



Fundamentos de GIT y GitHub

Fundamentos de GitHub (Parte II)

***Gestionar el código fuente
utilizando GitHub para
mantener un repositorio de
código remoto seguro y
permitir trabajo concurrente.***

- Unidad 1: Ambiente de desarrollo y sus elementos de configuración.
- Unidad 2: Elementos de la interfaz, navegación e interacción.
- Unidad 3: Fundamentos de GIT y GitHub.



Te encuentras aquí



¿Qué aprenderás en esta sesión?

Utilizar Android Studio para gestionar proyectos Git.

Git & GitHub

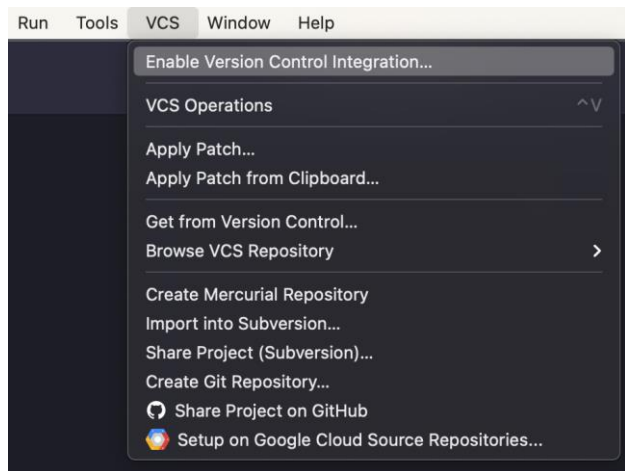
Git	GitHub
Software de versionamiento de archivos que registra los cambios en el tiempo.	Plataforma de desarrollo colaborativo que permite almacenar y gestionar proyectos Git. Se debe crear una cuenta.
Se instala y funciona de manera local en el computador. Toda la información del proyecto se encuentra en la carpeta .git dentro del proyecto.	Es una plataforma web en la que se debe crear una cuenta y permite tener una copia de nuestro proyecto.
Al ser local NO necesita de una conexión a internet para funcionar.	Al ser una plataforma web se necesita una conexión a internet.
La carpeta .git es visible solo para el usuario dueño del proyecto.	Administra los usuarios y los permisos para leer, escribir y modificar el proyecto compartido.

`/* Git & Android Studio */`

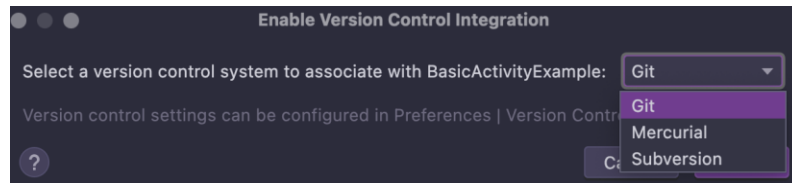
VCS

Version Control System

El menú VCS permite habilitar la integración con el sistema de control de versiones.

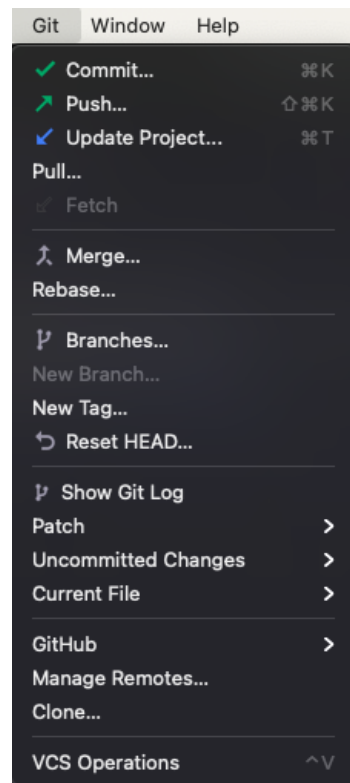


Para asociar el VCS con la raíz del proyecto, se debe seleccionar Git.

