



FACULTAD DE
INGENIERÍA
DE SISTEMAS

CARRERA DE SOFTWARE INGENIERÍA DE SOFTWARE Y REQUERIMIENTOS

LABORATORIO – CONFIGURACIÓN INICIALES

Integrantes:

- **Estudiante 1:** Betancourt Alison
- **Estudiante 2:** Huilca Fernando

Objetivo general

El objetivo de este laboratorio es realizar las primeras configuraciones iniciales en AzureDevOps, sobre la cual trabajaremos en este semestre en el desarrollo de requerimientos y construcción de aplicaciones.

Objetivos específicos.

1. **Crear una organización en Azure DevOps.**
2. **Crear un proyecto en Azure DevOps.**
3. **Añadir miembros al proyecto y realizar configuraciones**
4. **Clonar solución de Azure DevOps hacia disco local**
5. **Dentro de carpeta clonada, crear una solución con proyecto de consola inicial**
6. **Ejecución de proyecto de consola**
7. **Subir código fuente en AzureDevOps**

1. Crear una organización en Azure DevOps.

Crear una organización, el nombre debe ser su ISWR2024BNombreApellido, si ya está usado puede añadir un numero al final.

Azure DevOps

fernando.huilca@epn.edu.ec

Almost done...

Name your Azure DevOps organization *

dev.azure.com/ ISWR2024BFernandoHuilca

We'll host your projects in

Brazil

Enter the characters you see

New Audio

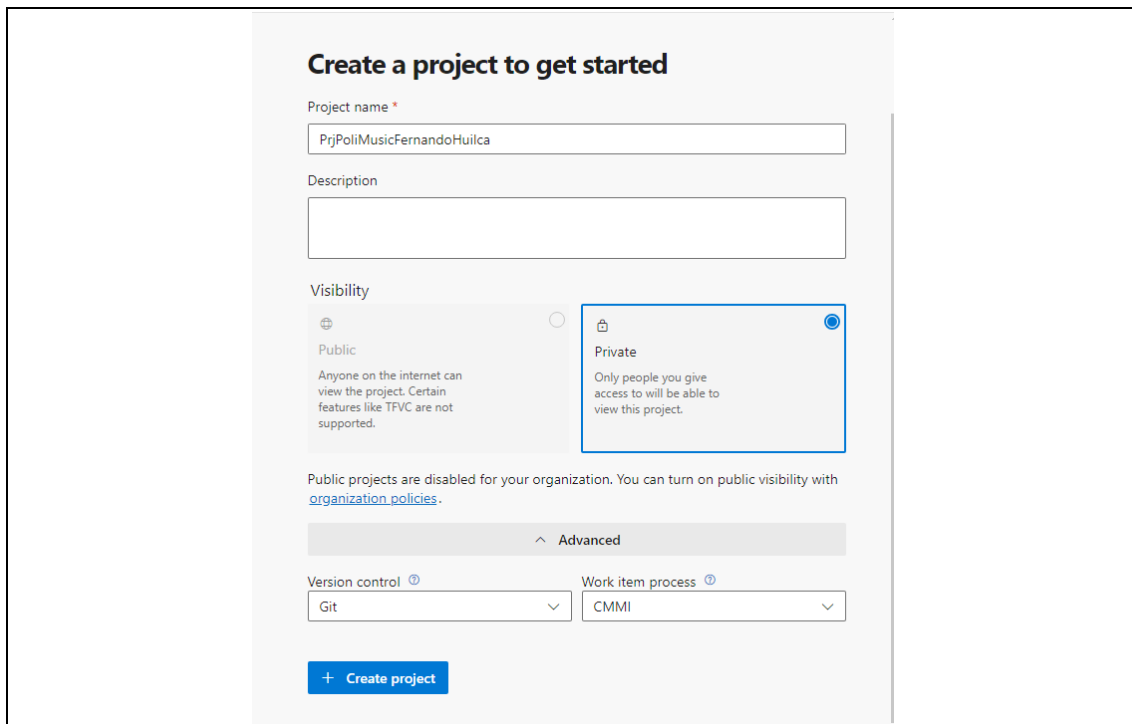
6GXXRSW

Challenge validation failed, please try again

Continue

2. Crear un proyecto en Azure DevOps.

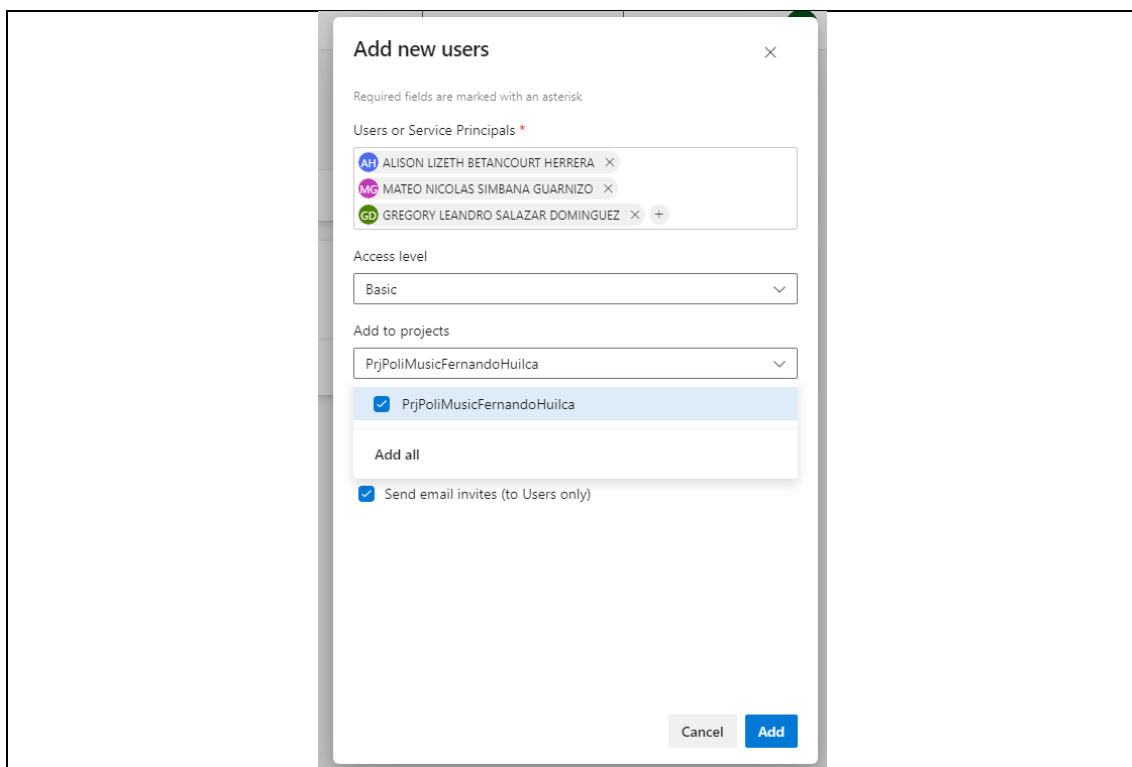
Crear un nuevo proyecto dentro de la organización, el nombre debe ser PrjPoliMusic**NombreApellido**, colocar su nombre y apellido. Escoger como “Version Control” a Git y como “Work Item process” a CMMI.



