



# TECNÓLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE

CENTRO DE GESTIÓN DE MERCADOS, LOGÍSTICA Y  
TECNOLOGÍAS DE LA INFORMACIÓN



@SENAComunica

[www.sena.edu.co](http://www.sena.edu.co)

# AGENDA.

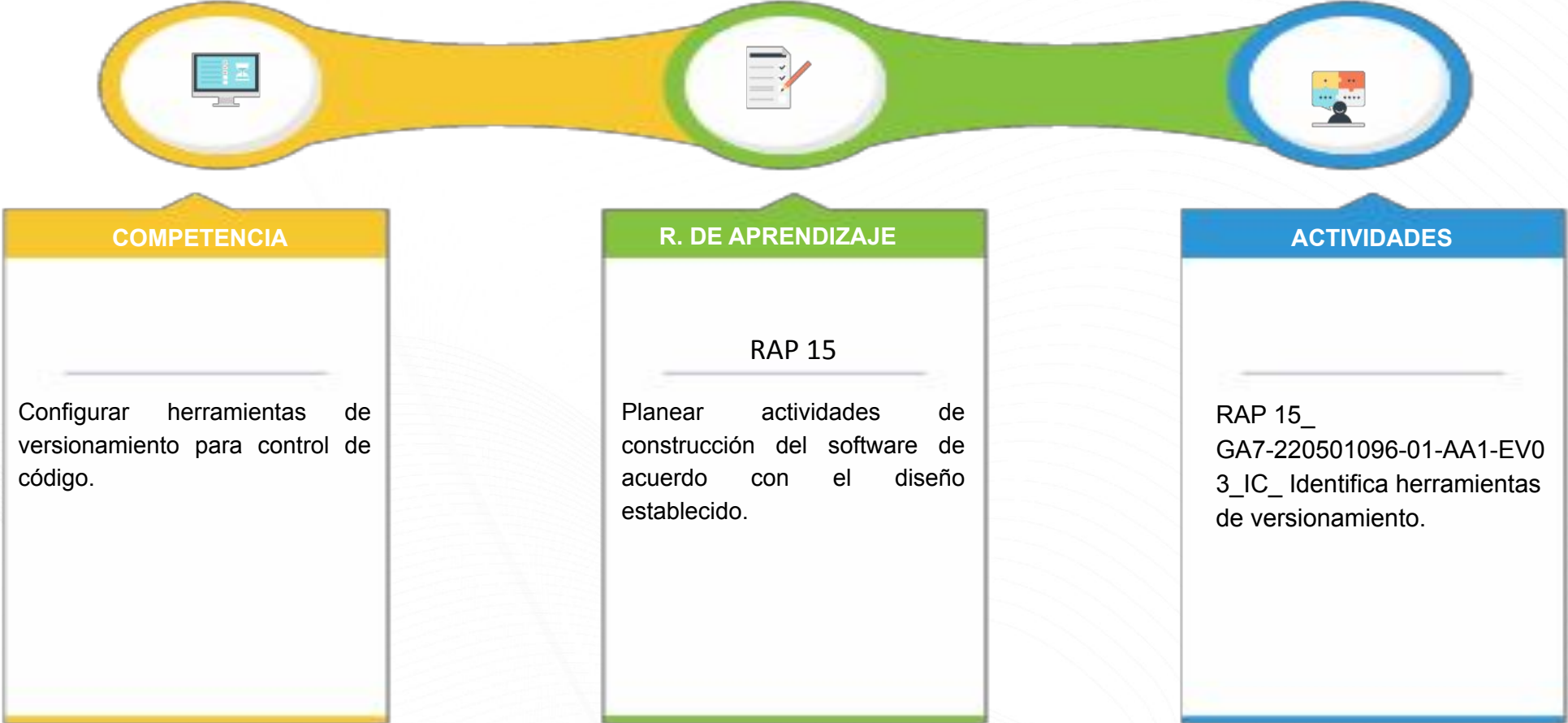
---

- RAP 15\_ GA7-220501096-01-AA1-EV03\_IC\_ Identifica herramientas de versionamiento.
- RAP 15\_ GA7-220501096-AA1-EV04\_RM\_ Instalación y configuración de herramienta de versionamiento (Local / Web).
- RAP 15\_ GA7-220501096-AA1-EV05\_IC\_ Herramientas de versionamiento (GIT) instalada y configurada.

# CRONOGRAMA



		<b>GA7-220501096-AA1-</b> Configurar herramientas de versionamiento para control de código.	<b>220501096-01.</b> Planear actividades de construcción del software de acuerdo con el diseño establecido.	<b>RAP 15</b>	<b>Evidencia de conocimiento:</b>  Identifica herramientas de versionamiento. <b>GA7-220501096-01-AA1-EV03.</b> Calificable.	13/11/2023	19/11/2023		Instructor Vocero
					<b>Evidencias de desempeño:</b>  Instalación y configuración de herramienta de versionamiento (Local / Web). <b>GA7-220501096-AA1-EV04.</b> Calificable.	13/11/2023	26/11/2023		Instructor Técnico
					<b>Evidencias de producto:</b>  Herramientas de versionamiento (GIT) instalada y configurada. <b>GA7-220501096-AA1-EV05.</b> Calificable.	13/11/2023	26/11/2023		Instructor Vocero







- **Evidencia de conocimiento:** (RAP 15) GA7-220501096-AA1-EV03 **identifica** herramientas de versionamiento

Tomando como referencia el componente formativo "Integración continua", realice una tabla con las diferencias entre el sistema de control de versionamiento git local y git remoto.