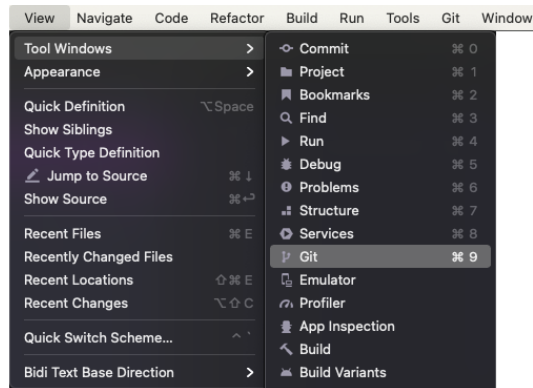


Menú Git

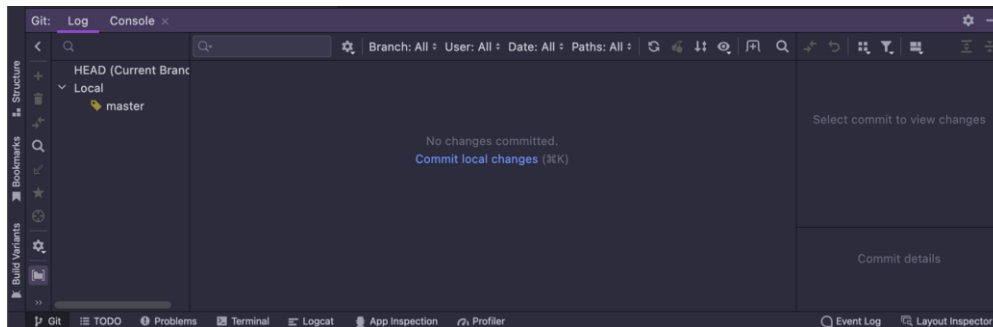
En el menú principal, se reemplaza VCS por Git que permite ejecutar comandos git desde una interfaz gráfica



Ventana de herramientas Git



En la parte inferior aparece el selector para la ventana de herramientas Git que contiene la lista de commits y permite ver los cambios y el detalle de cada commit



Ejercicio



Habilitar control de versiones

1. Abrir un proyecto existente.
2. Habilitar el sistema de control de versiones (VCS) para el proyecto abierto.

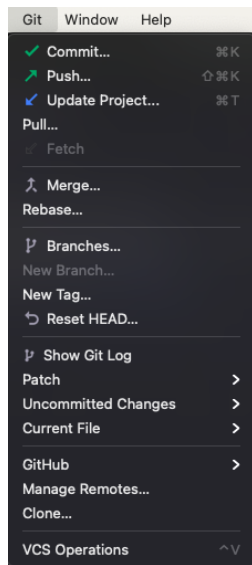


¿Qué etapa del flujo de Git acabamos de hacer?



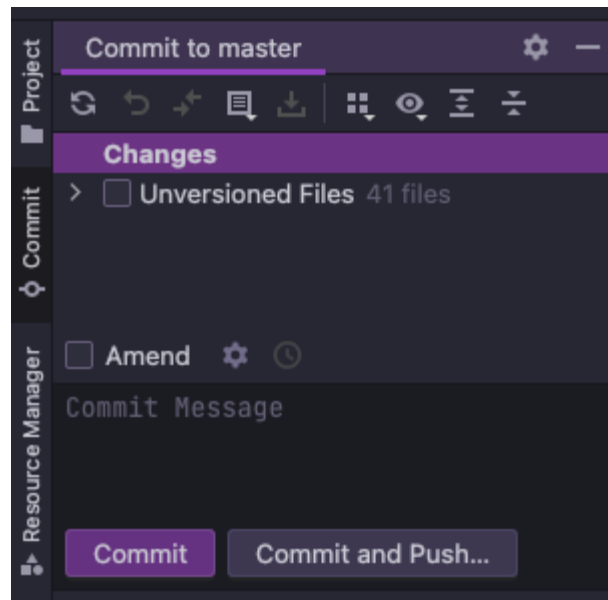
/* Guardando los cambios */

Guardando los cambios



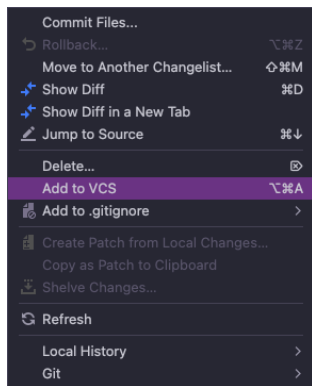
El menú Git de la barra permite hacer *Commit*.

