

Devops



Sobre el Expositor

Mct. Esteban Calabria

Microsoft Certified Trainer

Microsoft Certified: DevOps Engineer Expert

Microsoft Certified: Azure Solutions Architect Expert

Microsoft Certified: Azure Developer Associate

Microsoft Certified: Azure AI Fundamentals



<https://www.linkedin.com/in/esteban-calabria-7a44401a/>



<https://www.instagram.com/mct.esteban.calabria/>

OBJETIVOS

- Introducir el ecosistema de Azure DevOps.
- Mostrar el uso efectivo de las herramientas de desarrollo y despliegue de Azure DevOps.
- Demostrar cómo Azure DevOps puede mejorar la colaboración y la eficiencia en equipos de desarrollo de software.
- Entender los Conceptos básicos de Devops
- Mejores prácticas de Azure DevOps

DIRIGIDO

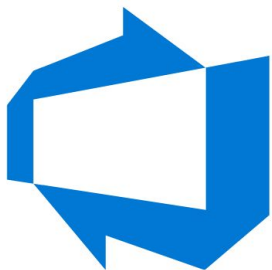
Este curso está dirigido a

- Profesionales de TI
- Desarrolladores de software
- Administradores de sistemas
- Cualquier persona interesada en aprender cómo implementar Azure DevOps.

REQUISITOS:

- Conocimientos básicos de desarrollo de software.
- Familiaridad con los conceptos básicos de la nube y Azure (preferible pero no obligatorio).

¿Qué es Azure DevOps?



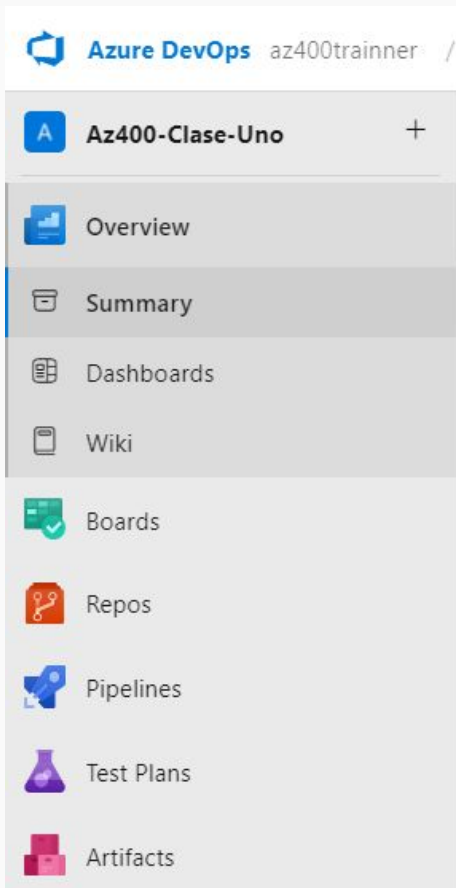
Azure DevOps

Beneficios y ventajas de usar Azure DevOps.

Visión general de las herramientas incluidas en Azure DevOps:

- Azure Boards
- Azure Repos,
- Azure Pipelines,
- Azure Test Plans
- Azure Artifacts

Conceptos Preliminares...

