

# GESTION DE CONFIGURACION DEL SOFTWARE

https://www.youtube.com/watch?v=xmT3OJpSTqk



# Introducción

Sistema: colección de componentes que se organizan con el objetivo de proporcionar una función o conjunto de funciones determinadas.

La configuración se puede considerar como una colección de versiones específicas de elementos de hardware, firmware o software que se combinan de acuerdo con un proceso de construcción específico para satisfacer un propósito particular.

La gestión de configuración (GC) es la disciplina de identificar la configuración de un sistema en momentos diferentes con el propósito de controlar de una manera sistemática los cambios en la configuración y mantener la integridad y el seguimiento de los cambios en la configuración durante el ciclo de vida del sistema.

Formalmente el IEEE 610.12-90, define GC como "Una disciplina que establece dirección y seguimiento técnicos y administrativos a: la identificación y documentación de las características funcionales y físicas de un elemento de configuración, toma notas y produce informes de cambios en el proceso y en el estado de implementación y verifica el cumplimiento de los requerimientos especificados."



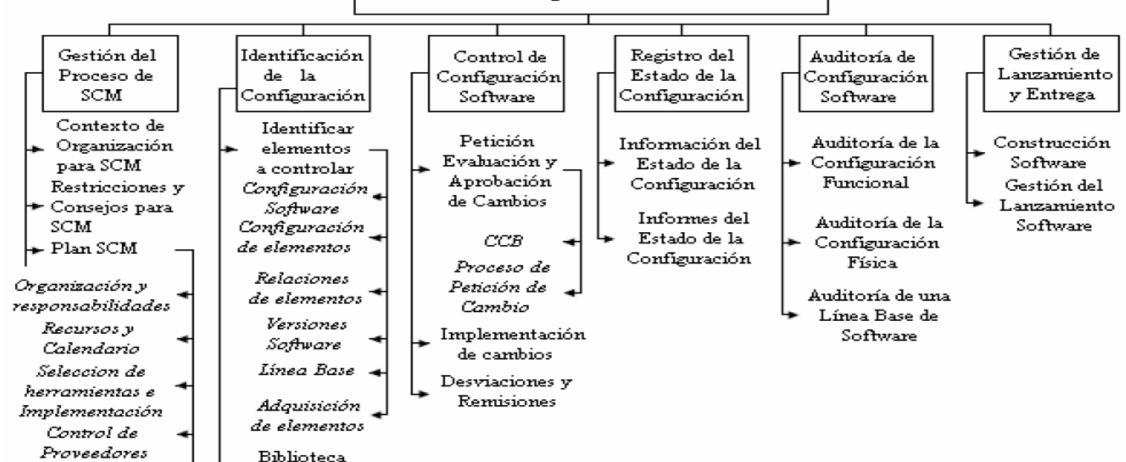
# Gestión de configuración del software

La gestión de la configuración del software (SCM) es un proceso que soporta el ciclo de vida del software (IEEE12207.0-96) que beneficia a la gestión de proyectos, las actividades de desarrollo y mantenimiento, las actividades de garantía y a los clientes y usuarios del producto final.

El concepto de gestión de configuración es aplicable a todos los elementos que se pueden controlar, aunque existen algunas diferencias de implementación entre gestión de configuración del hardware y gestión de configuración del software.

La GCS es una actividad de garantía de calidad del software que se aplica en todas las fases del proceso de ingeniería del software.

