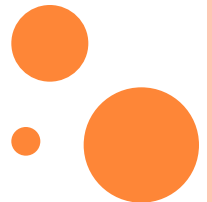
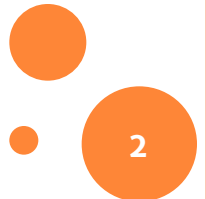


GIT EN ANDROID STUDIO



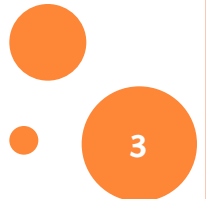
CONTROL DE VERSIONES CON GIT

- Utilizar Git es necesario para gestionar el código en las entregas de la asignatura.
- Este tema explica la integración Git en Android Studio
 - Asume conocimiento sobre Git
- Más información sobre Git:
 - Online, p.e. missing.csail.mit.edu/2020/version-control/
 - Material de la asignatura "Análisis y Diseño de Sistemas Información"



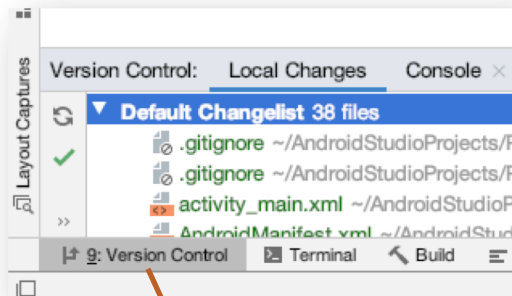
CONFIGURAR GIT EN ANDROID STUDIO

- Pre-requisito: Instalar y configurar Git en el sistema
- Abrir/Crear un proyecto que queramos controlar con Git en Android Studio
 - Menú VCS -> "Enable Version Control Integration"
 - Elegir "Git" en el desplegable
- El proyecto ya estaría controlado con git
 - Los archivos pasan a tener el nombre en color rojo (hasta que se sincronicen por primera vez)

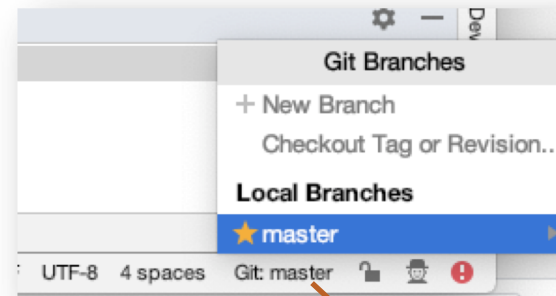


CONFIGURAR GIT EN ANDROID STUDIO

- Al activar la integración con Git, se crea automáticamente una rama local "master"
- En Android Studio, las pestañas inferiores dan información sobre el control de versiones



Información sobre
los archivos

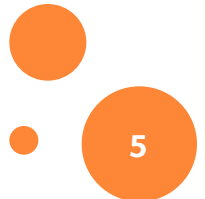


Información sobre
las ramas Git



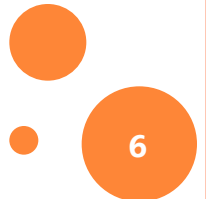
CONFIGURAR GIT EN ANDROID STUDIO

- El archivo **.gitignore** contiene nombres (o patrones de nombres) de archivos que Git omite al sincronizar
- Un proyecto Android con Git contiene 2 **.gitignore**: uno en la raíz del proyecto, otro en la carpeta *app*
- Durante el desarrollo de una aplicación, es recomendable actualizar el **.gitignore** ubicado en la raíz para que se omitan:
 - Código auto-generado
 - Archivos binarios
 - Archivos de configuración local (que podrían incluir contraseñas), ...



