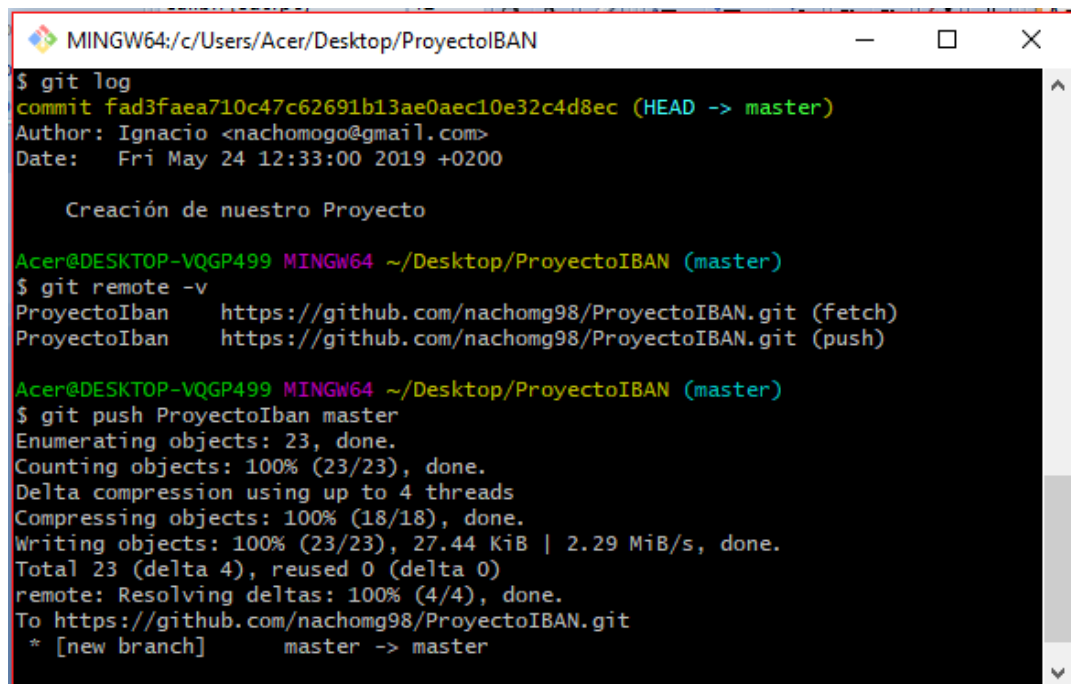
```
protected string calcularIban(string ccc)
{
    // Calculamos el IBAN
    ccc = ccc.Trim();
    if (ccc.Length != 20)
    {
        return "La CCC debe tener 20 dígitos";
    }
    // Le añadimos el código del país al ccc
    ccc = ccc + "142800";

    // Troceamos el ccc en partes (26 dígitos)
    string[] partesCCC = new string[5];
    partesCCC[0] = ccc.Substring(0, 5);
    partesCCC[1] = ccc.Substring(5, 5);
    partesCCC[2] = ccc.Substring(10, 5);
    partesCCC[3] = ccc.Substring(15, 5);
    partesCCC[4] = ccc.Substring(20, 6);

    int iResultado = int.Parse(partesCCC[0]) % 97;
    string resultado = iResultado.ToString();
    for (int i = 0; i < partesCCC.Length - 1; i++)
    {
        resultado = iResultado.ToString();
    }
    // Le restamos el resultado a 98
    int iRestoIban = 98 - int.Parse(resultado);
    string restoIban = iRestoIban.ToString();
    if (restoIban.Length == 1)
        restoIban = "0" + restoIban;

    return "ES" + restoIban + ccc;
}
```

