

## Subtarea 1: Uso de Git y GitHub

Creamos el repositorio público en GitHub que corresponde con el siguiente enlace:

[https://github.com/kelpis/ENDES\\_T2\\_tarea.git](https://github.com/kelpis/ENDES_T2_tarea.git)

Clonamos el repositorio “vacío” a nuestro PC y tomamos una captura de pantalla de los siguientes pasos.

- Dentro de la carpeta del repositorio, ejecutamos el comando “ls” que muestra el contenido de la carpeta, en este caso el readme.txt

```
MINGW64/c/Users/Villa/repos2/ENDES_T2_tarea
Villa@GEMMA MINGW64 ~
$ mkdir repos2
Villa@GEMMA MINGW64 ~
$ cd repos2
Villa@GEMMA MINGW64 ~/repos2
$ ls
Villa@GEMMA MINGW64 ~/repos2
$ git clone https://github.com/kelpis/ENDES_T2_tarea.git
Cloning into 'ENDES_T2_tarea'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
Villa@GEMMA MINGW64 ~/repos2
$ ls
ENDES_T2_tarea/
Villa@GEMMA MINGW64 ~/repos2
$ cd ENDES_T2_tarea/
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ ls
README.md
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ |
```

- Resultado de ejecutar git status cuando el repositorio está "al día".

```
MINGW64/c/Users/Villa/repos2/ENDES_T2_tarea
Villa@GEMMA MINGW64 ~
$ cd repos2
Villa@GEMMA MINGW64 ~/repos2
$ ls
Villa@GEMMA MINGW64 ~/repos2
$ git clone https://github.com/kelpis/ENDES_T2_tarea.git
Cloning into 'ENDES_T2_tarea'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
Villa@GEMMA MINGW64 ~/repos2
$ ls
ENDES_T2_tarea/
Villa@GEMMA MINGW64 ~/repos2
$ cd ENDES_T2_tarea/
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ ls
README.md
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ |
```

- Resultado de ejecutar git status cuando hay un archivo nuevo que git todavía no está rastreando.

```
MINGW64/c/Users/Villa/repos2/ENDES_T2_tarea
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ cd ..
Villa@GEMMA MINGW64 ~/repos2
$ ENDES_T2_tarea/
bash: ENDES_T2_tarea/: Is a directory
Villa@GEMMA MINGW64 ~/repos2
$ cd ENDES_T2_tarea/
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ pwd
/c/Users/Villa/repos2/ENDES_T2_tarea
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ code readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$
```

- Resultado de ejecutar git status cuando hay archivos en la fase staged, es decir; están preparados para hacer un commit.

```
MINGW64/c/Users/Villa/repos2/ENDES_T2_tarea
/c/Users/Villa/repos2/ENDES_T2_tarea
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ code readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git add readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ 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)
  new file:   readme.txt

Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ |
```

- Resultado de que el repositorio local este al día tras haber realizado cambios, pero no está sincronizado con GitHub.

```

MINGW64:/c/Users/Villa/repos2/ENDES_T2_tarea
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ pwd
/c/Users/Villa/repos2/ENDES_T2_tarea
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ code readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git add readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ 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)
    new file:   readme.txt

Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git commit -m "Empezamos con la subtask 2 del readme.txt"
[main 9c12680] Empezamos con la subtask 2 del readme.txt
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$

```

- Resultado de sincronizar nuestro repositorio Git local con el repositorio en GitHub

```

MINGW64:/c/Users/Villa/repos2/ENDES_T2_tarea
nothing to commit, working tree clean
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git add readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ 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)
    new file:   readme.txt

Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git commit -m "Empezamos con la subtask 2 del readme.txt"
[main 9c12680] Empezamos con la subtask 2 del readme.txt
 1 file changed, 1 insertion(+)
 create mode 100644 readme.txt
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 367 bytes | 367.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kelpis/ENDES_T2_tarea.git
   ed59af7..9c12680  main -> main
Villa@GEMMA MINGW64 ~/repos2/ENDES_T2_tarea (main)
$

```

## Subtarea 2: Comparación de dos IDEs o editores de texto

Para esta tarea me decante por los dos editores de texto que conozco y de momento he usado, el Bloc de notas (Notepad) y el Visual Studio Code.

El Bloc de notas es un editor de texto incluido en los sistemas operativos Windows desde 1985 con una funcionalidad muy simple.

Mientras que el Visual Studio Code es un editor de código fuente y una herramienta de programación avanzada de código abierto. Se caracteriza por su potencia y por ser altamente personalizable y extensible.

