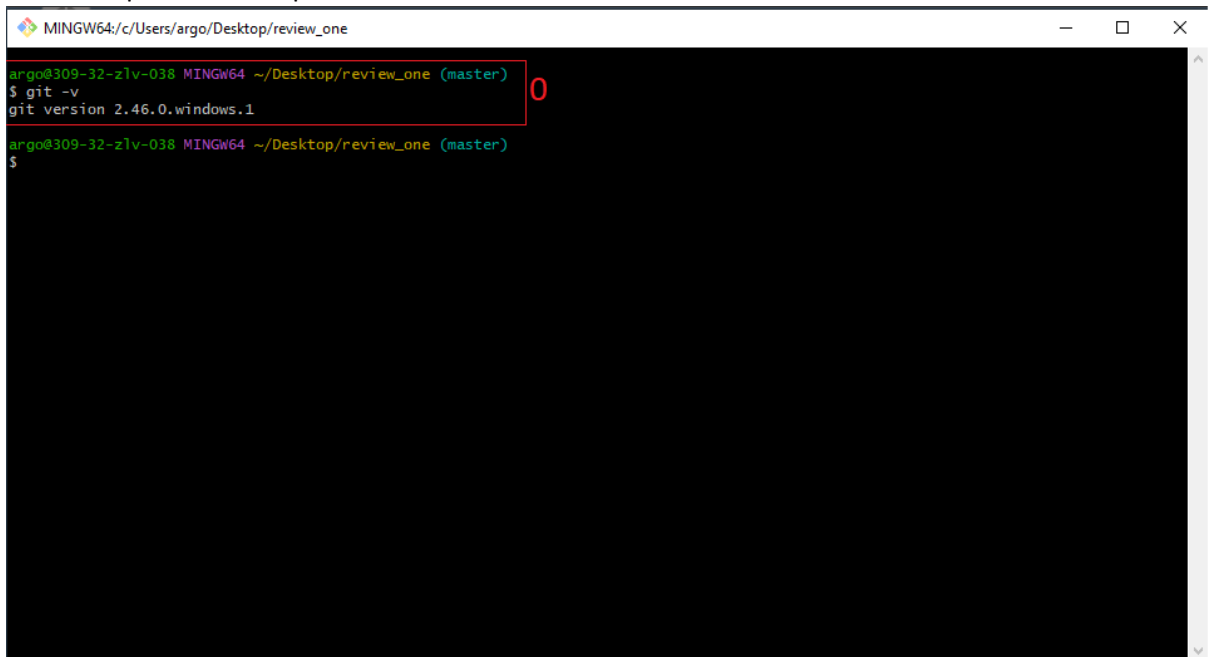


T0. Activitats de repàs I

1. Crea la carpeta de projecte “review_one” en la teva unitat i afegeix-la com a repositori de git.

PASOS

0. Para el correcto desarrollo de la actividad, primero verificamos si tenemos instalado Git en nuestro computador, para esto ejecutamos el comando mostrado en la terminal el cuál nos dará la versión instalada de Git, en caso de no reconocer git como un comando significa que tenemos que realizar el proceso de instalacion.



```
MINGW64:/c:/Users/argo/Desktop/review_one
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ git -v
git version 2.46.0.windows.1
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$
```

Una vez descargado Git en nuestro computador, veremos que nos descarga algunas dependencias para su correcto funcionamiento y una de ellas es la terminal Git bash, en esta terminal tendremos un entorno muy parecido a la terminal de Linux por lo que, si ya hemos usado linux previamente nos resultará amigable en cuanto a comandos e interfaz