The screenshot shows the 'Create a project to get started' form in Azure DevOps. The form includes the following fields and options:

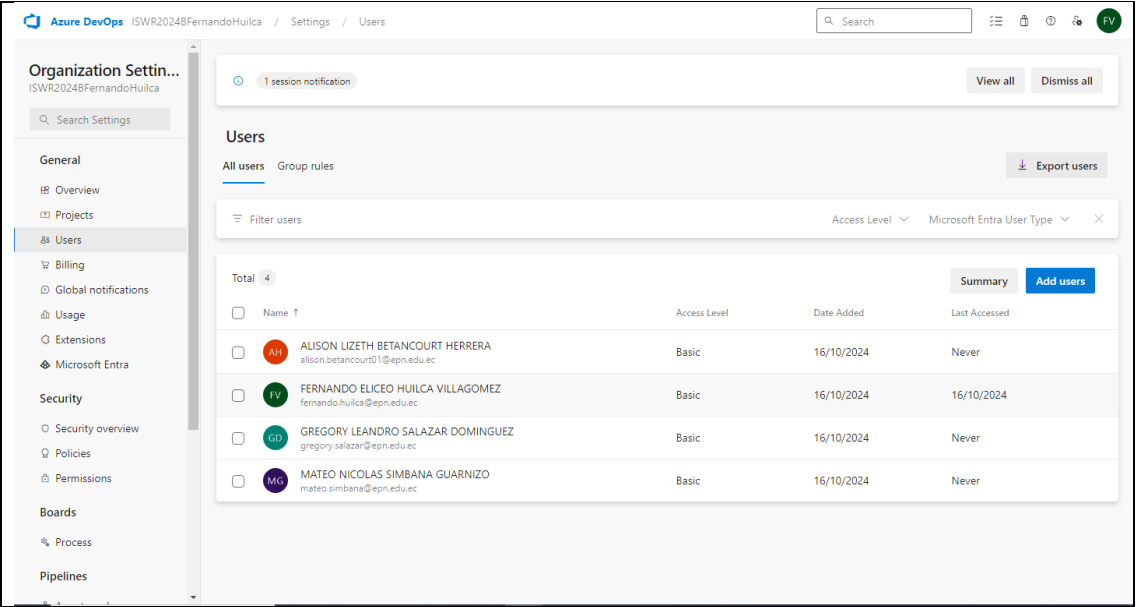
- Project name ***: A text input field containing 'PrjPoliMusicFernandoHuilca'.
- Description**: A large text area for project description.
- Visibility**: Two radio button options. 'Public' is disabled with a note: 'Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).' The 'Private' option is selected and highlighted with a blue border. Its description is: 'Only people you give access to will be able to view this project.'
- Advanced**: A section header for additional settings.
- Version control**: A dropdown menu set to 'Git'.
- Work item process**: A dropdown menu set to 'CMMI'.
- Create project**: A blue button at the bottom.

3. Añadir miembros al proyecto y realizar configuraciones



The screenshot shows the 'Add new users' dialog in Azure DevOps. The dialog includes the following fields and options:

- Required fields are marked with an asterisk**: A note at the top.
- Users or Service Principals ***: A list of three users: ALISON LIZETH BETANCOURT HERRERA, MATEO NICOLAS SIMBANA GUARNIZO, and GREGORY LEANDRO SALAZAR DOMINGUEZ. Each user has a small icon and a close button (X).
- Access level**: A dropdown menu set to 'Basic'.
- Add to projects**: A dropdown menu set to 'PrjPoliMusicFernandoHuilca'.
- Add all**: A button to add all selected users.
- Send email invites (to Users only)**: A checked checkbox.
- Cancel** and **Add**: Buttons at the bottom.



4. Clonar solución de Azure DevOps hacia disco local

Para la clonación, puede realizarlo de varias formas, por ejemplo, usando su IDE favorito, o directamente trabajando todo desde la línea de comandos. Para este ejemplo solo trabajaremos desde la línea de comandos.

Verifique primero que tiene instalado git con este comando: “git --version”

Si aparece una versión, es que ya está instalado GIT. Si no reconoce el comando, debe instalar primero GIT desde este link. <https://git-scm.com/downloads>

Caso éxito	Caso fallido
<pre>C:\Users\pc>git --version git version 2.39.1.windows.1</pre>	

Para la clonación, desde la línea de comandos, se recomienda si está usando Windows, hacerlo desde esta ruta:

En nuestro caso particular usaremos otra ruta para tener mejor organización de los repositorios y dado que estamos trabajando en una pc personal:

```
C:\Users\<yourusername>\source\repos

reposAzureDevOps

>: ls
Ahorcado_in_C-/      Electronica_and Software/
AppParty/            Emulador_Arquitectura_Computacional/
Arboles-AVL/         Estructura-de-Datos-y-Algoritmos/
Arboles_Binarios_de_Busqueda/ ListaSimple/
Arquitectura_de_Von_Neumann/ ListasDoblementeEnlazadas/
BadIceCream02/       Perseverance/
Busqueda-y-Ordenamiento/ PoliEats_ProyectoEDA/
CodeHTML/            ProyectoFinalProgramacion/
Cositas_Secretas/    desktop.ini
Da-Vinci/            reposAzureDevOps/

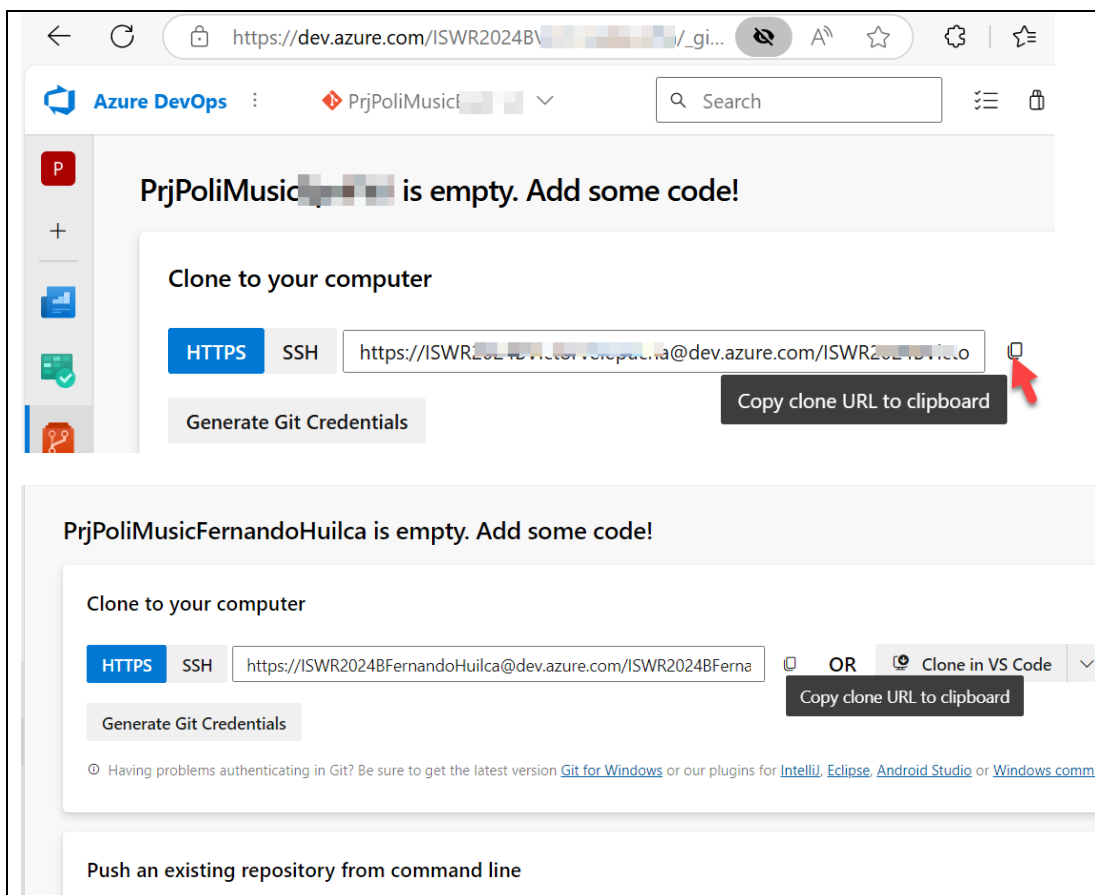
0.144s
home > Documents > Git-Fernando_Huilca
>: cd reposAzureDevOps/

0.054s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>: pwd
/c/Users/Fernando_Huilca/Documents/Git-Fernando_Huilca/reposAzureDevOps

0.043s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
```

