Ingeniería del Software

Práctica 3

Taller de iniciación a Git

2º Grado en Ingeniería Informática
Universidad de Córdoba
Curso 2024/25

1. **Sistemas de control de versiones**
2. Instalación y configuración de Git
3. Uso básico de Git
4. Comandos de GitHub
5. Markdown
6. Principales ejemplos
7. Referencias y tutoriales

Introducción a los VCS

- VCS, *Version Control System*
- Permiten el control y gestión de los cambios que experimenta un producto durante su desarrollo.
- Registra toda la actividad de los artefactos del software (crear, modificar, borrar).
- Cuentan con mecanismos para recuperar copias y trabajar de forma colaborativa.

Términos generales

- Repositorio: donde se almacenan los archivos que se encuentran bajo el control de versiones.
- Clientes: uno o más se conectan al repositorio para leer o escribir datos en él.

Términos generales

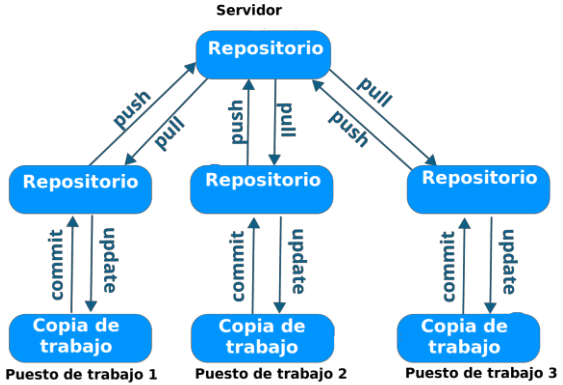
- Consignar (*commit*): publicar uno o más cambios en el repositorio mediante una transacción atómica.
- Revisión: estado que se crea cada vez que se realiza un *commit*. A cada revisión se le asigna un identificador único.
- Conflicto (*conflict*): Situación que ocurre cuando varios clientes realizan cambios en sus copias locales, solapándose al intentar publicarlos en el repositorio.
- Rama (*branch*): Línea de desarrollo que se crea a partir de una revisión concreta y que evoluciona de forma independiente.

Arquitecturas de los VCS

- Centralizado

  - Único repositorio central
  - Clientes visión completa del sistema
  - Punto crítico ante fallos
  - *Subversion*

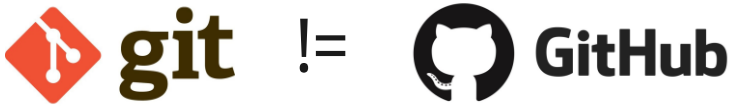
- Distribuido



  - Repositorio remoto
  - Cada cliente tiene copia completa
  - Capacidad de restauración
  - *Git*