### Lenguajes de programación compatibles

En el Bloc de notas se podría escribir cualquier lenguaje ya que esencialmente es un editor de texto, aunque básico. Algunos de los lenguajes de programación más comunes que incluyen serían:

- Python
- Java
- C/C++
- HTML, CSS, JavaScript

VS Code al tratarse de un entorno de desarrollo, es compatible con todos los lenguajes populares y se puede alternar entre dichos lenguajes. Gracias al soporte integrado que tiene y a las extensiones disponibles se pueden descargar lenguajes adicionales.

A parte de leer los lenguajes que el Bloc notas lee, VS Code también es compatible con:

- TypeScript
- C#
- PHP
- GO
- Ruby
- Rust
- F#
- Visual Basic

### Facilidad de uso

La facilidad de uso del Notepad deriva principalmente de su simplicidad y funcionalidad directa. Algunas características que contribuyen a su facilidad de uso son:

- Interfaz muy simple
- Sin características complicadas
- Rápido inicio
- No requiere conocimientos especiales

VS Code es conocido también por su facilidad de uso, aunque al tratarse de una herramienta más avanzada su curva de aprendizaje es ligeramente superior. Algunos aspectos de su uso son:

- Interfaz intuitiva
- Instalación sencilla
- Soporte multiplataforma
- Resaltado de sintaxis y autocompletado

### Depuración

Notepad no dispone de herramientas integradas de depuración como lo hace VS Code, que cuenta con características específicas para encontrar y solucionar errores en el código fuente. Nos da la opción de ir ejecutando los programas línea por línea y detener la ejecución en puntos específicos (llamados puntos de interrupción) y examinar el estado del programa en cualquier momento.

### Gestión de versiones

De nuevo el Bloc de notas no incluye funcionalidades avanzadas de gestión de versiones al ser una aplicación de texto simple. Respecto a VS Code, es compatible con varios sistemas de control de versiones, pero tampoco tiene un gestor de versiones incorporado que proporcione directamente esa funcionalidad. Lo que si integra, y el bloc de notas no, es un sistema de control de versiones, pero externo utilizando extensiones. Las extensiones más comunes son:

- Git
- GitHub
- Bitbucket
- Mercurial

### Personalización y extensibilidad

Aunque el Bloc de notas sea una aplicación simple, se puede personalizar y extender su funcionalidad en cierta medida. Por ejemplo:

- Cambiar fuente y el tamaño
- Configurar márgenes y saltos de línea
- Cambiar los temas del color

Por el contrario, VS Code se caracteriza por su énfasis en la personalización y extensibilidad. Algunas de sus características son:

- Permite instalar, explorar y administrar extensiones desde su mercado de extensiones.
- Cambiar de tema; con temas ya disponibles o nuevos con extensiones descargables.
- Permite ajustar aspectos específicos según las necesidades del usuario.
- Se pueden definir atajos para el teclado personalizados.
- Otros aspectos que se pueden personalizar y configurar también son:
  - Snippets
  - La apariencia de la terminal
  - El entorno de depuración

### Otros aspectos de interés

El VS Code se considera la evolución del Bloc de notas ya que ambos editores fueron desarrollados por Microsoft.

El primero se diseñó como sustituto y alternativa al primero para poder dar respuesta a tareas cada vez más avanzadas.

## Subtarea 3: Tu herramienta favorita.

Con los conocimientos que hasta ahora tengo no me atrevería a decir que IDE o herramienta CASE me podría ayudar más en mí el futuro. Me voy a basar en uno de los IDEs que de momento he podido utilizar, Apache NetBeans.

Netbeans es básicamente un entorno de desarrollo integrado (IDE) de código abierto que utilizamos para el desarrollo de aplicaciones en lenguaje de programación Java.

Aunque no se clasifica como una herramienta CASE, Netbeans tiene características que pueden ser consideradas como parte del desarrollo de software y del enfoque CASE.

Algunas de las características que tiene este IDE y que creo que me van a ahorrar trabajo son:

- Tiene soporte para varios lenguajes de programación; cuando aprenda más lenguajes será útil que sean compatibles en el mismo IDE.
- Su editor de código tiene la función de resaltado de sintaxis que facilita la correcta escritura del código.
- Autocompletado y plantillas de código; que ayudan a escribir más rápido el código.
- Integra sistemas de control de versiones en este caso Git, para poder crear repositorio de los proyectos que haga.
- Depurador; ayuda a encontrar y corregir los errores en los programas

Posee otras funcionalidades que aun no use, pero que puedo intuir que me van a facilitar la labor:

- Herramientas de base de datos: proporciona herramientas para conectar y gestionar bases de datos.
- Constructor de interfaz gráfica de usuario: facilita la creación de interfaces visuales para aplicaciones Java.
- Extensiones: para incorporar nuevas funcionalidades.