1. Se crea la carpeta “review_one” mediante comandos (mkdir <nombre_directorio>).
2. Se verifica que la carpeta se haya creado exitosamente, esto se realiza mediante el comando “ls -la” el cuál nos generará una lista detallada de los directorios y archivos existentes en nuestra ubicacion
3. Se accede a la carpeta creada (a partir de ahora la llamaremos directorio) mediante el comando “cd” el cual nos permite cambiar de ruta
4. Usando el comando “git init” generamos los archivos necesarios para que el Git bash reconozca el directorio como un repositorio Git, en este caso se crea el directorio “.git”.
5. Si usamos de nuevo el comando “ls -la” podremos ver el directorio creado .git. De igual manera un indicativo de que creamos un repositorio es que despues de la ruta en la que estamos ubicados aparecerá un “(master)”.

```
MINGW64: c:/Users/argo/Desktop/review_one
argo@309-32-z1v-038 MINGW64 ~/Desktop
$ mkdir review_one
argo@309-32-z1v-038 MINGW64 ~/Desktop
$ ls -la
total 17
drwxr-xr-x 1 argo 197121  0 Sep 18 16:26 ./
drwxr-xr-x 1 argo 197121  0 Sep 18 16:20 ../
drwxr-xr-x 1 argo 197121  0 Sep 17 19:45 LoveStone/
-rw-r--r-- 1 argo 197121 282 Feb  8 2022 desktop.ini
drwxr-xr-x 1 argo 197121  0 Sep 18 16:26 review_one/
argo@309-32-z1v-038 MINGW64 ~/Desktop
$ cd review_one/
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one
$ git init
Initialized empty Git repository in C:/Users/argo/Desktop/review_one/.git/
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ ls -la
total 4
drwxr-xr-x 1 argo 197121 0 Sep 18 16:27 ./
drwxr-xr-x 1 argo 197121 0 Sep 18 16:26 ../
drwxr-xr-x 1 argo 197121 0 Sep 18 16:27 .git/
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$
```

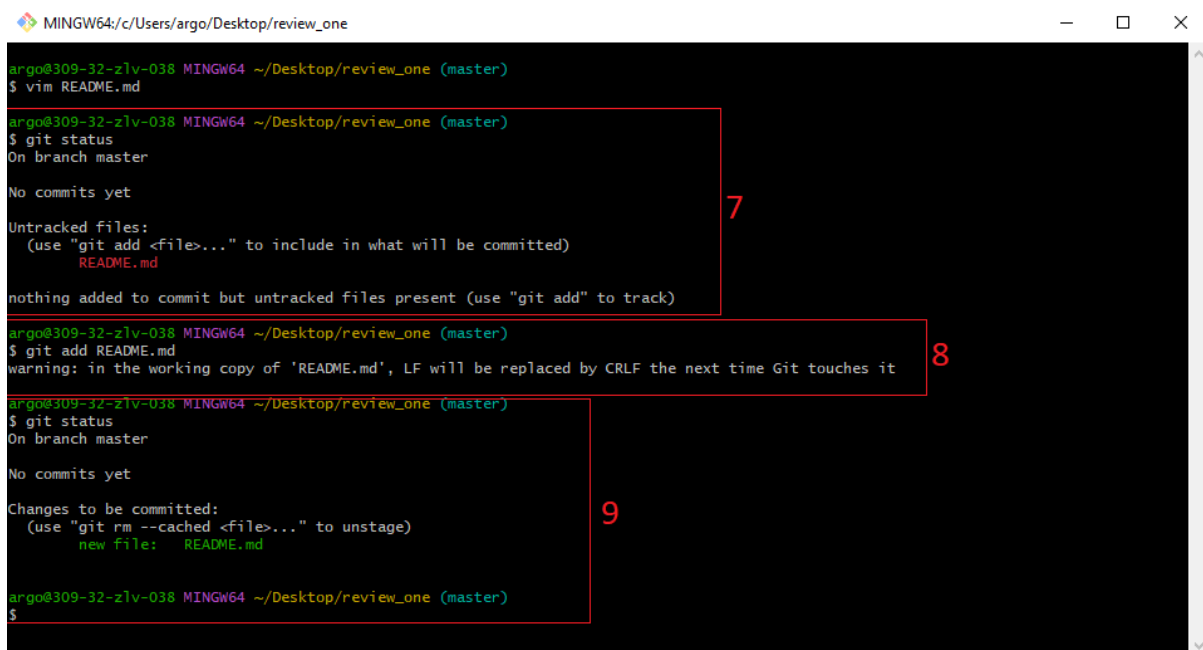


Aquest repositori conté les activitats de repàs i del mòdul de projectes

[illegible]

