Classguide 502

Crea la cuenta en GitHub y un proyecto en AndroidStudio y configura el entorno con tus credenciales en: File > Settings > Version Control > GitHub.

**3. Añadiendo la lógica**

Prueba las distintas opciones que ofrece Git y define cada una de las siguientes operaciones:

- Commit.

- Push.

- Pull.

- Stash.

- Merge.

- Checkout.

Busca o diseña una imagen/diagrama donde entiendas claros los conceptos mencionados.

Timeline

Description automatically generated

Ilustración 1: Diagrama commandos Git. Fuente: <https://www.junosnotes.com/git-commands/>

Los comandos descritos en el diagrama son los siguientes:

* *Commit*: Actualiza los cambios deseados en el repositorio local, también con este comando se suele describir con un mensaje dichos cambios.
* *Push*: Envía los cambios al repositorio remoto, en nuestro caso GitHub.
* *Pull*: Descarga posibles cambios desde el repositorio remoto a los archivos locales del proyecto.
* *Stash*: Almacena los cambios locales a parte y actualiza el proyecto al *commit* principal. De esta manera se puede cambiar de rama o comprobar el estado actual del proyecto sin crear conflictos ni perdiendo trabajo.
* *Merge*: Combina dos ramas en una de ellas. Se suele usar cuando una funcionalidad ya ha sido desarrollada, así se añade de manera efectiva al proyecto. En nuestro caso tenemos una rama para cada característica (*feature*) y una rama de desarrollo (*develop*) donde se hace el *merge*.
* *Checkout*: Sirve para cambiar de rama de trabajo.

Para la realización de la práctica final usamos el repositorio de GitHub de Javier Verón dedicado a la asignatura (<https://github.com/VeronLinks/Android/tree/main/Entrega>). En la carpeta “Entrega” están los ejercicios de cada módulo y el proyecto final hecho todo en grupo.

A screenshot of a computer

Description automatically generated with medium confidence

Ilustración 2: Repositorio en GitHub

Para el método de trabajo por ramas seguimos el siguiente diagrama. Donde existe una rama principal *develop* a la que se van añadiendo *features*.

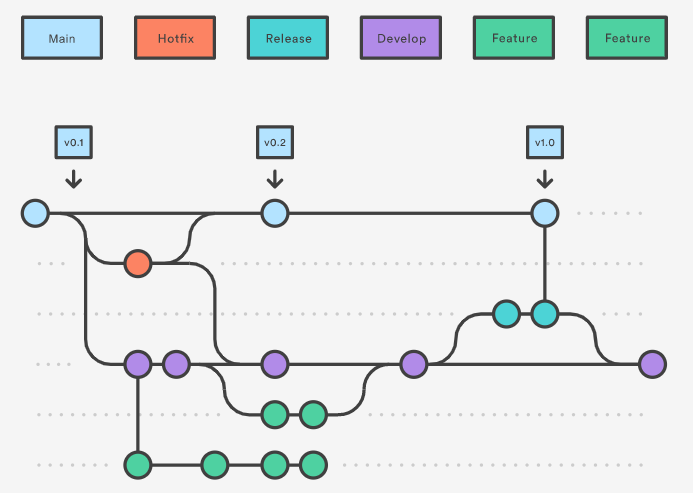


Ilustración 3: Diagrama de ramas

Primero empezamos usando la terminal de Windows para realizar los comandos de Git. Pero rápidamente pasamos a manejar el repositorio desde Android Studio.

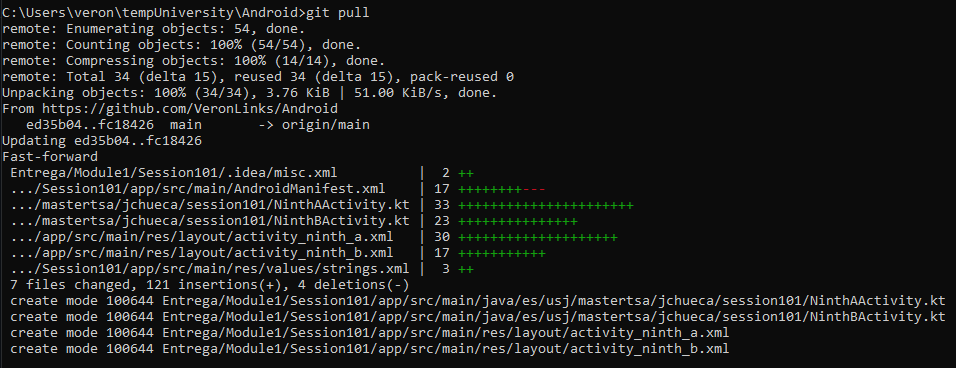


Ilustración 4: Uso de Git mediante la terminal de Windows

Para esto se añade la cuenta de GitHub a Android Studio mediante un token de autentificación y entonces ya se tiene acceso a toda la funcionalidad de Git y más. Cabe destacar la herramienta para hacer *merge* seguro de ciertos archivos que han sido editados al mismo tiempo por varias personas, dicha herramienta es muy potente, cómoda y fácil de usar.

Graphical user interface, text

Description automatically generated

Ilustración 5: Ejemplo de cuenta de GitHub añadida a Android Studio