



DEPARTAMENTO NACIONAL DE PLANEACIÓN

GUÍA PARA LA GESTIÓN DE PRUEBAS UNITARIAS AUTOMÁTICAS - PIPELINES

**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
1.0	25-09-2020	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.



1 OBJETIVO

En el presente documento tiene como objetivo dar a conocer el procedimiento en la gestión de pruebas unitarias automáticas 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

- **Pipelines.** Ofrece un conjunto completo de herramientas de automatización de CI/CD para implementaciones en máquinas virtuales, tanto en el entorno local como en cualquier nube.
- **GIT.** Es un software de control de versiones, pensado en la eficiencia y la confiabilidad del mantenimiento de versiones.
- **TFSVC.** Ayuda a la gestión de equipos de manera colaborativa, también ofrece control de versiones sobre los desarrollos.



4 Pre-requisitos

- a) Agente Pool: Debe contar como un agente administrador para poder realizar las compilaciones
- b) Repo: Debe contar con un repositorio ya sea en GIT o en TFSVC.

5. Pruebas Unitarias

Al desarrollo que está ejecutando se incluye el proyecto de pruebas unitarias de tipo “MSTest(.Net Core)” según la figura 1.

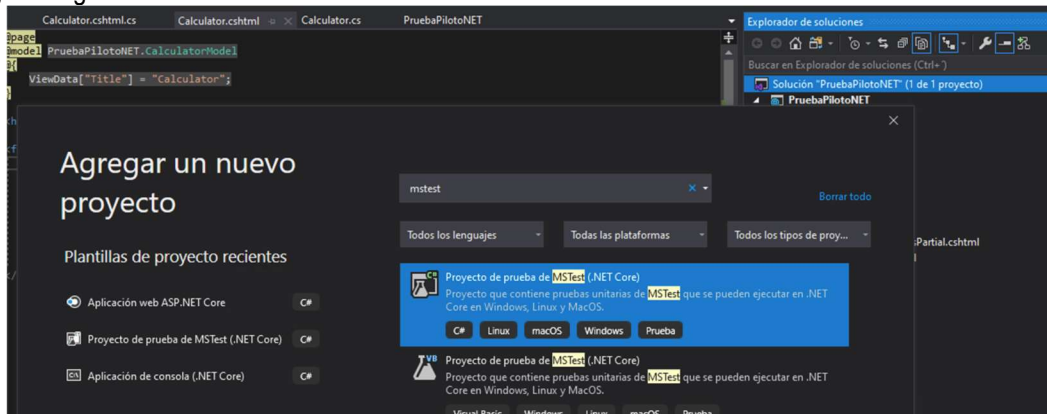


Figura 1. Inclusión de MSTest al proyecto

En el proyecto de MSTest debe agregar la referencia en la dependencia del proyecto. Para ello debe habilitar el nombre del proyecto el cual se ejecutará las pruebas unitarias. Figura 2

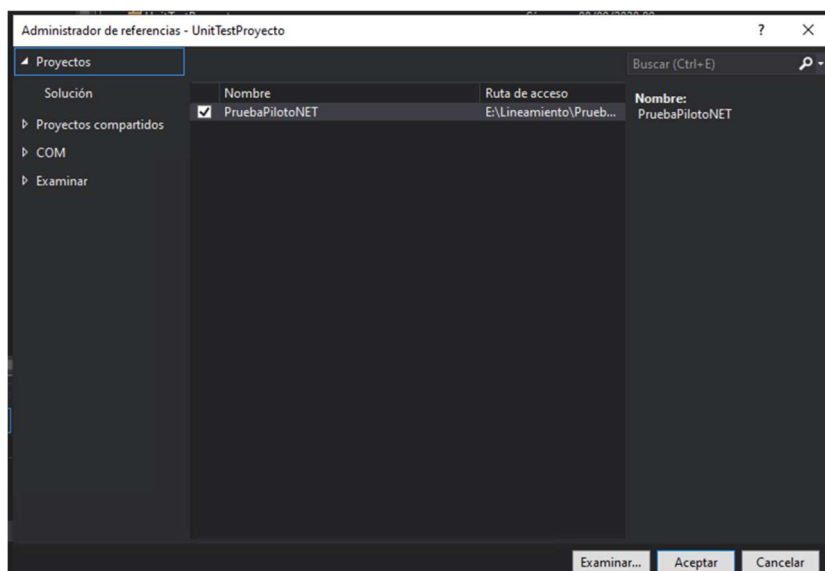


Figura 2. Vincular el proyecto a las pruebas unitarias.