3. Afegeix-lo al staging area.

7. Para identificar un archivo o directorio que no se haya agregado al staging area, usamos el comando “git status” el cual nos indicará en color rojo lo que nos hace falta para añadir.
8. Una vez identificado, en este caso el archivo README.md, lo añadimos con el comando “git add” seguido del nombre del archivo/directorio
9. Si volvemos a usar el git status, este nos indicará en color verde que se ha subido pero hace falta el commit para confirmar



```
MINGW64/c:/Users/argo/Desktop/review_one
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ vim README.md

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  README.md

nothing added to commit but untracked files present (use "git add" to track)

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ git status
On branch master

No commits yet

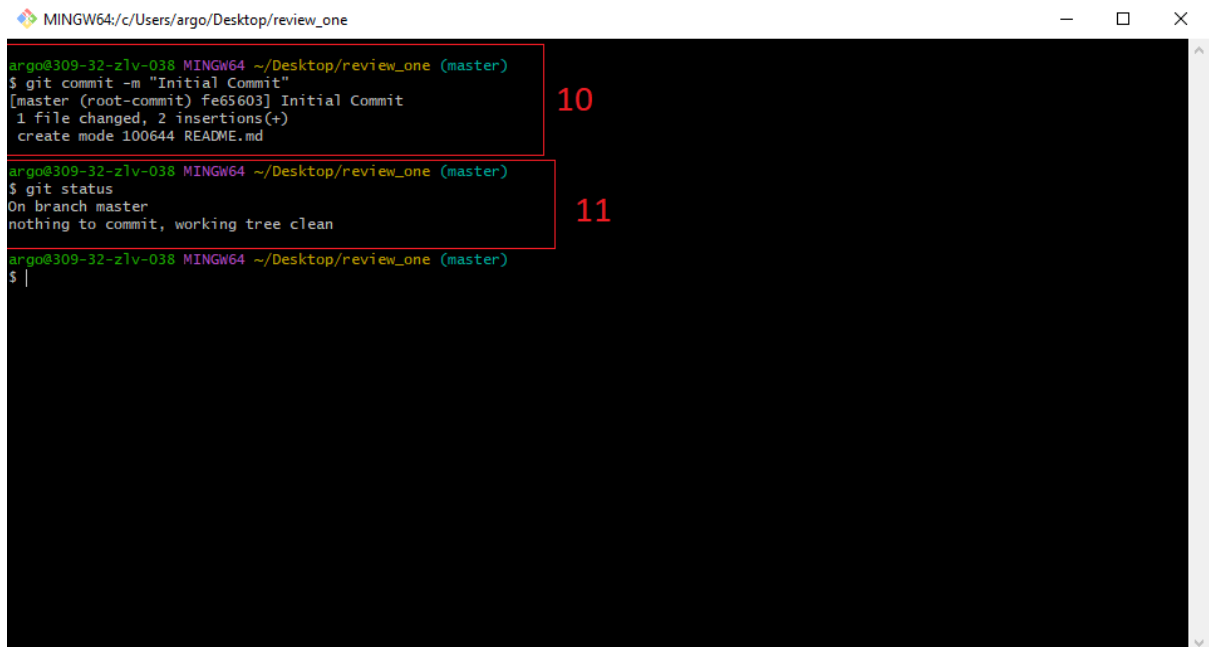
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$
```

4. Afegeix-lo a la commit area amb el comentari "Initial commit".

10. Para realizar los commit y “confirmar” los add’s, usamos el comando “git commit –m <Mensaje>”

11. Si usamos nuevamente git status veremos que no tenemos nada para realizar commit, este es un buen momento para realizar un push a nuestro repositorio remoto

A screenshot of a Windows terminal window titled 'MINGW64; c:/Users/argo/Desktop/review_one'. The terminal shows the execution of two git commands. The first command, 'git commit -m "Initial Commit"', is highlighted with a red box and labeled '10' in red text. Its output shows the creation of an initial commit for 'README.md'. The second command, 'git status', is also highlighted with a red box and labeled '11' in red text. Its output indicates that the working tree is clean and there is nothing to commit. The terminal text is as follows:

```
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ git commit -m "Initial Commit"
[master (root-commit) fe65603] Initial Commit
1 file changed, 2 insertions(+)
create mode 100644 README.md

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ git status
On branch master
nothing to commit, working tree clean

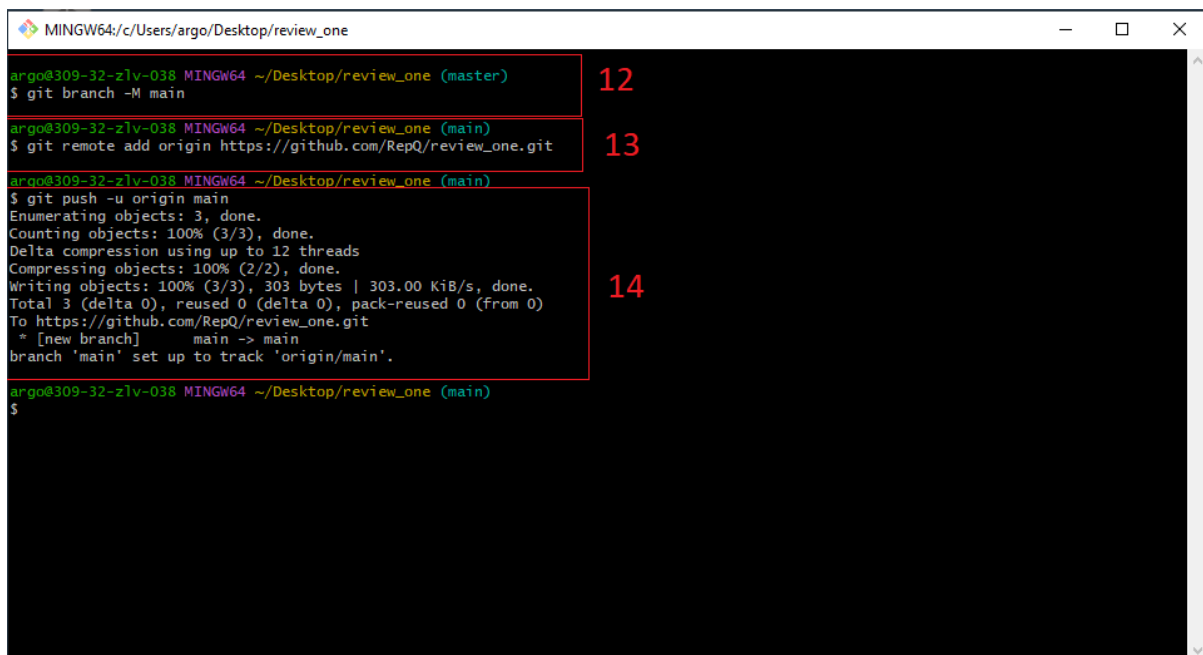
argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ |
```

