



DEPARTAMENTO NACIONAL DE PLANEACIÓN

GUÍA PARA LA GESTIÓN DE CONTROL DE CÓDIGO FUENTE

**Oficina de Tecnologías y sistemas de Información
Grupo de Gestión de Sistemas de Información**

**Departamento Nacional de Planeación
Bogotá, 2020**



Control de Versiones

Versión	Fecha	Descripción	Autores
0	15-09-2020	Elaboración del documento.	César Augusto Gómez Perilla

Derechos de Autor: La elaboración de este documento y sus diferentes componentes estuvo a cargo del Grupo de Gestión de Sistemas de Información de la Oficina de Tecnologías y Sistemas de Información del Departamento Nacional de Planeación, DNP, razón por la cual los Derechos de Autor y en lo particular los derechos patrimoniales de este documento y su contenido pertenece exclusivamente al DNP. Por lo tanto, su uso y reproducción por terceros, está sujeto a la autorización expresa de la Oficina de Tecnologías y Sistemas de Información, OTSI del DNP en cumplimiento de la Ley 23 de 1982 y demás que la modifican o adicionan. Siendo así, este documento está protegido por Derechos de Autor y no pueden ser copiados, ni reproducidos, ni distribuidos por personas o Entidades diferentes al DNP.



TABLA DE CONTENIDO

1.	OBJETIVO	4
2.	ALCANCE	4
3.	TÉRMINOS Y DEFINICIONES.....	4
4.	GIT	5
4.1	Configuración del GIT.....	5
4.2	Clonación del repositorio	5
4.3	Ubicarse en el directorio del repositorio.....	6
4.4	Creación de la rama de desarrollo “develop”	6
4.5	Historias de usuario	7
4.6	Ramas puntuales	8
4.7	Adicionar, modificar y comentar sobre el desarrollo	8
4.8	Sincronización rama <puntual> con la rama “develop”	9
4.9	Subir los cambios al repositorio	9
5.	TFS VC	11
5.2	Configuración la ruta local	11
5.3	Incluir el proyecto al repositorio asignado	11
5.4	Sincronizar código fuente al repositorio.....	12
5.5	Visualización del repositorio en Azure DevOps Server.....	13
5.6	Creación de la rama desarrollo “develop”	13
5.7	Creación de ramas puntuales	15
5.8	Sincronización rama <puntual> con la rama “develop”	15



1. OBJETIVO

En el presente documento tiene como objetivo dar a conocer el procedimiento en la gestión de control de código fuente en la elaboración y ejecución de un proyecto dentro del Grupo de Gestión de Sistemas de Información del Departamento Nacional de Planeación – DNP.

2. ALCANCE

Brindar una guía para ser aplicada desde los roles Líder Técnico, Líder de Desarrollo o quién designe el Jefe de la dependencia o en su defecto la firma externa contratada si aplica.

3. TÉRMINOS Y DEFINICIONES

- **Control de versiones.** Corresponde a la gestión de los diversos cambios que se realizan sobre elementos de algún producto.
- **GIT.** Es un software de control de versiones, pensado en la eficiencia y la confiabilidad del mantenimiento de versiones.
- **TFS VC.** Ayuda a la gestión de equipos de manera colaborativa, también ofrece control de versiones sobre los desarrollos.
- **Ramas.** Es un control de versiones que registra los cambios realizados a lo largo del tiempo, de modo que puede recuperar versiones específicas más adelante. Por defecto se tiene una rama llamada “master” que es la versión estable del producto.
- **Ramas puntuales.** Tiene la misma característica que la rama pero su ciclo de vida llega hasta cuando se fusiona con otra rama, eliminándola del control de versiones.
- **Épica.** Una Épica puede ser vista como una historia de usuario de alta complejidad (petición de negocio de alto nivel y complejidad). Para elaborarla se puede requerir un esfuerzo muy grande y no puede ser cubierta en un Sprint. Una épica debido a su gran tamaño es difícil de estimar y de abordar, así que es necesario descomponerla en varias características e historias. Implementar una épica suele llevar dos o más sprints.
- **Historia de Usuario.** Una historia de usuario representa una necesidad de negocio que puede ser implementada en un sprint y aporta valor al producto. Al final del sprint la historia añade una nueva funcionalidad o característica al producto y puede ser candidata para pasar a producción.



4. PRE-REQUISITOS

En Azure DevOps Server maneja permisos como “Readers”, “Contributors” y “Administrators” en los repositorios, en este caso, se debe tener en cuenta que el Líder de Desarrollo o quién designe el Jefe de la dependencia o en su defecto la firma externa contratada si aplica, debe manejar el permiso de “Contributors”.