Cuando se ejecuta las pruebas unitarias debe garantizar que las historias de usuario cumplan con lo solicitado. Ver Figura 3.

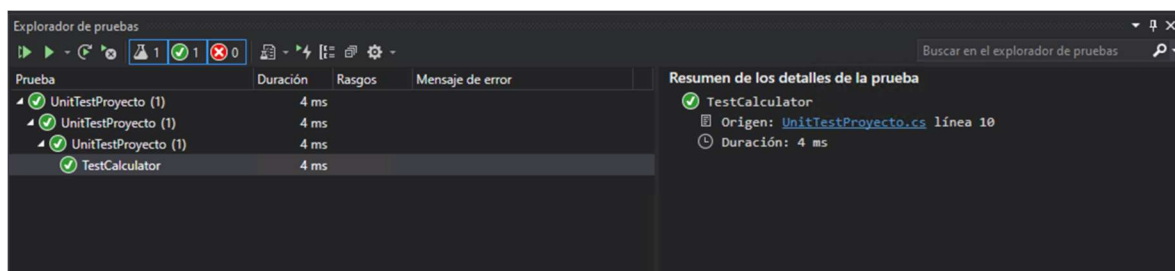


Figura 3. Resultados de las pruebas unitarias.

6. Pipelines

6.1 Creación de pipelines - Builds

Se selecciona el proyecto en Azure DevOps y va a “Pipelines” y crea un Pipeline de “Build”. Ver figura 4.

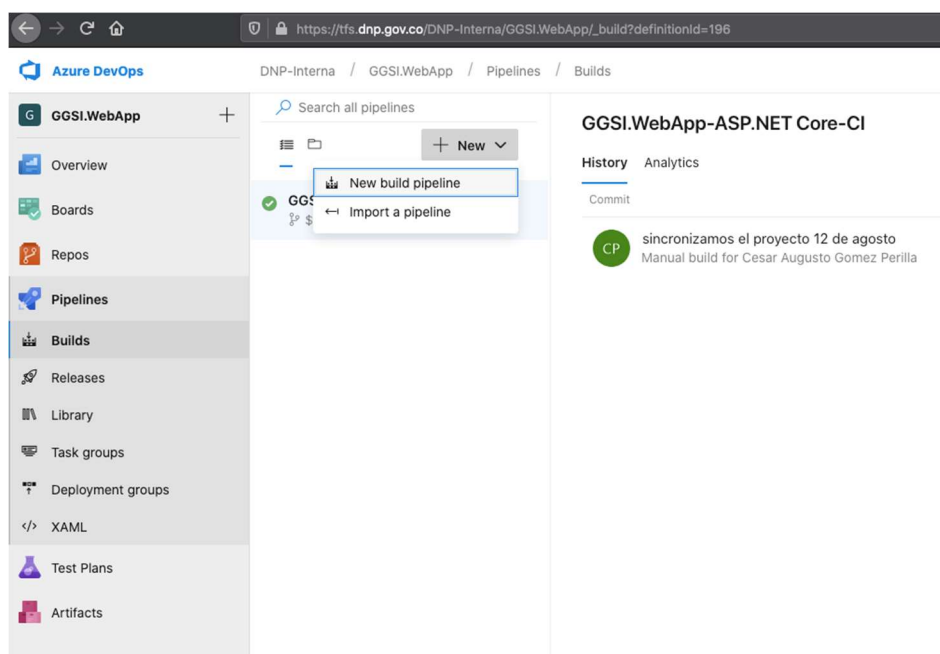


Figura 4. Nuevo Pipeline Build en Azure DevOps

6.2 Conexión a la versión de código

Se inicia conectandose a la versión de código. Para este caso será “Team Foundation Version Control” como se ve en la figura 5.