5. Puja el teu repositori local a un repositori remot (ha de tenir el mateix nom).

12. Usamos “git branch –M main” basicamente para seguir la convencion mas reciente de Git para cambiar de “master” a “main” como rama principal

13. Vinculamos el repositorio local a uno remoto. Acá creamos una referencia llamada en este caso origin que apunta a nuestro repositorio remoto (la url)

14. Estamos subiendo a nuestro repositorio remoto, vinculado con origin, nuestra rama main creada previamente en el paso 12



```
MINGW64: c:/Users/argo/Desktop/review_one

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (master)
$ git branch -M main

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (main)
$ git remote add origin https://github.com/RepQ/review_one.git

argo@309-32-z1v-038 MINGW64 ~/Desktop/review_one (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RepQ/review_one.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

6. Aplica les tècniques de refacció i optimització en el codi següent:

```
using System;

class Program
{
    static void Main(string[] args)
    {
        int nota1 = 0;
        int nota2 = 0;
        int nota3 = 0;

        Console.WriteLine("Introdueix la primera qualificació");
        nota1 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Introdueix la segona qualificació: ");
        nota2 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Introdueix la tercera qualificació: ");
        nota3 = Convert.ToInt32(Console.ReadLine());

        int suma = nota1 + nota2 + nota3;

        float promig = suma / 3;

        Console.WriteLine("El promig de les notes es: " + promig );

        if (promig > 6)
        {
            Console.WriteLine("Aprovat");
        }
        else if (promig < 6)
        {
            Console.WriteLine("Suspès");
        }
    }
}
```

Crea el projecte dins del mateix repositori i puja'l al repositori remot.

```
MINGW64:/c/Users/jmmte/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review_one
jmmte@LAPTOP-FD83TB4R MINGW64 ~/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review
$ ls
README.md
'T0. Activitats de repàs .docx'
jmmte@LAPTOP-FD83TB4R MINGW64 ~/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   "T0. Activitats de repàs\303\240s .docx"

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    T0_Project_SantiagoVergara/
    "~$. Activitats de repàs\303\240s .docx"

no changes added to commit (use "git add" and/or "git commit -a")

jmmte@LAPTOP-FD83TB4R MINGW64 ~/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review
$ git add .

jmmte@LAPTOP-FD83TB4R MINGW64 ~/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
```