Si no existe las subcarpetas source y repos, puede crearlas.

En AzureDevOps, nos copiamos la URL que queremos clonar así:



Finalmente, desde la ruta “c:\Users\yourusername\source\repos”, ejecute el comando

```
git clone SU_URL_DE_AZURE_DEVOPS
```

Le pedirá que se autentique, usar el correo y password de la universidad. Y finalmente aparecerá que se realizó la clonación con éxito.

```
C:\Users\pc>cd source  
  
C:\Users\pc\source>cd repos  
  
C:\Users\pc\source\repos> git clone https://ISWR2024BAlisonBetancourt@dev.azure.com/ISWR2024BAlisonBetancourt/PrjPoliMusicAlisonBetancourt/_git/PrjPoliMusicAlisonBetancourt  
Cloning into 'PrjPoliMusicAlisonBetancourt'...  
warning: You appear to have cloned an empty repository.  
  
C:\Users\pc\source\repos>
```

Ejecución



```
reposAzureDevOps  
Busqueda-y-Ordenamiento/ PoliEats_ProyectoEDA/  
CodeHTML/ ProyectoFinalProgramacion/  
Cositas_Secretas/ desktop.ini  
Da-Vinci/ reposAzureDevOps/  
  
0.144s  
home > Documents > Git-Fernando_Huilca  
> cd reposAzureDevOps/  
  
0.054s  
home > Documents > Git-Fernando_Huilca > reposAzureDevOps  
> pwd  
/c/Users/Fernando_Huilca/Documents/Git-Fernando_Huilca/reposAzureDevOps  
  
0.043s  
home > Documents > Git-Fernando_Huilca > reposAzureDevOps  
> git clone https://ISWR2024BFernandoHuilca@dev.azure.com/ISWR2024BFernandoHuilca/PrjPoliMusicFernandoHuilca/_git/PrjPoliMusicFernandoHuilca  
Cloning into 'PrjPoliMusicFernandoHuilca'...  
warning: You appear to have cloned an empty repository.  
  
11.124s  
home > Documents > Git-Fernando_Huilca > reposAzureDevOps  
> :
```

Finalmente, ingresamos a la carpeta clonada con “cd PrjPoliMusicNombreApellidoEstudiante”

```
C:\Users\pc\source\repos>cd PrjPoliMusicAlisonBetancourt  
  
C:\Users\pc\source\repos\PrjPoliMusicAlisonBetancourt>
```

Ejecución:

```
PrjPoliMusicFernandoHuilca
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>: pwd
/c/Users/Fernando_Huilca/Documents/Git-Fernando_Huilca/reposAzureDevOps

0.043s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>: git clone https://ISWR2024BFernandoHuilca@dev.azure.com/ISWR2024BFernandoHuilca/PrjPoliMusicFernandoHuilca/_git/PrjPoliMusicFernandoHuilca
Cloning into 'PrjPoliMusicFernandoHuilca'...
warning: You appear to have cloned an empty repository.

11.124s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>: ls
PrjPoliMusicFernandoHuilca/

0.163s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>: cd PrjPoliMusicFernandoHuilca/

0.046s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (master)
>: |
```

5. Crear aplicación de consola

Puede crear una aplicación de consola usando el IDE o Editor de código que usted prefiera, igual está en total libertad de escoger el lenguaje de programación y/o framework que usted prefiera.

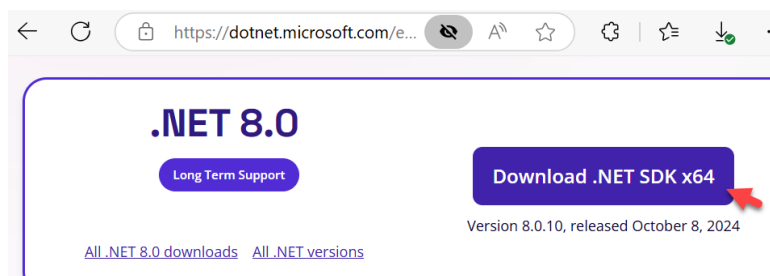
Para seguir con este ejercicio, vamos a usar .NET 8.0 y el lenguaje C#.

Primero nos aseguraremos de que tenemos instalado el SDK del .NET 8.0, para ello desde la consola ejecutamos este comando: dotnet --version

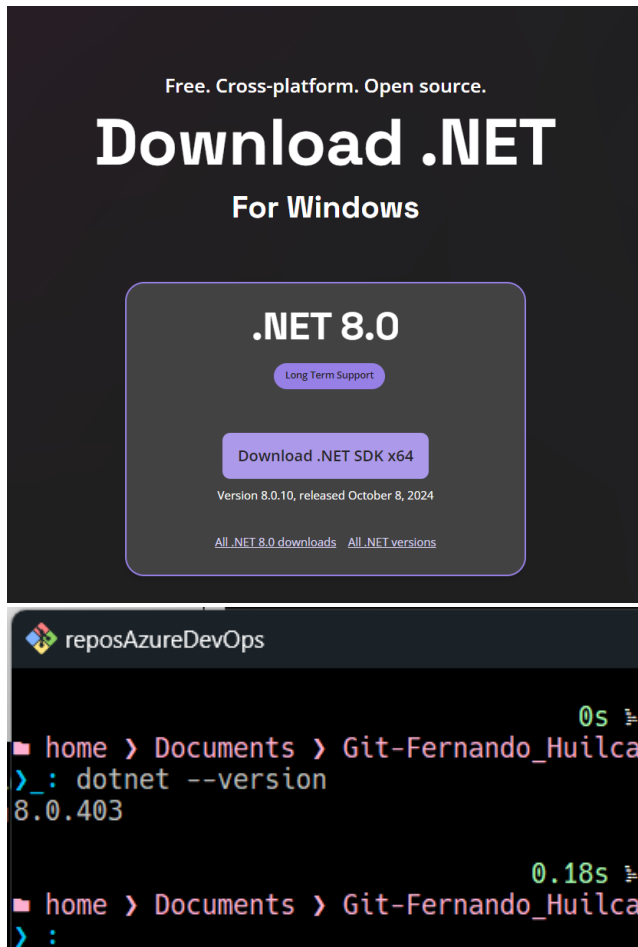
Caso éxito	Caso fallido
<pre>C:\>dotnet --version 8.0.402</pre> <p>Ejecución:</p> <pre>home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (master) >: dotnet --version 7.0.402 home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (master) >: </pre>	<pre>C:\>dotnet --version 'dotnet ' is not recognized as an internal or external command, operable program or batch file.</pre>

