

Guía Repositorio

Darly Julieth Cuberos Rojas

SENA

Análisis y Desarrollo de Software

Ing. Fabian David Barreto Sánchez

1 de octubre de 2024

Actividad de aprendizaje GA7-220501096-AA1 configurar herramientas de versionamiento para control de código

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.

Características	Git Local	Git remoto
Función Principal	Sistema de control de versiones en el equipo local.	Almacenamiento y colaboración de repositorios en la nube.
Almacenamiento	Los repositorios se almacenan en el sistema local.	Los repositorios se almacenan en servidores remotos (en línea).
Conectividad a Internet	No requiere conexión a Internet para gestionar cambios.	Requiere conexión a Internet para sincronizar y compartir cambios.
Acceso a Repositorios	Solo accesible desde el equipo donde está instalado.	Accesible desde cualquier lugar con conexión a Internet.
Colaboración	Principalmente individual; se puede compartir el repositorio local manualmente.	Facilita la colaboración con múltiples usuarios en línea.
Interfaz	Se interactúa a través de la terminal o CLI (Git Bash).	Tiene una interfaz web gráfica para visualizar y gestionar repositorios.
Historial y Versionado	Control de versiones local sin compartir.	Control de versiones compartido y visible para colaboradores.
Copia de Seguridad	Riesgo de pérdida si no se respalda manualmente.	Copia de seguridad automática en la nube.
Gestión de Pull Requests	No se soporta.	Soporta solicitudes de cambios (Pull Requests) para revisión y aprobación de código.
Forks y Contribuciones	No es aplicable.	Permite a los usuarios hacer forks (copias) y contribuir a proyectos públicos.

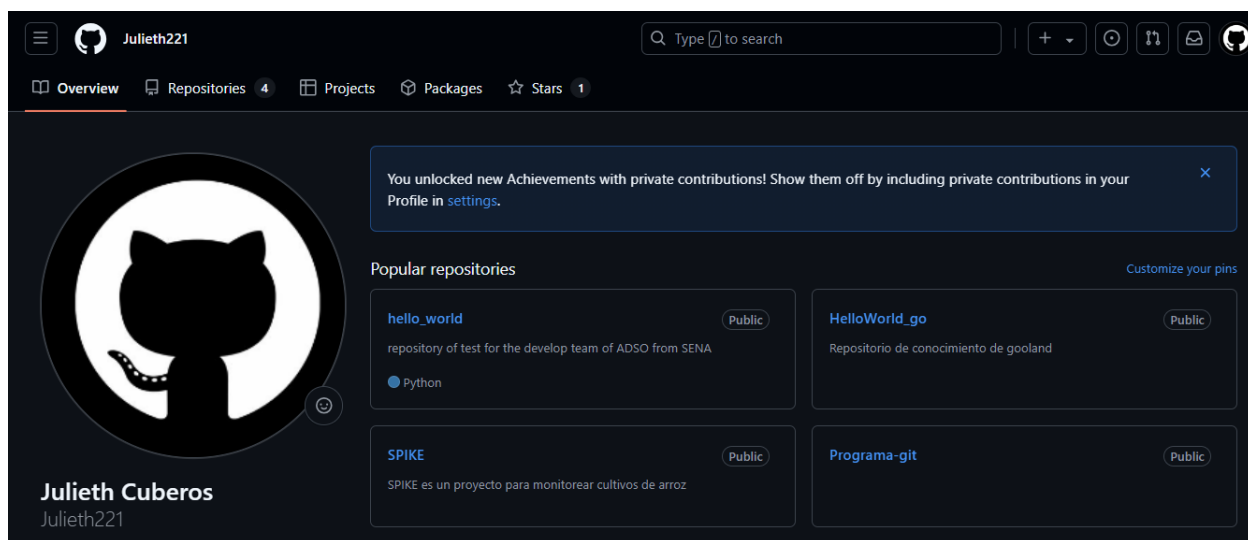
Privacidad	Todo el control es local y privado.	Puede ser público o privado, dependiendo de la configuración del repositorio.
Acceso a Gráficos y Métricas	Solo disponible en comandos manuales.	Proporciona gráficos de actividad, contribuciones, y análisis de código.