```
jmmte@LAPTOP-FD83TB4R MINGW64 ~/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review
$ git commit -m "Finish T0"
[main 938eaf3] Finish T0
40 files changed, 465 insertions(+)
 create mode 100644 T0_Project_SantiagoVergara/.vs/ProjectEvaluation/t0_project_santiagovergara.metadata.v9.bin
 create mode 100644 T0_Project_SantiagoVergara/.vs/ProjectEvaluation/t0_project_santiagovergara.projects.v9.bin
 create mode 100644 T0_Project_SantiagoVergara/.vs/ProjectEvaluation/t0_project_santiagovergara.strings.v9.bin
 create mode 100644 T0_Project_SantiagoVergara/.vs/T0_Project_SantiagoVergara/De
signTimeBuild/.dtbcache.v2
 create mode 100644 T0_Project_SantiagoVergara/.vs/T0_Project_SantiagoVergara/Fin
leContentIndex/15ab2ca2-ee52-4a69-b94f-a9da58c78e60.vsidx
 create mode 100644 T0_Project_SantiagoVergara/.vs/T0_Project_SantiagoVergara/v1
7/.futdcache.v2
 create mode 100644 T0_Project_SantiagoVergara/.vs/T0_Project_SantiagoVergara/v1
7/.suo
 create mode 100644 T0_Project_SantiagoVergara/.vs/T0_Project_SantiagoVergara/v1
7/DocumentLayout.backup.json
```



```
jmmte@LAPTOP-FD83TB4R MINGW64 ~/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review
_one (main)
$ git push
Enumerating objects: 55, done.
Counting objects: 100% (55/55), done.
Delta compression using up to 8 threads
Compressing objects: 100% (48/48), done.
Writing objects: 100% (53/53), 141.90 KiB | 2.03 MiB/s, done.
Total 53 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
To https://github.com/RepQ/review_one.git
   9a90503..938eaf3  main -> main
```

```
jmmte@LAPTOP-FD83TB4R MINGW64 ~/OneDrive/Escritorio/SANTIAGO/ITB/Proyecto/review
_one (main)
```

\$ | página: 9 de 11 | Palabras: 737 |  Español (España, internacional) |

7. Afegeix un
respectius co
un valor per f

7. Afegeix una funcionalitat al programa anterior que permeti emmagatzemar ciutats i els seus respectius codis postals. El programa ha de demanar el nom i el codi fins que l'usuari introdueixi un valor per finalitzar la introducció de dades i mostrar un llistat amb format dels valors introduïts.

```

Program.cs
T0_Project_SantiagoVergara
Program
Promig(int[] notas)

1  using System;
2  using System.Collections.Generic;
3  class Program
4  {
5      //Funcion encargada de calcular el promedio de las notas
6      public static void Promig(int[] notas)
7      {
8          int suma;
9          float promig;
10
11          suma = notas[0] + notas[1] + notas[2];
12          promig = (float)suma / 3;
13          Console.WriteLine("El promig de les notes es: " + promig);
14          if (promig >= 6)
15              Console.WriteLine("Aprovat");
16          else
17              Console.WriteLine("Suspés");
18      }
19
20      //Funcion encargada de obtener las 3 calificaciones
21      public static void Qualificacion(int[] notas)
22      {
23          Console.WriteLine("Introdueix la primera qualificació");
24          notas[0] = Convert.ToInt32(Console.ReadLine());
25
26          Console.WriteLine("Introdueix la segona qualificació: ");
27          notas[1] = Convert.ToInt32(Console.ReadLine());
28
29          Console.WriteLine("Introdueix la tercera qualificació: ");
30          notas[2] = Convert.ToInt32(Console.ReadLine());
31      }

```

```

33      //Funcion encargada de Obtener las ciudades y sus respectivos codigos postales
34      public static void Cities()
35      {
36          bool stop;
37          Dictionary<String, int> Cities;
38
39          stop = false;
40          Cities = new Dictionary<string, int>();
41          do
42          {
43              Console.WriteLine($"Ingrese el nombre de la ciudad: ");
44              String City = Console.ReadLine();
45              Console.WriteLine($"Ingrese el codigo postal de la Ciudad {City}: ");
46              int CityPostal = Convert.ToInt32(Console.ReadLine());
47              Cities.Add(City, CityPostal);
48              Console.WriteLine("Presione 0 para continuar, cualquier otra tecla para parar: ");
49              if (Convert.ToInt32(Console.ReadLine()) != 0)
50                  stop = true;
51          } while (!stop);
52
53          ShowCities(Cities);
54      }
55
56      //Funcion encargada de mostrar por pantalla las ciudades y codigos postales obtenidos en un formato Ciudad - Codigo Postal
57      public static void ShowCities(Dictionary<String, int> Cities)
58      {
59          foreach (var entry in Cities)
60          {
61              Console.WriteLine($"Ciudad: {entry.Key} - Codigo Postal {entry.Value}");
62          }
63      }
64

```

```
0 referencias
65  static void Main(string[] args)
66  {
67      //Array de notas tipo int
68      int[] notas;
69
70      //Creacion del array
71      notas = new int[3];
72
73      //Llamado a funciones
74      Cities();
75      Qualificacion (notas);
76      Promig(notas);
77  }
78  }
79
```