Si el comando no es reconocido, nos descargamos el instalador desde este link:

<https://dotnet.microsoft.com/en-us/download>



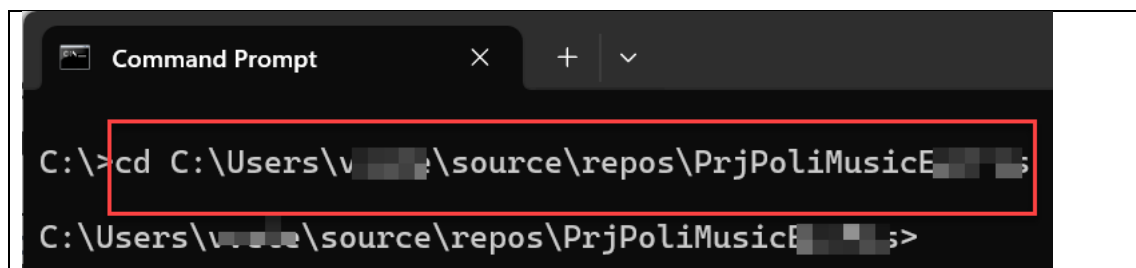
Dado que la versión que poseo no es la mas reciente, procedo a descargarme la actualización:



Una vez instalado, cerramos todas las consolas (cmd) que tengamos abiertas, y abrimos **una nueva consola** (cmd) y verificamos que el comando se ejecute correctamente y nos presente la versión 8.

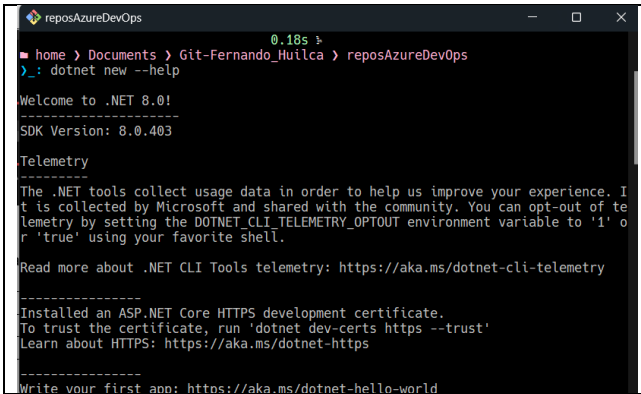
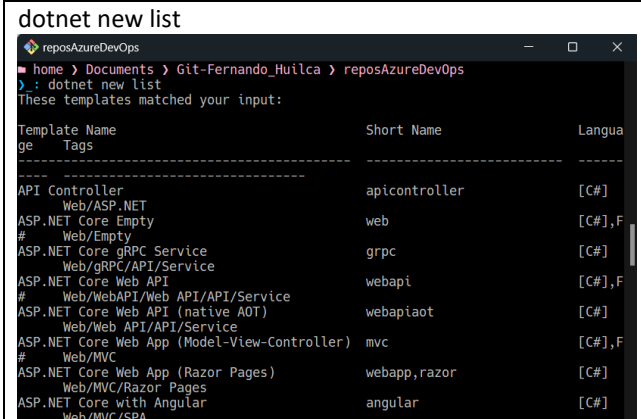
Inicialmente en este ejercicio, sólo crearemos una aplicación de consola, y lo vamos a realizar únicamente por esta vez desde la línea de comandos, sin necesidad de usar un IDE o Editor de Código como VSCode. Si usted decide usar un IDE o Editor puede hacerlo de forma alternativa.

Desde la consola ingresamos a la ruta de la carpeta clonada, por ejemplo, así:



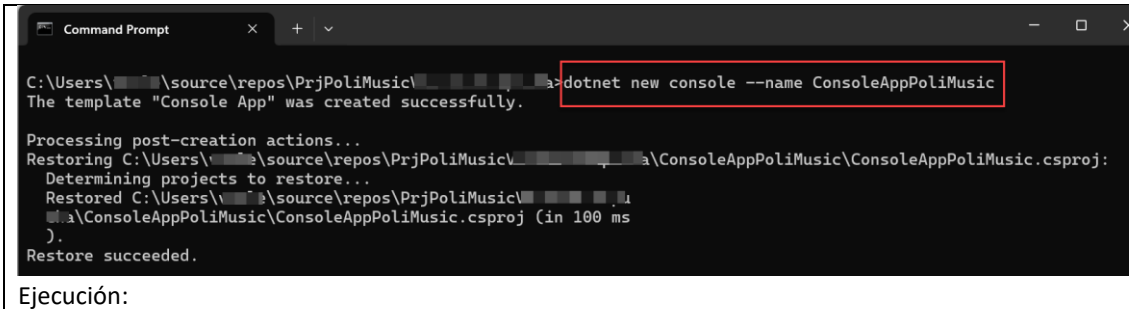
Y desde la consola podemos ejecutar estos comandos:

Comando	Descripción
dotnet new --help	Lista las opciones de ayuda para "dotnet new"

	
<p>dotnet new list</p> 	<p>Lista los proyectos que puede crear</p>

De todo el listado de proyectos que puede crear, crearemos uno de tipo “console” con lenguaje C# con el nombre **“ConsoleAppPoliMusic”**, por lo que el comando a ejecutar es:

`dotnet new console --name ConsoleAppPoliMusic`


--

Ejecución:


```
reposAzureDevOps

For more information, run:
  dotnet new console -h

For details on the exit code, refer to https://aka.ms/templating-exit-codes#127

1.889s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>_: dotnet new console --name ConsoleAppPoliMusic
The template "Console App" was created successfully.

Processing post-creation actions...
Restoring C:\Users\Fernando_Huilca\Documents\Git-Fernando_Huilca\reposAzureDevOps\ConsoleAppPoliMusic\ConsoleAppPoliMusic.csproj:
  Determining projects to restore...
  Restored C:\Users\Fernando_Huilca\Documents\Git-Fernando_Huilca\reposAzureDevOps\ConsoleAppPoliMusic\ConsoleAppPoliMusic.csproj (in 46 ms).
Restore succeeded.

0.999s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>: |
```

6. Colocar su nombre y apellido en archivo "Program.cs"

En la consola, ingresamos a la carpeta "ConsoleAppPolimusic" con el comando "cd ConsoleAppPolimusic"

```
Command Prompt

C:\Users\...source\repos\PrjPoliMusic>cd ConsoleAppPoliMusic
C:\Users\...source\repos\PrjPoliMusic\ConsoleAppPoliMusic>

0.999s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>_: pwd
/c/Users/Fernando_Huilca/Documents/Git-Fernando_Huilca/reposAzureDevOps

0.055s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps
>_: cd ConsoleAppPoliMusic/

0.057s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > ConsoleAppPoliMusic
>: |
```

Abra el archivo Program.cs, puede hacerlo desde la consola con este comando: notepad Program.cs

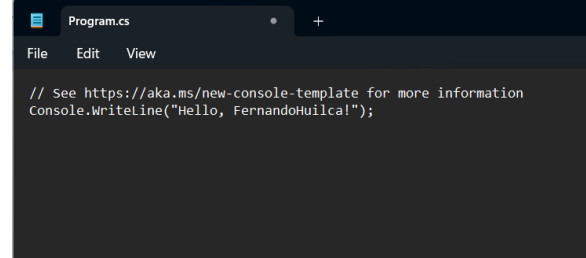
```
C:\Users\...source\repos\PrjPoliMusic\ConsoleAppPoliMusic>notepad Program.cs
```

Realice estos cambios.