Por defecto, aparece la ventana *Commit* en la barra izquierda de Android Studio.



Guardando los cambios

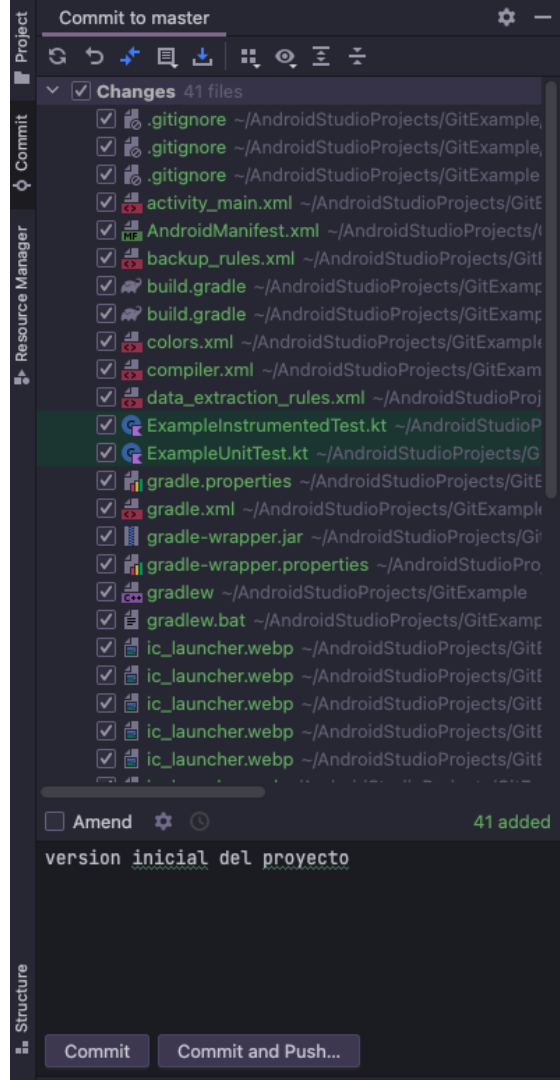
Agregar y confirmar



Con el botón derecho, accedemos al menú contextual para agregar nuestro proyecto completo (Add to VCS).

Luego de agregar los archivos que queremos, se agrega el mensaje y se confirman los cambios.

Por el momento, el proyecto está versionado localmente, así que debemos utilizar la opción Commit.



Ejercicio



Versión inicial del proyecto

Subtítulo

Siguiendo con el versionamiento del proyecto:

1. Agregar todos los archivos del proyecto al control de versiones.
2. Indicar un mensaje de commit y confirmar los cambios.
3. Ir a la ventana de herramientas Git y verificar el commit inicial.



¿Qué etapa del flujo de Git acabamos de hacer?