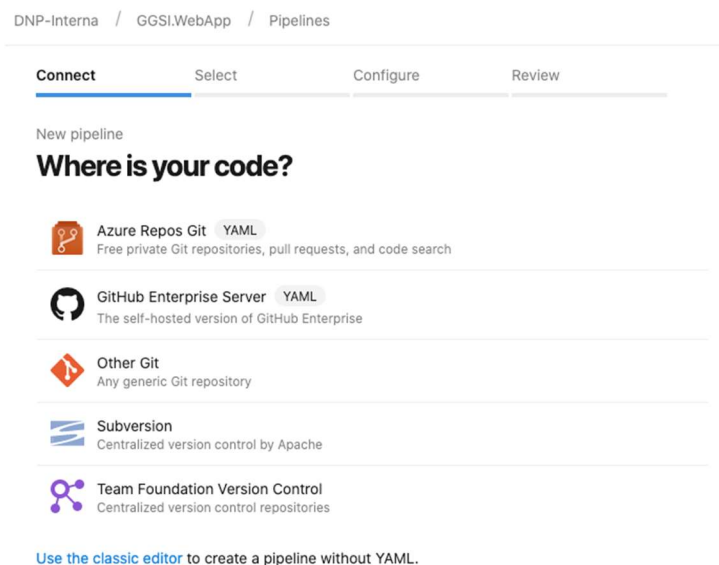


Figura 5. Conectarse al repositorio

6.3 Selección del repositorio

Una vez se conecte, se selecciona el repositorio “TFVC”. Ver figura 6.

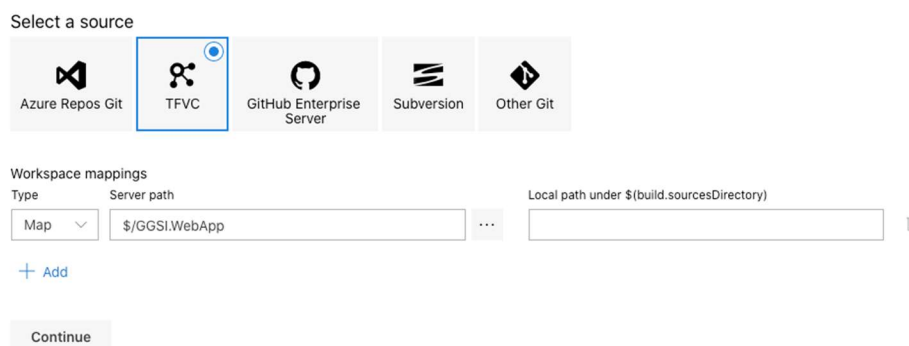


Figura 6. Trazabilidad de los comentarios del proyecto

6.4 Selección de la plantilla.

Según el lenguaje de desarrollo, se selecciona la plantilla, para este caso será con “ASP.NET Core”. Ver figura 7.

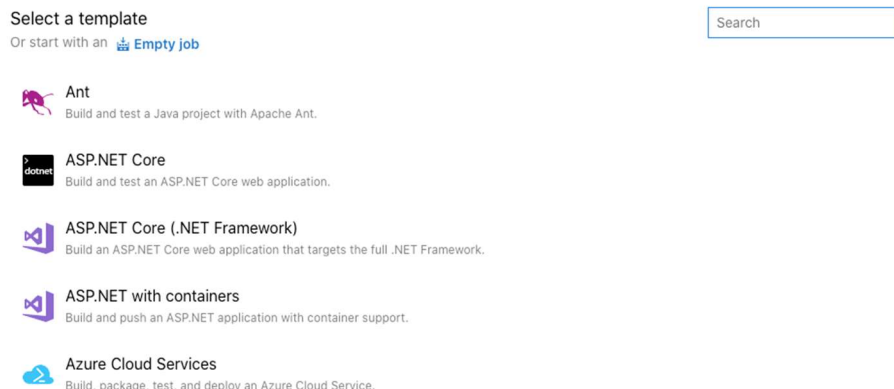


Figura 7. Selección plantilla

6.5 Configurar el agente de ejecución para el pipeline

Se debe definir el nombre y agente. Para el agente se puede dejar el que viene por **default**. Ver figura 8.