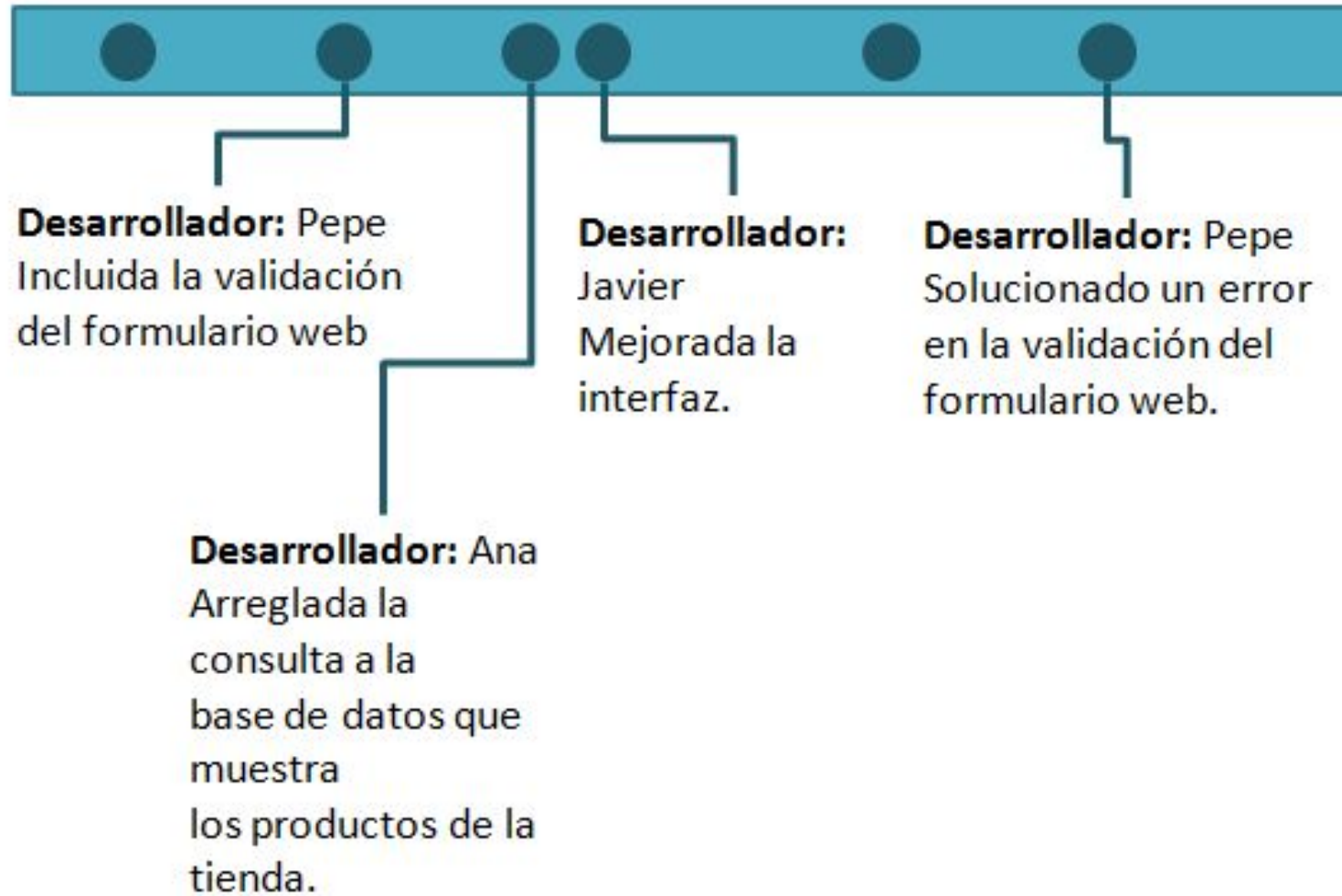
**Elementos para tener en cuenta en el documento:**

- Se deben seguir las normas básicas de presentación de un documento escrito, es decir el documento debe tener como mínimo una portada, introducción, objetivo, tablas con diferencias y características y comandos de git local y git remoto.
- Realice una tabla con la descripción de los comandos básicos de git remoto y git local

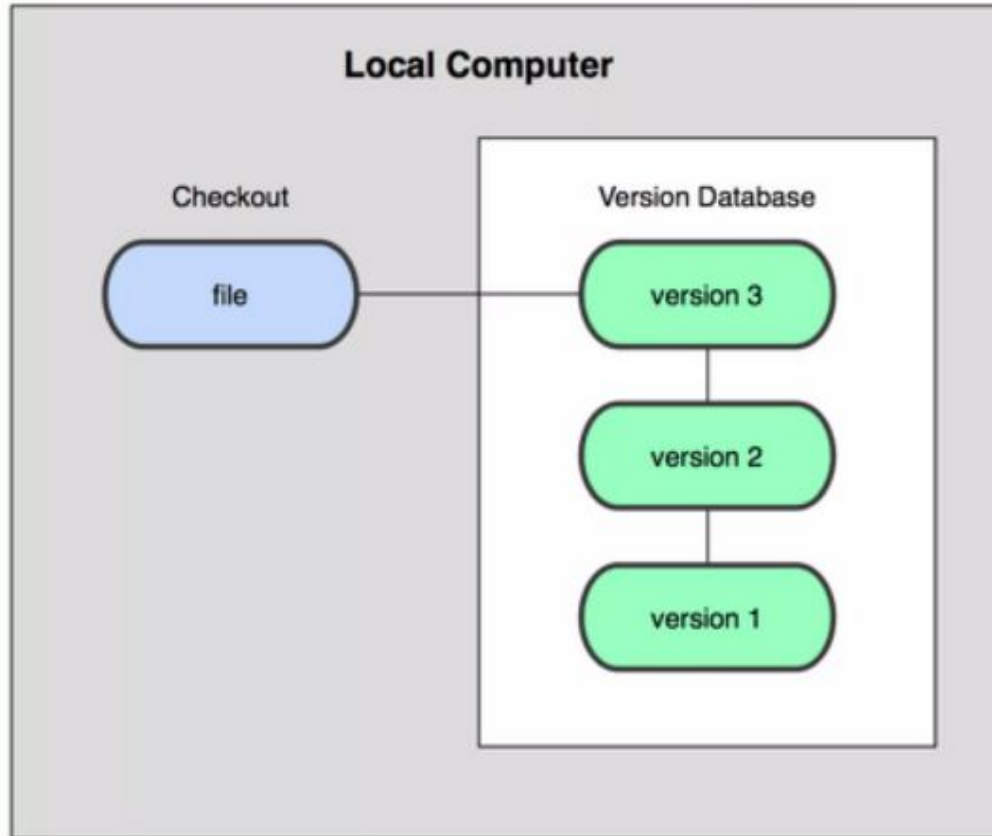
**Lineamientos generales para la entrega de la evidencia:**

- **Productos para entregar:** documento informe técnico
- **Extensión:** Libre.
- **Formato:** PDF.
- Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio: **identifica herramientas de versionamiento GA7-220501096-AA1-EV03.**

