Subtarea 1: Uso de GitHuby Git

Con el comando “ls” compruebo que el repositorio está vacío.

Texto

Descripción generada automáticamente

Con el comando “git status” compruebo que el repositorio esta al día.

Texto

Descripción generada automáticamente

Subo un archivo al repositorio.

Texto

Descripción generada automáticamente

Utilizo el comando “git add .” para poner el repositorio en la fase de lanzamiento.

Texto

Descripción generada automáticamente

Uso “git commit -m “primer commit”” para guardar los cambios ya hechos.

Texto

Descripción generada automáticamente

A partir de aquí el programa me empieza a fallar y “crashear”, cambio del master al main para poder hacer “git push”, pero no me deja, el programa me empieza a dar errores por cada instrucción y la infalible técnica de cerrarlo y abrirlo de nuevo no ofrece resultados. Ante la imposibilidad de continuar decido hace un nuevo repositorio (T2) y realizar todos los pasos anteriores de nuevo. GIT sigue dando muchos problemas en este enésimo repositorio, como no realizar ninguna acción al introducir cualquier comando, ni siquiera dando error. Finalmente pude hacer el push y sincronizar el repositorio con GitHub, esto ultimo no lo pude capturar pues a GIT le pareció oportuno crashear en ese preciso instante, pero al volver a poner la instrucción de nuevo pone que todo está al día y en la página web aparece el documento.

Texto

Descripción generada automáticamente con confianza media

Aquí un ejemplo de GIT “funcionando” en mi ordenador.

Imagen que contiene pájaro

Descripción generada automáticamente

Esto es una segunda sincronización .

Texto

Descripción generada automáticamente

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

NeatBeans y Bloc de Notas.

NetBeans es un IDE especifico para Java pero ofrece un amplio soporte para más lenguajes de programación, incluyendo Java, JavaScript, PHP, HTML, CSS, C, C++, y más. Mientras que el Bloc de Notas es un simple editor de texto plano y de los más básicos, debido a su naturaleza de editor de texto no se puede hablar de compatibilidad con lenguajes de programación ya que tu puedes escribir el código en el lenguaje que quieras, pero no lo podrás ni compilar ni ejecutar sin ayuda externa.

Respecto a la facilidad de uso el Bloc de Notas carece de dificultad de aprendizaje ya que solo se puede escribir, pero eso significa que no hay ninguna curva de aprendizaje y que la tarea de programar será mucho más difícil que en cualquier IDE por la falta de ayuda, autocompletado o señalización de errores de sintaxis. Al contrario, NeatBeans posee algo de dificultad inicial ya que tiene abundantes funcionalidades que hay que aprender, pero al hacerlo estas solo te facilitan la programación. NeatBeans es más difícil de usar, pero es mucho más fácil programar en él.

En el ámbito de la depuración hay un claro ganador pues el Bloc de Notas no tiene esa capacidad, NeatBeans sin embargo ofrece herramientas avanzadas de, especialmente en el contexto de Java. Este permite establecer puntos de interrupción, inspección de variables, y seguimiento de la ejecución del código.

En la gestión de versiones el Bloc de Notas, debido a su simplicidad y robustez, no tiene ninguna capacidad de gestión y no la necesita ni se la echa en falta. NeatBeans si que necesita esa funcionalidad y la posee, desde internet y gratuitamente puedes bajar tanto versiones anteriores como las ultimas, NeatBeans se actualiza con frecuencia, aunque no siempre es recomendable bajarse la última.

NeatBeans es un IDE altamente personalizable, puedes modificar infinidad de cosas, desde su funcionamiento hasta como lo visualizas, cambiando colores, fuente, tamaño, etc. El Bloc de notas no se puede modificar tanto, pero si que se puede personalizar algo, como el tamaño de la fuente o el tipo.

**Subtarea 3: Tu herramienta favorita.**

Creo que la herramienta CASE que mas me ayudara en mi vida como programador todavía no se ha desarrollado, o al menos no a un buen nivel. Pienso que será una herramienta que utilizara la IA (inteligencia artificial) para programar, actualmente esto ya se puede hacer, pero la IA no está suficientemente desarrollada como para que sea eficaz, actualmente comete muchos errores, pero en un futuro entenderá lo que el programador le comunique y lo traducirá a código mejor de lo que haría una persona. Lo que nunca logrará o al menos no lo hará en un futuro próximo será dominar la lógica tal y como lo hace un humano. Esta IA será una especie de super compilador que pueda traducir lenguaje natural humano a lenguaje máquina, el programador podrá trabajar al más alto nivel sin perder eficacia pues la IA lo convertiría a lenguaje máquina de manera precisa. Esta herramienta nos ahorrará tener que aprender las diferentes sintaxis y particularidades de cada lenguaje de programación. Actualmente ya hay IA que son bastante decentes traduciendo de un lenguaje a otro. La herramienta nos ahorrará las tareas más repetitivas y tediosas, lo que nos permitirá centrarnos en la lógica que, por otra parte es lo más complejo de la programación, y en el diseño del programa.

Esta herramienta también podrá traducir de lenguaje de programación a lenguaje humano, esto será muy útil para entender código escrito por otra persona sin tener que perder tiempo descifrando lo que intentaban hacer.