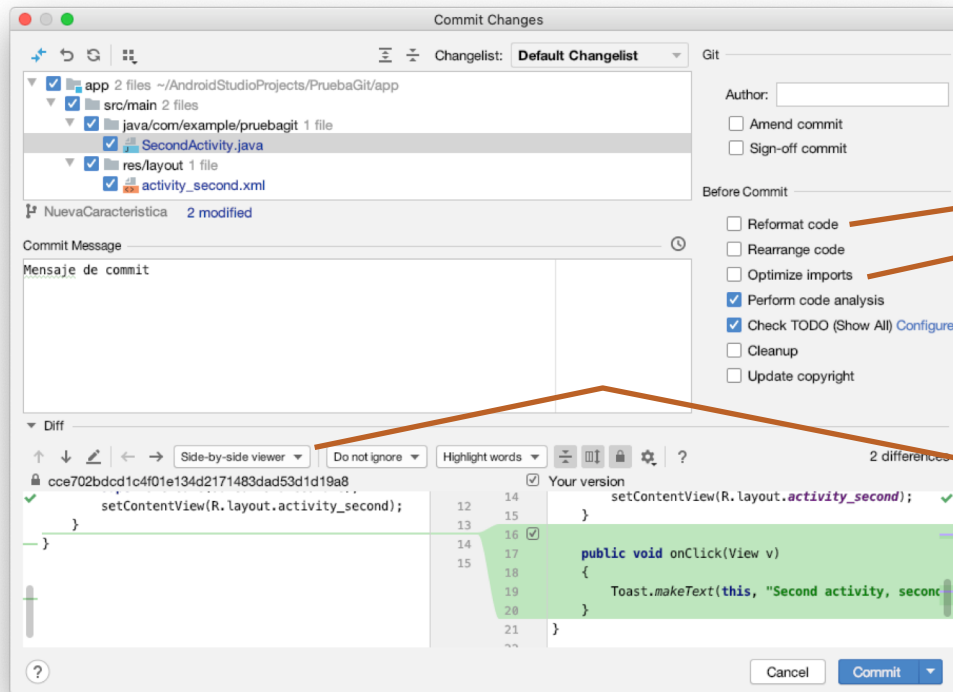
CONTROL DE CAMBIOS LOCAL

- Registrar últimos cambios locales (commit)
 - Menu VCS -> "Commit"
 - Elegir archivos y escribir mensaje de "commit"
 - Recomendable escribir un mensaje corto y significativo
 - Evitar mensajes genéricos, p.e., "Problemas arreglados"
 - Requiere un nombre y cuenta de e-mail para Git
 - Es recomendable NO indicar que se usen globalmente
 - Indicará si hay algún conflicto
- Se guardan los cambios en el repositorio local
 - Los archivos sincronizados pasarán a color verde



CONTROL DE CAMBIOS LOCAL

○ Algunas opciones de la ventana "commit":