e l



Interacción

Plan de Gestión de Configuración

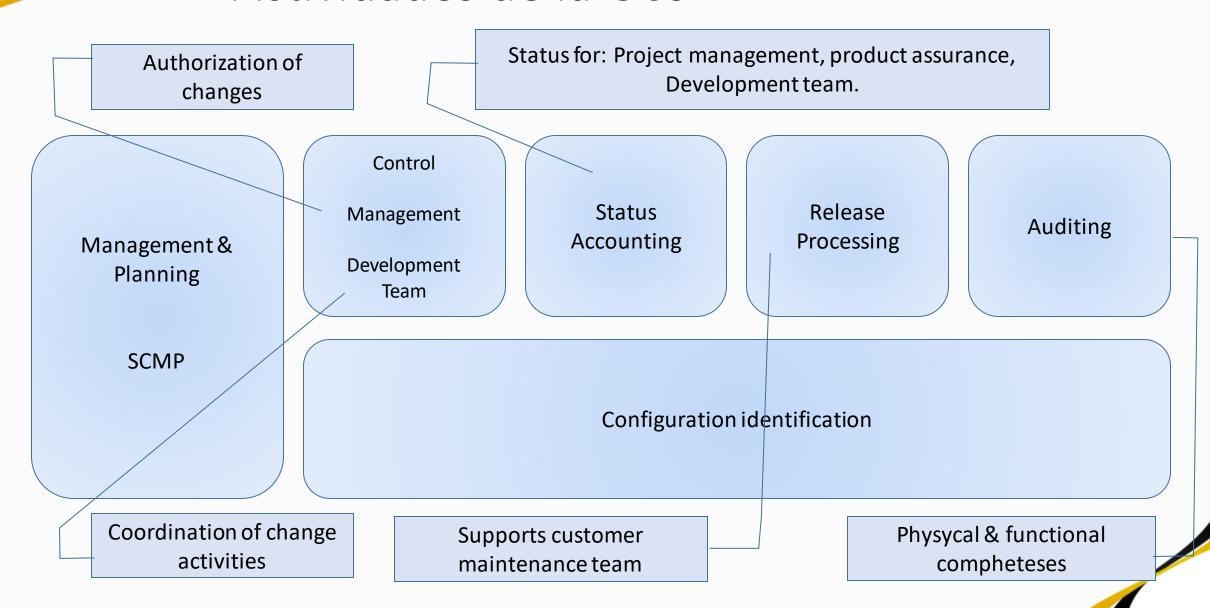
Seguimiento de la SCM

Medidas de SCIM Auditoría de SCIM Software



#### Ingeniería de Sistemas y Computación Ingeniería del Software I

# Actividades de la GCS





Ya que los cambios dentro del desarrollo del SW pueden ocurrir en cualquier momento, se debe estar preparado para:

- Identificar el cambio del software.
- Controlar ese cambio.
- Garantizar que el cambio quede bien implantado.
- Informar el cambio.



# ¿Qué permite conocer la GCS?

- ¿Quién hizo los cambios?
- ¿Qué cambios se hicieron al software?
- ¿Cuándo se hicieron los cambios?
- ¿Por qué se hicieron los cambios?

#### Ingeniería de Sistemas y Computación Ingeniería del Software I

## PROCESO DE GCS

#### **IEEE**

Identificación de la Configuración Control de Cambios en la Configuración Generación de Informes de Estado Auditoria de la Configuración

#### **ISO**

Identificación de la configuración Control de cambios a la configuración Informe del estado de la Configuración Auditoria de la configuración

#### **CMM**

Planificación de las actividades de Gestión de Configuración Identificación de los ECS Control de cambios a los ECS Informar a los grupos e individuos involucrados de los cambios a los ECS Auditoria de la Configuración