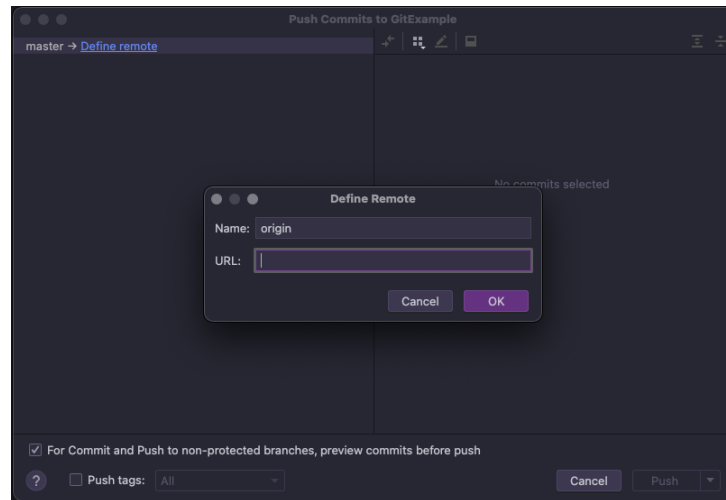


/* Android Studio & GitHub */

Empujando los cambios

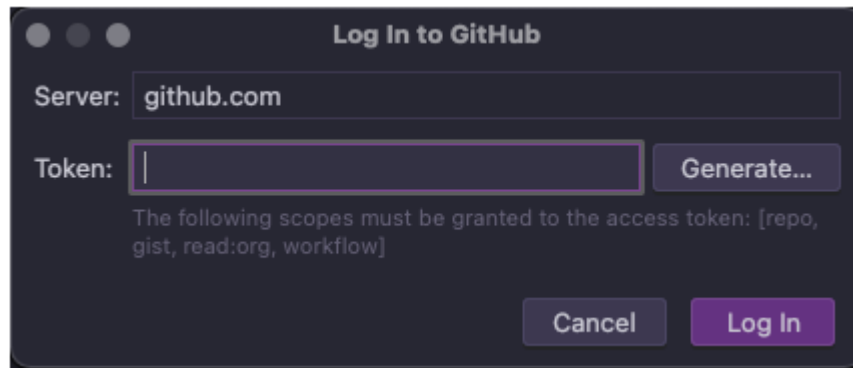
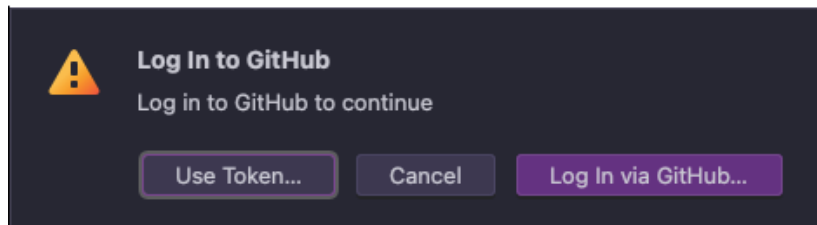
Si no se tiene un remoto configurado, en la pantalla Push Commits se puede definir uno directamente seleccionando *Define remote*.

La URL a ingresar corresponde a un repositorio GitHub existente.



Configurando el acceso a GitHub

Elegimos la opción “*Use token*” y usamos la opción “*Generate*” para crear un nuevo token directamente en [GitHub](#).



Configurando un token en GitHub

<https://github.com/>


Es importante mantener un periodo de duración del token por motivos de seguridad.

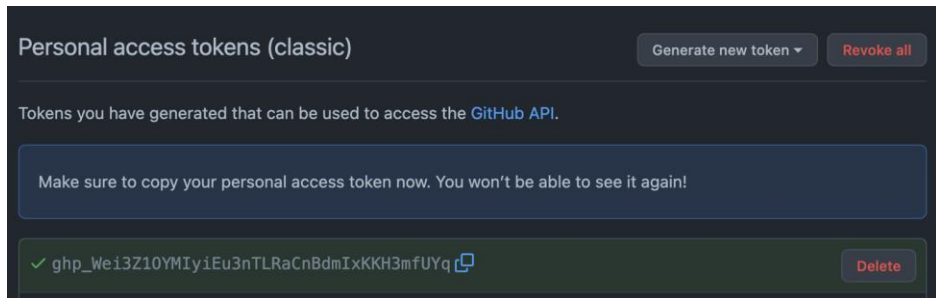
Select scopes permite configurar los permisos específicos que tiene el token y que permiten modificar el código.

The screenshot shows the GitHub 'Settings / Developer settings' page. On the left sidebar, 'Personal access tokens' is selected, with 'Tokens (classic)' highlighted. The main area is titled 'New personal access token (classic)'. It includes a description of personal access tokens, a 'Note' section with a text input field containing 'Android Studio GitHub integration plugin', an 'Expiration' dropdown set to '90 days' with a note that the token will expire on Mon, Feb 13 2023, and a 'Select scopes' section with a link to 'Read more about OAuth scopes'.