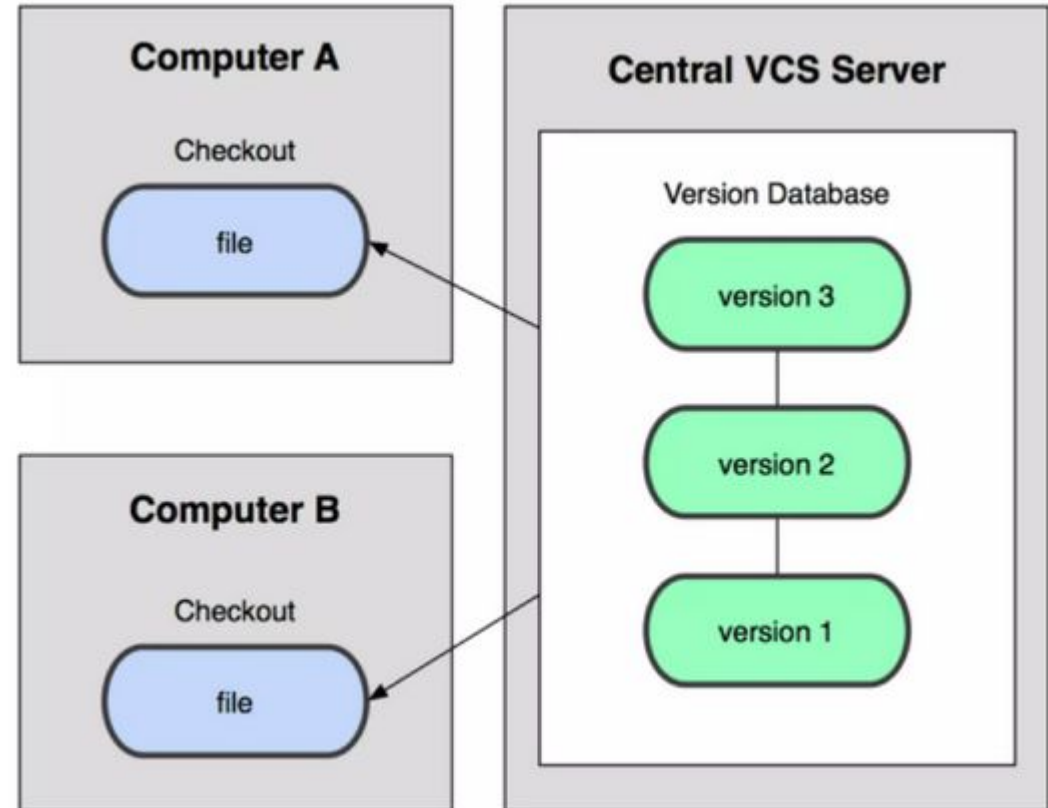
# HERRAMIENTAS DE VERSIONAMIENTO



# HERRAMIENTAS DE VERSIONAMIENTO



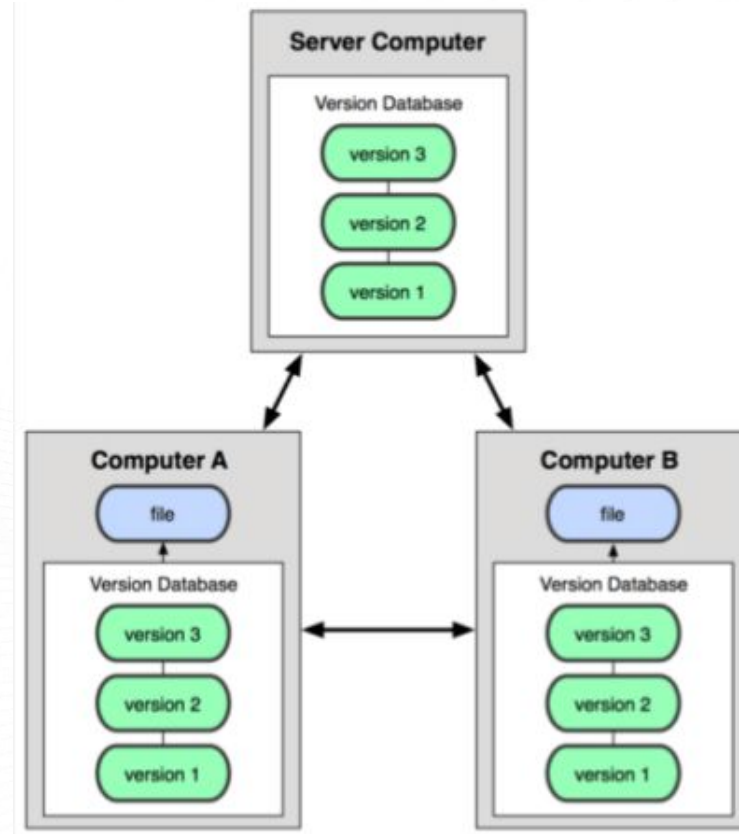
LOCALES



CENTRALIZADOS