#### Actividades básicas de la GCS

Identificación Control Auditoria



## IDENTIFICACION DE LA CONFIGURACION

#### **OBJETIVO:**

Identificar la estructura del Software

#### META:

Tener la capacidad de identificar los componentes del SW

#### PREGUNTAS:

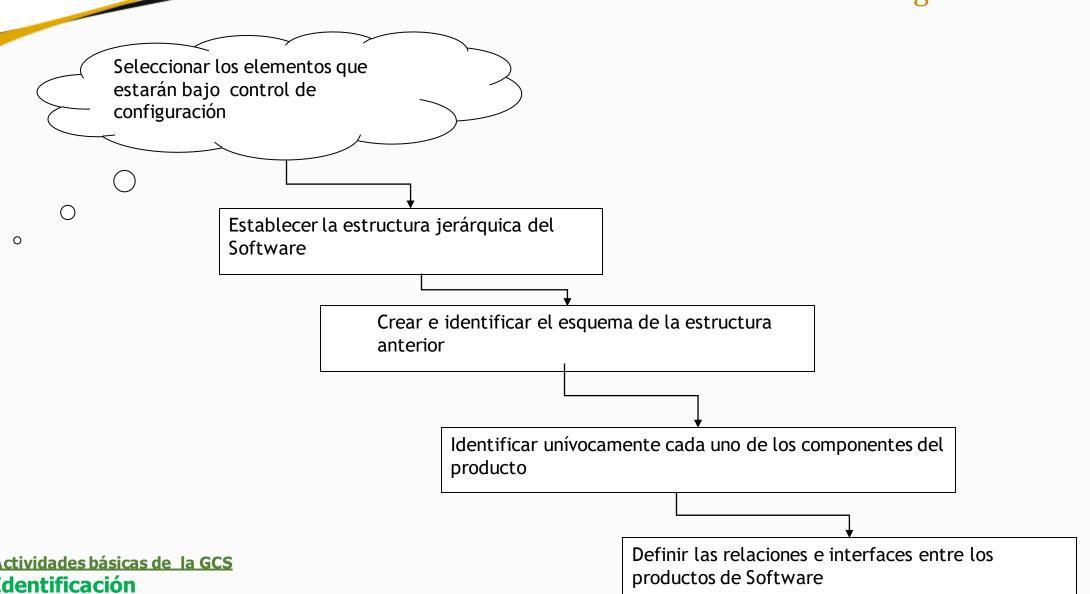
- ¿Cuál es la configuración del Software?
- ¿Que versión de archivo es ésta?
- ¿Cuáles son los componentes del Software?

Actividades básicas de la GCS
Identificación
Control
Auditoria



#### Pasos a seguir:

#### Ingeniería de Sistemas y Computación Ingeniería del Software I



Actividades básicas de la GCS **Identificación** 

Control Auditoria



## CONTROL DE CAMBIOS

#### **OBJETIVO**

Controlar los cambios y la liberación de los productos durante el ciclo de vida.

#### **META**

Establecer un mecanismo que asegure la producción del SW de calidad.

#### PREGUNTAS:

¿Qué esta controlado?

¿Cómo son controlados los cambios a los productos?

¿Quién controla los cambios?

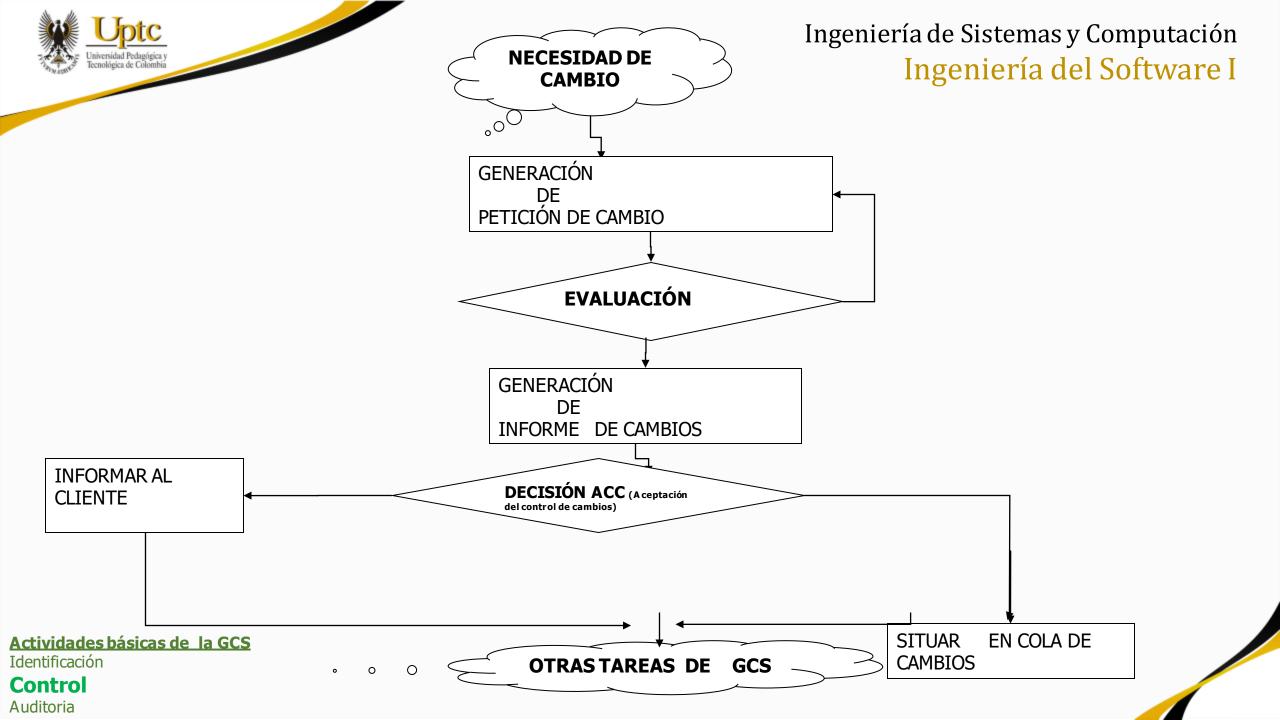
Actividades básicas de la GCS
Identificación
Control
Auditoria