Comando	Descripción	Git
git init	Inicializa un nuevo repositorio de Git en el directorio actual.	Local
git commit	Guarda los cambios en el historial del repositorio local, creando un nuevo commit.	Local
git log	Muestra el historial de commits del repositorio, incluyendo los detalles de cada uno.	Local
git status	Muestra el estado del repositorio: archivos modificados, añadidos, y no rastreados.	Local
git checkout	se utiliza para cambiar entre ramas	local
git branch	Crea, lista o elimina ramas (branches) en el repositorio.	Local
git clone	Clona un repositorio remoto a una copia local.	Remoto
git pull	Descarga y fusiona los cambios del repositorio remoto con la rama local.	Remoto
git push	Sube los commits del repositorio local al repositorio remoto.	Remoto

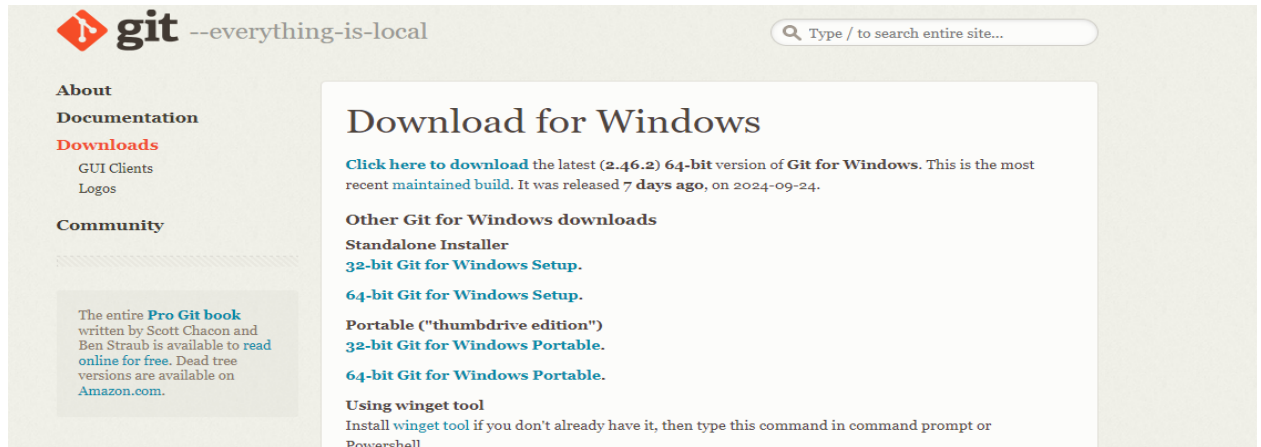
<code>git config --global user.name "Tu Nombre"</code>	Configura el nombre del usuario globalmente (para todos los repositorios).	Local
<code>git config --global user.email "tu_email@example.com"</code>	Configura el correo electrónico del usuario globalmente (para todos los repositorios).	Local

Evidencia de desempeño: GA7-220501096-AA1-EV02 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.



1. Instalar git



The screenshot shows the Git website's 'Download for Windows' page. The header features the Git logo and the tagline '--everything-is-local'. A search bar is located in the top right corner. The left sidebar contains links for 'About', 'Documentation', 'Downloads' (highlighted), 'GUI Clients', 'Logos', and 'Community'. The main content area is titled 'Download for Windows' and provides instructions on how to download the latest version of Git for Windows. It includes links for the 'Standalone Installer', 'Portable ("thumbdrive edition")', and 'Using winget tool'. A sidebar on the left mentions the 'Pro Git book'.

git --everything-is-local

Type / to 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 Windows

[Click here to download](#) the latest (2.46.2) 64-bit version of **Git for Windows**. This is the most recent **maintained build**. It was released 7 days ago, on 2024-09-24.