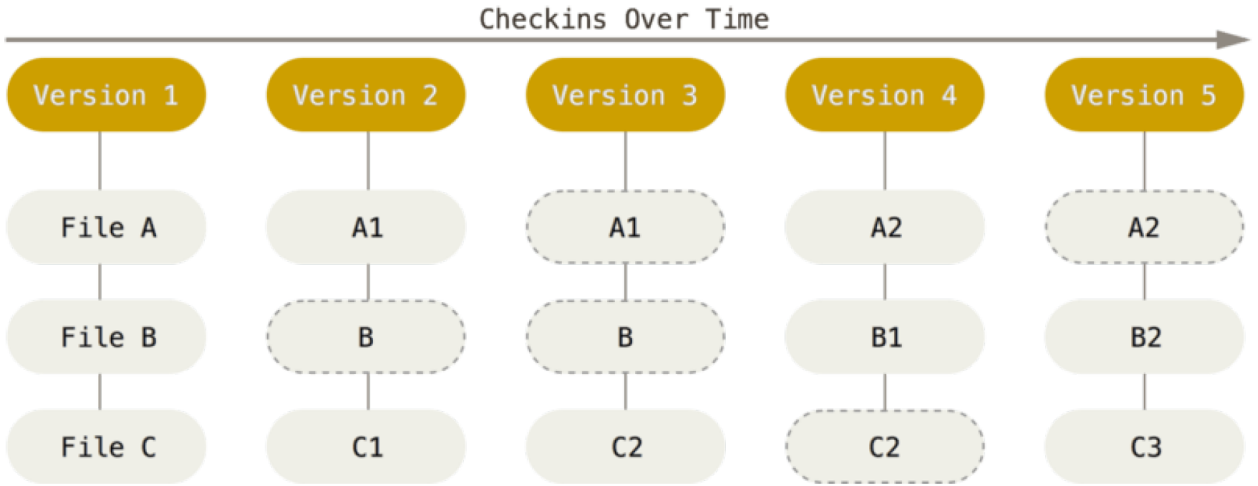
Conceptos básicos Git

- Git
  - Sistema para el control distribuido de versiones de código.
  - Permite dar seguimiento a los cambios realizados sobre un archivo, y almacenar una copia de los cambios.
- GitHub
  - Sitio web donde podemos subir una copia de nuestro repositorio Git.



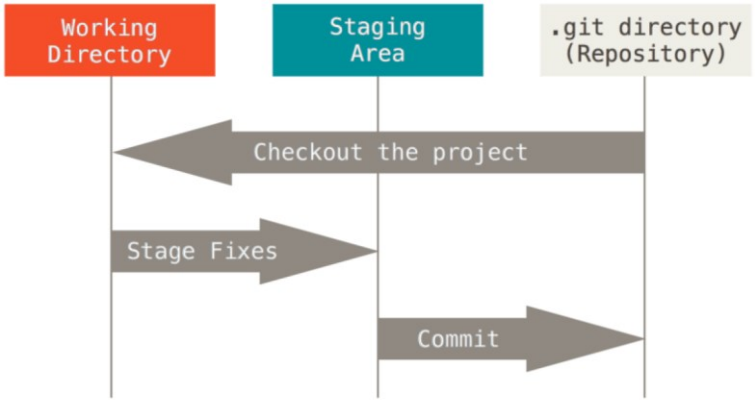
Conceptos básicos Git

- Se basa en el concepto de instantánea (*snapshot*):



Conceptos básicos Git

- En *Git* un archivo se encuentra en uno de los siguientes estados:
  - Confirmado (*commited*): Datos actualizados en la copia central del repositorio.
  - Modificado (*modified*): El archivo tiene cambios que aún no han sido confirmados.
  - Preparado (*staged*): El archivo modificado ha sido marcado para su confirmación.



Conceptos básicos Git

- master: Rama predeterminada (y principal) que se crea al crear el repositorio.
  - También puede llamarse main (por defecto en GitHub).
- origin: Nombre predeterminado para el repositorio remoto principal.
  - Se le da ese nombre automáticamente al clonar un repositorio.
- HEAD: Commit en el que se posiciona el repositorio en un momento concreto.
  - Por lo general, es el último commit.

1. Sistemas de control de versiones

2. Instalación y configuración de Git

3. Uso básico de Git

4. Comandos de GitHub

5. Markdown

6. Principales ejemplos

7. Referencias y tutoriales

Instalación de Git

- https://git-scm.com/

- Comprobar instalación

```
git --version
```

Configuración

- Nombre de usuario

```
git config --global user.name “Luis Martinez”
```

- Email del usuario

```
git config --global user.email in1macal@uco.es
```

- Editor de texto

```
git config --global core.editor “gedit”
```

Configuración

- Habilitar color en la interfaz

```
git config --global color.ui true
```

- Listar la configuración

```
git config --list
```

1. Sistemas de control de versiones

2. Instalación y configuración de Git

3. Uso básico de Git

4. Comandos de GitHub

5. Markdown

6. Principales ejemplos

7. Referencias y tutoriales

Primeros pasos

- Crear un repositorio

```
mkdir proyecto-git
git init proyecto-git
cd proyecto-git
```

Primeros pasos

- Añadir ficheros al área de *staging* o preparación

```
git add fichero.txt
git add -A
```

- Confirmar los cambios

```
git commit -m “mensaje descriptivo”
```

Primeros pasos

- Ver estado o cambios en el directorio

```
git status
```

- Ver diferencia entre ficheros en el directorio y el repositorio de git

```
git diff
git diff --color-words
```

- Ver diferencia entre ficheros preparados (*staged*) y el repositorio

```
git diff --staged
```

Primeros pasos

- Historial de commits

```
git log
git log -n 5
git log --since=2021-09-01
git log --author="Luis Martinez"
git log --oneline
```