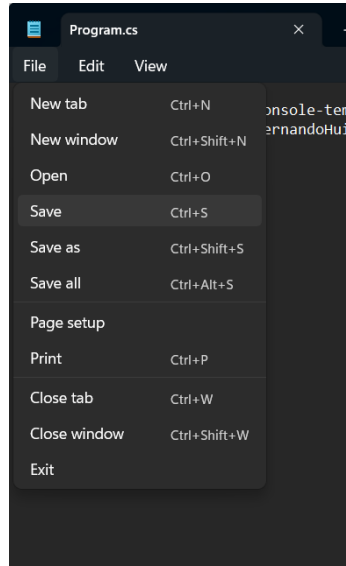
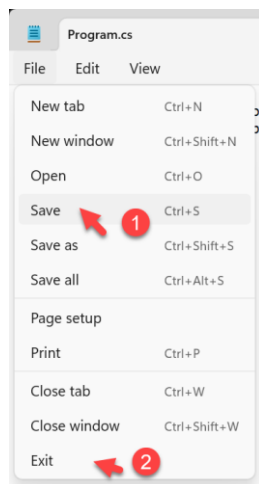
Antes	Después
// See https://aka.ms/new-console-template for more information	// See https://aka.ms/new-console-template for more information

```
Console.WriteLine("Hello, World!");
```

```
Console.WriteLine("Hello, NombreApellidoEstudiante!");
```



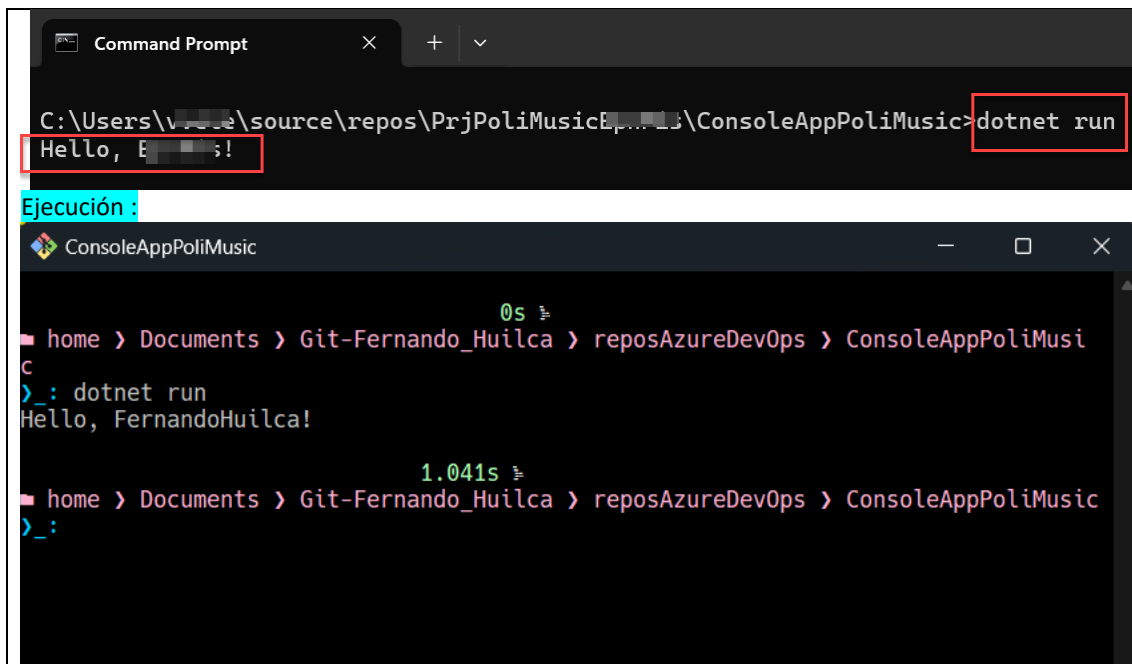
Y finalmente, guarde los cambios del archivo Program.cs y cierre el Notepad.



7. Ejecución de proyecto de consola

Ejecutar el comando “dotnet run”

El comando “dotnet run”, internamente realiza la compilación, es decir un “dotnet build” y si no existen errores ejecuta el programa. A continuación, se muestra la ejecución del programa, donde se muestra su nombre y apellido.



The image shows a Windows Command Prompt window with the following text:

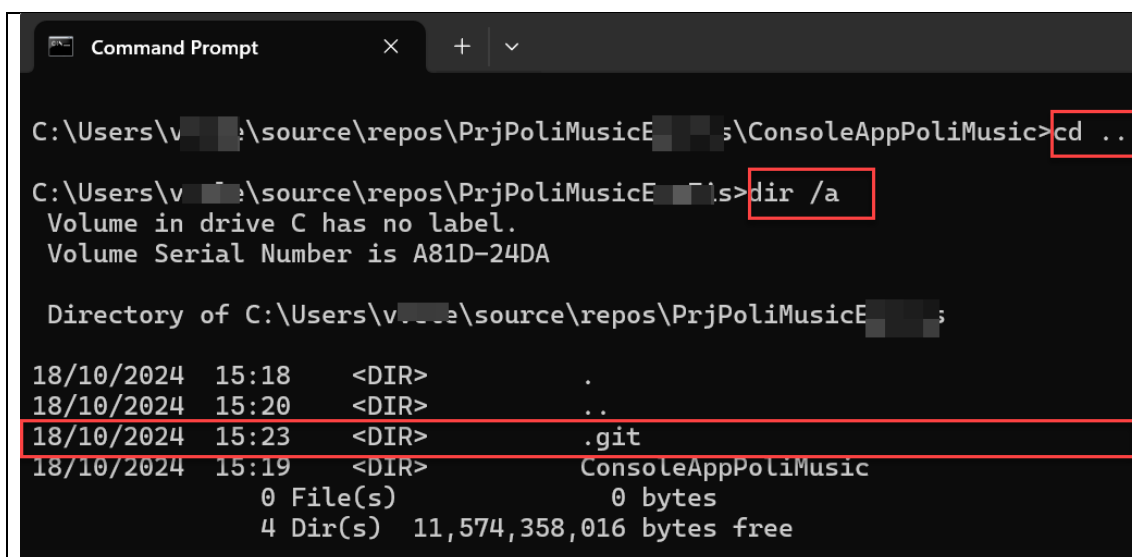
```
C:\Users\fernando\source\repos\PrjPoliMusicEjemplo\ConsoleAppPoliMusic>dotnet run
Hello, FernandoHuilca!
```

Below the Command Prompt, there is a console application window titled "ConsoleAppPoliMusic". It shows the same output as the Command Prompt, with a timestamp of 0s and 1.041s.

8. Subir código fuente en AzureDevOps

Para subir el código, lo primero que debemos hacer añadir al escenario (en inglés **stage**) los archivos que queremos versionar. Se versiona solo código fuente (archivos .cs), archivos de configuración (.config), el archivo de proyecto (.csproj) y el archivo de la solución (.sln). Las carpetas bin, obj y su contenido interno nunca se añade al Git.