# HERRAMIENTAS DE VERSIONAMIENTO

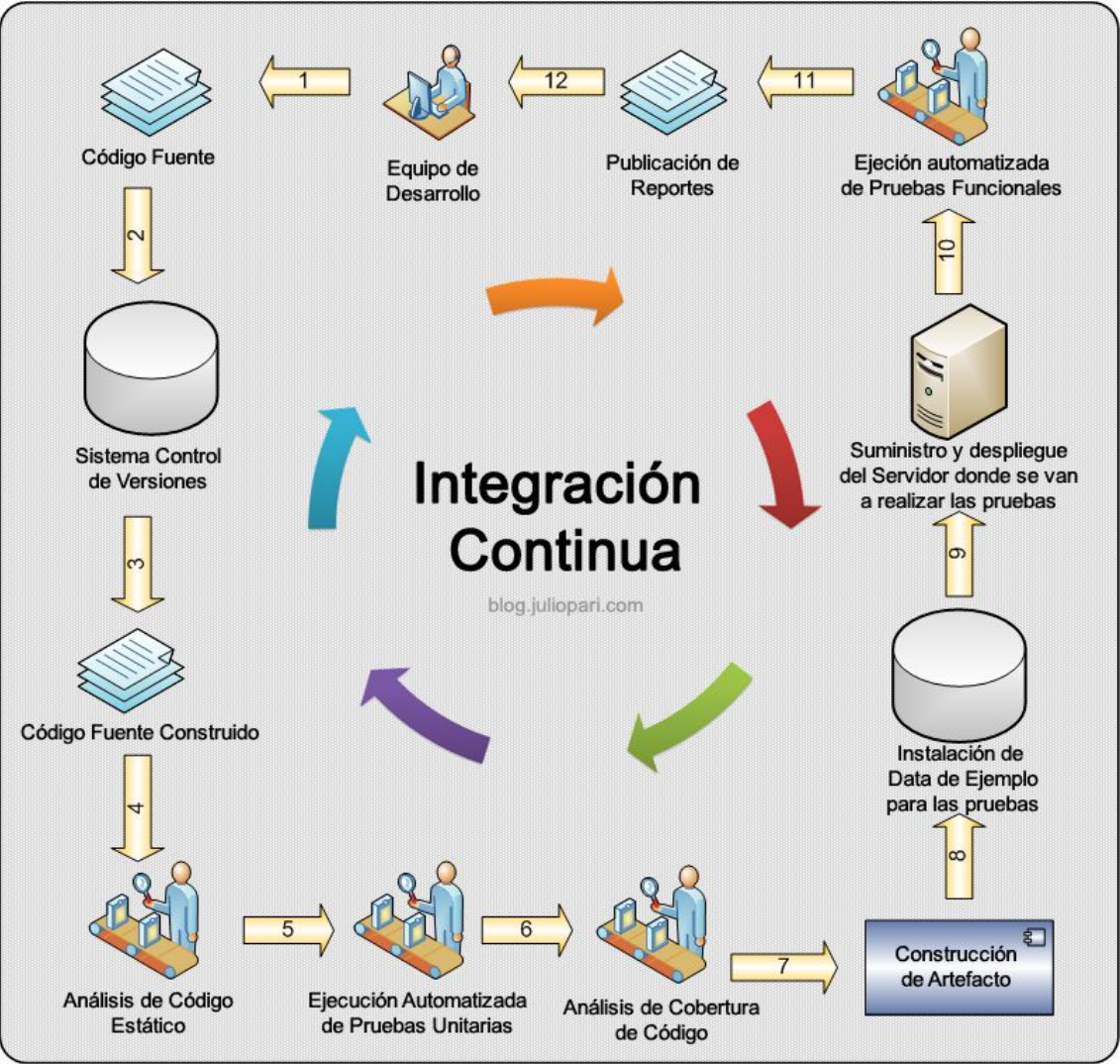


DISTRIBUIDOS

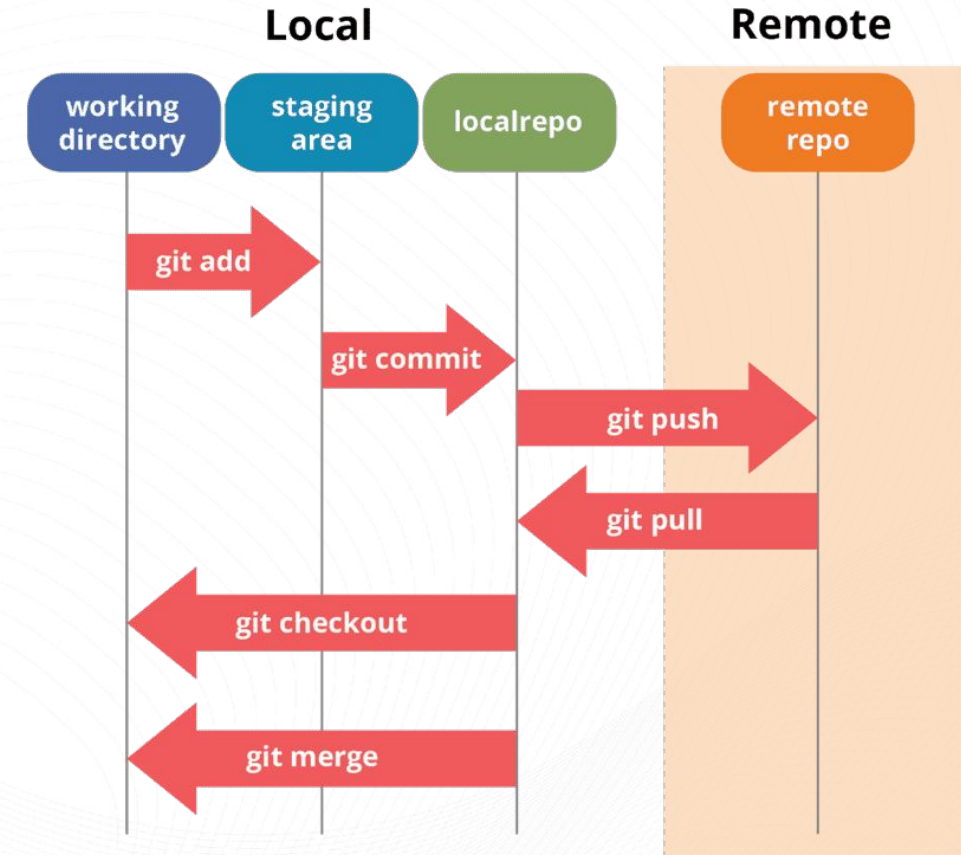
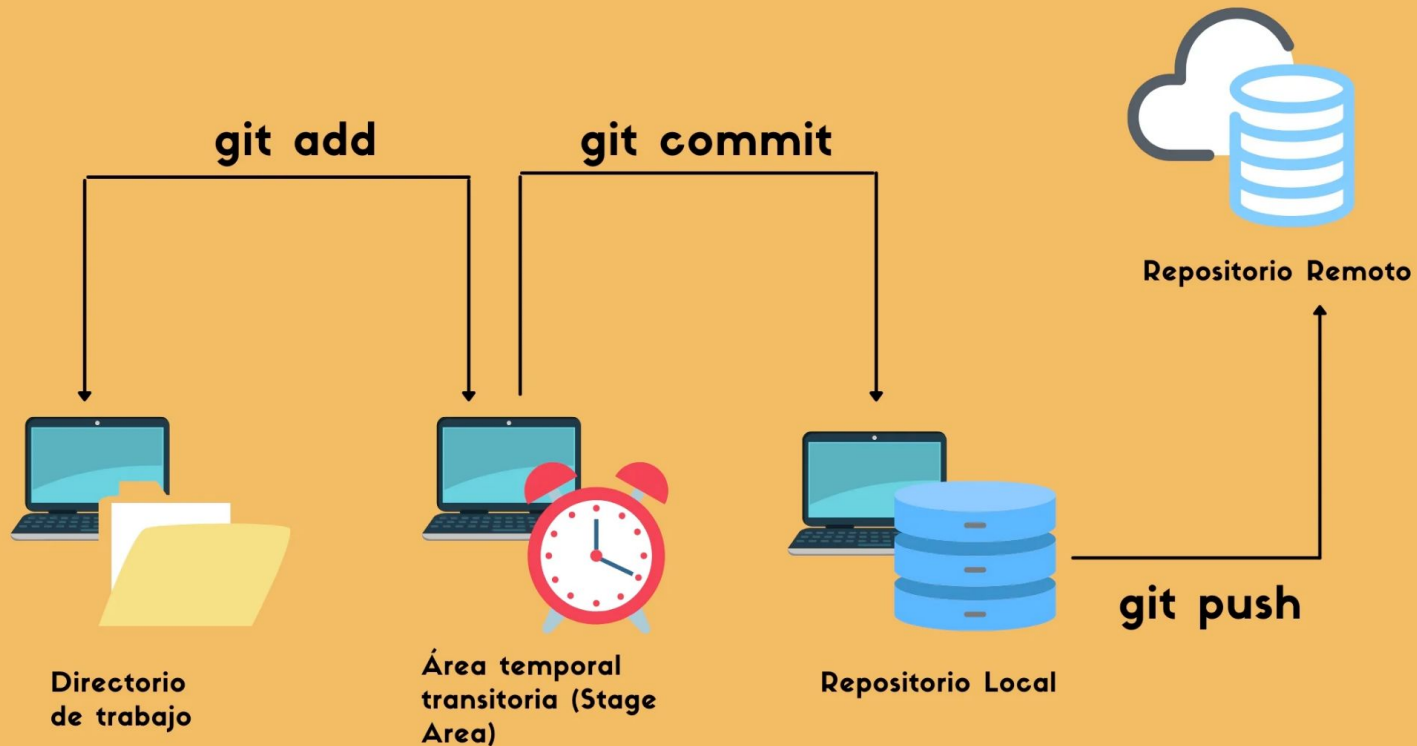