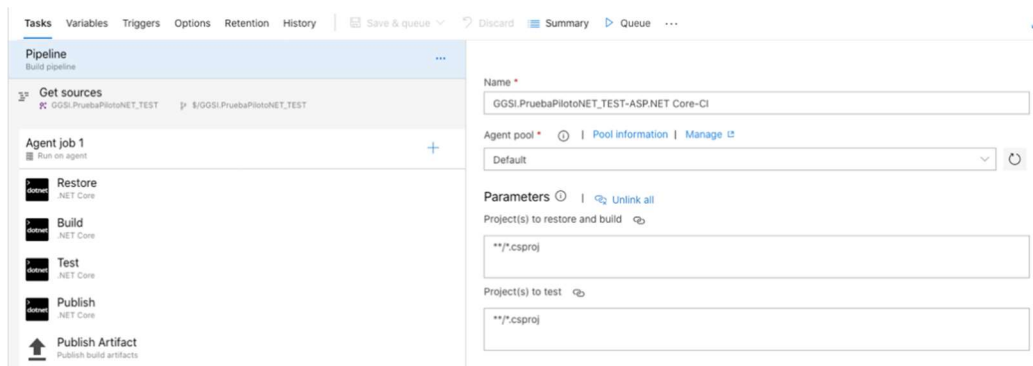


Figura 8. Definir el agente

6.6. Tarea Restaurar dependencias

Se coloca la primera tarea que tiene como finalidad restaurar las dependencias del proyecto. Ver figura 9. La configuración es la siguiente:

- **Display name:** Restore
- **Command:** restore
- **Feeds to use:** habilita "Feed(s) I select here" y "Use packages from NuGet.org"

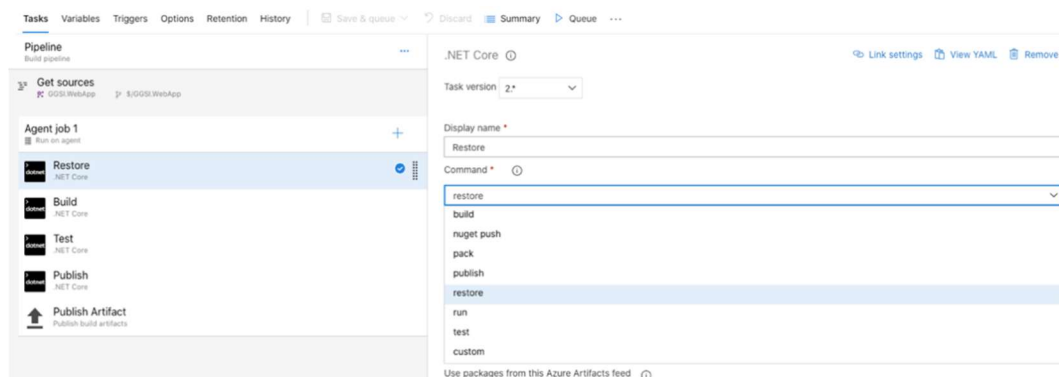


Figura 9. Tarea restaurar dependencias

6.7. Tarea Compilar

Luego se coloca la tarea de compilar el proyecto. Ver figura 10. La configuración ese la siguiente:

- **Display name:** Build
- **Command:** build

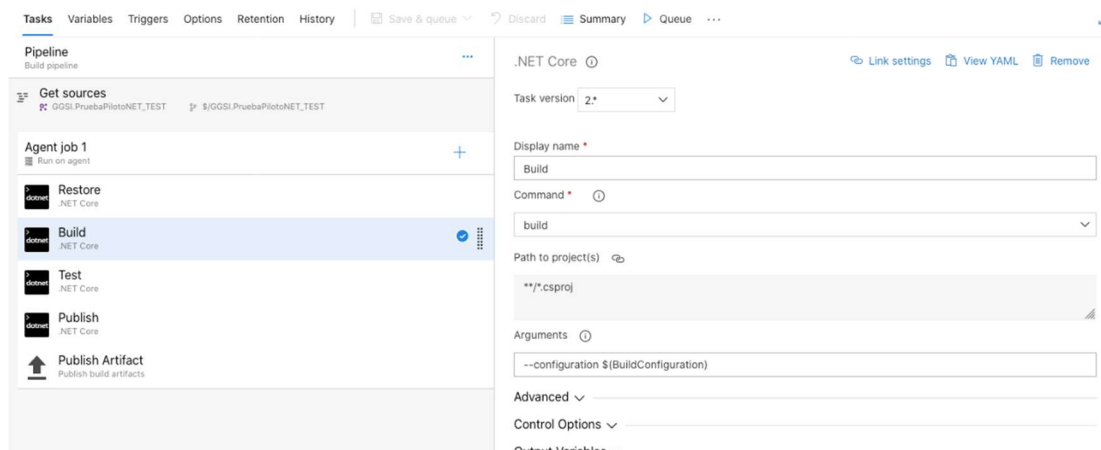


Figura 10. Tarea de compilar

6.8. Tarea Pruebas Unitarias

Se coloca la tarea de pruebas unitarias para ejecutar proyectos vinculados al mismo. Ver figura 11. La configuración ese la siguiente:

- **Display name:** Test
- **Command:** test
- **Arguments:** habilita "Publish test results and code coverage"

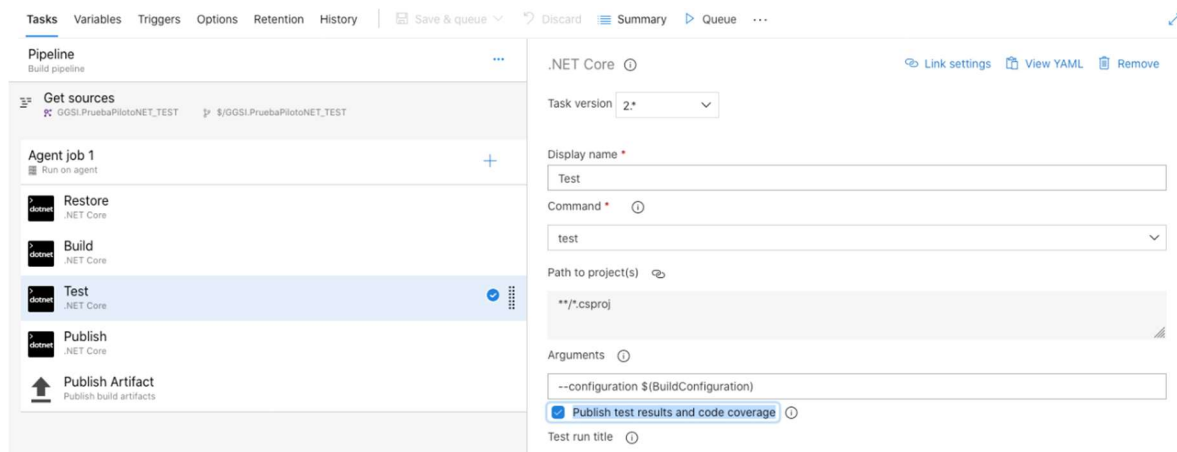


Figura 11. Pruebas unitarias

6.9. Tarea Despliegue

Una vez pasé la tarea de pruebas unitarias, se debe comprimir el proyecto para cuando se vaya hacer despliegues en ambientes ya sea en Nube o en máquinas virtuales. Ver figura 12

La configuración es la siguiente:

- **Display name:** Publish
- **Command:** publish
- **Publish Web Projects:** habilitado
- **Zip Publish Projects:** habilitado
- **Add Project name to publish path:** habilitado