Lo primero que debemos realizar es desde la consola, subírnos 1 nivel, usando el comando "cd .." y listamos el contenido de esta carpeta incluyendo archivos ocultos, con el comando "dir /a". Se mostrará la carpeta oculta ".git" como se indica en la siguiente imagen:



The image shows a Windows Command Prompt window with the following text:

```
C:\Users\fernando\source\repos\PrjPoliMusicEjemplo\ConsoleAppPoliMusic>cd ..
C:\Users\fernando\source\repos\PrjPoliMusicEjemplo>dir /a
Volume in drive C has no label.
Volume Serial Number is A81D-24DA

Directory of C:\Users\fernando\source\repos\PrjPoliMusicEjemplo;

18/10/2024  15:18    <DIR>          .
18/10/2024  15:20    <DIR>          ..
18/10/2024  15:23    <DIR>          .git
18/10/2024  15:19    <DIR>          ConsoleAppPoliMusic
               0 File(s)                0 bytes
               4 Dir(s)  11,574,358,016 bytes free
```

```
PrjPoliMusicFernandoHuilca

0s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca > ConsoleAppPoliMusic (master)
>: cd ..

0.056s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (master)
>: ls -a
./ ../ .git/ ConsoleAppPoliMusic/

0.069s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (master)
>:
```

Entonces, una vez que estamos a la altura (mismo nivel) de la carpeta “.git”, debemos crear un archivo llamado “.gitignore”, para lo cual desde la terminal podemos hacerlo así: “echo. > .gitignore” (de forma alternativa podría usar “type NUL > .gitignore” o en Linux/Mac o Windows desde Git bash con “touch .gitignore”), y verificamos que se haya creado con el comando “dir /a”

```
Command Prompt

C:\Users\vele\source\repos\PrjPoliMusicEpnFis>echo. > .gitignore

C:\Users\vele\source\repos\PrjPoliMusicEpnFis>dir /a
Volume in drive C has no label.
Volume Serial Number is A81D-24DA

Directory of C:\Users\vele\source\repos\PrjPoliMusicEpnFis

18/10/2024  15:32    <DIR>          .
18/10/2024  15:20    <DIR>          ..
18/10/2024  15:23    <DIR>          .git
18/10/2024  15:32             3 .gitignore
18/10/2024  15:19    <DIR>          ConsoleAppPoliMusic
                1 File(s)                3 bytes
                4 Dir(s)  11,556,859,904 bytes free

C:\Users\vele\source\repos\PrjPoliMusicEpnFis>
```

```
PrjPoliMusicFernandoHuilca
doHuilca > ConsoleAppPoliMusic (% master)
>: cd ..

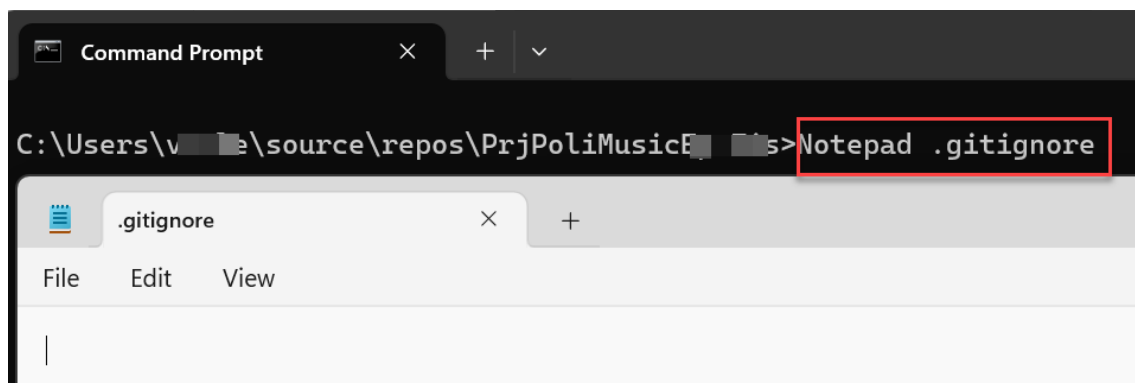
0.056s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernan
doHuilca (% master)
>: ls -a
./ ../ .git/ ConsoleAppPoliMusic/

0.069s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernan
doHuilca (% master)
>: touch .gitignore

0.133s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernan
doHuilca (% master)
>: ls -a
./ ../ .git/ .gitignore ConsoleAppPoliMusic/

0.07s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernan
doHuilca (% master)
>: |
```

Editamos el archivo creado escribiendo desde la consola "Notepad .gitignore"



En su contenido colocamos lo siguiente:

```
# Ignore bin and obj folders
bin/
obj/

# Ignore build results
*.exe
*.dll
*.pdb

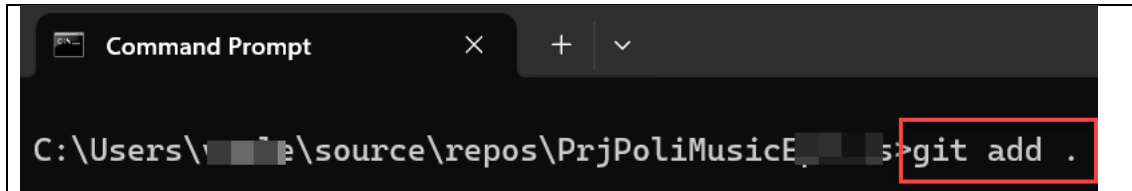
# Ignore temporary files
*.tmp
*.log
*.user
*.suo
*.cache
*.csproj.user

# Ignore publish outputs
```

```
publish/
```

