

# DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

# INGENIERIA EN SISTEMAS COMPUTACIONALES

# NOMBRE DE LA MATERIA:

***Ingeniería de Software***

***Docente: Euri Salgado Escobar***

***Unidad 3 - Desarrollo***

**Práctica: P1 Instala Git**

**Alumno: Álvarez Ortiz Juan Manuel 18070519**

**PRÁCTICA**

|  |  |  |  |
| --- | --- | --- | --- |
| Departamento Académico | Sistemas y Computación | | |
| Programa Académico | Plan de Estudios | Formato | |
| Ing. en Sistemas Computacionales | ISIC-2010-224 | SyC-03-2016-P | |
| Asignatura | Clave de la Asignatura | Créditos SATCA | Semestre |
| Ingeniería de software | SCD-1011 | 2-3-5 | 7mo |

|  |  |
| --- | --- |
| No. de Práctica | Nombre de la Práctica |
| P\_01\_01 | Instalación de Git |
| Competencia de la Práctica | |
| Conoce cómo instalar la herramienta Git | |

1. **INTRODUCCIÓN**

**Sistema de control de versiones**

Un controlador de versiones es un sistema que nos permite guardar un registro de las modificaciones que realizamos sobre un fichero o conjunto de ficheros a lo largo del tiempo de tal manera que sea posible recuperar versiones específicas más adelante. Ejemplos de estos sistemas son: Git, CVS, Apache Subversión (SVN), Mercurial y Monotone.

Un sistema de control de versiones debe proporcionar:

* Mecanismo de almacenamiento de los elementos que deba gestionar (ej. archivos de texto, imágenes, documentación...).
* Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos).
* Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto).

**Git**

Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

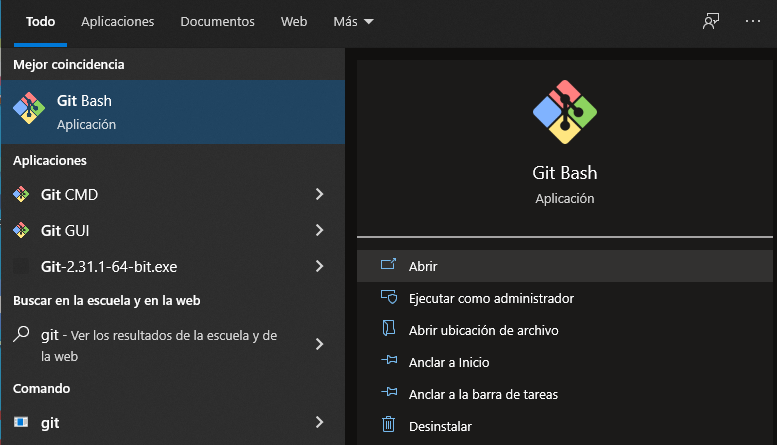
En pocas palabras, GIT es un sistema que te ayuda a organizar el código de trabajo, el historial y evolución del mismo. Funciona como una máquina del tiempo que te permite navegar a diferentes versiones de tu proyecto.

Es multiplataforma, así que está disponible para sistemas como Linux, Windows, Mac OS, Solaris, OpenBSD.

1. **REQUERIMIENTOS**

|  |  |
| --- | --- |
| Equipo | Software |
| Computadora con sistema operativo compatible con Git. | -Software Git.  -Editor de texto (Bloc de notas, Notepad++, Atom, Sublime Text). |

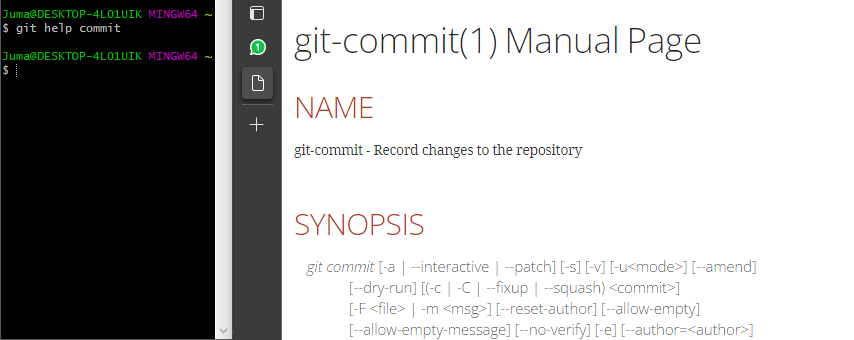
1. **DESCRIPCIÓN DE LA PRÁCTICA.**
2. Descargar e instalar Git. El sitio web es <https://git-scm.com/>. Después de realizar la instalación, nos quedaran dos programas: Git GUI (con interface gráfica) y Git Bash (con línea de comandos).