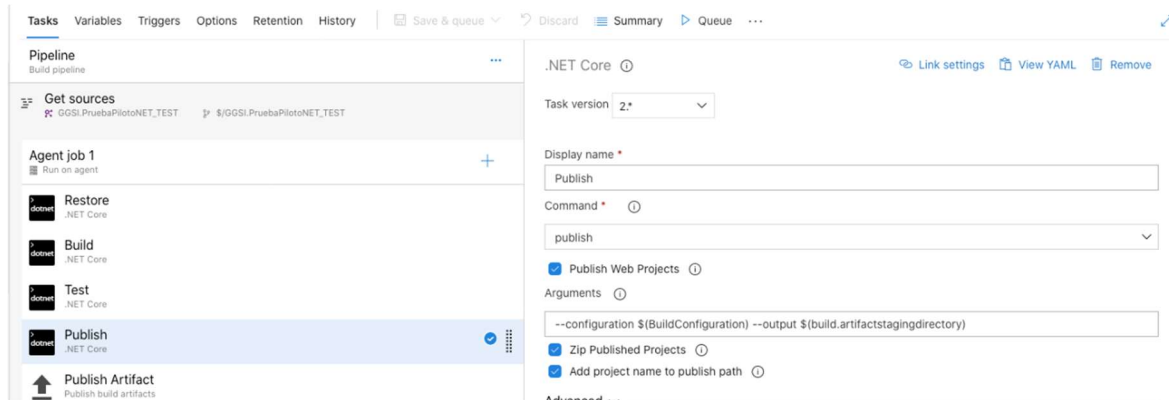


Figura 12. Despliegue

6.10. Guardar el pipeline y ejecutarlo

Una vez se tenga configurado el pipeline, se guarda y se ejecuta, se visualiza el proceso de cada tarea ya programada. ver figura 13

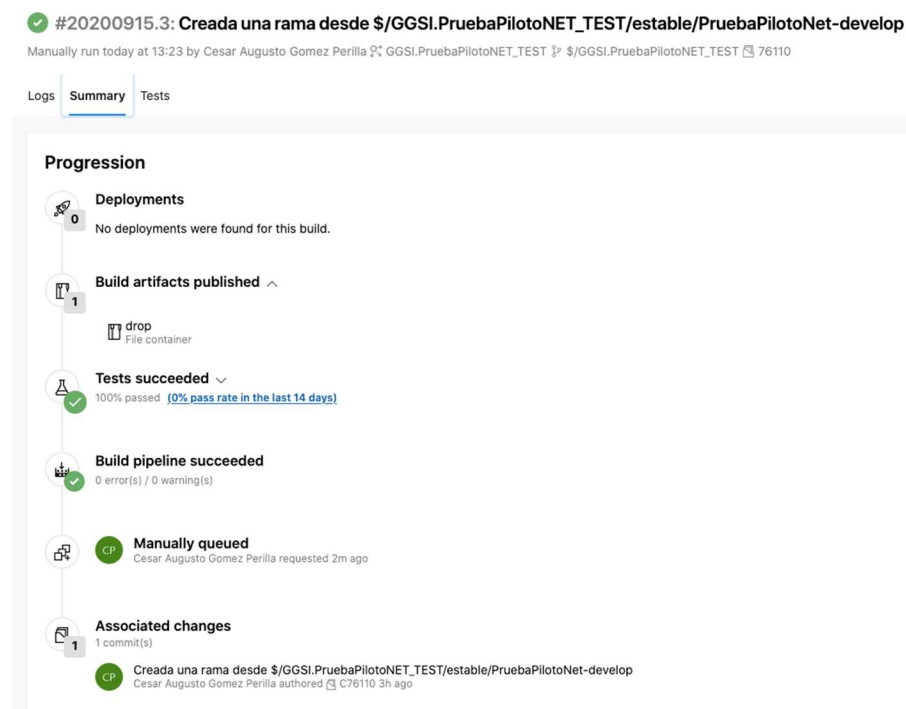


Figura 13. Proceso de ejecución del pipeline

En el proceso del pipeline en la tarea de “Test” se ejecutó la prueba unitaria que se encuentra en el proyecto. Ver figura 14.