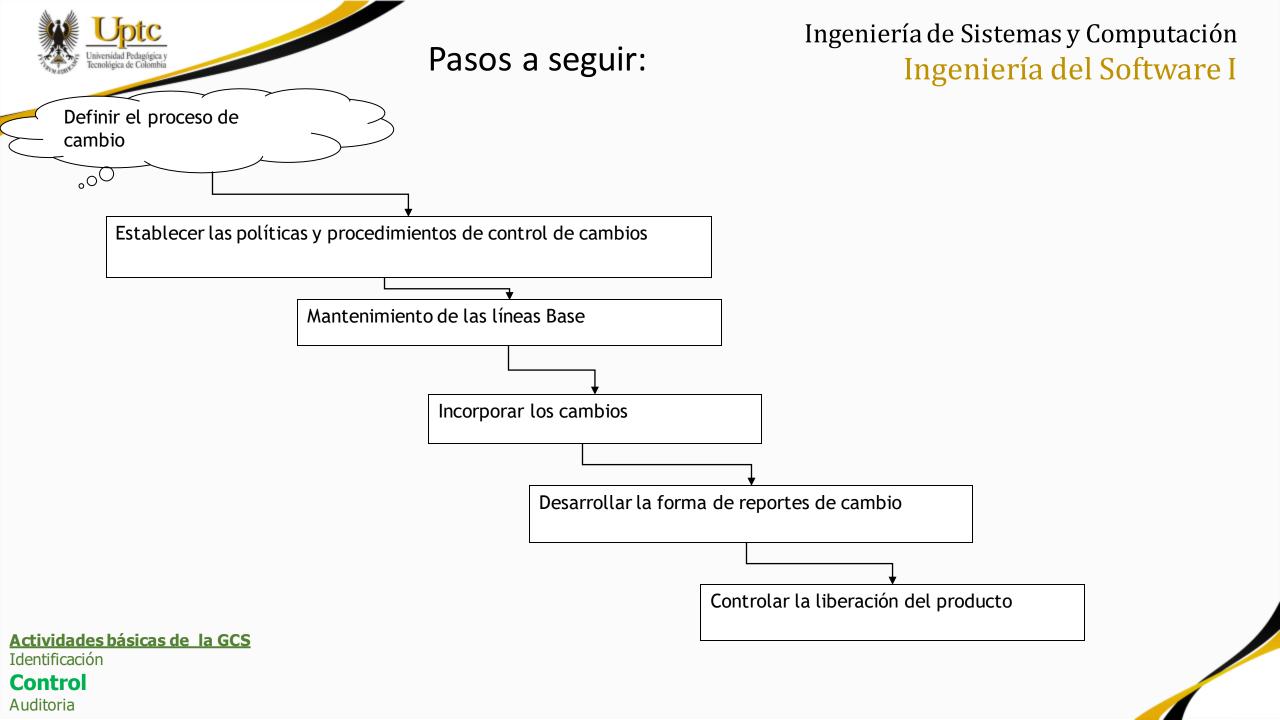


#### CONTROL DE CAMBIOS

Ante un gran esfuerzo de desarrollo de Software, el cambio incontrolado lleva rápidamente al caos.