Elimina espacios o tabuladores extra e indenta el código

Elimina "import"s innecesarios

Selector de vista: ficheros separado o vista unificada

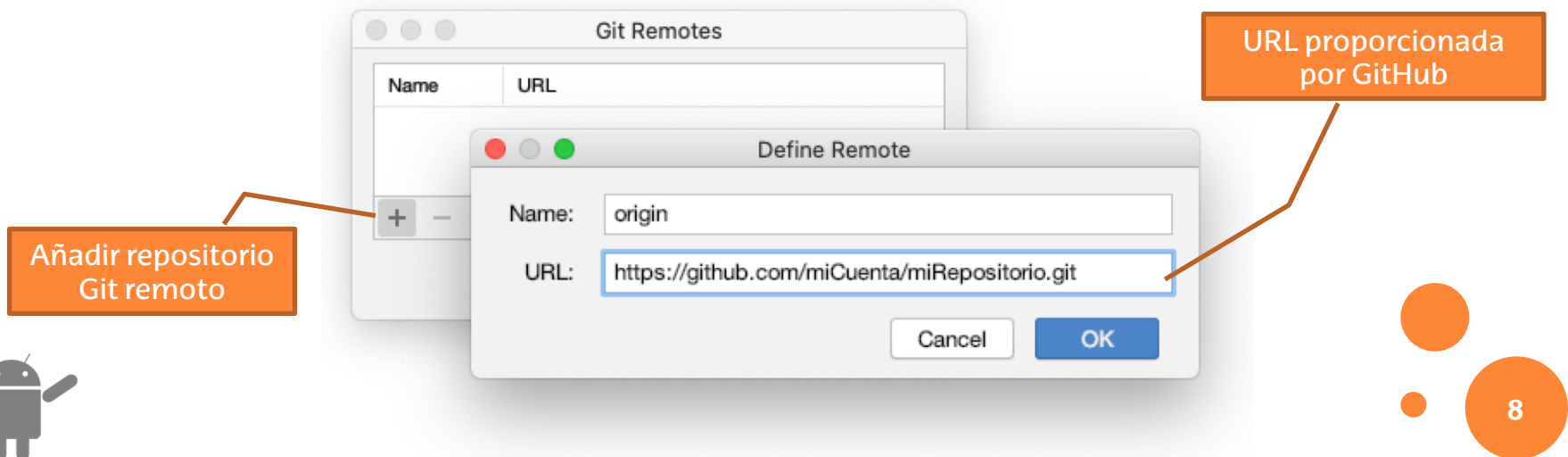
○ Recomendación de la filosofía Git:

- "commit early, commit often"