Para cualquier sistema de control de versiones moderno tiene algún mecanismo para soportar el uso de ramas. Cuando se habla de ramificaciones, significa que se toma la rama principal de desarrollo (master) y a partir de ahí continúa trabajando sin seguir la rama principal de desarrollo. Para este caso se manejará la rama “master”(producción), “qa”(pruebas), “develop”(desarrollo), “tb-n”(historias de usuario), según la figura 1.

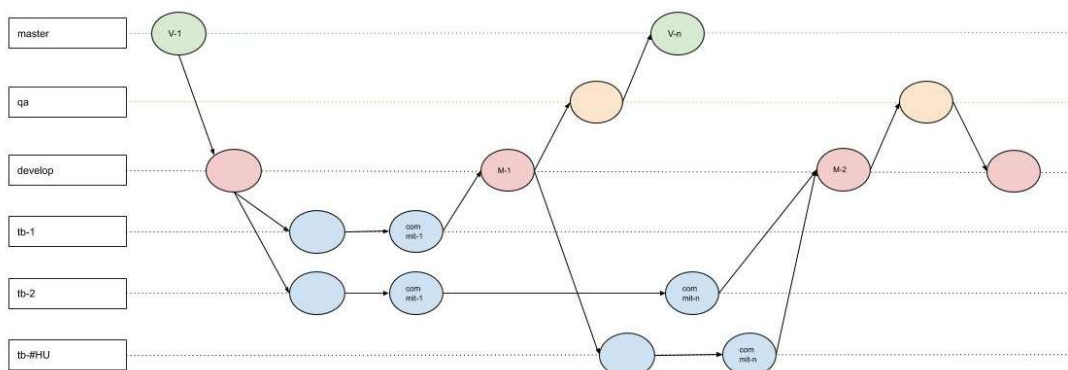


Figura 1. Estructura de ramas para el proyecto

Teniendo en cuenta lo anterior se explicará el versionamiento de código fuente desde GIT y TFSVC.

5. GIT

4.1 Configuración del GIT

Se debe inicializar y configurar el nombre y correo de quien realizará el control de versiones. Ver figura 2.

```
Cesars-MacBook-Pro:repositorios cesargomezperilla$ git init
Reinitialized existing Git repository in /Applications/XAMPP/xamppfiles/htdocs/repositorios/.git/
Cesars-MacBook-Pro:repositorios cesargomezperilla$ git config user.name "Cesar Gomez Perilla"
Cesars-MacBook-Pro:repositorios cesargomezperilla$ git config user.email "cesgomez@dnpp.gov.co"
Cesars-MacBook-Pro:repositorios cesargomezperilla$
```

Figura 2. Configuración del GIT

4.2 Clonación del repositorio

Para la clonación del repositorio se requiere tener las credenciales y permisos dados por el Líder Técnico del Grupo de Gestión de Sistemas de Información del Departamento Nacional de Planeación.

Una vez tenga los permisos, ingresa a la terminal y clona el repositorio como se ve en la figura 3.



```
repositorios
Username
Git: https://tfs.dnp.gov.co (Press 'Enter' to confirm or 'Escape' to cancel)

PROBLEMS 1K+ OUTPUT DEBUG CONSOLE TERMINAL 1: git

Cesars-MacBook-Pro:repositorios cesargomezperilla$ git clone https://tfs.dnp.gov.co/DNP-Interna/GGSI.PruebaPiloto_PHP/_git/GGSI.PruebaPiloto_PHP
Cloning into 'GGSI.PruebaPiloto_PHP'...
Cesars-MacBook-Pro:repositorios cesargomezperilla$ git clone https://tfs.dnp.gov.co/DNP-Interna/GGSI.PruebaPiloto_PHP/_git/GGSI.PruebaPiloto_PHP
Cloning into 'GGSI.PruebaPiloto_PHP'...
remote: Azure Repos
remote: Found 47 objects to send. (112 ms)
Unpacking objects: 100% (47/47), done.
Cesars-MacBook-Pro:repositorios cesargomezperilla$
```

Figura 3. Clonación del proyecto

4.3 Ubicarse en el directorio del repositorio

Una vez se clone el repositorio a su terminal local, de debe ubicar en el directorio y con el comando de “log” puede ver el detalle de quienes han hecho “commit” al proyecto. Ver figura 4.

```
index.php — GGSI.PruebaPiloto_PHP

index.php x
index.php ...
1 Cesar Gomez, 13 days ago | 1 author (Cesar Gomez)
2 [DOCTYPE html] Cesar Gomez, 13 days ago • Se inicializa el proyecto en la rama desarrollo
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Prueba de concepto</title>
8 </head>
9 <body>
10   <p>Es una prueba</p>
11 </body>
12 </html>

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208850.
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git log --pretty=oneline
a4e059df8f5846174a11b4bb3b1f94b77038459 (HEAD -> master, origin/master, origin/HEAD) Se inicializa el proyecto en la rama desarrollo
6a5213cd66d7221427d123fabda74f2a284df49e Added README.md file
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$
```