Figura 14. Resumen de las pruebas unitarias

En caso de que las pruebas unitarias encuentren un error detiene el proceso del pipeline e informa cual fue el método que fallo en las pruebas unitarias. Ver figura 15, 16 y 17.

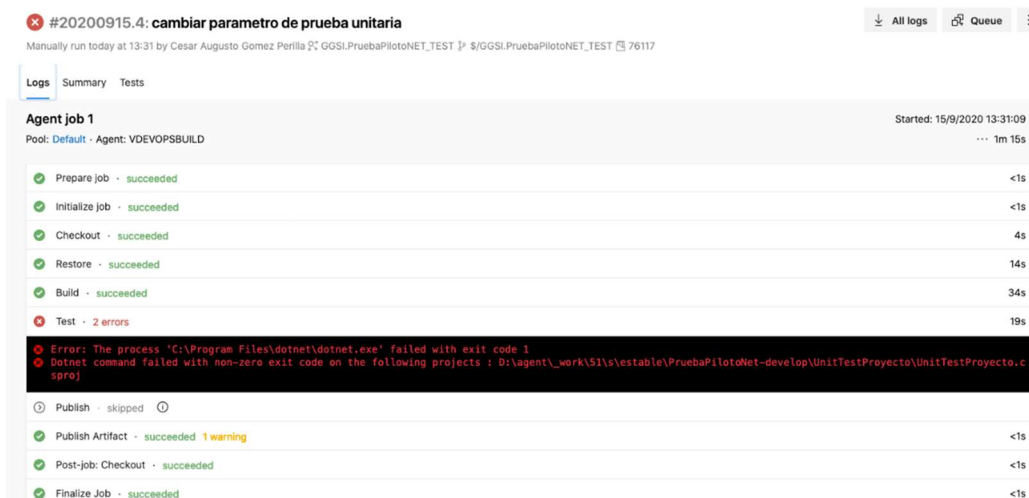


Figura 15. Logs del pipeline

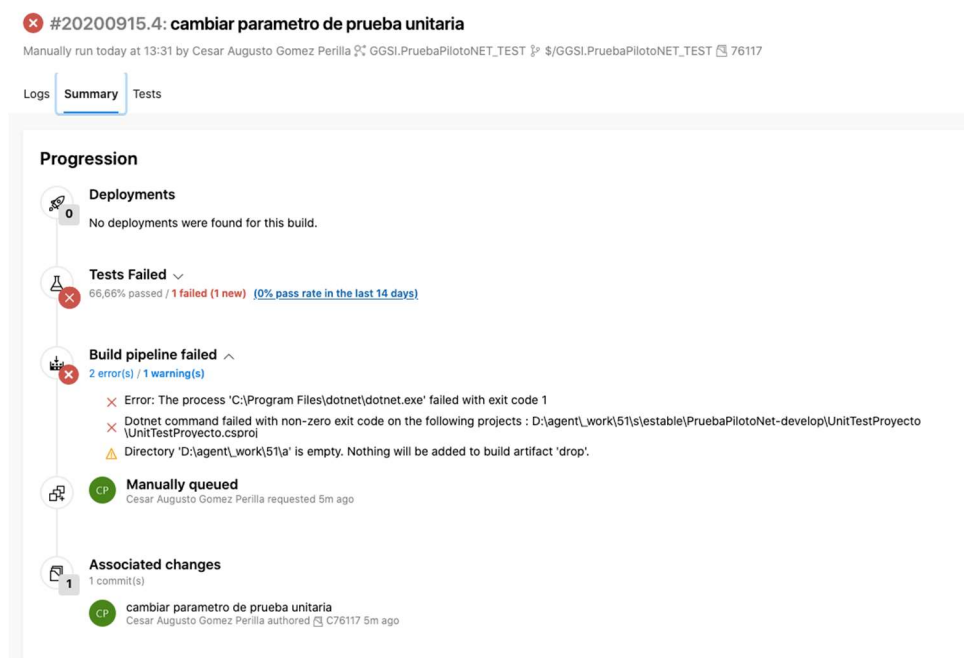


Figura 16. Resumen del pipeline

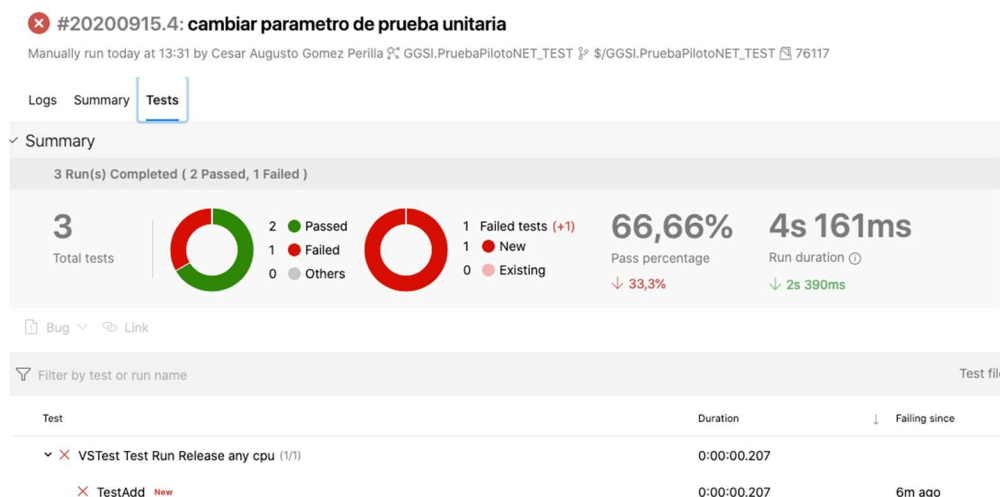


Figura 17. Tiempos de ejecución

6.6 Configuración de pipelines por rama