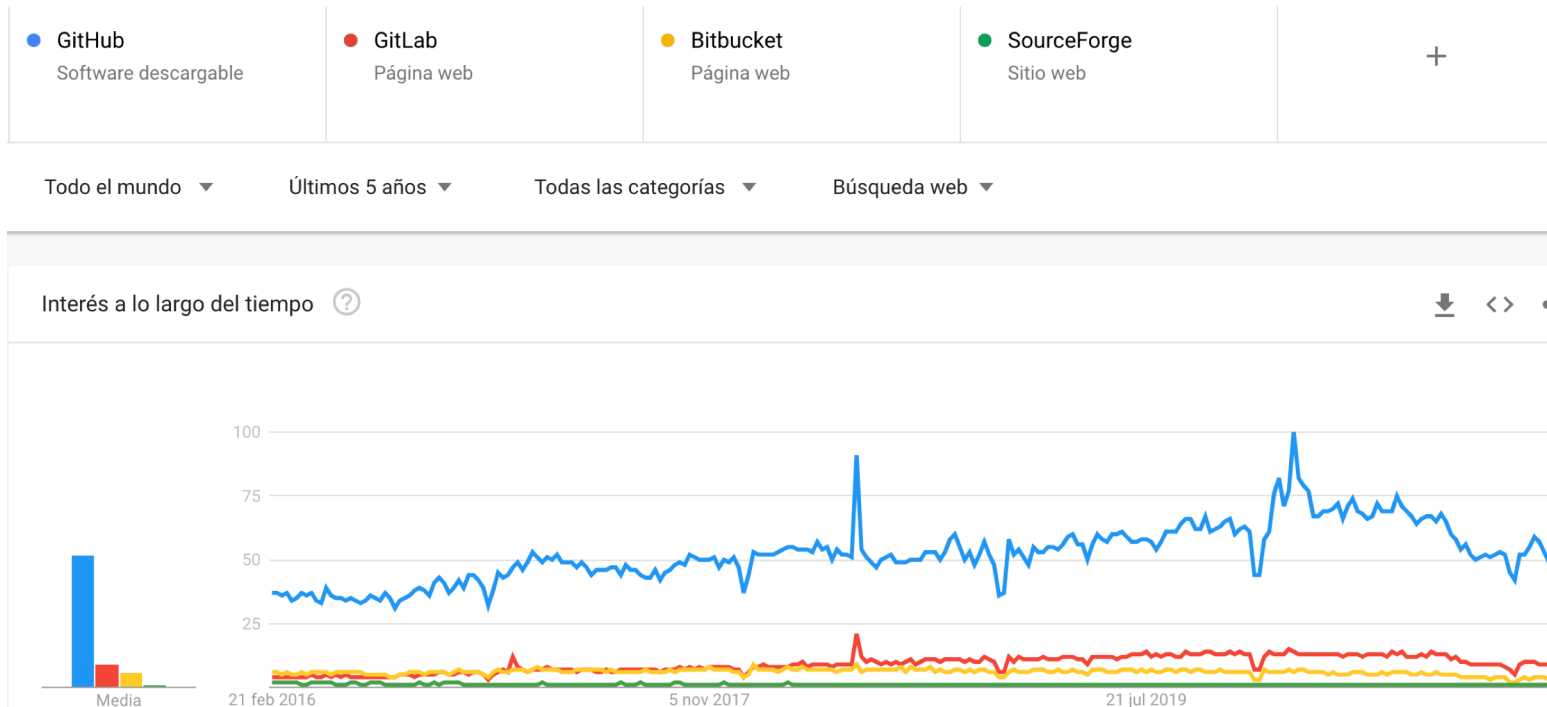
AÑADIR REPOSITORIO REMOTO

- Pre-requisito: crear repositorio Git en GitHub
 - Será el repositorio remoto con el que sincronizar cambios
 - No hace falta inicializarlo con ningún contenido
- En Android Studio hay que indicar la URL del repositorio remoto
 - Menú VCS -> Git -> Remotes



REPOSITORIOS REMOTOS

- Comparativa de popularidad sobre diferentes plataformas de repositorios de código.
 - Según Google Trends (18 de febrero de 2021)



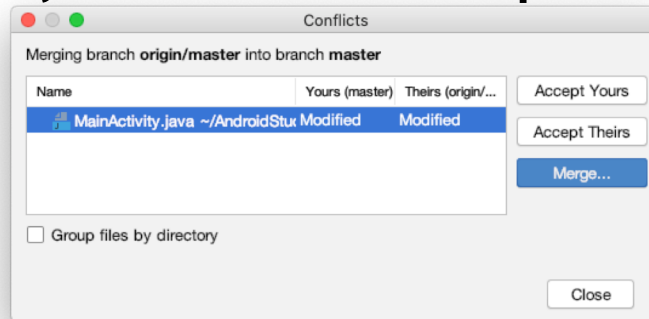
USO DEL REPOSITORIO REMOTO

- Subir cambios al repositorio remoto
 - Pre-requisito: "commit" con lo que queremos sincronizar
 - Menú VCS -> Git -> Push
 - Elegir "commit"(s) a sincronizar
 - Si hubiera conflictos entre versiones de código, aparecerían en este momento
- Recuperar última versión del repositorio remoto
 - Menú VCS -> Git -> Pull
 - Si hubiera conflictos entre versiones de código, aparecerían en este momento



RESOLVIENDO CONFLICTOS

- Cuando surjan conflictos, aparecerá esta ventana:

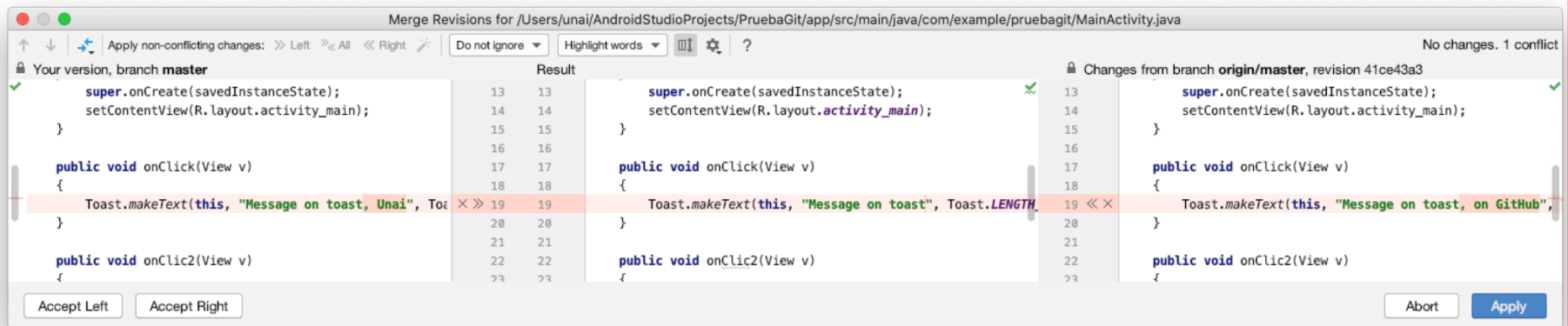


Aceptar nuestra versión,
descartar código remoto

Aceptar cambios remotos,
descartar nuestra versión

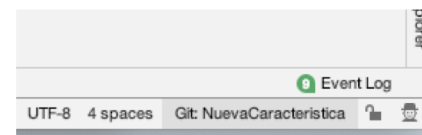
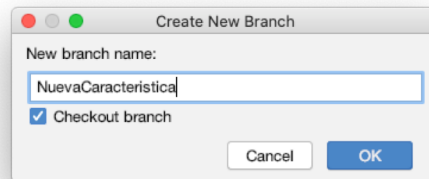
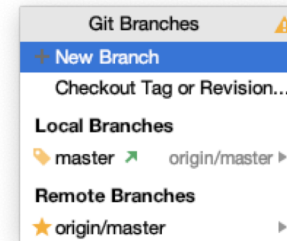
Ir a ventana de fusión

- Ventana de fusión:

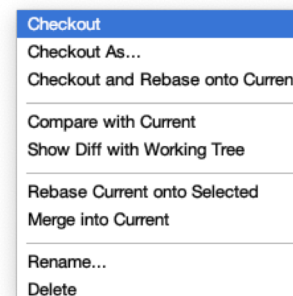


RAMAS

- Gestión de las ramas
 - Menú VCS -> Git -> Branches
- Crear una nueva rama
 - Elegir "New Branch" en el menú
 - Una vez creada, aparecerá en la esquina inferior derecha



- Gestionar ramas existentes
 - Elegir rama en el menú



COMPARAR CAMBIOS ENTRE VERSIONES

- Para comparar código con versiones anteriores:
 - Abrir el archivo a comparar en el editor
 - Menú VCS -> Git -> Compare with ...
 - Elegir del listado la versión a comparar

File Revisions		
c5a2abd3813...	4 minutes ago	ulopeznovoa
1c6645d7953...	5 minutes ago	ulopeznovoa
41ce43a3df8...	9 minutes ago	Unai Lopez N...
9c7797fbd1af...	14 minutes ago	ulopeznovoa
630cc885a09...	16 minutes ago	Unai Lopez N...
4423d513b6a...	19 minutes ago	ulopeznovoa
20d76f768ec...	14/04/2020, 1...	ulopeznovoa
a6f05e314e8f...	14/04/2020, 1...	ulopeznovoa
Commit message		
Added 2nd button		

- Se muestra una ventana con ambas versiones y las diferencias resaltadas



GIT EN ANDROID

○ Ejercicio 1

- Abrir un proyecto existente (p.e. el último ejercicio o laboratorio completado) y añadir control de versiones con Git.
- Añadir una funcionalidad (p.e. un botón en un layout y código asociado en la actividad) y registrar el cambio con un "commit".
- Crear un repositorio en GitHub y sincronizar el proyecto.



BIBLIOGRAFÍA

- <https://medium.com/@apeapius/git-and-github-with-android-studio-gui-tool-3af48b663da2>
- <https://medium.com/better-programming/how-to-use-git-in-android-studio-part-1-a8a554006aad>
- <https://medium.com/better-programming/how-to-use-git-in-android-studio-part-2-93cec67b91b0>
- Material de la asignatura “Análisis y Diseño de Sistemas de Información”
 - Por Mikel Villamañe y Juanan Pereira
- Consultados en abril 2020