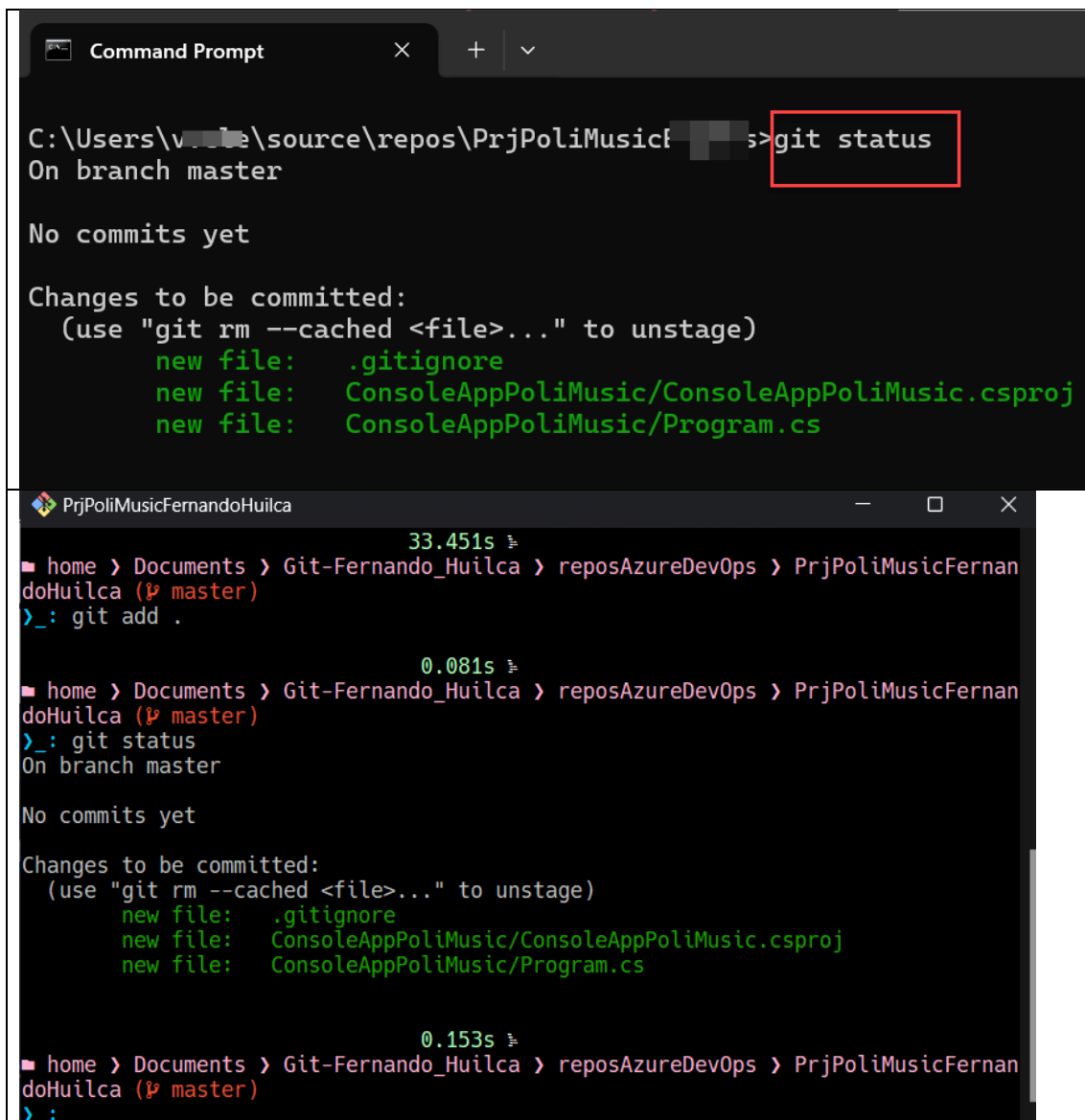
Cerramos el archivo “.gitignore” guardando los cambios y cerramos el notepad.

Ahora desde la consola ejecutamos el comando “git add .”, tomando en cuenta que el carácter “.” en git add significa que añadirá al stage todos los archivos, exceptuando los que estén especificados en el archivo .gitignore.



```
Command Prompt
C:\Users\...source\repos\PrjPoliMusicE...>git add .
```

Verificamos los archivos que están en el stage con el comando “git status”, deberá aparecer una pantalla similar a la siguiente:



```
Command Prompt
C:\Users\...source\repos\PrjPoliMusicE...>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore
    new file:   ConsoleAppPoliMusic/ConsoleAppPoliMusic.csproj
    new file:   ConsoleAppPoliMusic/Program.cs

PrjPoliMusicFernandoHuilca
33.451s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (P master)
>: git add .

0.081s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (P master)
>: git status
On branch master

No commits yet

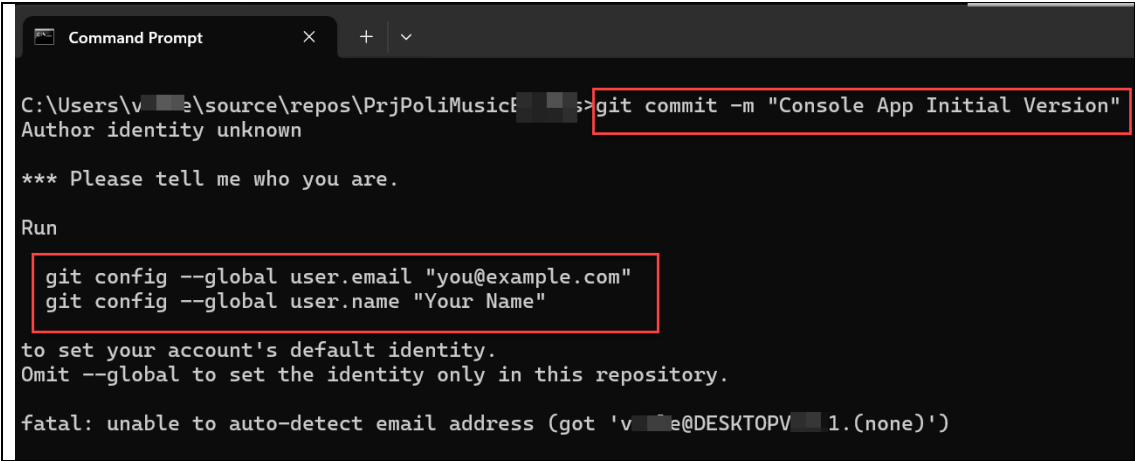
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore
    new file:   ConsoleAppPoliMusic/ConsoleAppPoliMusic.csproj
    new file:   ConsoleAppPoliMusic/Program.cs

0.153s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (P master)
>:
```

Nota: Si ejecuta “git add .”, sin tener el archivo “.gitignore” puede deshacer los cambios con el comando “git reset”. En este caso los archivos están correctos, por lo que **no** usaremos el comando “git reset”

Una vez verificado que los archivos correctos están en el stage, el próximo paso es realizar un commit, para indicar que esta es nuestra primera versión, para ello usamos el comando “git commit -m "Console App Initial Version"”.

Es posible que, si es la primera vez que usa git, es decir, que recién instaló git en el computador, le presente este mensaje solo la primera vez:



```
Command Prompt
C:\Users\...e\source\repos\PrjPoliMusicf...s>git commit -m "Console App Initial Version"
Author identity unknown

*** Please tell me who you are.

Run

git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'v...e@DESKTOPV... 1.(none)')
```

Si no le aparece este mensaje, ignore estos pasos indicados a continuación, de configuración del nombre y correo. Para solventar este inconveniente seguimos los pasos indicados a continuación, en este caso añadimos el “Nombre Apellido” preferentemente si tildes, y el correo electrónico de la universidad:

Comando	Descripción
git config --global user.name "Nombre Apellido"	Establece valores de configuración para tu usuario
git config --global user.email "usuario@dominio.com"	Establece valores de configuración para tu email
git config --list	Lista configuración de git, donde a más de la configuración por defecto debe visualizar sus datos.

```
Command Prompt
C:\Users\v\source\repos\PrjPoliMusic\>git config --global user.email "v@epn.edu.ec"
C:\Users\v\source\repos\PrjPoliMusic\>git config --global user.name "v"
C:\Users\v\source\repos\PrjPoliMusic\>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=v@epn.edu.ec
user.name=v
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://ISWR2@dev.azure.com/ISWR2/PrjPoliMusic/_git/PrjPoliMusic
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
```

Personalmente ya tenía configurado mi nombre y correo con el git bash, entonces solo adjunto la captura del commit

```
PrjPoliMusicFernandoHuilca
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore
    new file:   ConsoleAppPoliMusic/ConsoleAppPoliMusic.csproj
    new file:   ConsoleAppPoliMusic/Program.cs

0.153s
■ home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (P master)
> : git commit -m "Console App Initial Version"
[master (root-commit) b25f04d] Console App Initial Version
3 files changed, 31 insertions(+)
create mode 100644 .gitignore
create mode 100644 ConsoleAppPoliMusic/ConsoleAppPoliMusic.csproj
create mode 100644 ConsoleAppPoliMusic/Program.cs

0.17s
■ home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (P master)
> :
```

