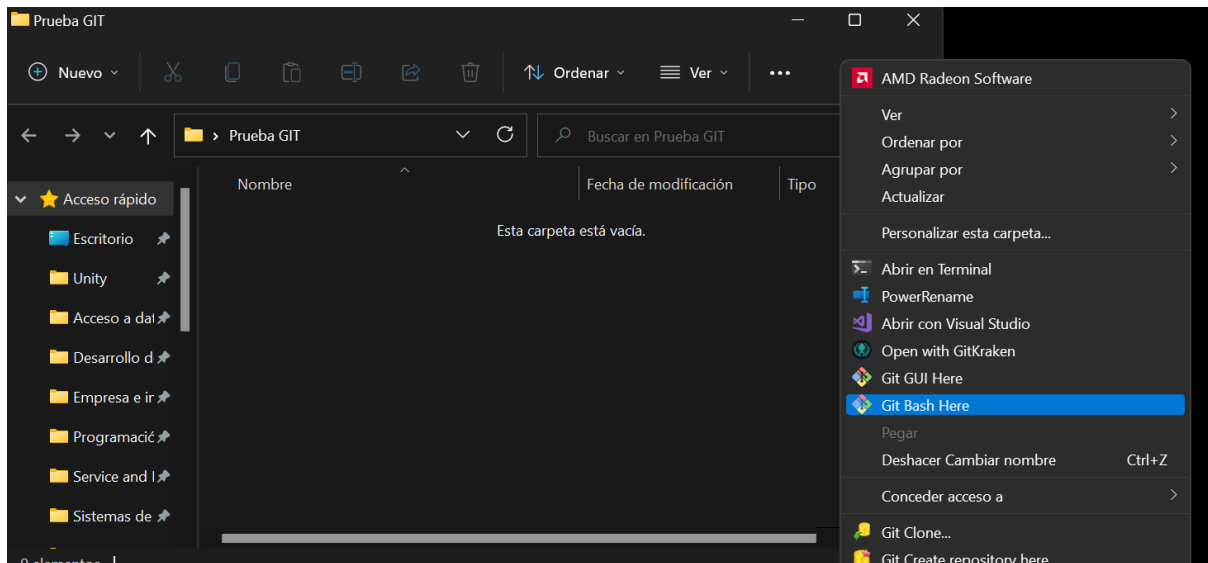


Pulsamos botón derecho dentro de la carpeta que quieras versionar (meter a GIT), y le damos a Git Bash Here (esa opción la tenemos con lo que instalamos el año pasado de Git en la asignatura de entornos):



Lo primero creamos el .gitignore, para excluir del control de versiones las carpetas que no queremos (en el caso de PMDM, las carpetas Temp, Build y Library).

Con el touch creamos el archivo y con el nano lo modificamos:

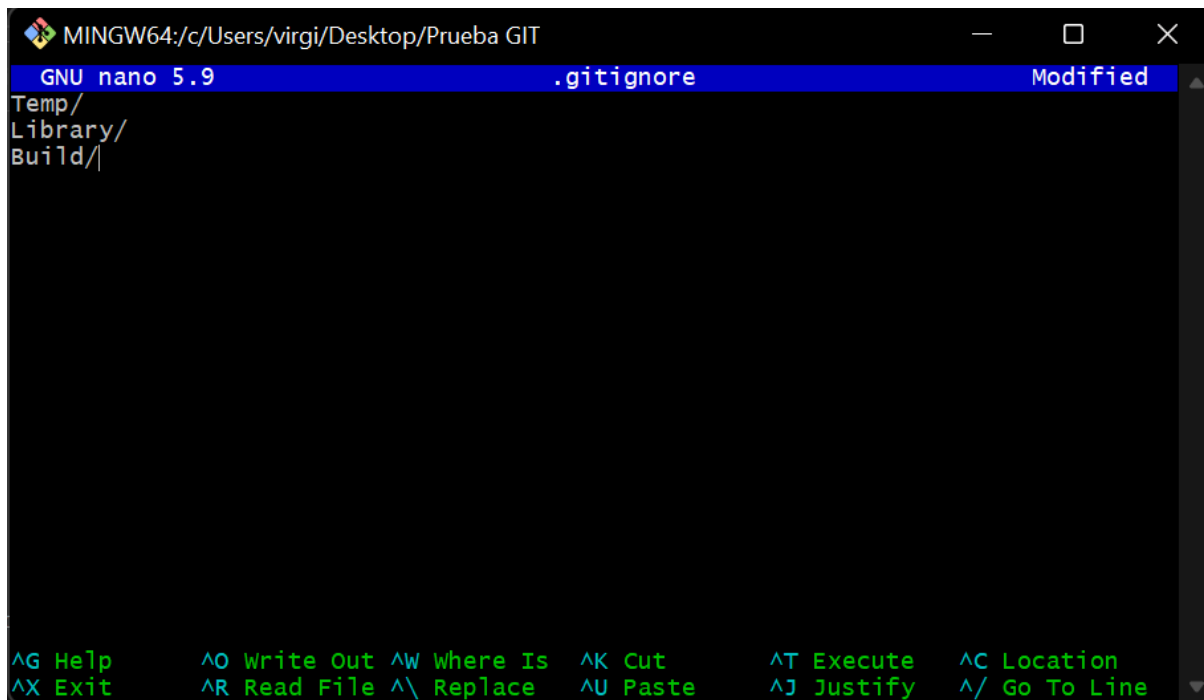
```
MINGW64:/c/Users/virgi/Desktop/Prueba GIT

virgi@Virginia-pc MINGW64 ~/Desktop/Prueba GIT
$ touch .gitignore

virgi@Virginia-pc MINGW64 ~/Desktop/Prueba GIT
$ nano .gitignore

virgi@Virginia-pc MINGW64 ~/Desktop/Prueba GIT
$ |
```

Añadimos esas líneas en la ventana que nos aparece (Control X para salir y guardar):



The screenshot shows a terminal window titled 'MINGW64:/c/Users/virgi/Desktop/Prueba GIT'. Inside, the GNU nano 5.9 editor is open, editing the .gitignore file. The file content is: Temp/, Library/, and Build/. The nano editor's status bar at the bottom shows various keyboard shortcuts like ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^_/ Go To Line.

```
MINGW64:/c/Users/virgi/Desktop/Prueba GIT
GNU nano 5.9 .gitignore Modified
Temp/
Library/
Build/

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_/ Go To Line
```

Cuando ya tenemos el .gitignore creado, vamos a decirle que añada el contenido de la carpeta en el control de versiones con la instrucción:

```
virgi@Virginia-pc MINGW64 ~/Desktop/Prueba GIT
$ git init .
```

Eso inicializa Git en la carpeta.

Luego le tenemos que decir que añada todos los archivos de la carpeta al control de versiones (lo que hay en el .gitignore lo va a ignorar):

```
virgi@Virginia-pc MINGW64 ~/Desktop/Prueba GIT
$ git add .
```

Para hacer un commit:

```
virgi@Virginia-pc MINGW64 ~/Desktop/Prueba GIT
$ git commit -am "Primer commit"
```

Y si quieres subirlo a algun sitio como Git Hub tienes que poner esta instrucción:
git remote add origin <https://github.com/virginia-oc/Prueba.git>

En la URL pones la de tu repositorio, que has tenido que crear previamente.
Y con: git push , subes todo al repositorio remoto.

Cuando estés trabajando y hagas algún cambio en el proyecto siempre hay que seguir los mismos pasos:

git add . -> Para añadir todos los archivos nuevos que se hayan podido crear.

git commit -am "Descripcion del commit" -> para registrar el cambio.

git push -> subir al repositorio remoto los cambios hechos.