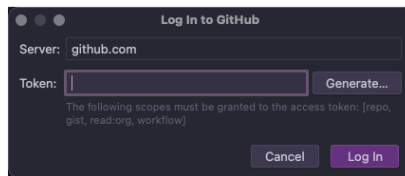
Generate token

Utilizando el token

El token generado **solo se muestra 1 vez** luego de ser creado, por lo que debe ser copiado en ese momento usando este ícono 



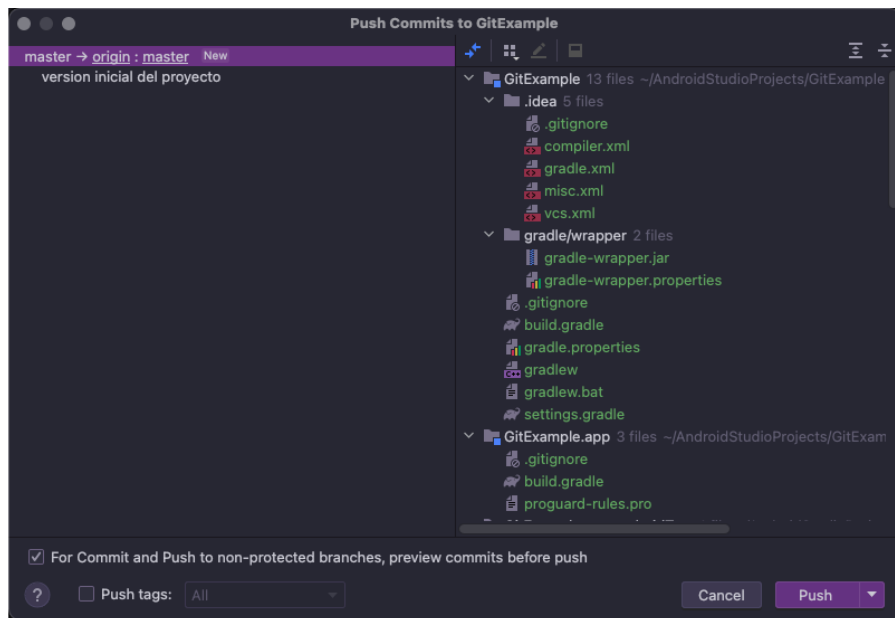
El token lo utilizamos en la pantalla de login de GitHub.



Empujando los cambios

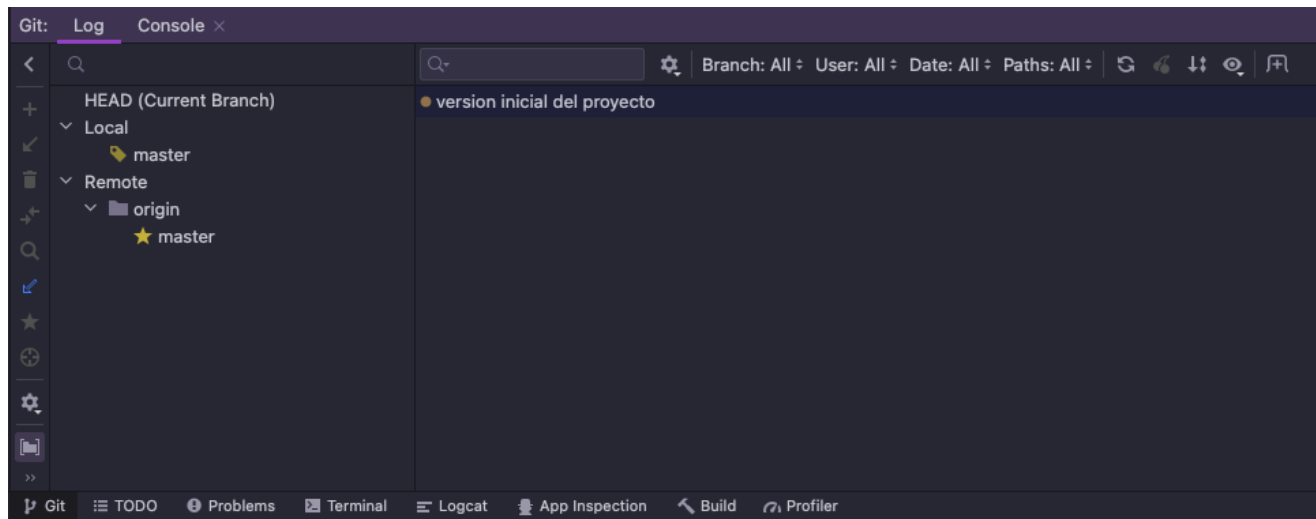
Con el proyecto enlazado con el remoto se pueden empujar los cambios con el botón