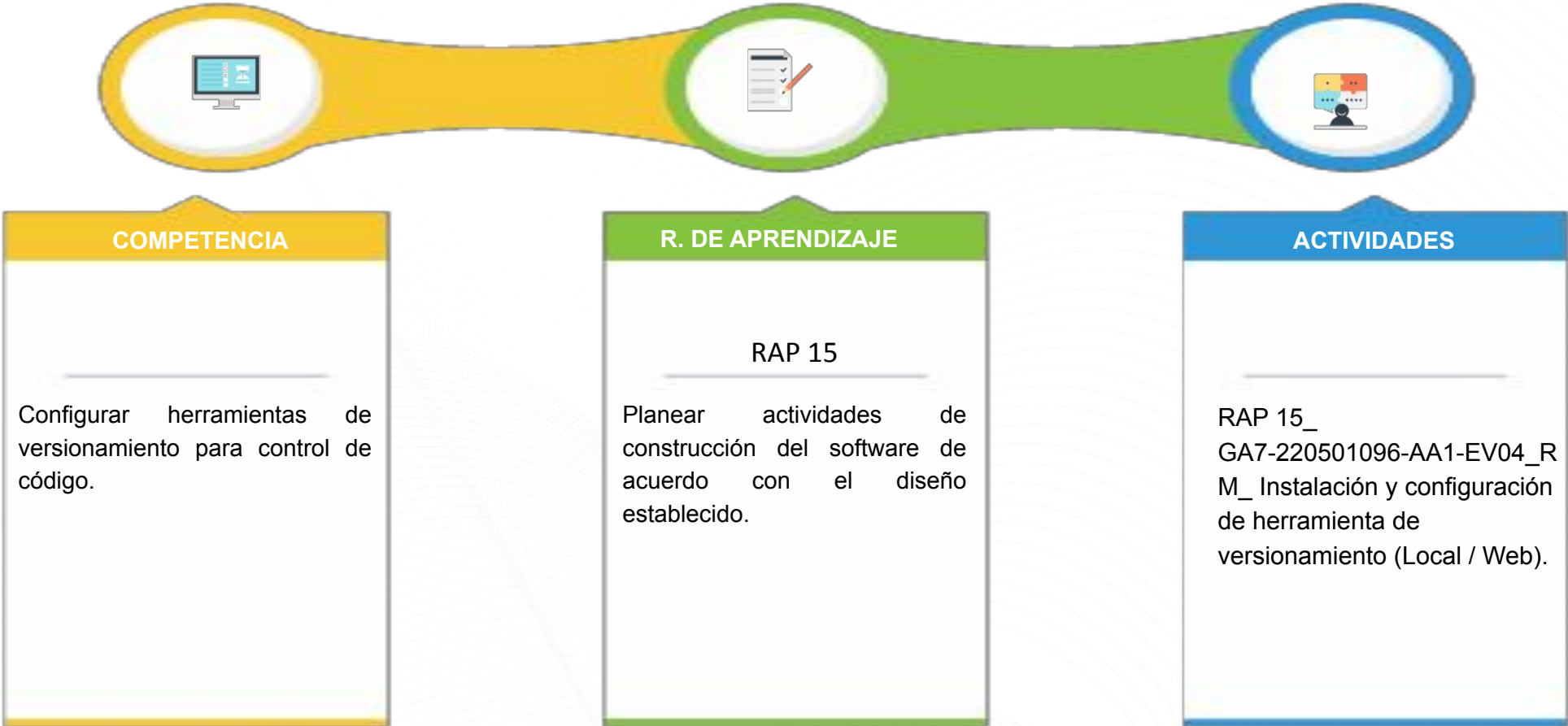
# HERRAMIENTAS DE VERSIONAMIENTO



# HERRAMIENTAS DE VERSIONAMIENTO









- Evidencia de desempeño: **(RAP 15)** GA7-220501096-AA1-EV04 instalación y configuración de herramienta de versionamiento (Local / Web)

Tomando como referencia el componente formativo "Integración continua", realice la instalación y configuración de las herramientas de control de versionamiento tanto local como remoto.

GFPI-F-135 V01



#### Elementos para tener en cuenta en el documento:

- Se deben seguir las normas básicas de presentación de un documento escrito, es decir el documento debe tener como mínimo una portada, introducción, objetivo, paso a paso con pantallazos de la instalación de las herramientas de control de versionamiento tanto local como remota.

#### Lineamientos generales para la entrega de la evidencia:

- **Productos para entregar:** documento instalación
- **Extensión:** Libre.
- **Formato:** PDF.
- Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio: **Instalación y configuración de herramienta de versionamiento (Local / Web) GA7-220501096-AA1-EV04.**





## ¿Cómo empezar con GIT?



Empieza creando una cuenta en GitHub e instalando Git en tu equipo. Crea un repositorio local en Git y un nuevo repositorio en Github. Añade archivos a tu repositorio, crea commits y crea nuevas ramas.



https://github.com/signup?source=login

Already have an account? [Sign in](#) →

Welcome to GitHub!  
Let's begin the adventure

Enter your email\*

→

Continue

Ver credenciales guardadas

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.



Already have an account? [Sign in](#)

Welcome to GitHub!

Let's begin the adventure

Enter your email\*

✓ JJDONATO@MISENA.EDU.CO

Create a password\*

✓ .....

Enter a username\*

✓ JHONDONATOAR

Would you like to receive product updates and announcements via email?