Finalmente intentamos nuevamente realizar el commit, ahora ya debe funcionar, como se muestra en la imagen:


```
Command Prompt
C:\Users\...\source\repos\PrjPoliMusicE...>git commit -m "Console App Initial Version"
[master (root-commit) 6ddf049] Console App Initial Version
3 files changed, 31 insertions(+)
create mode 100000 .gitignore
create mode 100000 ConsoleAppPoliMusic/ConsoleAppPoliMusic.csproj
create mode 100000 ConsoleAppPoliMusic/Program.cs
C:\Users\...\source\repos\PrjPoliMusicE...>
```

Para verificar que efectivamente todo esta correcto, nos apoyamos del comando “git log” o si queremos ver información alterna usamos el comando “git reflog” como se muestra a continuación:

```
Command Prompt
C:\Users\...\source\repos\PrjPoliMusicE...>git log
commit 6ddf049abb5fec7d11440bfd7ea278154ba6c82e (HEAD -> master)
Author: V... <v...@epn.edu.ec>
Date: Fri Oct 18 15:58:00 2024 -0500

    Console App Initial Version
C:\Users\...\source\repos\PrjPoliMusicE...>git reflog
6ddf049 (HEAD -> master) HEAD@{0}: commit (initial): Console App Initial Version
```

Hasta aquí, su código fuente está únicamente en su GIT de su computadora local, pero necesitamos que usted pueda continuar trabajando sobre este código fuente desde cualquier computadora y desde cualquier parte del mundo, para lo cual debemos subir este código en un repositorio en la nube como GitHub, GitLab, BitBucket, o en nuestro caso será en AzureDevOps. Para hacerlo finalmente ejecutamos este comando “git push” Como habíamos clonado anteriormente de AzureDevOps en el comando “git push” no necesitamos especificar más valores.

```
Command Prompt
C:\Users\...\source\repos\PrjPoliMusicE...>git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 772 bytes | 257.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Analyzing objects... (6/6) (5 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (75 ms)
remote: Storing index... done (32 ms)
To https://dev.azure.com/ISWR.../PrjPoliMusicE.../_git/PrjPoliMusicE...
* [new branch]      master -> master
```

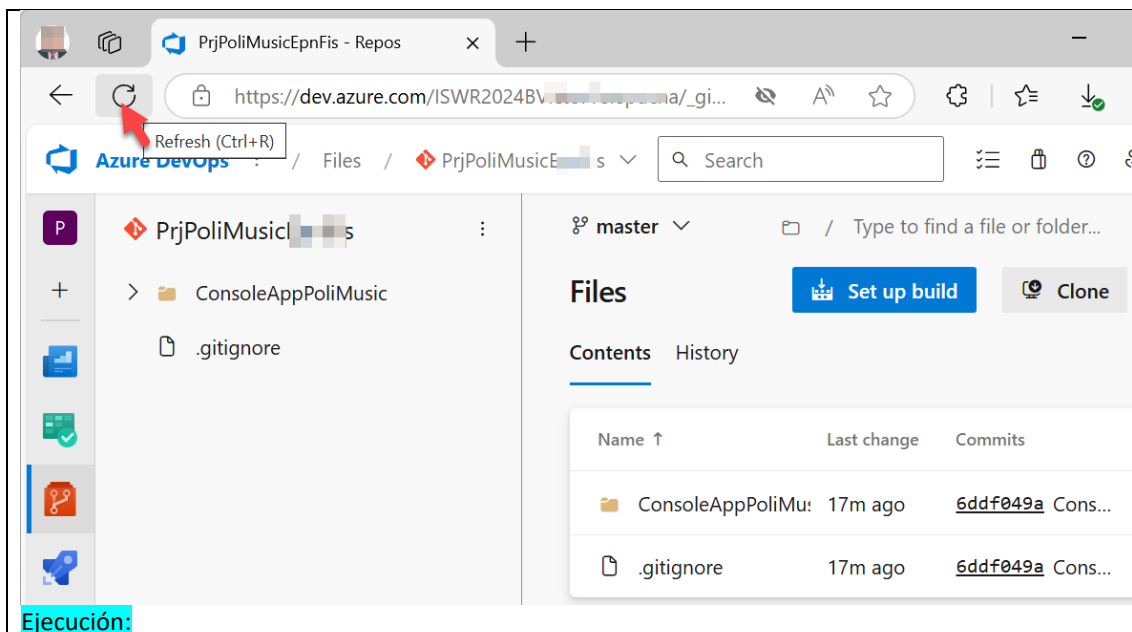
Ejecución:

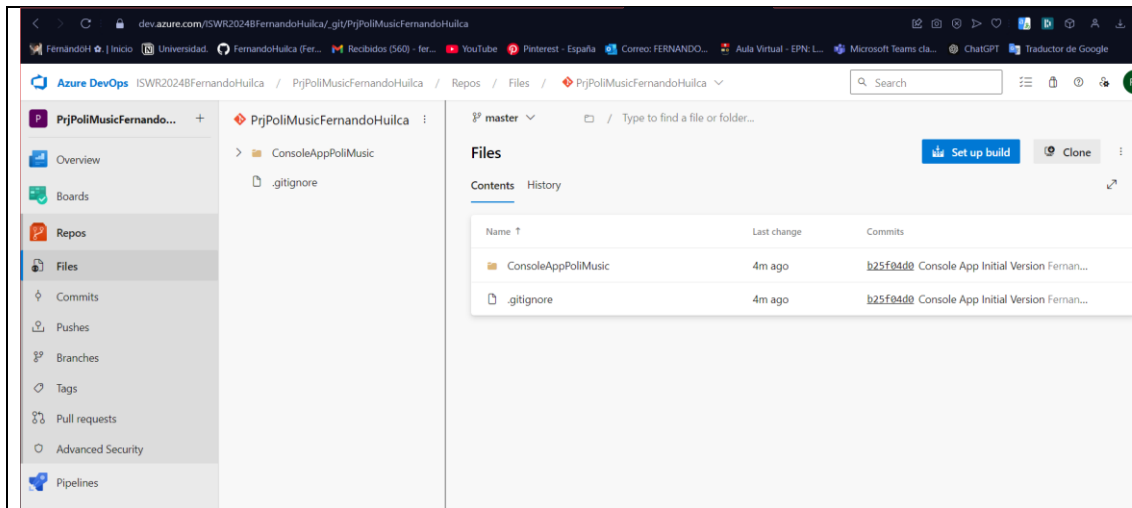
```
PrjPoliMusicFernandoHuilca
create mode 100644 ConsoleAppPoliMusic/Program.cs

0.17s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (master)
> git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 777 bytes | 777.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Analyzing objects... (6/6) (4 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (73 ms)
remote: Storing index... done (32 ms)
To https://dev.azure.com/ISWR2024BFernandoHuilca/PrjPoliMusicFernandoHuilca/_git/PrjPoliMusicFernandoHuilca
* [new branch]      master -> master

2.635s
home > Documents > Git-Fernando_Huilca > reposAzureDevOps > PrjPoliMusicFernandoHuilca (master)
> :
```

Finalmente, corroboramos que nuestro código se encuentre en la nube de Azure DevOps, refrescando desde el navegador.





Y ahora como su código ya se encuentra en Azure DevOps, usted desde la universidad, su casa o cualquier lugar del mundo, puede clonarlo y seguir trabajando para añadir más funcionalidad.