Las pruebas automáticas se pueden configurar por ramas según así lo desea. Para este caso se hará pruebas con la rama <puntual>, para ello se crea un pipeline como se muestra en el punto 6.1 y 6.2, luego de seleccionar TFVC cambia el directorio de “Server path” por la rama que se desea ejecutar para las pruebas automáticas como se ve en la figura 18, y continua con los puntos 6.4 y 6.5

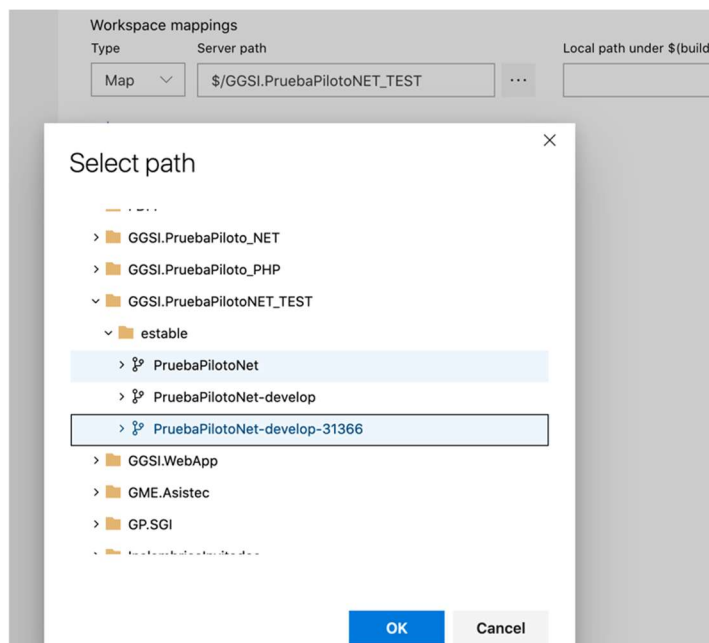


Figura 18. Seleccionando la rama puntual para el pipeline