Type "y" for yes or "n" for no

✓ y

Verify your account

Protección de su cuenta



Dashboard

Search

Type to search

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repositoryImport repository

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Home

Send feedbackFilter 8

Updates to your homepage feed

We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design.

Learn more

Start writing code

Start a new repository

A repository contains all of your project's files, revision history, and collaborator discussion.

JHONDONATO

AR /

name your new repository...

Public

Anyone on the internet can see this repository

Private

You choose who can see and commit to this repository

Create a new repository

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

JHONDONATOAR / README.md

Create

1 - Hi, I'm @JHONDONATOAR

2 - I'm interested in ...

3 - I'm currently learning ...

4 - I'm looking to collaborate on ...

5 - How to reach me ...

6

Latest changes

3 hours ago

Hallo & 안녕하세요 – GitHub Mobile is now also available in German and Korean.

Yesterday

Copilot content exclusions – Temporary rollback and upcoming fix

4 days ago

Dependabot auto-triage rules support CVE IDs and GHSA IDs

5 days ago

Custom Organization Roles are now GA

View changelog

Explore repositories

comixed / comixed

The ComiXed Digital Comic Management System






https://github.com/new

☆

📧 ⬇️ 📄 ☰

☰

 New repository

🔍 Type to search


>

+

🕒

🔗

📧




### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

Owner \*

Repository name \*

 JHONDONATOAR

/


ADSOSENA

✔️ ADSOSENA is available.


Great repository names are short and memorable. Need inspiration? How about [solid-spork](#) ?

Description (optional)

Repositorio de prueba

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

RAP 15\_ GA7-220501096-AA1-EV04\_RM\_ Instalación y configuración de herramienta de versionamiento (Local / Web).

A screenshot of a web browser showing the GitHub repository page for "JHONDONATOAR / ADSOSENSA". The browser's address bar shows the URL "https://github.com/JHONDONATOAR/ADSOSENSA". The repository page includes a header with the repository name, a search bar, and navigation links like "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". Below the header, there are buttons for "Pin", "Unwatch" (1), "Fork" (0), and "Star" (0). The main content area features two cards: "Set up GitHub Copilot" and "Add collaborators to this repository". Below these is a "Quick setup" section with a light blue background, showing options to "Set up in Desktop" or "HTTPS SSH" with the repository URL. It also includes a command line section for creating a new repository and a section for pushing an existing repository from the command line.

← → ↻ https://github.com/JHONDONATOAR/ADSOSENSA ☆

JHONDONATOAR / ADSOSENSA 🔍 Type to search + ⌵ ⌵ ⌵ ⌵

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ADSOSENSA Public Pin Unwatch 1 Fork 0 Star 0

**Set up GitHub Copilot**  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
[Get started with GitHub Copilot](#)

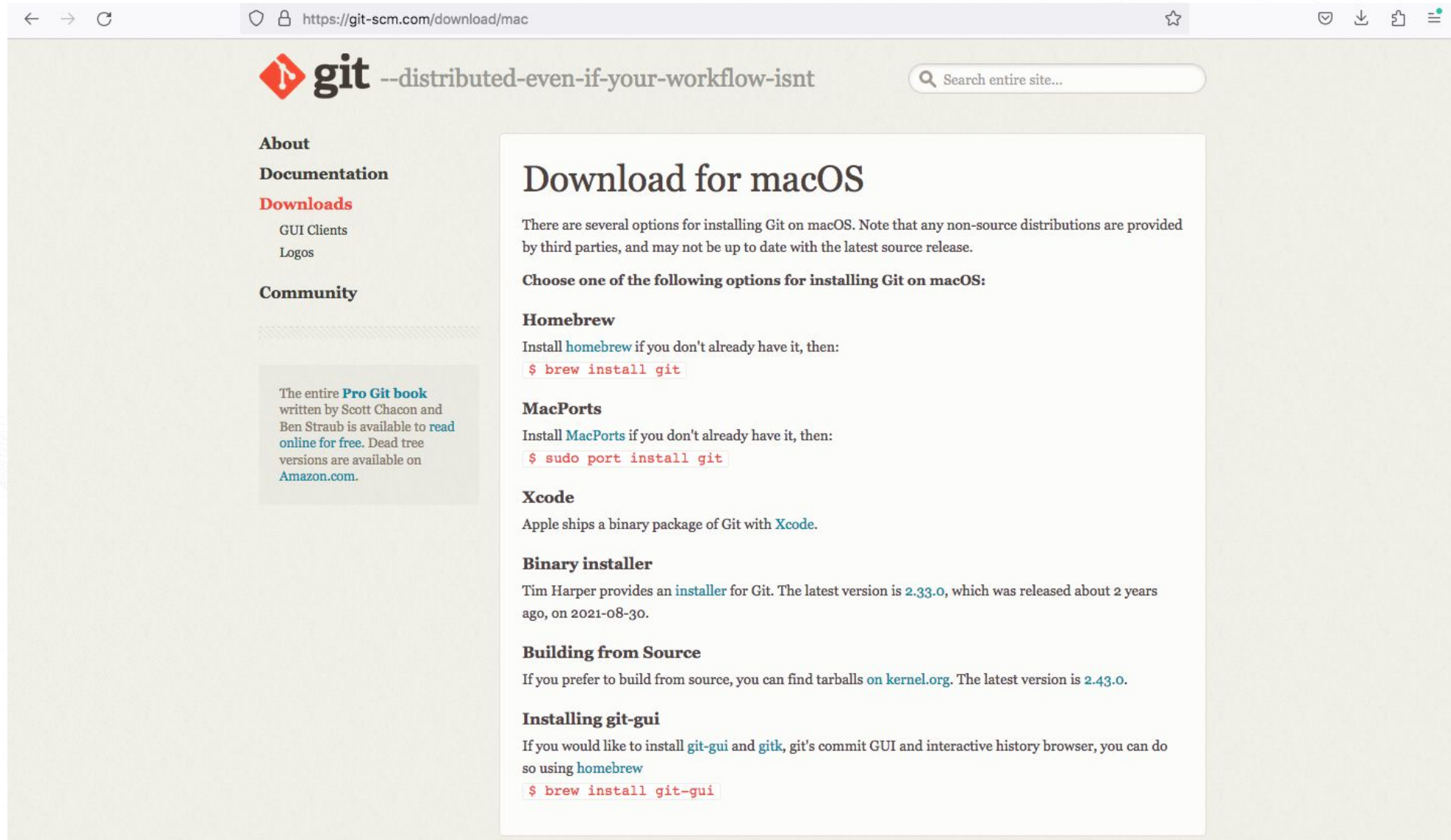
**Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**  
[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/JHONDONATOAR/ADSOSENSA.git>  
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# ADSOSENSA" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/JHONDONATOAR/ADSOSENSA.git
git push -u origin main
```

**...or push an existing repository from the command line**



The screenshot shows the Git website's download page for macOS. The browser's address bar displays "https://git-scm.com/download/mac". The page features the Git logo and the tagline "--distributed-even-if-your-workflow-isnt". A search bar is located in the top right. On the left sidebar, there are links for "About", "Documentation", "Downloads" (highlighted in red), "GUI Clients", "Logos", and "Community". Below these links is a box containing text about the "Pro Git book". The main content area is titled "Download for macOS" and provides instructions for installing Git on macOS. It lists several options: Homebrew, MacPorts, Xcode, Binary installer, Building from Source, and Installing git-gui. Each option includes a brief description and a terminal command to install Git.

**git** --distributed-even-if-your-workflow-isnt

Search entire site...

**About**  
**Documentation**  
**Downloads**  
GUI Clients  
Logos  
**Community**

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Download for macOS

There are several options for installing Git on macOS. Note that any non-source distributions are provided by third parties, and may not be up to date with the latest source release.

**Choose one of the following options for installing Git on macOS:**

### Homebrew

Install [homebrew](#) if you don't already have it, then:

```
$ brew install git
```

### MacPorts

Install [MacPorts](#) if you don't already have it, then:

```
$ sudo port install git
```

### Xcode

Apple ships a binary package of Git with [Xcode](#).

### Binary installer

Tim Harper provides an [installer](#) for Git. The latest version is [2.33.0](#), which was released about 2 years ago, on 2021-08-30.

### Building from Source

If you prefer to build from source, you can find tarballs [on kernel.org](#). The latest version is [2.43.0](#).

### Installing git-gui

If you would like to install [git-gui](#) and [gitk](#), git's commit GUI and interactive history browser, you can do so using [homebrew](#)

```
$ brew install git-gui
```



## RAP 15\_ GA7-220501096-AA1-EV04\_RM\_ Instalación y configuración de herramienta de versionamiento (Local / Web).



```
Last login: Tue Sep 19 13:18:18 on console
[MacBook-Pro-de-JHON:~ jhondonato$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
        [--exec-path=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset      Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index


examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status


grow, mark and tweak your common history
  branch     List, create, or delete branches
  checkout   Switch branches or restore working tree files
  commit     Record changes to the repository
  diff       Show changes between commits, commit and working tree, etc
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  tag        Create, list, delete or verify a tag object signed with GPG


collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
MacBook-Pro-de-JHON:~ jhondonato$ █
```





```
MacBook-Pro-de-JHON:PROYECTO jhondonato$ git init
Reinitialized existing Git repository in /Users/jhondonato/Desktop/PROYECTO/.git/
MacBook-Pro-de-JHON:PROYECTO jhondonato$ █
```

```
MacBook-Pro-de-JHON:PROYECTO jhondonato$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    demo/

nothing added to commit but untracked files present (use "git add" to track)
MacBook-Pro-de-JHON:PROYECTO jhondonato$ █
```



```
[MacBook-Pro-de-JHON:PROYECTO jhondonato$ git add .
```

```
[MacBook-Pro-de-JHON:PROYECTO jhondonato$ git commit -m "SE AGREGA EL PROYECTO"
```

```
[master (root-commit) a0f9aad] SE AGREGA EL PROYECTO
```

```
Committer: JHON DONATO <jhondonato@MacBook-Pro-de-JHON.local>
```

```
Your name and email address were configured automatically based  
on your username and hostname. Please check that they are accurate.  
You can suppress this message by setting them explicitly. Run the  
following command and follow the instructions in your editor to edit  
your configuration file:
```

```
git config --global --edit
```

```
After doing this, you may fix the identity used for this commit with:
```

```
git commit --amend --reset-author
```

```
6 files changed, 86 insertions(+)  
create mode 100755 demo/core/data.js  
create mode 100755 demo/index.html  
create mode 100755 demo/package.json  
create mode 100755 demo/ui/main.reel/main.css  
create mode 100755 demo/ui/main.reel/main.html  
create mode 100755 demo/ui/main.reel/main.js  
MacBook-Pro-de-JHON:PROYECTO jhondonato$
```



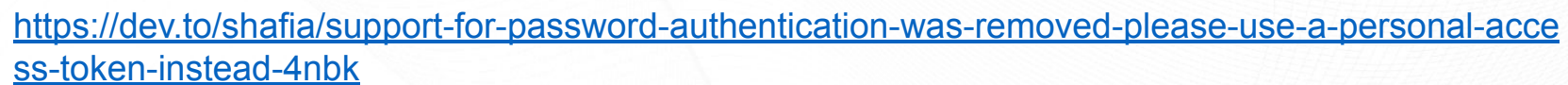