Figura 4. Trazabilidad de los comentarios del proyecto

4.4 Creación de la rama de desarrollo “develop”

Para iniciar la codificación se debe crear la rama con el nombre “develop”, el cual se hará todas las integraciones de las historias de usuario del proyecto, para los desarrolladores puedan trabajar sin ningún problema y desplegar los avances. Ver figura 5.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git log --pretty=oneline
a4e059dff8f5046174a11b4bb3b1fd4bf7038459 (HEAD -> master, origin/master, origin/HEAD) Se inicializa el proyecto en la rama desarrollo
6a5213cd66d7221427d123fabda74f2a284df49e Added README.md file
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch
* master
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch develop
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch
develop
* master
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git checkout develop
Switched to branch 'develop'
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch
* develop
  master
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$
```

Figura 5. Creación de la rama “develop”

Luego de crear la rama, se debe mover sobre ella con el comando “checkout <rama>”

4.5 Historias de usuario

Se debe revisar las historias de usuario asignadas al desarrollador, para ello en DevOps en la sección “Boards/Plans” encontrará las historias asignadas. Ver figura 6 y 7.

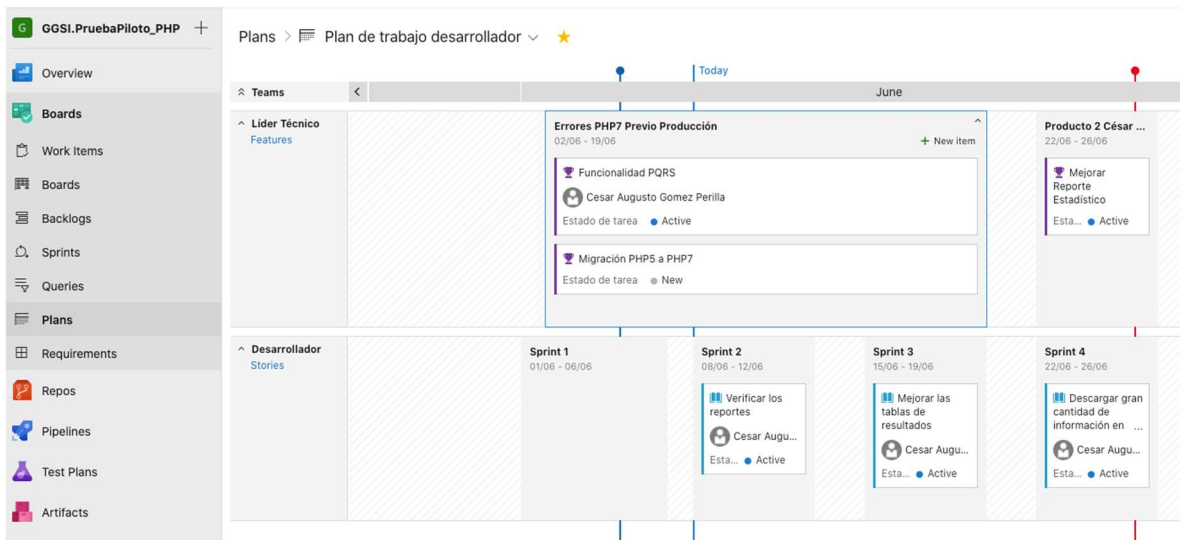


Figura 6. Historias de usuario en el planeador

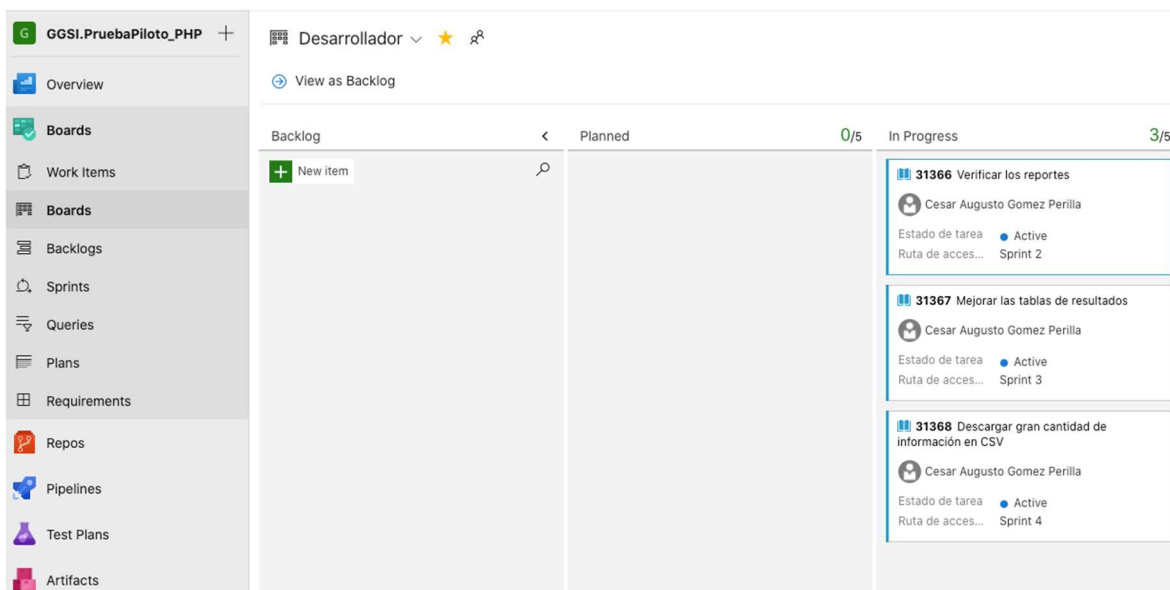


Figura 7. Historias de usuarios asignados

4.6 Ramas puntuales

Una vez este sobre la rama “develop”, según la historia de usuario asociado en DevOps, se debe crear una rama específica, Ejemplo: Para el caso anterior se tomará la historia de usuario #31366 Verificar los reportes. Ver figura 8.