Subimos los cambios a Git y GitHub. Añadimos el proyecto a git y realizamos un commit con el mensaje "Creación del nuestro Proyecto". Luego lo subimos a nuestro repositorio remoto con git push.



```
MINGW64:/c:/Users/Acer/Desktop/ProyectoIBAN
$ git log
commit fad3faea710c47c62691b13ae0aec10e32c4d8ec (HEAD -> master)
Author: Ignacio <nachomogo@gmail.com>
Date:   Fri May 24 12:33:00 2019 +0200

    Creación de nuestro Proyecto

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git remote -v
ProyectoIban      https://github.com/nachomg98/ProyectoIBAN.git (fetch)
ProyectoIban      https://github.com/nachomg98/ProyectoIBAN.git (push)

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git push ProyectoIban master
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 4 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (23/23), 27.44 KiB | 2.29 MiB/s, done.
Total 23 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/nachomg98/ProyectoIBAN.git
 * [new branch]      master -> master
```

- **PROCESO DE REFACTORIZACIÓN:**

Como vemos el código no está refactorizado, por ello pasamos al proceso de la refactorización.

```
//Quitamos las instrucciones using que no utilizemos.
using System;

namespace ProyectoIBAN
{
    public class Iban
    {
        /// <summary> ...
        public bool EsIBANvalido(string IBAN)
        {
            return IBAN == calcularIban(IBAN.Substring(4));
        }

        private string calcularIban(string ccc)
        {
            // Calculamos el IBAN
            string numeroCalcular = ccc.Trim();
            if (numeroCalcular.Length != 20)
            {
                return "La CCC debe tener 20 dígitos";
            }
            // Le añadimos el código del país al ccc
            numeroCalcular = numeroCalcular + "142800";

            // Troceamos el ccc en partes (26 dígitos)
            string[] partesCCC = new string[5];
            partesCCC[0] = numeroCalcular.Substring(0, 5);
            partesCCC[1] = numeroCalcular.Substring(5, 5);
            partesCCC[2] = numeroCalcular.Substring(10, 5);
            partesCCC[3] = numeroCalcular.Substring(15, 5);
            partesCCC[4] = numeroCalcular.Substring(20, 6);

            int miResultado = int.Parse(partesCCC[0]) % 97;
            string resultado = miResultado.ToString();
            for (int i = 0; i < partesCCC.Length - 1; i++)
```

```
int miResultado = int.Parse(partesCCC[0]) % 97;
string resultado = miResultado.ToString();
for (int i = 0; i < partesCCC.Length - 1; i++)
{
    miResultado = int.Parse(resultado + partesCCC[i + 1]) % 97;
    resultado = miResultado.ToString();
}
// Le restamos el resultado a 98
int miRestoIban = 98 - int.Parse(resultado);
string restoIban = miRestoIban.ToString();
if (restoIban.Length == 1)
    restoIban = "0" + restoIban;

//Devolvemos el Iban
return "ES" + restoIban + ccc;
}
}
```

Volvemos a subir los cambios al git:

```
Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git add -A

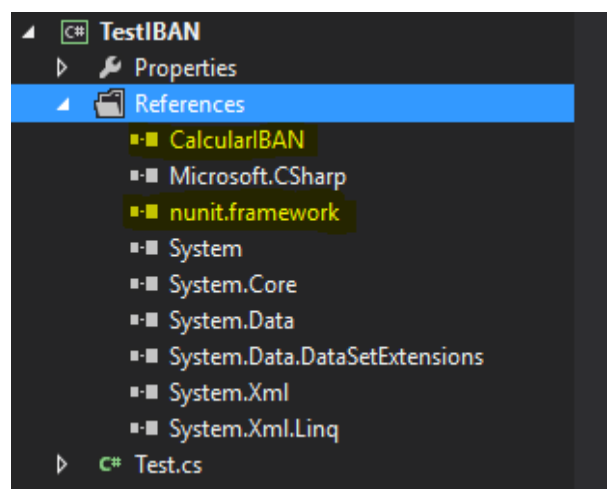
Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git commit -m "Refactorización de la clase Iban.cs"
[master 9fb4380] Refactorización de la clase Iban.cs
 2 files changed, 0 insertions(+), 0 deletions(-)

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git push proyecto master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 53.99 KiB | 7.71 MiB/s, done.
Total 4 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/nachomg98/ProyectoIban.git
 a6c8749..9fb4380 master -> master

Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ |
```

- **IMPLEMETACIÓN DE LOS TEST UNITARIOS:**

Una vez que tenemos la clase Iban.cs implementada y refactorizada, implementamos en la dll TestIBAN, los test unitarios del testeo que tenemos que hacer a la clase Iban.cs. Para ello lo primero que tenemos que hacer es referenciar la dll CalcularIban en Test y agregar la referencia Nunit.Framework.