1. Abrir git bash y verificar la versión instalada mediante el comando git --version.



1. Git nos proporciona ayuda para conocer el uso de sus diferentes comandos, por ejemplo, utilizar el comando git help commit para ver la documentación de commit.



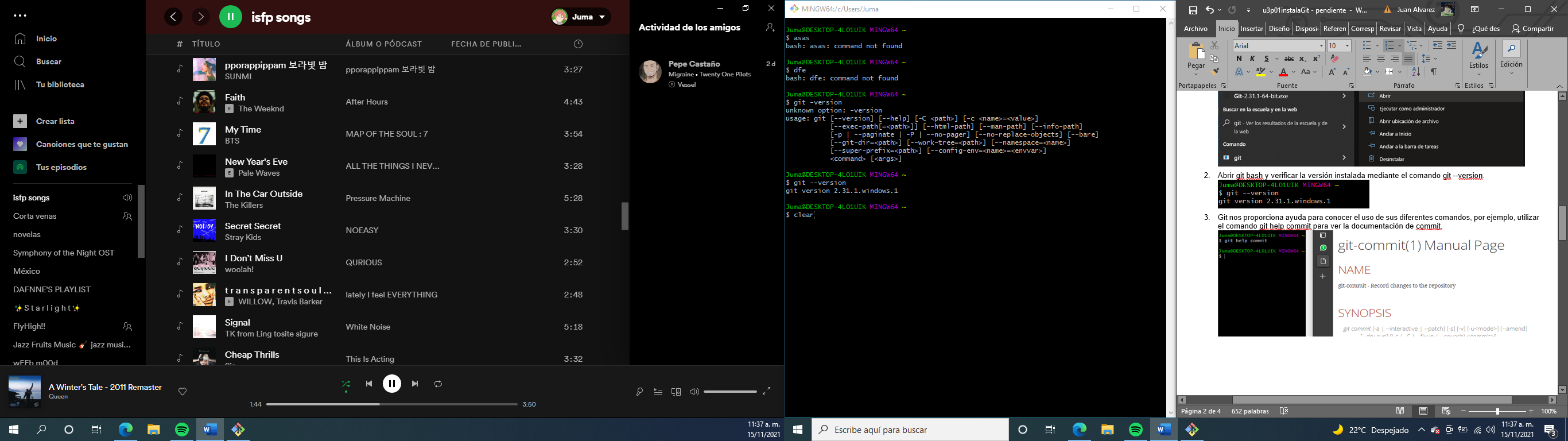
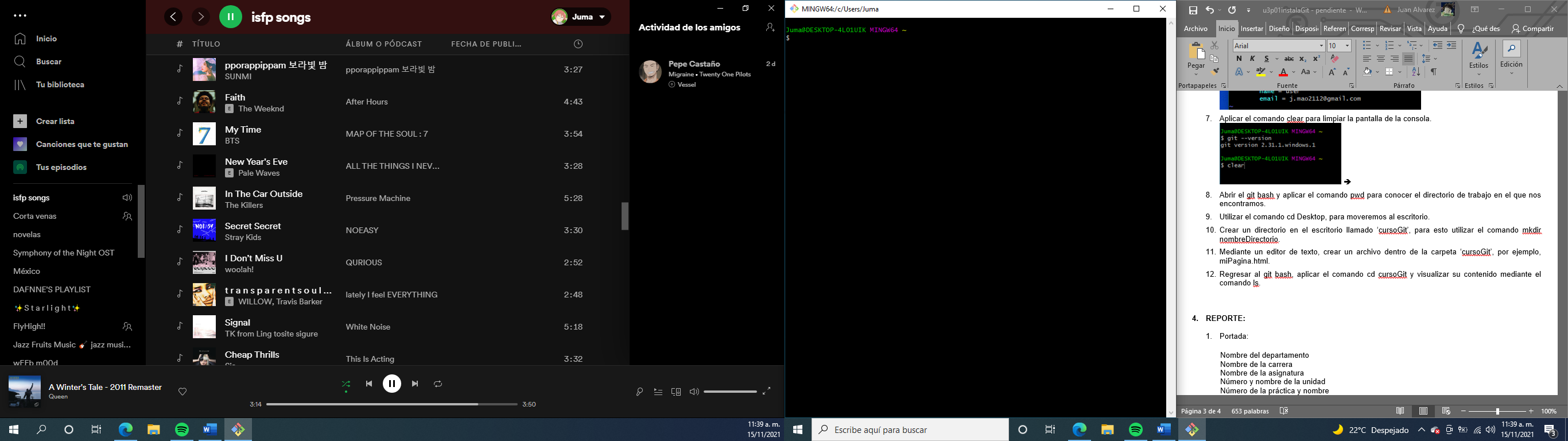
1. Establecer nuestro nombre de usuario en Git, para esto, utilizar el comando git config --global user.name “user”, siendo user, nuestro nombre de usuario.