El control de cambios en la GCS, es de las tareas más importantes para prevenir impacto negativo (o fracaso) en el desarrollo y resultado del proyecto.







## AUDITORIA DE LA CONFIGURACION

#### OBJETIVO

Verificar que el producto de SW integrado satisface los requerimientos estándares o acuerdos contractuales y que los componentes que se integran corresponden con las versiones vigentes.

#### — META

Verificar que todos los productos de SW han sido producidos descritos e identificados correctamente y que todas las solicitudes de cambio han sido procesadas.



## **AUDITORIA DE CONFIGURACION**

La auditoria se plantea las siguientes interrogantes:

- ¿Se ha hecho el cambio especificado en la Orden de Cambio de Ingeniería (OCI)?
- ¿Se han incorporado modificaciones adicionales?
- ¿Se ha llevado a cabo una revisión técnica formal para comprobar la corrección técnica?
- ¿Se han seguido adecuadamente estándares de ingeniería de Software?
- ¿Se han remarcado los cambios en el documento de Elementos de configuración de Software (ECS)?
- ¿Se han especificado la fecha del cambio y el autor del cambio?
- ¿Refleja la identificación del ECS los cambios?
- ¿Se han seguido procedimientos del GCS para señalar el cambio, registrarlo y divulgarlo?
- ¿Se han actualizado adecuadamente todos los ECS relacionados?



## AUDITORIA DE LA CONFIGURACION

¿Cómo asegurar que el cambio se ha implementado correctamente?

- 1) Revisiones técnicas formales : se centran en la corrección técnica del elemento de configuración que ha sido modificado.
- 2) Auditorias de configuración del software: complementa la revisión técnica formal



Ingeniería del Software I

# Herramientas para Gestión de Configuración

Git

GitHub

**CVS** 

Subversión

• • •

Identificación Control

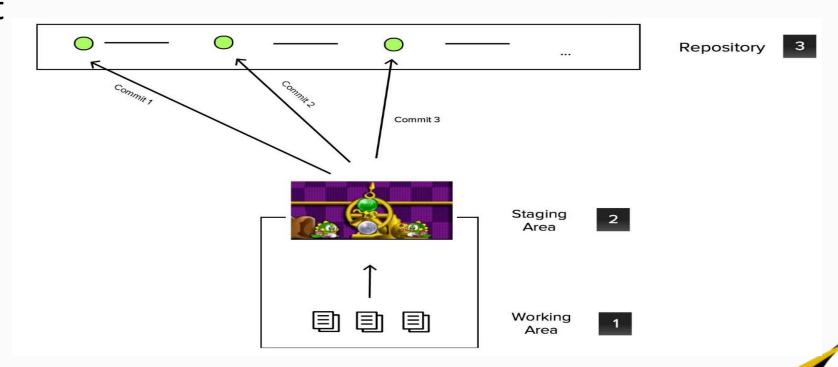


# Taller con Git

Git – Repositorio de escritorio

Github – Ambiente web colaborativo

• Arquitectura de Git





## Pasos

#### 1. Instalar Git

Descargar instalador: <a href="https://git-scm.com/download/win">https://git-scm.com/download/win</a>

### 2. Configurar Git

En windows se trabaja principalmente en GitBash : ambiente de trabajo modo consola



# Comandos iniciales

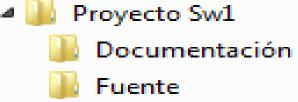
- git --version (muestra la version de Git instalada)
- git config --global user.name "Nombre Apellido" (Usuario autorizado en Git)
- git config --global user.name (Ver Usuario)
- git config --global user.email "cuenta@dominio" (Email usuario autorizado en Git)
- git config --global --list (Ver configuración Git)
- git help (consulta general de la ayuda)
- git help COMANDO (consulta particular de la ayuda)



# Crear proyecto

 Crear una carpeta Raíz donde se almacenará la información del proyecto.