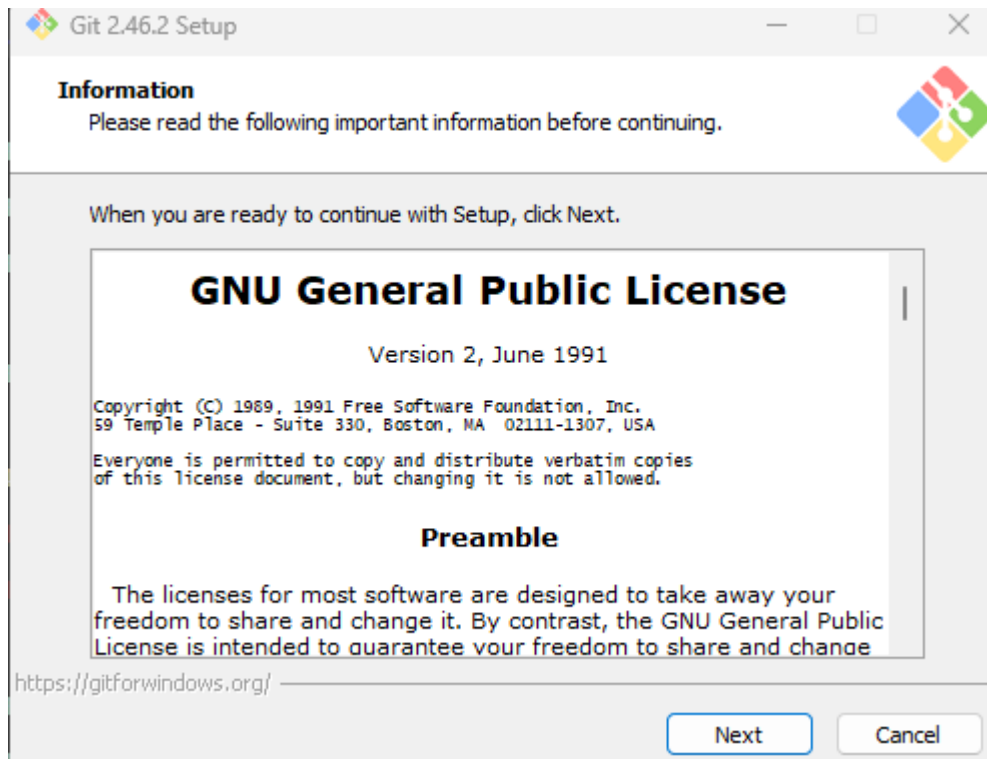
Other Git for Windows downloads

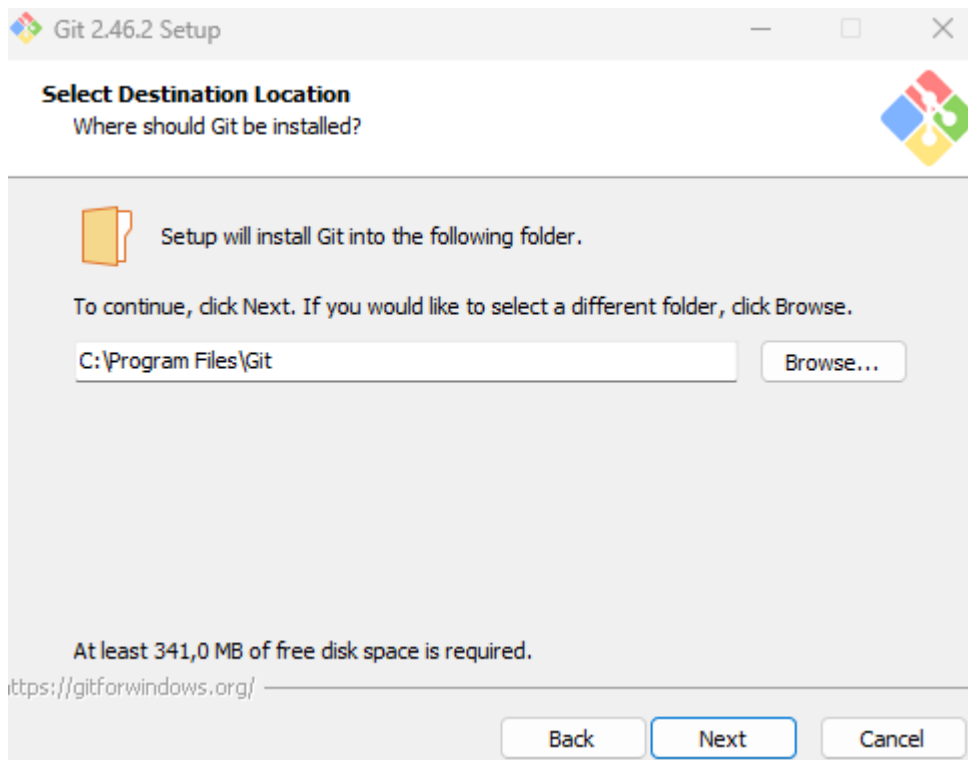
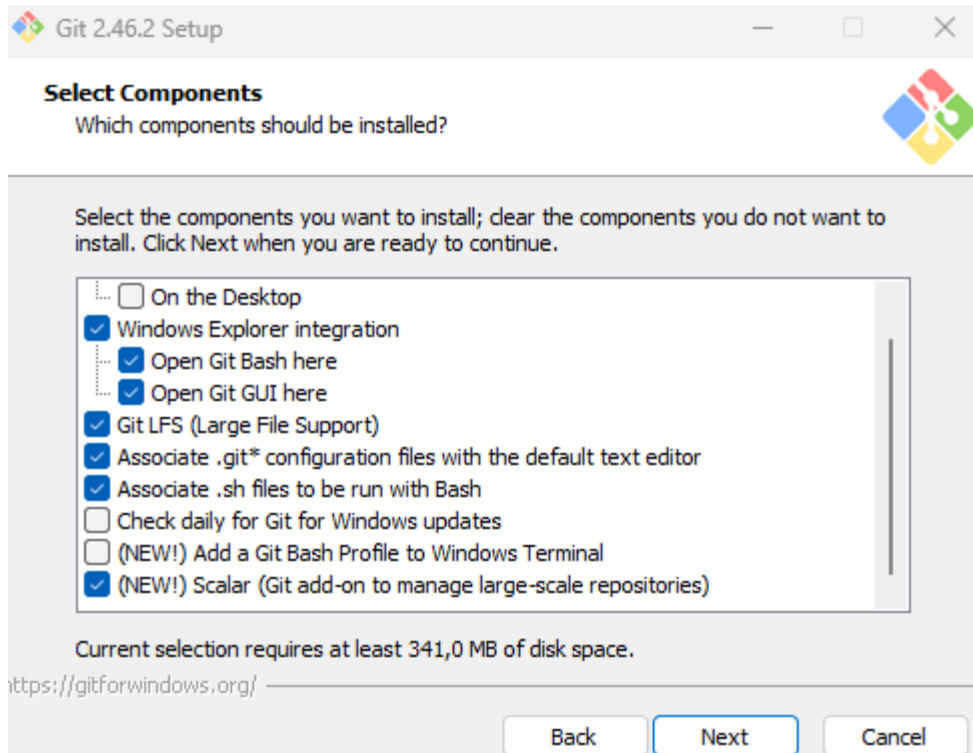
Standalone Installer
[32-bit Git for Windows Setup](#).
[64-bit Git for Windows Setup](#).

Portable ("thumbdrive edition")
[32-bit Git for Windows Portable](#).
[64-bit Git for Windows Portable](#).