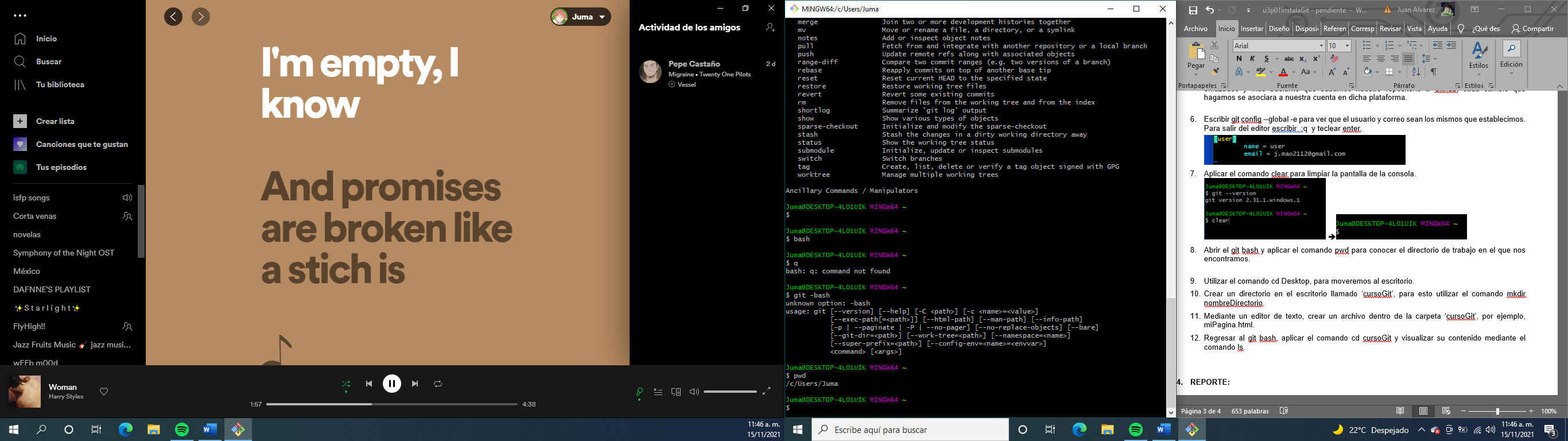
1. Establecer un correo, con el comando git config --user.mail direcciónCorreo. Si ya se tiene una cuenta en Github, utilizar el correo de dicha cuenta, si no, podemos crear una para estar enlazados y más adelante que subamos nuestro repositorio a Github, cada cambio que hagamos se asociara a nuestra cuenta en dicha plataforma.
2. Escribir git config --global -e para ver que el usuario y correo sean los mismos que establecimos. Para salir del editor escribir :q y teclear enter.