Push



Ventana de herramientas Git

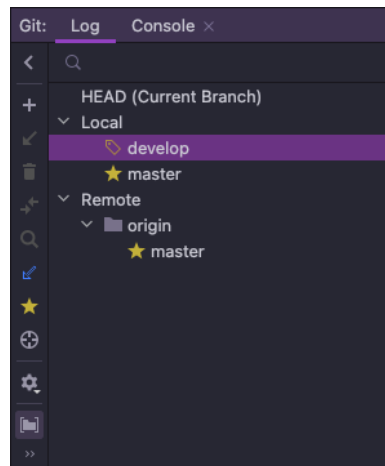
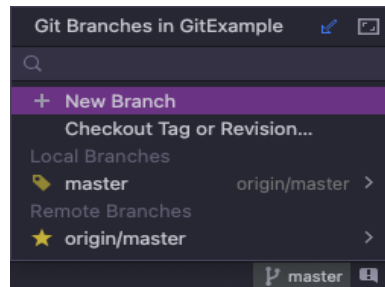
En la ventana de herramientas Git aparece dividido el proyecto en Local y Remoto (Remote).



Nueva rama

En la parte inferior derecha aparece el nombre de la rama actual. Al hacer clic sobre el nombre, despliega el menú contextual que permite crear una nueva rama (+ New Branch).

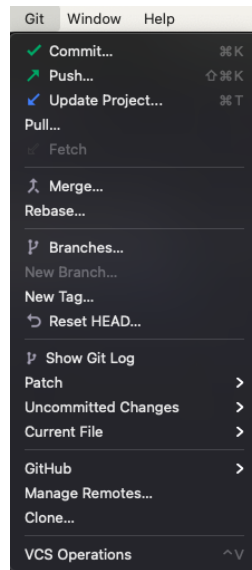
Todas las ramas son listadas en la ventana de herramientas Git que incluye Android Studio.



Menú Git

GitHub

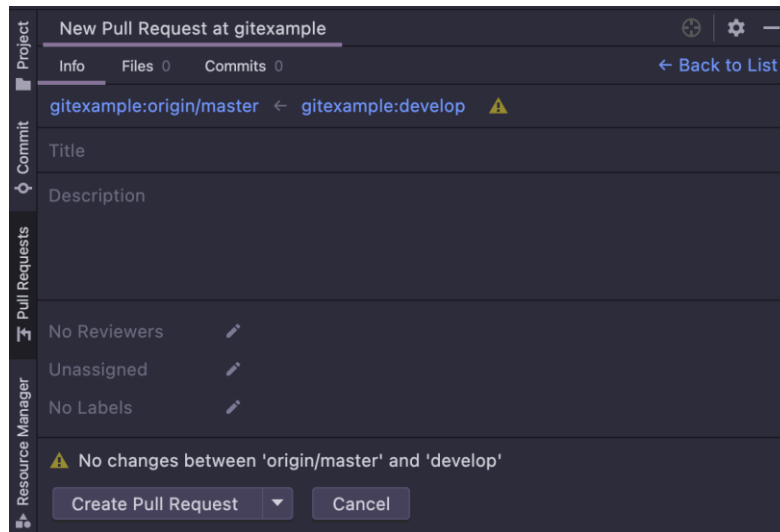
La integración con GitHub permite crear Pull request directamente desde el menú Git.



Menú Git

GitHub

En la parte izquierda aparece la nueva ventana **Pull Requests** del IDE que permite la creación de PR.



Ejercicio



Actividad

1. Crear repositorio en GitHub.
2. Crear (ocupar algún proyecto existente) proyecto con Android Studio.
3. Inicializar el sistema de control de versiones con git.
4. Agregar y confirmar los cambios.
5. Agregar permisos en github (en caso de ser necesario).
6. Empujar los cambios.
7. Crear la rama develop.
8. Empujar la rama develop.



¿Terminal o IDE?





Próxima sesión...

Realizaremos la prueba del módulo.

{desafío}
latam_

*Academia de
talentos digitales*