```
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch tb-31366
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch
* develop
  master
  tb-31366
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git checkout tb-31366
Switched to branch 'tb-31366'
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch
  develop
  master
* tb-31366
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$
```

Figura 8. Ramas puntuales de desarrollo.

4.7 Adicionar, modificar y comentar sobre el desarrollo

El desarrollador una vez termine su labor debe incluir su trabajo en la rama <puntual>, ya sea con nuevos archivos o actualizando los existentes. Se recomienda que en los comentarios hacer referencia a que requerimiento funcionales puntuales. Ver figura 9.



```
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git branch
develop
* tb-31366
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git status
On branch tb-31366
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.php

no changes added to commit (use "git add" and/or "git commit -a")
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git add .
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git commit -m "Se elaboró el desarrollo de la historia de usuario verificar reportes"
git: 'commit' is not a git command. See 'git --help'.

The most similar command is
    commit
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git commit -m "Se elaboró el desarrollo de la historia de usuario verificar reportes"
[tb-31366 b5763c1] Se elaboró el desarrollo de la historia de usuario verificar reportes
1 file changed, 1 insertion(+), 1 deletion(-)
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git status
On branch tb-31366
nothing to commit, working tree clean
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$
```

Figura 9. Trazabilidad de los comentarios

4.8 Sincronización rama <puntual> con la rama “develop”

Una vez terminado de desarrollar sobre la rama <puntual>, debe realizar la integración con la función “merge” con la rama “develop”. Ver figura 10.

```
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git checkout develop
Switched to branch 'develop'
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git merge tb-31366
Updating a4e059d..b5763c1
Fast-forward
 index.php | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git commit -m "Se realiza el merge con la rama tb-31366"
On branch develop
nothing to commit, working tree clean
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$
```

Figura 10. Integración entre ramas.

4.9 Subir los cambios al repositorio

Luego de tener el desarrollo debe sincronizar el trabajo que tiene en la terminal local a repositorio en DevOps. Ver figura 11 y 12.



```
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git push
fatal: The current branch develop has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin develop

Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$ git push --set-upstream origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 417 bytes | 417.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Analyzing objects... (3/3) (8 ms)
remote: Storing packfile... done (47 ms)
remote: Storing index... done (48 ms)
To https://tfs.dnp.gov.co/DNP-Interna/GGSI.PruebaPiloto_PHP/_git/GGSI.PruebaPiloto_PHP
 * [new branch]      develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
Cesars-MacBook-Pro:GGSI.PruebaPiloto_PHP cesargomezperilla$
```

Figura 11. Sincronización al repositorio DevOps

Branch	Commit	Author	Authored Date	Behind Ahead	Status	Pull Request
develop	b5763c14	Cesar Gomez	16 minutes ago	0 1		
master	a4e059df	Cesar Gomez	26/05/2020			

Figura 12. Visualización de la sincronización en DevOps



6. TFS VC

5.2 Configuración la ruta local

Configurar la ruta del proyecto que se encuentra Azure DevOps Server para trabajar en ambiente local. Debe seleccionar el proyecto y relacionar la carpeta local como se ve en la figura 13.

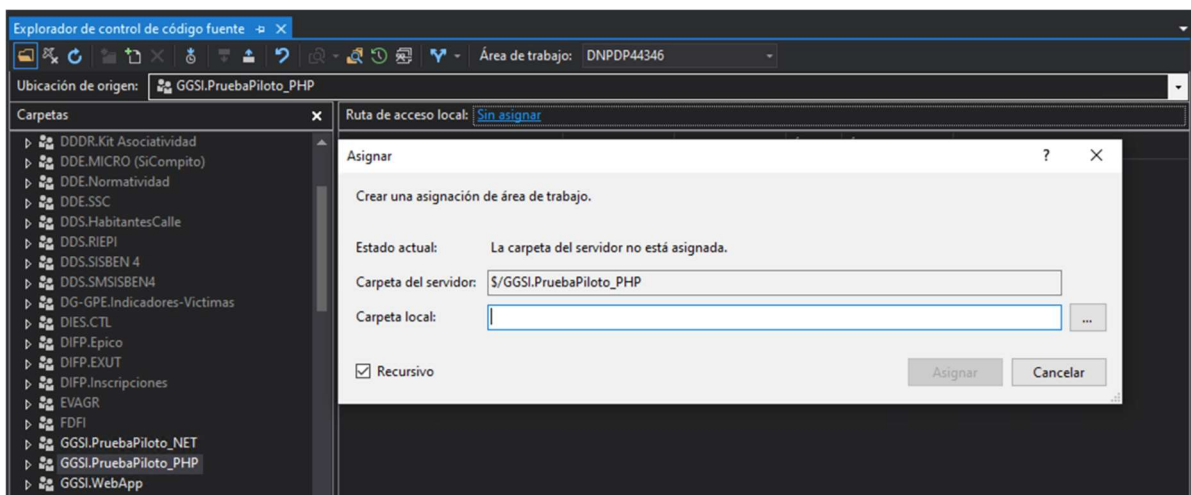


Figura 13. Asignar ruta local

5.3 Incluir el proyecto al repositorio asignado

Para incluir el proyecto debe dar clic derecho, en la opción “Agregar elementos a carpeta” y seleccionar el código que quiere agregar al control de código fuente. Ver en la figura 14 y 15.