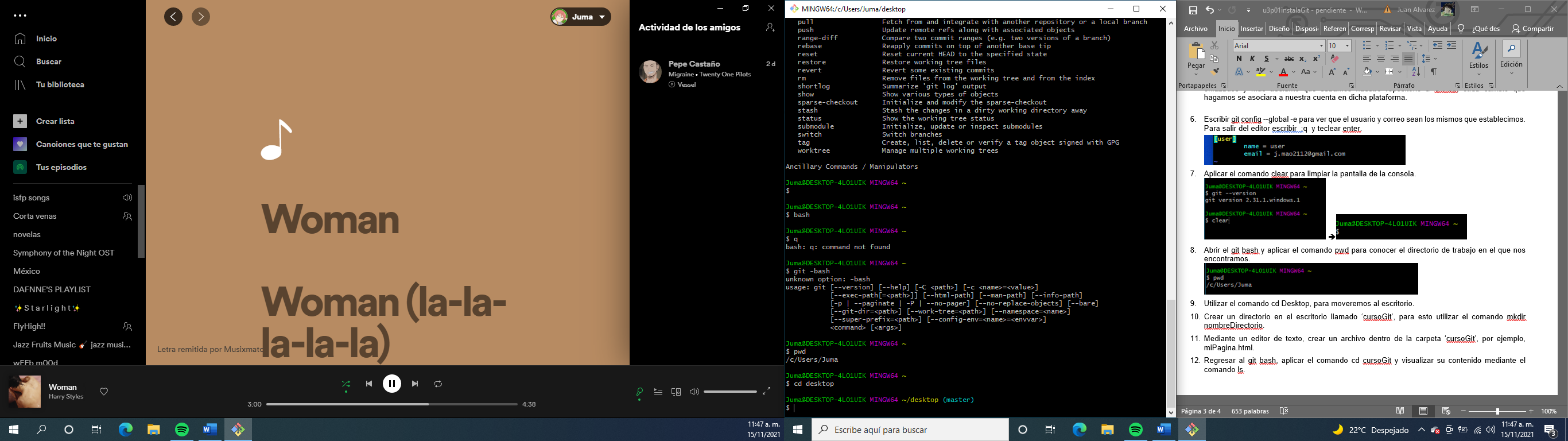
1. Aplicar el comando clear para limpiar la pantalla de la consola.

 🡺

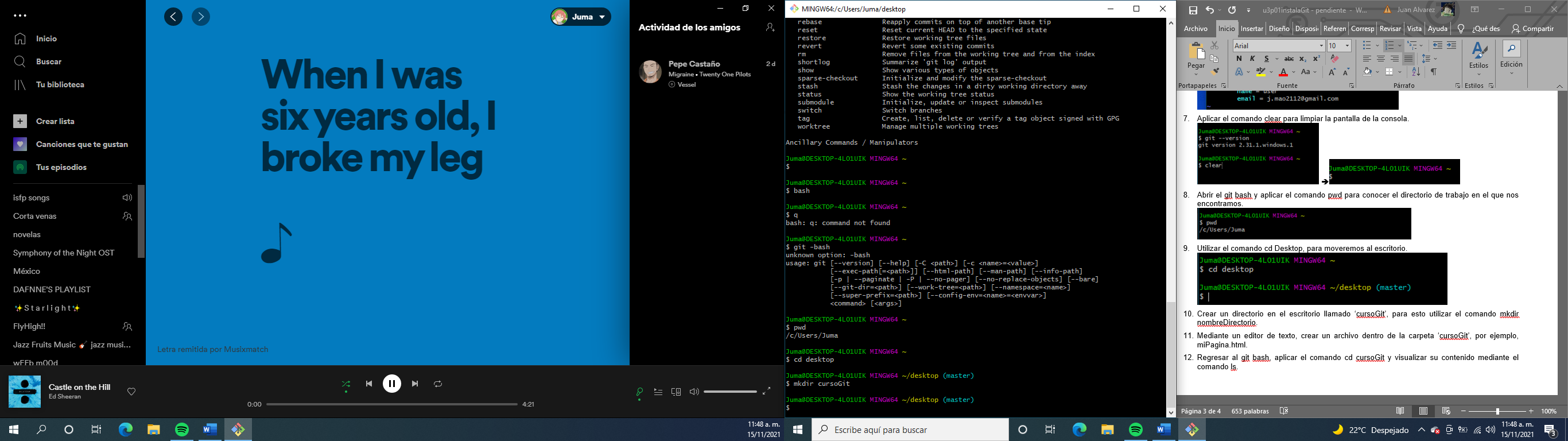
1. Abrir el git bash y aplicar el comando pwd para conocer el directorio de trabajo en el que nos encontramos.



