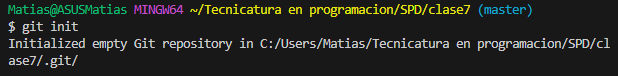
TRABAJO GRUPAL-3

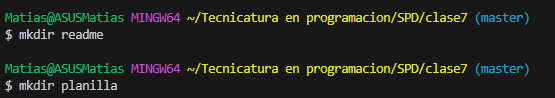
**Capturas Clase 7 Grupo CodeSprinters**

1. **Crear un repositorio nuevo con el nombre clase 7​.**

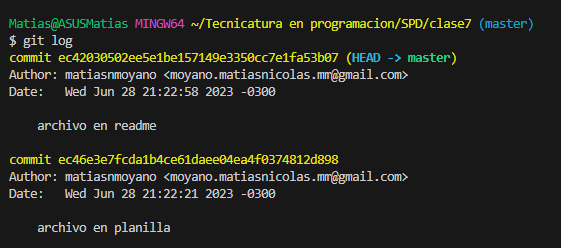
****

****

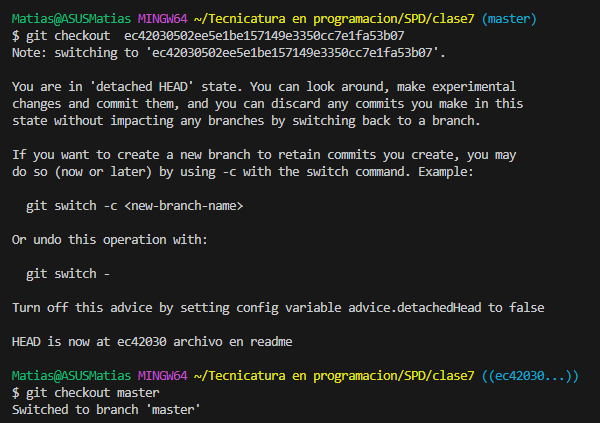
1. **Crear dos carpetas - readme - planilla ​**

****

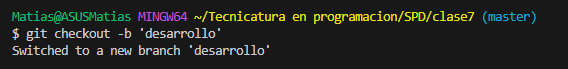
1. **Añadimos dos documentos de texto en ambas carpetas, añadimos con git add cada modificación y commiteamos. ​**

****

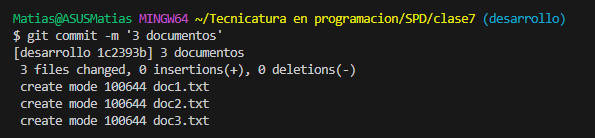
1. **Aplicar los comandos git checkout y seleccionar dos n° # para volver en el tiempo en nuestro proyecto​**
2. **Volver al último commit que realizamos anteriormente. ​**

****

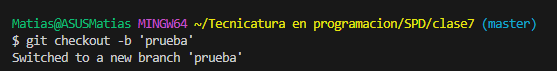
1. **Creamos una 1ra rama aux.: - desarrollo ​**

****

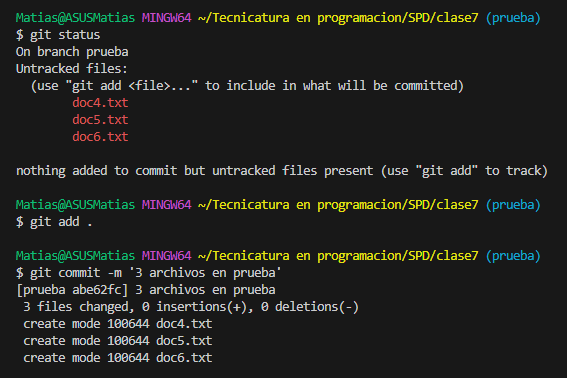
1. **Creamos 3 documentos de texto , añadimos y commiteamos. ​**

****

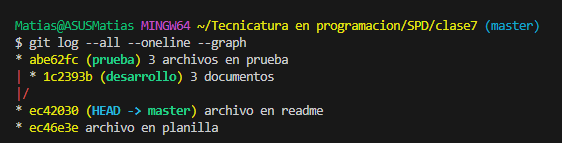
1. **Creamos una 2da rama aux.: - prueba ​**

****

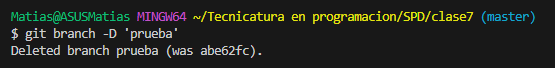
1. **Creamos 3 documentos de texto, añadimos y commiteamos.**

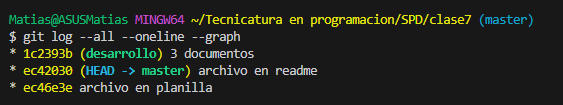
**​**

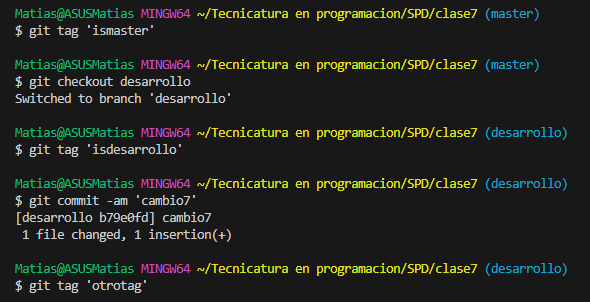
1. **Aplicamos el comando que permite ver todas las ramas y commit ​**

****

1. **Eliminamos una rama. ​**

****

1. **Aplicamos nuevamente el comando que nos permite visualizar las ramas y los commit. ​**
2. **Generar 3 etiquetas en nuestro proyecto​**

****

1. **Desarrollar cuál es la importancia de utilizar Git. (Visto en la clase n°6) Puede acompañar de imágenes o capturas de pantalla. Formato de Entrega : Doc. Word, Pdf, video, etc.​**

Distintos puntos que hacen que git- y github sea muy importante de saber utilizar bien y sea muy utilizado:

· Git facilita la colaboración en proyectos de software, permitiendo que varios programadores trabajen en el mismo código al mismo tiempo.

· Los cambios realizados se pueden fusionar de forma controlada.

· Git proporciona herramientas para resolver conflictos que puedan surgir al combinar diferentes cambios. Esto asegura una colaboración sin problemas entre los miembros del equipo.

· Git permite controlar las versiones de los archivos a lo largo del tiempo, lo que resulta fundamental para seguir el historial de modificaciones y revertir a versiones anteriores si es necesario.

· Con Git es fácil crear ramas, lo que permite trabajar en diferentes características o solucionar problemas sin afectar la rama principal del proyecto.

· Git mejora la gestión de proyectos de software al mantener un registro detallado de los cambios realizados por cada programador. Por lo que le sirve para controlar quién trabajo y quien no en un proyecto además de saber en qué volumen trabajo cada uno, lo que hace que git sea muy bueno para controlar empleados o alumnos y generar un trabajo grupal mucho más fluido.

· Git permite respaldar y recuperar proyectos de forma sencilla, minimizando el riesgo de pérdida de trabajo o errores graves. Al utilizar repositorios remotos y realizar copias de seguridad periódicas.

· Se garantiza la disponibilidad del código en diferentes ubicaciones. Lo que es muy importante y facilita que se trabaje en grupo a grandes distancias unas personas de otras.