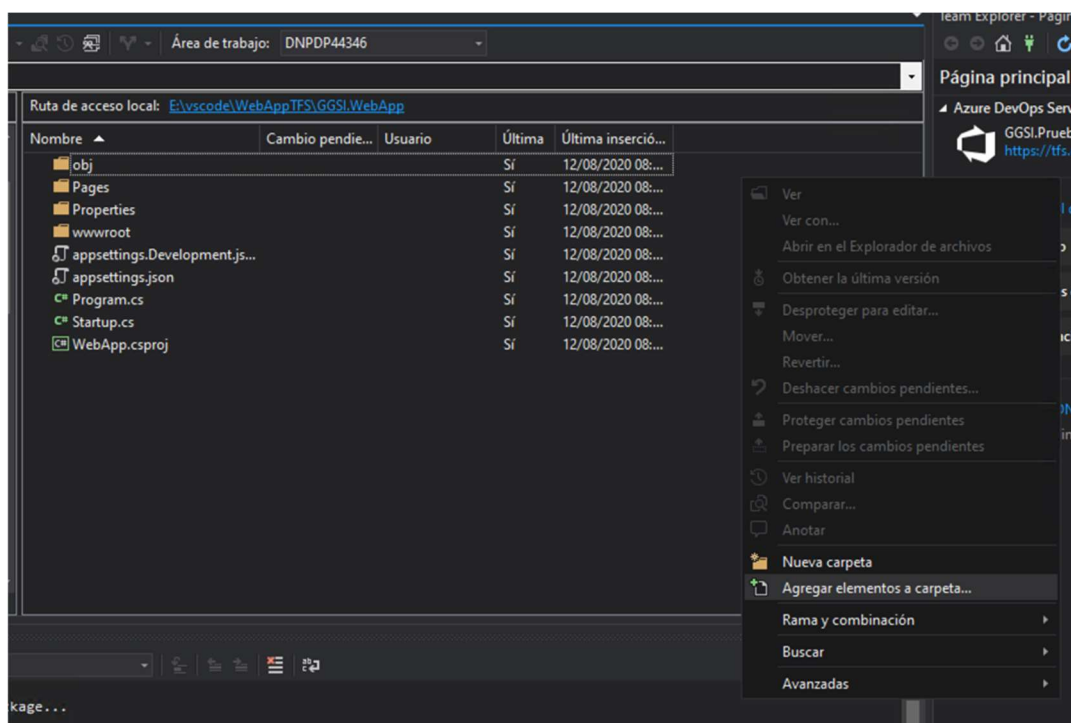


Figura 14. Agregar elementos a carpeta

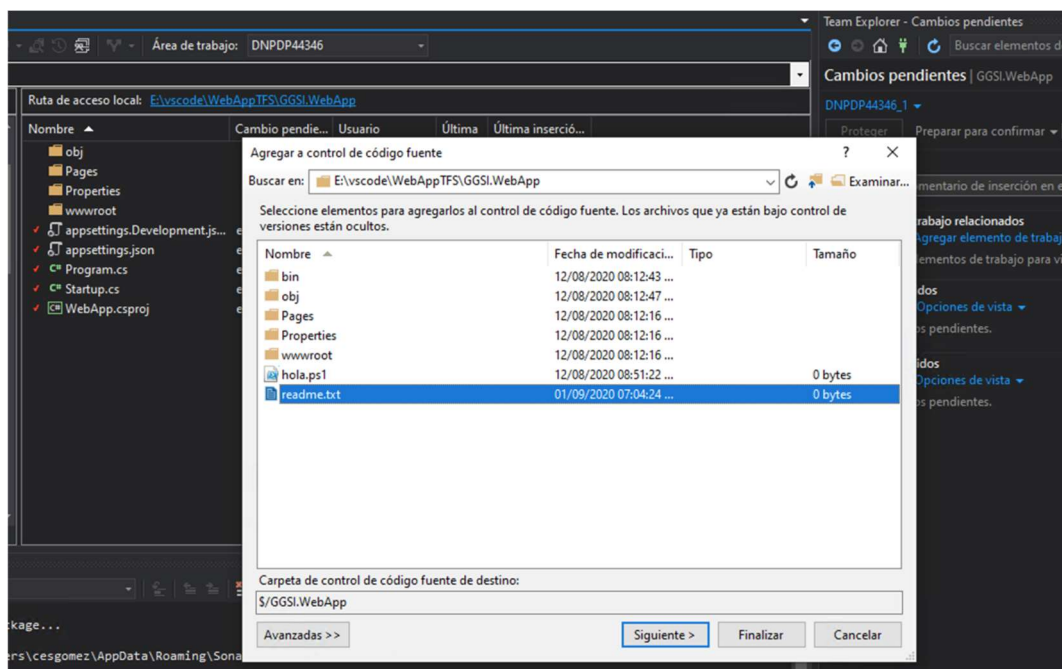


Figura 15. Agregar a control de código fuente

5.4 Sincronizar código fuente al repositorio

El desarrollador una vez termine su labor debe incluir su trabajo en el repositorio, ya sea con nuevos archivos o actualizando los existentes. Se recomienda que en los comentarios hacer referencia a que requerimiento



funcional puntuales. Ver figura 16.

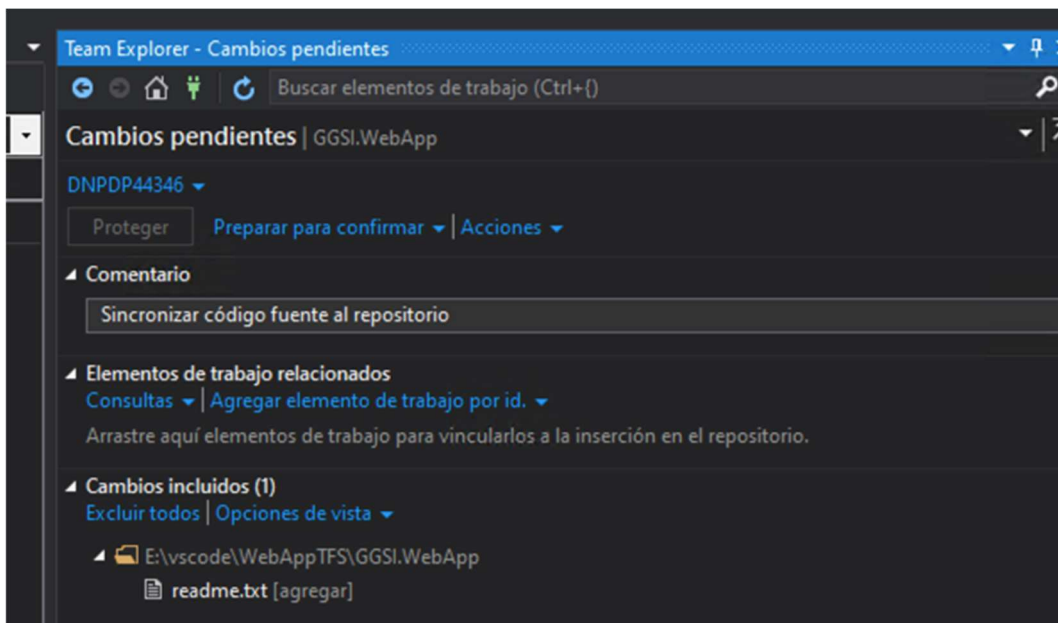


Figura 16. Sincronizar código fuente

5.5 Visualización del repositorio en Azure DevOps Server

Para ver la trazabilidad del código fuente se puede dirigir desde Azure DevOps Server en la opción “Repos”, luego en “Files”, encontrando el repositorio TFSVC con su respectivo historial. Ver figura 17.

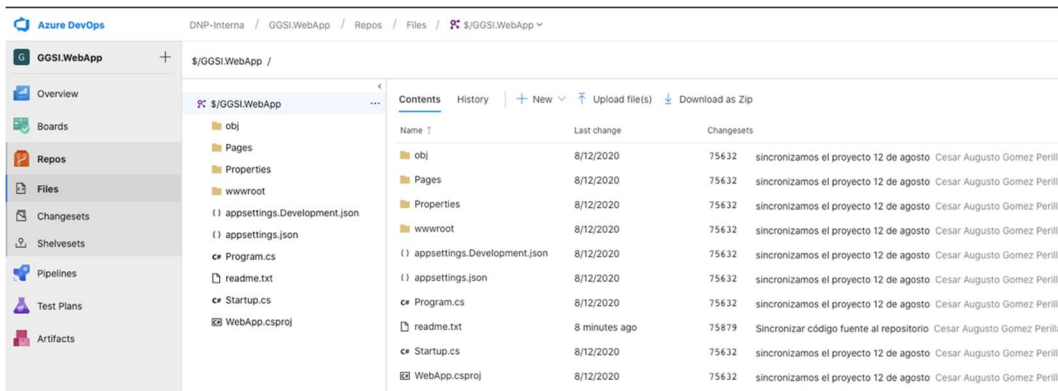


Figura 17. Trazabilidad de versión de código en DevOps

5.6 Creación de la rama desarrollo “develop”

Para iniciar la codificación se debe crear la rama con el nombre “develop”, el cual se hará todas las integraciones de las historias de usuario del proyecto, para los desarrolladores puedan trabajar sin ningún problema y desplegar los avances. Ver figura 18.

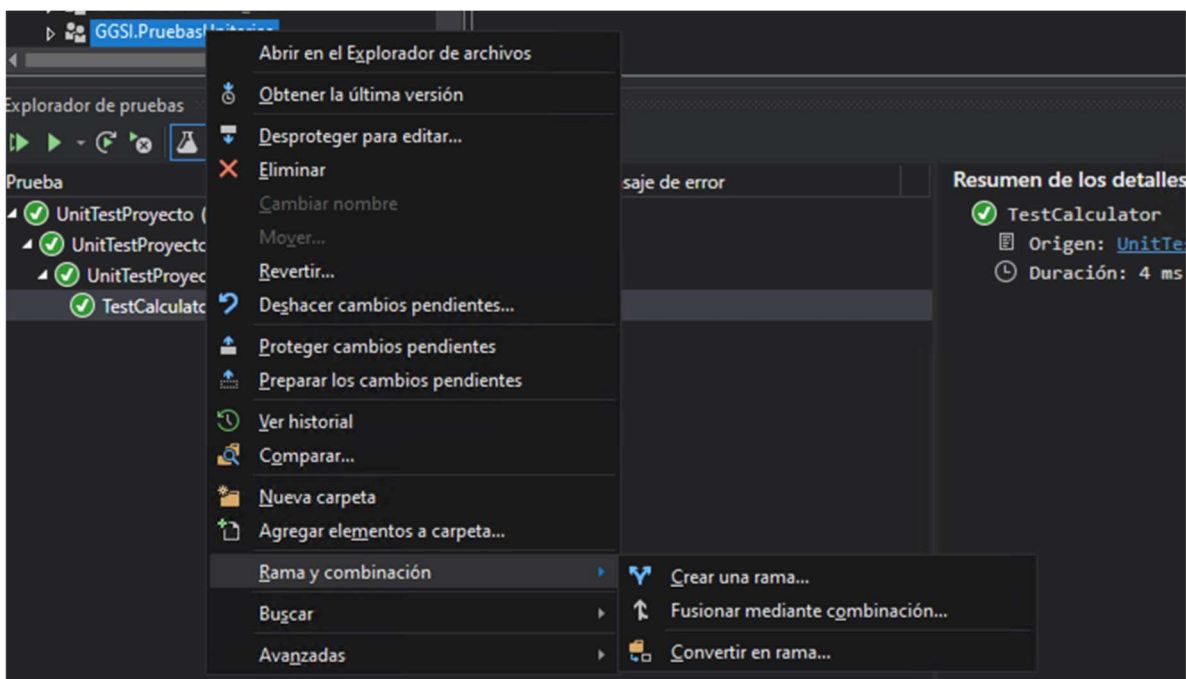


Figura 18 Crear rama de desarrollo