Implementamos la clase Test.cs:

```
namespace TestIBAN
{
    [TestFixture]
    public class Test
    {
        [Test]
        public void CuentaCorrienteVacía()...

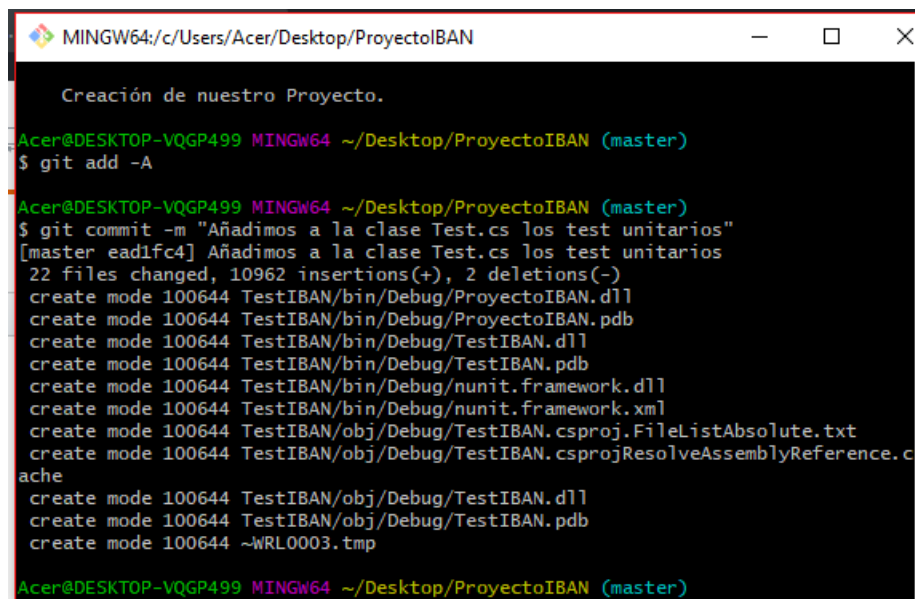
        [Test]
        public void ValidarIbanEsValido()...

        [Test]
        public void ValidarIbanNoEsValido()...

        [Test]
        public void CalculoIBANCorrecto()...

        [Test]
        public void CalculoIBANNoEsCorrecto()...
    }
}
```

Subimos los cambios a Git:



```
MINGW64:/c/Users/Acer/Desktop/ProyectoIBAN
Creación de nuestro Proyecto.
Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git add -A
Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
$ git commit -m "Añadimos a la clase Test.cs los test unitarios"
[master ead1fc4] Añadimos a la clase Test.cs los test unitarios
22 files changed, 10962 insertions(+), 2 deletions(-)
create mode 100644 TestIBAN/bin/Debug/ProyectoIBAN.dll
create mode 100644 TestIBAN/bin/Debug/ProyectoIBAN.pdb
create mode 100644 TestIBAN/bin/Debug/TestIBAN.dll
create mode 100644 TestIBAN/bin/Debug/TestIBAN.pdb
create mode 100644 TestIBAN/bin/Debug/nunit.framework.dll
create mode 100644 TestIBAN/bin/Debug/nunit.framework.xml
create mode 100644 TestIBAN/obj/Debug/TestIBAN.csproj.FileListAbsolute.txt
create mode 100644 TestIBAN/obj/Debug/TestIBAN.csprojResolveAssemblyReference.cache
create mode 100644 TestIBAN/obj/Debug/TestIBAN.dll
create mode 100644 TestIBAN/obj/Debug/TestIBAN.pdb
create mode 100644 ~\WRL0003.tmp
Acer@DESKTOP-VQGP499 MINGW64 ~/Desktop/ProyectoIBAN (master)
```