Primeros pasos

- Dar formato al log

```
git config --global alias.hist
  "log --pretty=format:'%C(yellow)[%ad]%C(reset)
  %C(green)[%h]%C(reset) | %s %C(bold)[%an]%C(reset)
  %C(blue)%d%C(reset)' --graph --date=short"
```

https://gist.github.com/ecasilla/9669241

```
git hist
```

Primeros pasos

- Añadir etiqueta en el commit actual

```
git tag etiqueta
```

Primeros pasos

- Ignorar archivos o directorios en el repositorio
    - Fichero *.gitignore*

Primeros pasos

- Retirar archivos del *staging*

```
git reset HEAD archivo
```

- Complementar/modificar último commit

```
git commit --amend -m “Mensaje”
```

Primeros pasos

- Revertir último commit

```
git revert HEAD
```

- Revertir a un commit específico (sin dejar huella)

```
git reset --soft id_commit
git reset --mixed id_commit
git reset --hard id_commit
```

Primeros pasos

- Eliminar archivos

```
git rm archivo
  git rm --cached archivo
git commit -m “Mensaje”
```

- Mover o renombrar archivos

```
git mv archivoAntiguo archivoNuevo
git commit -m “Mensaje”
```

Primeros pasos

- Listar el contenido del repositorio

```
git ls-tree master
```

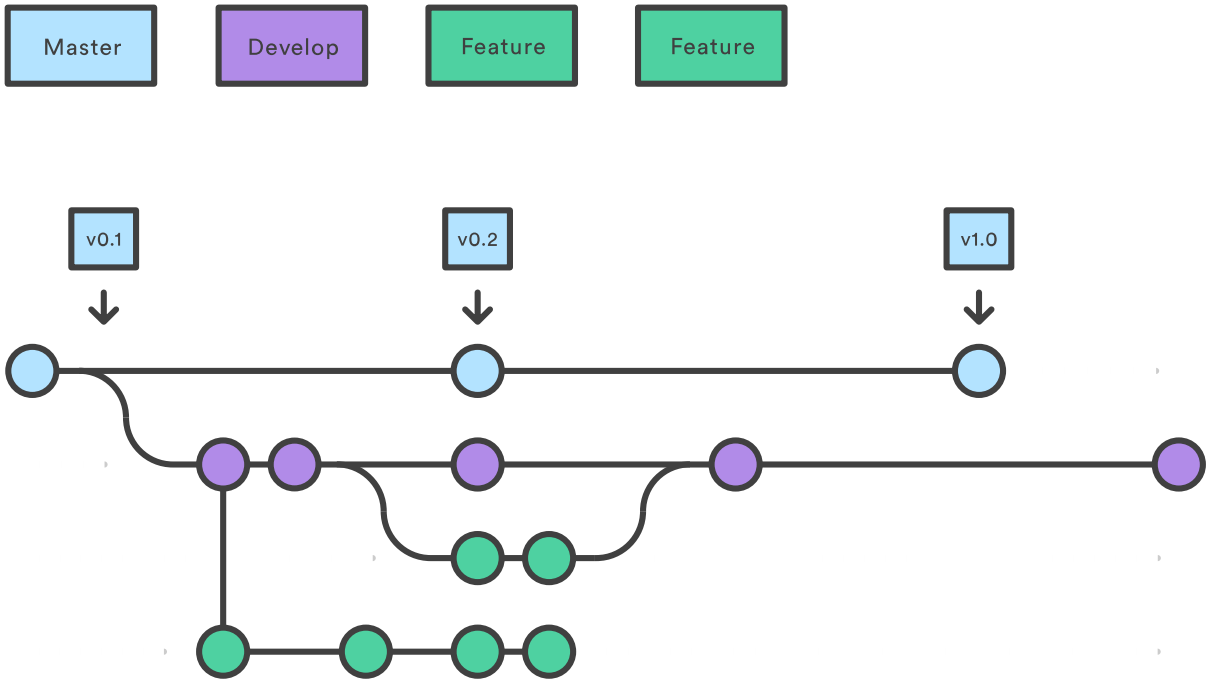
- Examinar el contenido de un commit

```
git show <id>
```