En la generación de la rama de desarrollo, se selecciona la rama de origen, la última versión de la rama y el nombre de destino, para este caso será “develop”. Ver figura 19.

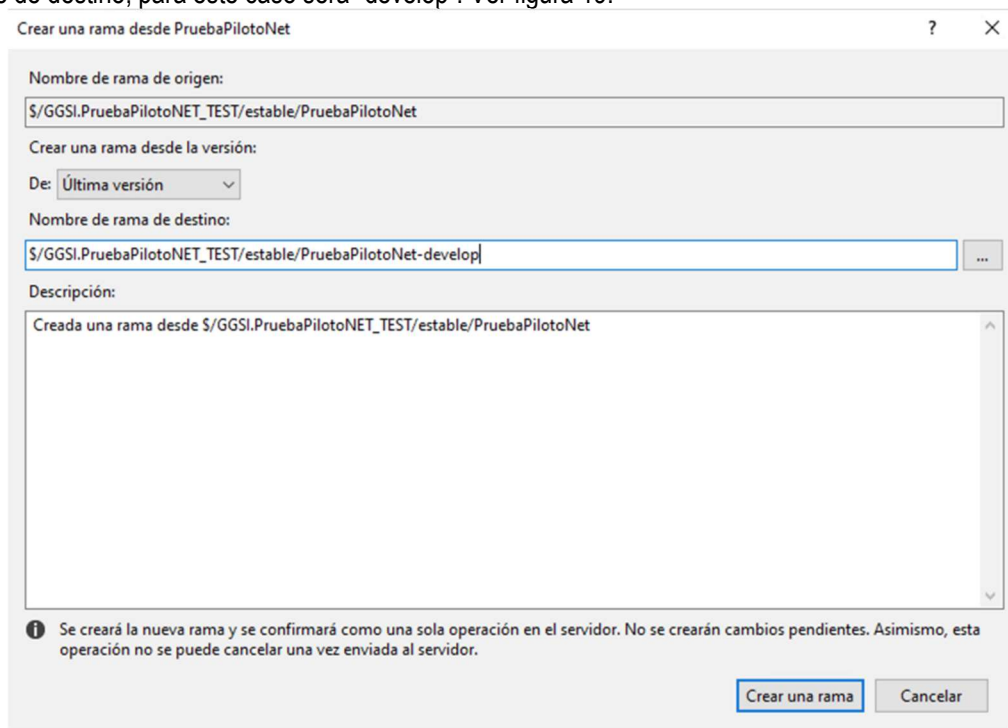


Figura 19. Indica la rama destino “develop”

5.7 Creación de ramas puntuales

La generación de las ramas puntuales está orientado a las historias de usuario de DevOps. Una vez este la rama de “develop”, para el caso del numeral 4.5 se tomará la historia de usuario #31366 Verificar reportes. Ver figura 20.

Ruta de acceso local: <E:\Lineamiento\otro\GGSI.PruebaPilotoNET TEST\estable>

Nombre	Cambio pendie...	Usuario	Última	Última inserció...
PruebaPilotoNet			Sí	15/09/2020 09:...
PruebaPilotoNet-develop			Sí	15/09/2020 09:...
PruebaPilotoNet-develop-31366			Sí	15/09/2020 10:...

Figura 20. Listado de las ramas

5.8 Sincronización rama <puntual> con la rama “develop”

Cuando se finaliza el desarrollo con su respectivo pruebas en la rama <puntual>, se debe sincronizar el desarrollo con la rama “develop”. Ver figura 21 y 22.

Asistente para la fusión mediante combinación del control de código fuente (área de trabajo: DNPD... ? X)

Seleccionar las ramas de origen y de destino para la operación Merge

Seleccione la rama que contiene los cambios que quiere fusionar mediante combinación.

Rama de origen:

Seleccione los cambios de la rama de origen que quiera fusionar mediante combinación:

☒ Cambios hasta una versión específica

☐ Conjuntos de cambios seleccionados

Seleccione la rama de destino para la operación Merge. La lista desplegable contiene todas las ramas de destino aplicables para la rama de origen seleccionada.

Rama de destino:

Asistente para la fusión mediante combinación del control de código fuente (área de trabajo: DNPD... ? X)

Seleccionar las versiones de los elementos de origen

Especifique la versión hasta la que deberían fusionarse mediante combinación los cambios de rama de origen. En la mayoría de los casos, los cambios anteriormente combinados no volverán a combinarse.

Tipo de versión:

Figura 21. Selección de la rama origen, destino y tipo de versión.

Asistente para la fusión mediante combinación del control de código fuente (área de trabajo: DNPD... ? ✕



Realizar la operación Merge

Se ha recopilado toda la información necesaria para la operación Merge.

La operación Merge se llevará a cabo cuando seleccione el botón "Finalizar". Se realizará una operación Merge intermedia en su área de trabajo. Debe realizar una operación de inserción en el repositorio para confirmar los cambios en el servidor.

Si se encuentran conflictos de fusión mediante combinación, tendrá la oportunidad de resolverlos.

< Anterior

Siguiente >

Finalizar

Cancelar

Figura 22. Sincronización "Merge" entre las ramas