**Subtarea 1: Uso de GitHub y Git**

Tras haber creado nuestro perfil, y clonado el repositorio al nuestro local mediante el comando *git clone:*

5. Ejecutamos el comando *ls -a*, para mostrar todo el contenido del repositorio, incluido los archivos ocultos y comprobamos que está vacío.

Texto

Descripción generada automáticamente

6. Ejecutamos el comando *git status* para comprobar el estado del repositorio, al no contener ningún archivo lógicamente está “al día”.

Texto

Descripción generada automáticamente

7. Pegamos el trabajo en la carpeta, y ejecutamos de nuevo el comando *git status*. Podemos comprobar que ahora en rojo aparece el archivo copiado y así nos indica que no se está rastreando.

Texto

Descripción generada automáticamente

8. Usando el comando *git add*, adjuntamos al rastreo el archivo de la tarea, y mediante *git status* comprobamos que estaría ya preparado para un commit.

Texto

Descripción generada automáticamente

9. Después, con el comando *git commit -m*, realizamos un commit de todos los archivos que tengamos, en este caso sólo el .doc de la tarea, y además añadimos el txt de “Añadir tarea 2”, tras usar *git status*, comprobamos que no tiene nada para hacer commit, y avisa que la rama se encuentra limpia, el repositorio local está al día.

Texto

Descripción generada automáticamente

10. Finalmente, para poder sincronizar todo lo que tenemos en el commit con nuestro repositorio de GitHub tenemos que utilizar el comando *git push*, y así subiríamos todo lo que tenemos en local.

Texto

Descripción generada automáticamente

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

Notepad++:

* Lenguajes de programación compatibles:

Soporta una gran variedad de lenguajes de programación, alrededor de unos 70, aun así permite crear diferentes esquemas de color para otros tipos de lenguajes según necesidad de los usuarios.

* Facilidad de uso

Dispone de una interfaz de usuario simple y sencilla, para poder ser usada por cualquier tipo de usuario. Tiene pestañas para facilitar la navegación entre documentos, los menús son fáciles de navegar y está todo perfectamente etiquetado.

* Depuración:

No tiene un depurador propio, depende de otras herramientas externas para ello.

* Gestión de versiones:

No dispone de ningún tipo de soporte integrado para la gestión de versiones, pero tiene la opción de añadir complementos externos para ello.

* Personalización y extensibilidad:

Es bastante personalizable, mediante códigos de color, modificaciones en las preferencias de edición… Se adecua bastante a las necesidades de un programador novato.

* Otros aspectos que te resulten de interés:
  + Sólo esta disponible para Windows.
  + Se puede ejecutar desde una unidad USB.
  + Está programado en C++.

Visual Code:

* Lenguajes de programación compatibles:

Soporta aun más lenguajes que Notepad++, incluyendo aquellos que son mas específicos científicamente.

* Facilidad de uso:

Es relativamente mas complejo que Notepad++, no está enfocado a usuarios principiantes, pero igualmente su interfaz no es muy compleja de aprender, dispone además de muchas más opciones para el usuario.

Un poco mas complejo que Notepad++ pero bastante fácil de usar igualmente. Interfaz amigable y con sugerencias de código que facilitan la programación.

* Depuración:

Tiene herramienta de depuración integrada: puede poner puntos de interrupción, inspeccionar variables, realizar seguimiento de la ejecución del código…

* Gestión de versiones:

Tiene integración directa para el control de versiones con Git.

* Personalización y extensibilidad:

Tiene muchas opciones de personalización, tanto propios como a través de extensiones de terceros que le permiten añadir hasta nuevas funciones.

* Otros aspectos que te resulten de interés:
  + Disponible para Windows, MacOS y Linux.

**Subtarea 3: Tu herramienta favorita**

Dentro de las herramientas CASE, creo que las que me serían de mayor utilidad son las M-Case principalmente porque son las que se