- Comparar un commit con el actual

```
git diff <id> fichero
```

Gestión de ramas



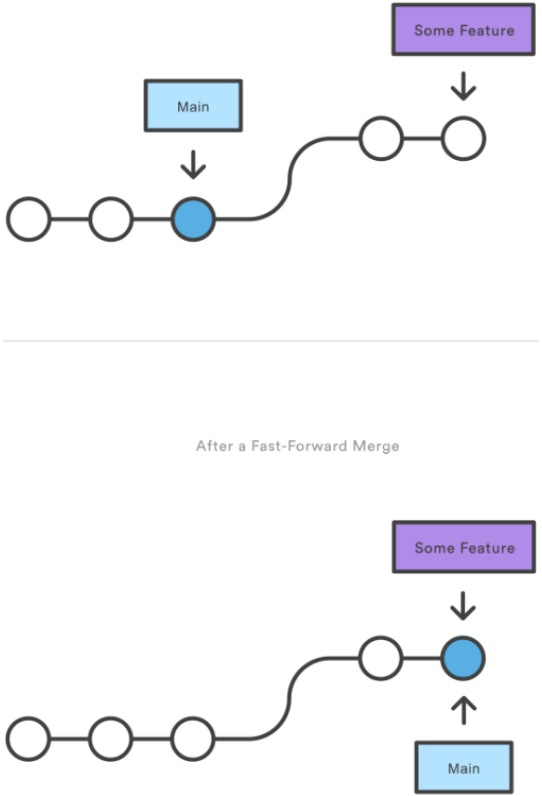
Gestión de ramas

Fusión con avance rápido (fast forward merge).

Ocurre cuando podemos recorrer el historial de revisiones de forma lineal entre ambas ramas.

Por ejemplo, hemos creado una rama **Some Feature** a partir de la master, pero la rama master no ha sufrido cambios después. En este caso, Git añade los commit de la rama Some Feature a la rama master. La creación del punto de fusión es opcional.

Esta estrategia es adecuada para cambios pequeños o corrección de errores.



(a) Fusión con avance rápido

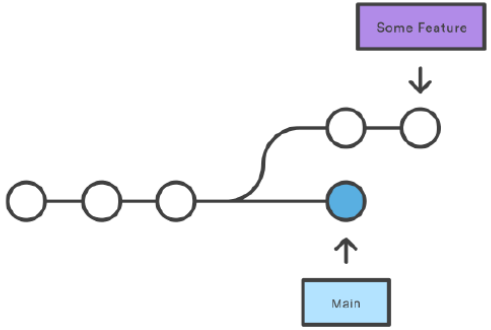
Gestión de ramas

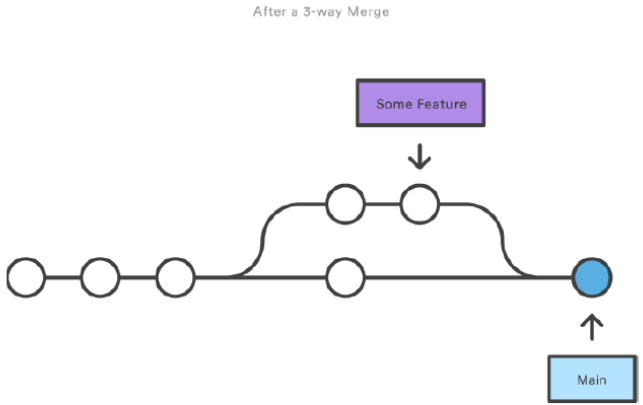
Fusión de tres vías (3-way merge).

Si la rama master ha sufrido modificaciones, hacen falta tres confirmaciones (en los dos extremos y en el predecesor común).

Se creará explícitamente un punto de fusión a continuación de los dos extremos.

Esta estrategia es más habitual cuando se integran funcionalidades de mayor complejidad o que implican mayor tiempo de desarrollo.