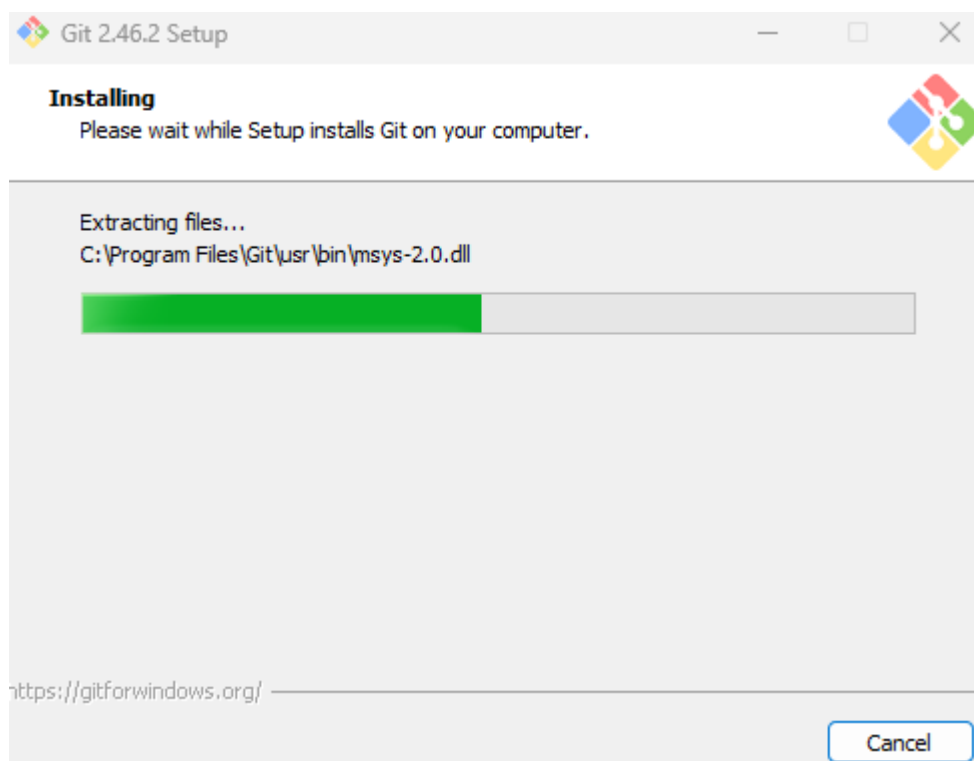
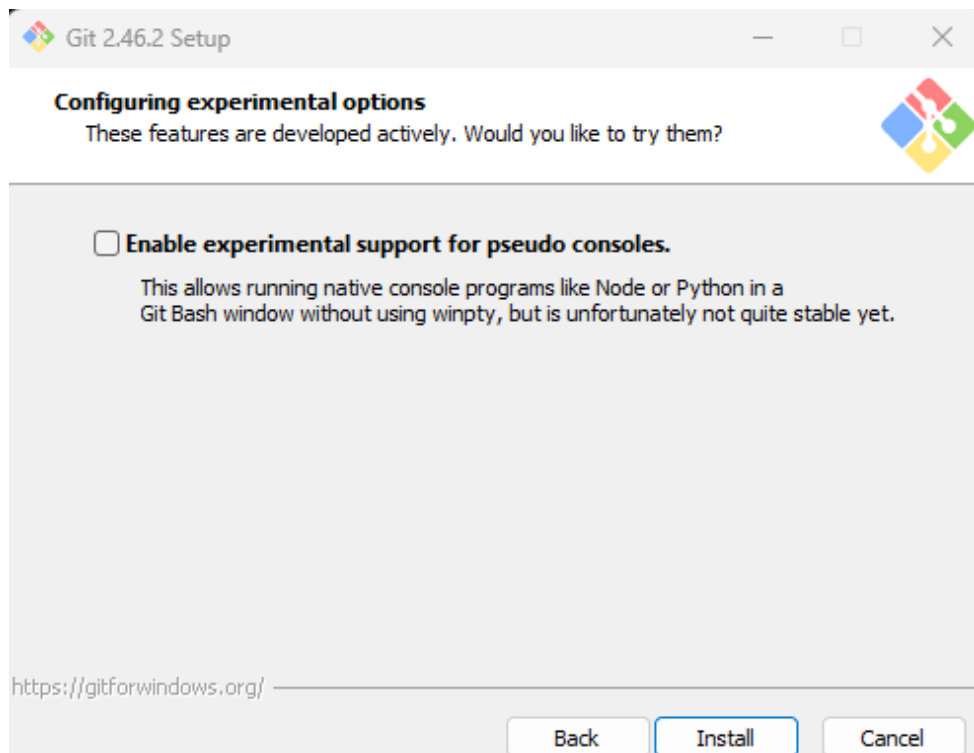
Using winget tool
 Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

2. Abrir el instalador





3. instalar



4. Verificar que git se haya instalado correctamente y posteriormente configurarlo con el usuario y correo del remoto

```

CommandType      Name                      Version      Source
-----
Application      git.exe                2.44.0.1     C:\Program Files\Git\cmd\git.exe

PS C:\Users\darly> git config --global user.name
Julieth221
PS C:\Users\darly> git config --global user.email
darlycuberos17@gmail.com
PS C:\Users\darly> |

```

Evidencia de producto: GA7-220501096-AA1-EV03 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

Se crea un repositorio remoto y se clona en el local

```

PS C:\Users\darly\OneDrive - SENA\ADSO\INFO> cd .\Programa-git\
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git> git clone https://github.com/Julieth221/Programa-git.git
Cloning into 'Programa-git'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

```

2. git remote -v El remoto llamado origin está asociado con la URL tanto para obtener actualizaciones (fetch) como para enviar tus cambios (push).