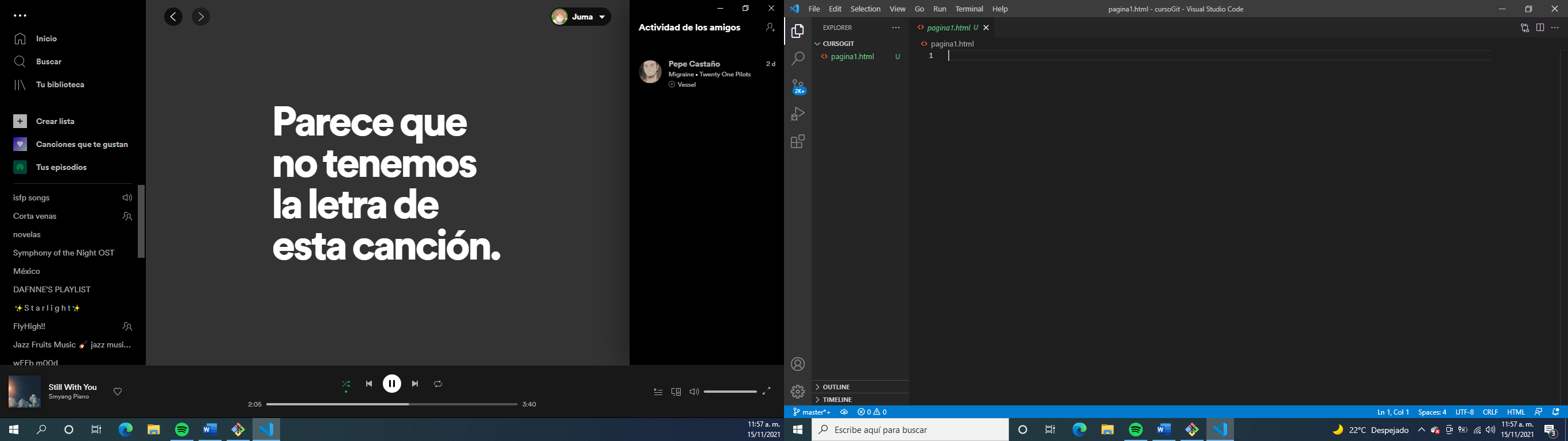
1. Utilizar el comando cd Desktop, para moveremos al escritorio.



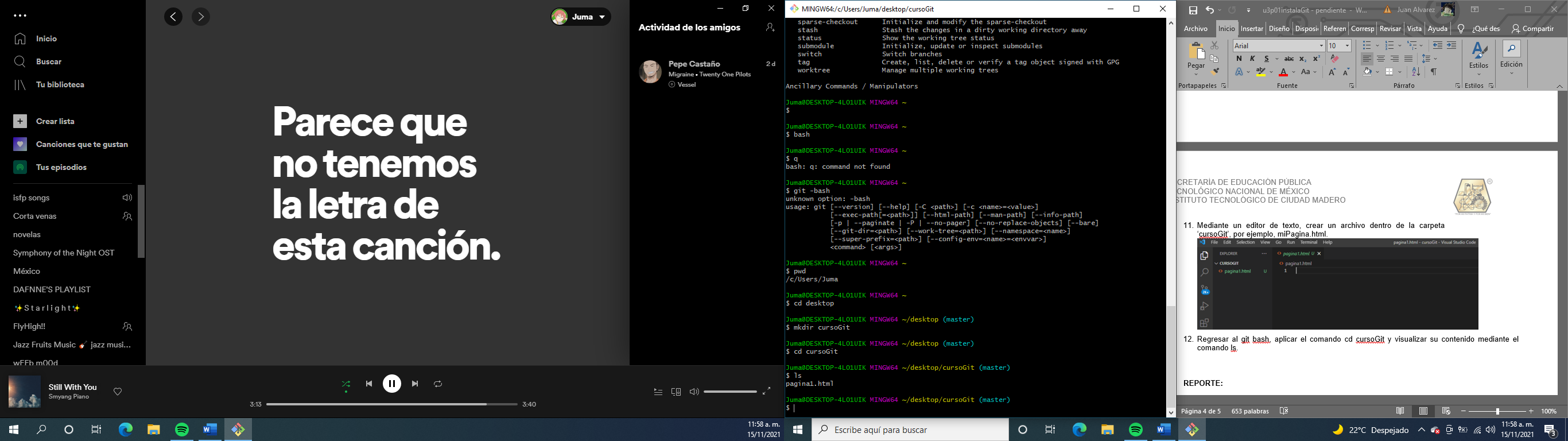
1. Crear un directorio en el escritorio llamado ‘cursoGit’, para esto utilizar el comando mkdir nombreDirectorio.



1. Mediante un editor de texto, crear un archivo dentro de la carpeta ‘cursoGit’, por ejemplo, miPagina.html.



1. Regresar al git bash, aplicar el comando cd cursoGit y visualizar su contenido mediante el comando ls.



1. **FUENTES DE INFORMACIÓN**

<file:///C:/Program%20Files/Git/mingw64/share/doc/git-doc/git-commit>.