(b) Fusión de tres vías

Gestión de ramas

- Ver listado de ramas

```
git branch
```

- Crear una rama

```
git branch nombre_rama
```

- Cambiarse a una rama

```
git checkout nombre_rama
```

Gestión de ramas

- Integrar una rama en la actual

```
git merge nombre_rama
```

Gestión de ramas

- Comparar ramas

```
git diff nombre_rama_1..nombre_rama_2
```

- Renombrar ramas

```
git branch -m nombre_antiguo nombre_nuevo
```

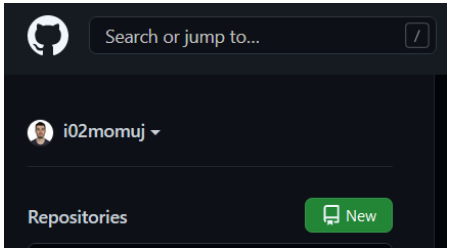
- Eliminar ramas

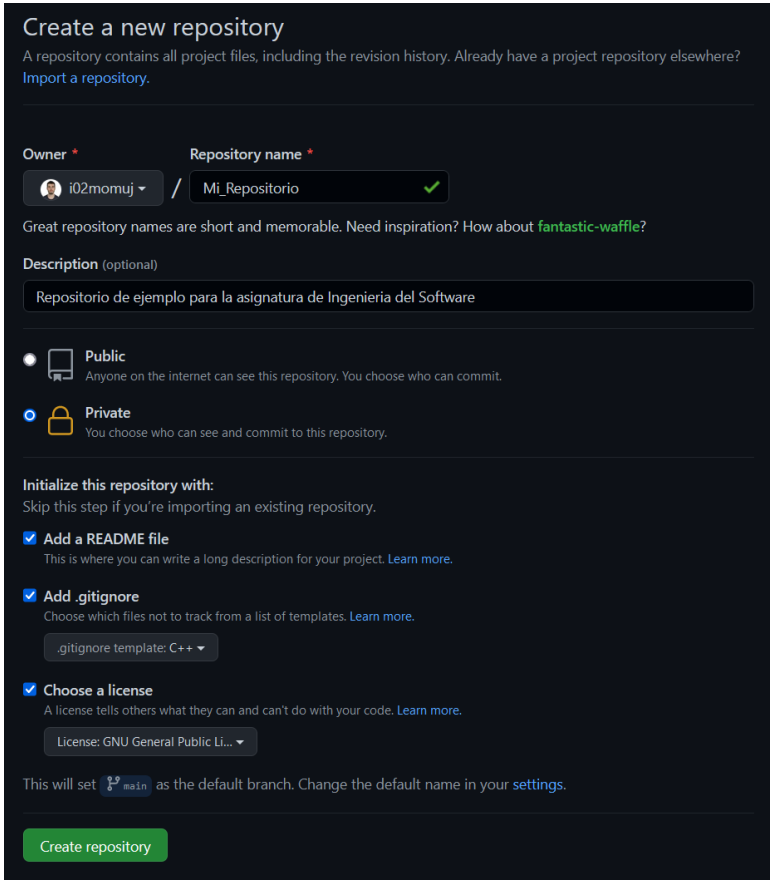
```
git branch -d nombre_rama
git branch -D nombre_rama
```

1. Sistemas de control de versiones
2. Instalación y configuración de Git
3. Uso básico de Git
4. **Comandos de GitHub**
5. Markdown
6. Principales ejemplos
7. Referencias y tutoriales

Crear/clonar repositorio

- https://github.com/





Crear/clonar repositorio

- Clonar repositorio remoto

```
git clone url
```

Crear/clonar repositorio

- Clonar repositorio (I)
  - Crear token personal: *Settings* → *Developer settings*
  - Al hacer *git clone*, nos pide usuario y contraseña
  - La contraseña es el token anterior

  - Si el repositorio es privado, pedirá siempre la contraseña

Crear/clonar repositorio

- Clonar repositorio (II)
  - Obtener clave ssh del ordenador personal

```
cd ~/.ssh
ssh-keygen
cat ~/.ssh/id_rsa.pub
```

En Windows:
* Ejecutar ssh-keygen
* Abrir como fichero de texto

https://phoenixnap.com/kb/generate-ssh-key-windows-10

  - Pegar en GitHub: *Settings → SSH and GPG keys → New ssh key*
  - Clonar con la opción ssh

```
git clone git@github.com:i02momuj/repositorio.git
```

  - Ya no pedirá la contraseña

Comandos básicos

- Añadir repositorio remoto (se hace automáticamente al clonar)

```
git remote add origin url
```

- Ver repositorios remotos

```
git remote -v
```

- Eliminar repositorio remoto

```
git remote rm origin
```

Comandos básicos

- Añadir cambios del repositorio local al remoto

```
git push -u origin master
```

- Comprobar si existen cambios en el repositorio remoto

```
git fetch origin
```