• Ejemplo:



• En consola, ubicarse en la carpeta creada

```
Otalora@Otalora-HP MINGW64 /c

$ cd Proyecto\ Sw1/

Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1

$ ls

Documentación/ Fuente/

Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1

$ |
```



# Comando para el seguimiento del proyecto

- Comando para iniciar el rastreo de los cambios en el proyecto
  - git init
- Comando para revisar el estado del proyecto
  - git status

```
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1

$ git init
Initialized empty Git repository in C:/Proyecto Sw1/.git/
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1 (master)

$ git status
On branch master
Initial commit
Untracked files:
    (use "git add <file>..." to include in what will be committed)

    "Documentaci\303\263n/"
Fuente/
nothing added to commit but untracked files present (use "git add" to track)

Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1 (master)

$ |
```

Aún no está en el repositorio.



# Subiendo la Línea Base

• Se sube la primera versión de los artefactos, así:

git add –A

Comando: add (adicionar).

Parametro A (All) subir todo.

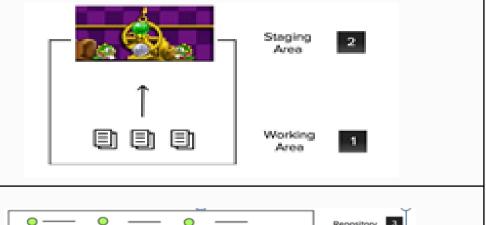
git add NOMBRE\_ARCHIVO

Comando para adicionar un solo archivo

git commit -m "Línea Base Proyecto"

Comando: commit (actualizar)

Parametro m (message) mensaje del cambio





git commit - am "MENSAJE"

Concatena los dos comandos anteriores y presenta mensaje al equipo de trabajo



# Subiendo la línea base

```
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1 (master)
$ cd Documentación
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1/Documentación (master)
 'Documentación Requisitos del proyecto.docx' 'Documento de arquitectura.docx'
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1/Documentación (master)
$ git status
On branch master
Untracked files:
   (use "git add <file>..." to include in what will be committed)
nothing added to commit but untracked files present (use "git add" to track)
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1/Documentación (master)
$ git add Documento de arquitectura.docx
fatal: pathspec 'Documento' did not match any files
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1/Documentación (master)
$ git add 'Documento de arquitectura.docx'
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1/Documentación (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
          new file: Documento de arquitectura.docx
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1/Documentación (master)
$ git commit -m "Se adicionó el archivo de arquitectura del sistema"
[master b8f5303] Se adicionó el archivo de arquitectura del sistema
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Documentaci\303\263n/Documento de arquitectura.docx"
```



# Revisar historial

• git log (comando para revisar historial de cambios)

```
Otalora@Otalora-HP MINGW64 /c/Proyecto Sw1/Documentación (master)
$ git log
commit b8f53032bd34995864630efb513d688b6b70ad47
Author: jorge otalora <jorge.otalora@uptc.edu.co>
Date: Tue Feb 28 14:56:21 2017 -0500

Se adicionó el archivo de arquitectura del sistema

commit 1f5b97152d728b05c98d8f474a97878ddff8acac
Author: jorge otalora <jorge.otalora@uptc.edu.co>
Date: Tue Feb 28 14:37:44 2017 -0500

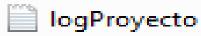
Linea Base Proyecto
```

• git log > LogProyecto.txt (comando para crear un archivo con el historial de cambios

del proyecto)







```
Date: Tue Feb 28 14:56:21 2017 -0500

Se adicionð el archivo de arquitectura del sistema

commit 1f5b97152d728b05c98d8f474a97878ddff8acac

Author: jorge otalora <jorge.otalora@uptc.edu.co>

Date: Tue Feb 28 14:37:44 2017 -0500
```

commit b8f53032bd34995864630efb513d688b6b70ad47
Author: jorge otalora <jorge.otalora@uptc.edu.co>



# Navegar entre versiones

git checkout CODIGO\_COMMIT

(El código commit se encuentra en el archivo Log) → Permite regresar a una versión anterior del proyecto.

• git checkout master

(Comando que permite regresar a la última versión del proyecto).

Documentación Requisitos del proyecto

Documentación Requisitos del proyecto

Documento de arquitectura



# Trabajo individual

- Instalar y configurar Git
- Crear el ambiente de trabajo de un proyecto
- Explorar y plantear un ejemplo del uso de ramas (branch) y fusiones (merge) en Git.