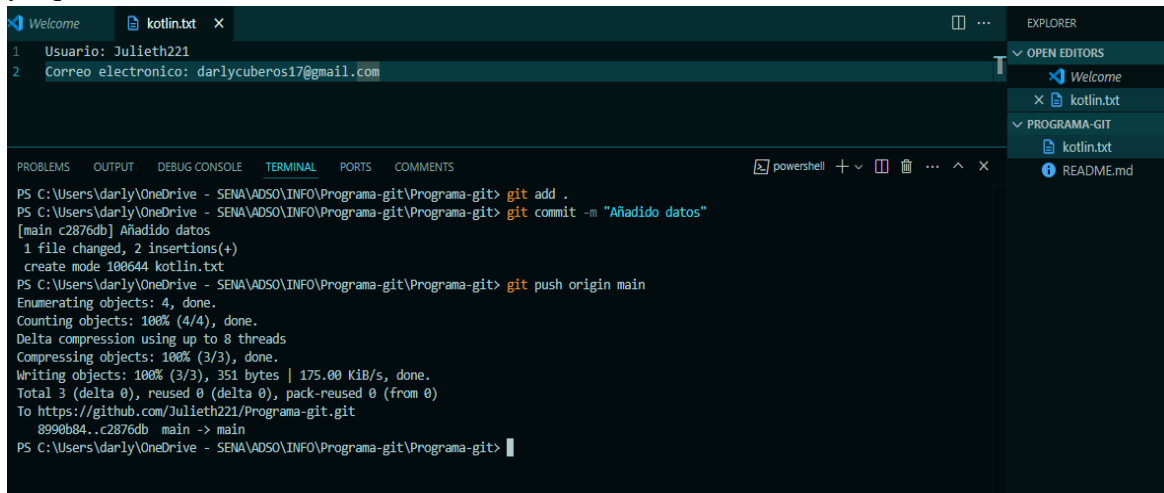
```

PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> git remote -v
origin https://github.com/Julieth221/Programa-git.git (fetch)
origin https://github.com/Julieth221/Programa-git.git (push)

```

3. Se crea el archivo kotlin.text y se añade en su contenido el usuario y correo configurados en el git remoto y local

4. se agregan los cambios con git add .
5. se guardan los cambios con git commit y se deja un breve mensaje de lo que se realizo
6. para subir los cambios a nuestro repo remoto usamos el comando git push origin main, ya que nos encontramos en la rama main



```

Welcome | kotlin.txt x
1 Usuario: Julieth221
2 Correo electronico: darlycuberos17@gmail.com

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> git add .
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> git commit -m "Añadido datos"
[main c2876db] Añadido datos
1 file changed, 2 insertions(+)
create mode 100644 kotlin.txt
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 351 bytes | 175.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Julieth221/Programa-git.git
8990b84..c2876db main -> main
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git>

```

Actividad de aprendizaje GA7-220501096-AA2 - Aplicar estándares de codificación

Esta actividad se centra en la descripción del plan de trabajo que se debe definir según las características del software a desarrollar, detallando estándares de codificación y lenguaje a utilizar.

Ejercicio: Generar una Factura Simple en PDF

Repo: <https://github.com/Julieth221/Programa-git.git>

1. Se crea un archivo en el repo mencionado anteriormente para el programa de factura
2. Se instalan las dependencias necesarias para usar la librería **gopdf**, y se añaden los archivos go.mod y go.sum
3. Se crea el programa para facturar productos
4. Se agregan los cambios con git add .
5. Se guardan los cambios con git commit y se deja un breve mensaje que describe lo realizado
6. Finalmente, se suben los cambios al repo remoto a la rama main con git push origin main

```
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> git add .
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> git commit -m "Se creo el programa para facturar pr
oductos"
[main 204c6f2] Se creo el programa para facturar productos
5 files changed, 180 insertions(+)
create mode 100644 Ver.pdf
create mode 100644 factura.pdf
create mode 100644 go.mod
create mode 100644 go.sum
create mode 100644 programa.go
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 4.93 KiB | 4.93 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Julieith221/Programa-git.git
c2876db..204c6f2 main -> main
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> █
```