- Añadir cambios del repositorio remoto al local

```
git pull
```

Comandos básicos

- Ver ramas remotas

```
git branch -r
```

- Ver todas las ramas

```
git branch -a
```

1. Sistemas de control de versiones
2. Instalación y configuración de Git
3. Uso básico de Git
4. Comandos de GitHub
5. **Markdown**
6. Principales ejemplos
7. Referencias y tutoriales

Introducción

- Lenguaje pensado para conversión entre texto plano y HTML.
- Sintaxis sencilla para dar formato a contenido web.
- Soportado, entre otros, por GitHub.
    - El repositorio suele tener un fichero README.md

Sintaxis básica

Formato	Sintaxis
Encabezados	`# Primer nivel` `## Segundo nivel`
Negrita	`**Texto negrita**`
Cursiva	`*Texto cursiva*`
Lista numerada	`1. Primer elemento` `2. Segundo elemento`

Lista no numerada	`* Primer elemento`
	`* Segundo elemento`
	`  * Sub-elemento`

Sintaxis básica

Formato	Sintaxis
Línea de código	`‘Línea de código’`
Bloque de código	`‘’’Bloque de código’’’`
Enlace	`[Texto alternativo](URL)`
Imagen	`[Texto alternativo](URL o ruta imagen)`

1. Sistemas de control de versiones
2. Instalación y configuración de Git
3. Uso básico de Git
4. Comandos de GitHub
5. Markdown
6. **Principales ejemplos**
7. Referencias y tutoriales

Creación de repositorio y primer commit

```
mkdir proyecto-git
git init proyecto-git
cd proyecto-git

Creamos un fichero en la carpeta. Se puede hacer usando nano fichA.txt

git add fichA.txt
git commit -m “Añadiendo fichero A”
Modificamos el fichero
git add fichA.txt
git commit -m “Segunda versión del fichero A”
```

Eliminar ficheros del repositorio

```
Partimos de un repositorio donde ya existen fichB.txt y fichC.txt
git rm --cached fichC.txt
git commit -m “Eliminando fichero C del repositorio”

git rm fichB.txt
git commit -m “Eliminando fichero B tanto del repositorio como local”

ls
Vemos como en nuestra carpeta el fichero C aun existe, el B no.
git ls-tree master
En el repositorio han desaparecido ambos ficheros, B y C.
```

Revertir commits

```
Modificamos el fichero fichA.txt
git add -A
git commit -m “Nueva versión del fichero A”

git revert HEAD --no-edit

Si abrimos el fichero fichA.txt, vemos como ya no están los cambios
git hist (o git log)
Vemos que existe un nuevo commit dejando huella del revert
```

Volver a versiones anteriores

```
git hist (o git log)
Copiamos el identificador del commit al que queremos movernos

git reset --hard id_commit

git hist (o git log)
Vemos como el historial de commits llega hasta el que nos hemos movido
```

Gestión básica de ramas

```
git branch rama_A
git checkout rama_A

Modificamos el fichero fichA.txt
git add -A
git commit -m “Commit en rama A”
git hist

git checkout master
git merge rama_A
git hist
```

Crear conflicto en ramas (I)

```
git checkout rama_A
Modificamos el fichero fichA.txt
git add -A
git commit -m “Nueva versión de fichero A en rama A”

git checkout master
Modificamos el fichero fichA.txt
git add -A
git commit -m “Nueva versión de fichero A en master”
```

Crear conflicto en ramas (II)

```
git merge rama_A
Abrimos fichA.txt para solucionar el conflicto

git add -A
git commit -m “Resolviendo conflicto”

git hist
```

1. Sistemas de control de versiones
2. Instalación y configuración de Git
3. Uso básico de Git
4. Comandos de GitHub
5. Markdown
6. Principales ejemplos
7. **Referencias y tutoriales**

7. Referencias y tutoriales

- Atlassian
  - https://www.atlassian.com/es/git
  - https://www.atlassian.com/es/git/tutorials/setting-up-a-repository
- Aula Software Libre UCO
  - https://aulasoftwarelibre.github.io/taller-de-git/
- Diego C. Martín
  - https://www.diegocmartin.com/tutorial-git/
- Youtube Pelado Nerd
  - https://www.youtube.com/watch?v=kEPF-MWGq1w

Ingeniería del Software

Práctica 3

Taller de iniciación a Git

2º Grado en Ingeniería Informática
Universidad de Córdoba
Curso 2024/25