```
MacBook-Pro-de-JHON:PROYECTO jhondonato$ git config --global user.email jjdonato@misena.edu.co
MacBook-Pro-de-JHON:PROYECTO jhondonato$ git config --global user.name jhondonatoar
MacBook-Pro-de-JHON:PROYECTO jhondonato$
```

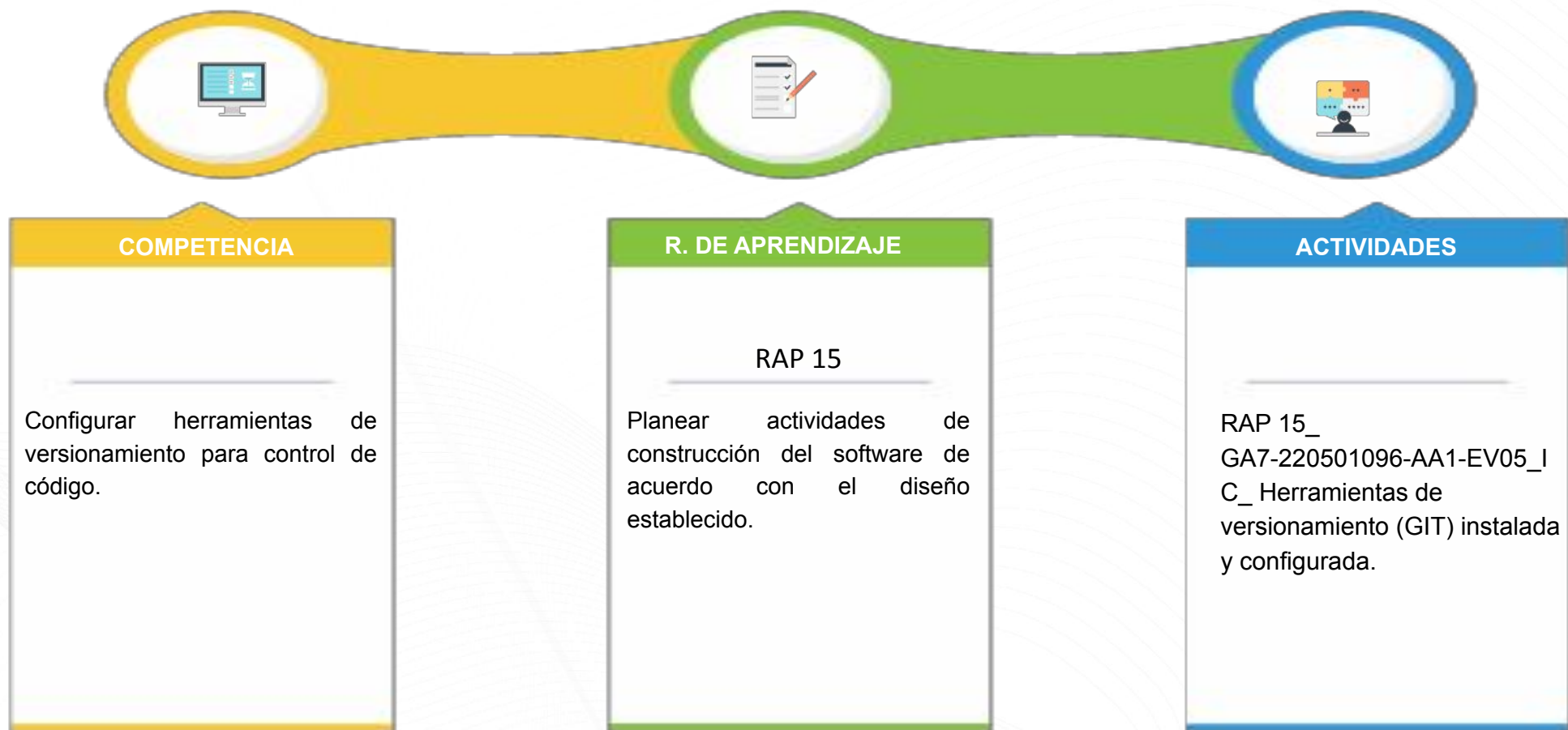
```
MacBook-Pro-de-JHON:PROYECTO jhondonato$ git commit -m "SE AGREGA EL PROYECTO"
On branch master
nothing to commit, working tree clean
```

```
MacBook-Pro-de-JHON:PROYECTO jhondonato$ git branch -m main
```

```
MacBook-Pro-de-JHON:PROYECTO jhondonato$ git push -u origin main
Username for 'https://github.com': jjdonato@misena.edu.co
Password for 'https://jjdonato@misena.edu.co@github.com':
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.48 KiB | 506.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To https://github.com/JHONDONATOAR/ADSOSENA.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
MacBook-Pro-de-JHON:PROYECTO jhondonato$
```









- Evidencia de producto: **(RAP 15)** GA7-220501096-AA1-EV05 herramientas de versionamiento (GIT) instalada y configurada.

Tomando como referencia el componente formativo "Integración continua", conectar el equipo local con el repositorio remoto por medio de Git usando los comandos vistos anteriormente.

Realice la ejecución de cada uno de los comandos básicos de git remoto y local teniendo en cuenta el ejercicio propuesto a continuación:

#### Paso 1

1. Crear un nuevo repositorio público en GitHub, gitLab o herramienta de su selección con el nombre Programa-git.
2. Añadirlo al repositorio local del Programa.
3. Mostrar todos los repositorios remotos configurados.

#### Paso 1

*# Crear el repositorio en GitHub y copiar su url con el protocolo https.*

```
> git remote add github url
```

```
> git remote -v
```

#### Paso 2

1. Agregar los cambios del repositorio local al repositorio remoto de GitHub o GitLab.
2. Acceder a GitHub o GitLab y comprobar que se han subido los cambios mostrando el historial de versiones.

```
>git push github master
```

#### Paso 3

- Agregar el archivo **kotlin.txt** que contenga el nombre del usuario y su correo electrónico.

- Agregar los cambios.
- Realizar un commit con el mensaje "Añadido datos."
- Cargar cambios al repositorio remoto.

```
> cat > kotlin.txt
```

```
git add .  
> git commit -m  
"Añadido datos."
```

```
> git push origin master.
```

**Elementos para tener en cuenta:**

- Se debe realizar el ejercicio en las herramientas de versionamiento instaladas y grabar un video ejecutando cada uno de los puntos solicitados en el ejercicio propuesto.
- Debe ir explicando cada uno de los requerimientos.
- El video debe tener una presentación con su nombre y los ejercicios a realizar.


**Lineamientos generales para la entrega de la evidencia:**

- **Productos para entregar:** video
- **Extensión:** Mp4.
- Para hacer el envío de la evidencia remítase al área de la actividad correspondiente y acceda al espacio: herramientas de versionamiento (GIT) instalada y configurada. GA7-220501096-AA1-EV05





**PORTADA DE UN DOCUMENTO EN UN SISTEMA DE GESTIÓN DE LA CALIDAD ISO 9001:2015**



MANUAL DE CALIDAD

Rev. 0  
Página 1 de 16

Identificación de  
revisión del  
documento

**MANUAL DE CALIDAD**

☐ COPIA CONTROLADA  
☐ COPIA NO CONTROLADA

Nº COPIA: \_\_\_\_\_

Control de las  
copias distribuidas

Histórico de revisiones  
y de cambios del  
documento

TABLA DE REVISIONES		
REVISIÓN	FECHA	DESCRIPCIÓN DE LA MODIFICACIÓN
0	31/01/2011	Revisión inicial

ELABORADO Y REVISADO: R. C.  
FECHA: 31/01/2011  
FIRMA: \_\_\_\_\_

APROBADO: DIRECCIÓN  
FECHA: 31/01/2011  
FIRMA: \_\_\_\_\_

Cuadros para  
evidenciar revisión y  
aprobación de  
documentos

[www.hederaconsultores.com](http://www.hederaconsultores.com)



# EVIDENCIA.

Documento Word o pdf.

- ◆ Portada
- ◆ Introducción
- ◆ Tabla de contenido
- ◆ Objetivos
- ◆ Desarrollo de las evidencias
- ◆ Conclusiones
- ◆ Referencias bibliográficas.



# ¿Preguntas e inquietudes?







# GRACIAS

Línea de atención al ciudadano: 01 8000 910270  
Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)