Evidencia de producto: GA7-220501096-AA2-EV02 módulos de software codificados y probados

Pruebas

Escribimos por consola un nombre y posteriormente vamos agregando los productos, se escribe el nombre del producto, luego la cantidad y finalmente el precio unitario, vuelve a mostrarnos un mensaje para seguir agregando productos, si se quiere agregar más productos simplemente ingresamos el nombre del producto y seguirá el mismo flujo para pedir la cantidad y el precio unitario, si quiere finalizar la compra simplemente escribimos f, la tecla f solo se podrá escribir cuando aparezca el mensaje “Ingrese el nombre del producto (o ‘f’ para terminar):” de otro modo, no podrá terminar.

```
PS C:\Users\darly\OneDrive - SENA\ADSO\INFO\Programa-git\Programa-git> go run .\programa.go
Ingrese el nombre del cliente:
Sofia Ramirez
Ingrese el nombre del producto (o 'f' para terminar):
mani
Ingrese la cantidad:
2
Ingrese el precio unitario:
1200
Ingrese el nombre del producto (o 'f' para terminar):
panela
Ingrese la cantidad:
3
Ingrese el precio unitario:
2500
Ingrese el nombre del producto (o 'f' para terminar):
arina pan
Ingrese la cantidad:
1
Ingrese el precio unitario:
4500
Ingrese el nombre del producto (o 'f' para terminar):
azucar
Ingrese la cantidad:
2
Ingrese el precio unitario:
3000
Ingrese el nombre del producto (o 'f' para terminar):
Ingrese el nombre del producto (o 'f' para terminar):
sal
Ingrese la cantidad:
1
Ingrese el precio unitario:
1000
Ingrese el nombre del producto (o 'f' para terminar):
arandanos
Ingrese la cantidad:
1
Ingrese el precio unitario:
5500
Ingrese el nombre del producto (o 'f' para terminar):
manzana
Ingrese la cantidad:
4
Ingrese el precio unitario:
2200
```

Cuando se escribe f en el mensaje “Ingrese el nombre del producto (o ‘f’ para terminar):”

termina la compra y muestra un resumen de la factura con el nombre del cliente, la fecha actual y los productos registrados, junto con la cantidad, el precio unitario y el total, despues de los productos aparece el valor total de la compra y finalmente nos muestra un mensaje si queremos imprimir la factura, escribimos si para que nos muestre la factura en pdf

```
Cliente: Sofia Ramirez
Fecha: 03/10/2024
Productos:
- mani: 2 x 1200.00 = 2400.00
- panela: 3 x 2500.00 = 7500.00
- arina pan: 1 x 4500.00 = 4500.00
- azucar: 2 x 3000.00 = 6000.00
- sal: 1 x 1000.00 = 1000.00
- arandanos: 1 x 5500.00 = 5500.00
- manzana: 4 x 2200.00 = 8800.00
- carne: 1 x 20000.00 = 20000.00
Valor Total: 55700.00
¿Quiere imprimir la factura? (si/no):
si
Ver.pdf
```

Al darle clic en ver.pdf, podemos observar el resumen de la factura en pdf

Resumen de la Factura

Cliente: Sofia Ramirez

Fecha: 03/10/2024

Producto	Cantidad	Precio Unitario	Total
mani	2	1200.00	2400.00
panela	3	2500.00	7500.00
arina pan	1	4500.00	4500.00
azucar	2	3000.00	6000.00
sal	1	1000.00	1000.00
arandanos	1	5500.00	5500.00
manzana	4	2200.00	8800.00
carne	1	20000.00	20000.00
Valor Total:			55700.00

si en vez darle si en generar factura en pdf, escribimos no, nos muestra un mensaje” No se ha generado la factura PDF”

```

3000
Ingrese el nombre del producto (o 'f' para terminar):
f

Resumen de la factura:
Cliente: Sofia
Fecha: 03/10/2024
Productos:
- mani: 2 x 3000.00 = 6000.00
- manzana: 2 x 2500.00 = 5000.00
- azucar: 1 x 3000.00 = 3000.00
Valor Total: 14000.00
¿Quiere imprimir la factura? (si/no):
no
No se ha generado la factura en PDF.

```