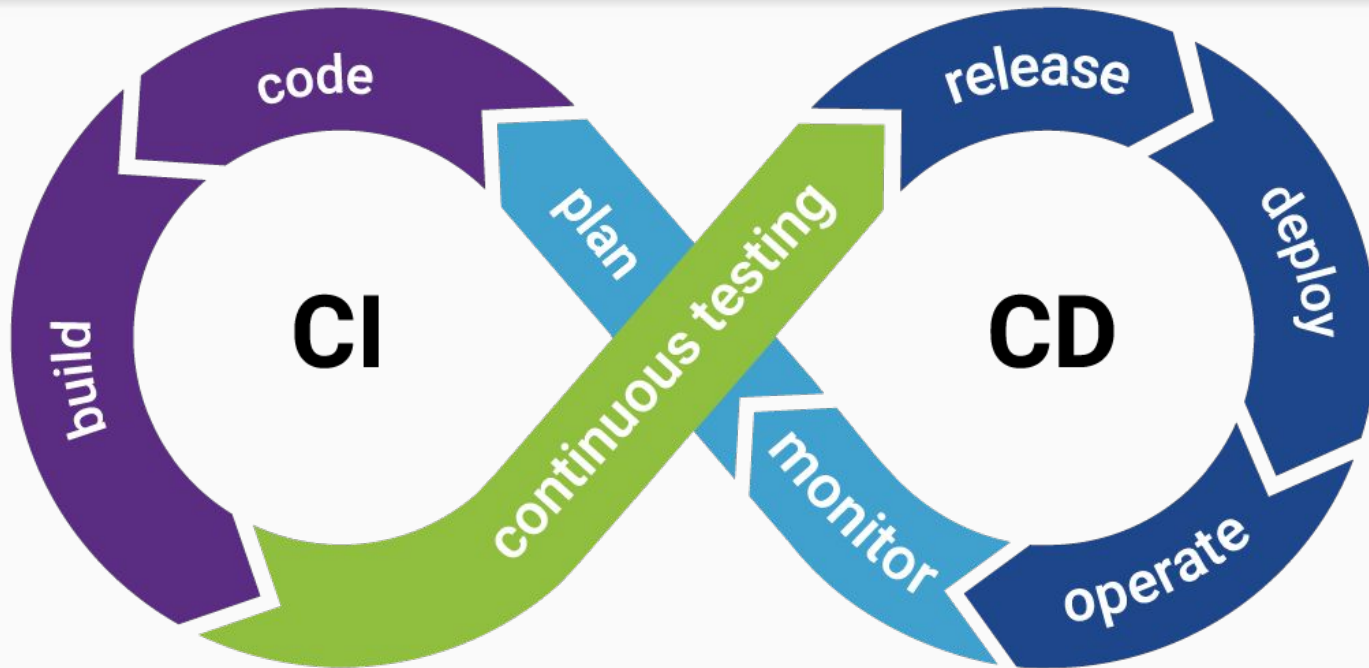


- Integración Continua
- 👉👉 Pruebas Unitarias
- Organización en Azure DevOps
 - Organización
 - Proyecto
- Tipos de Proyectos
- Continuous Deployment
- Continuous Integration
- (IaC)
 - Infraestructura como Código
- Monitoreo y Logging

Integración Continua



Continuous Integration vs Continuous Deployment





Azure Repos

Azure Repos

Azure Repos

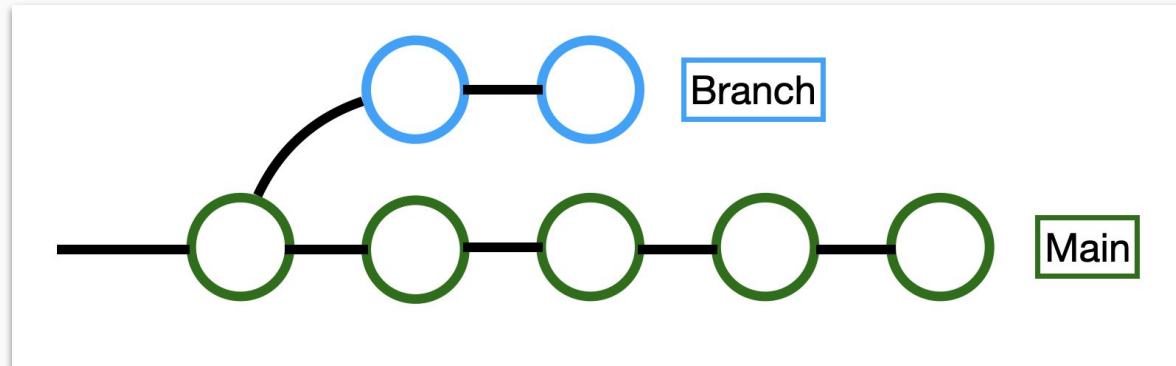
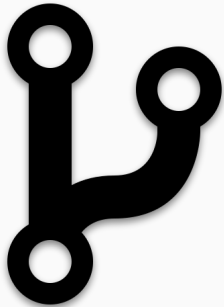


- Control de versiones con Git en Azure Repos.
- Creación y gestión de repositorios.
- Trabajo con ramas (branching) y solicitudes de extracción (pull requests).
- Integración con Visual Studio Code y otras herramientas de desarrollo.

Azure Repos

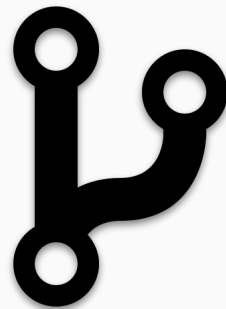
Ramas (Branches)

En Git, una rama es una línea de desarrollo independiente que permite trabajar en nuevas características, correcciones de errores o experimentar sin afectar la rama principal del proyecto. Cada rama es una secuencia de commits, y puedes fusionar ramas para integrar los cambios realizados.



Buenas Practicas

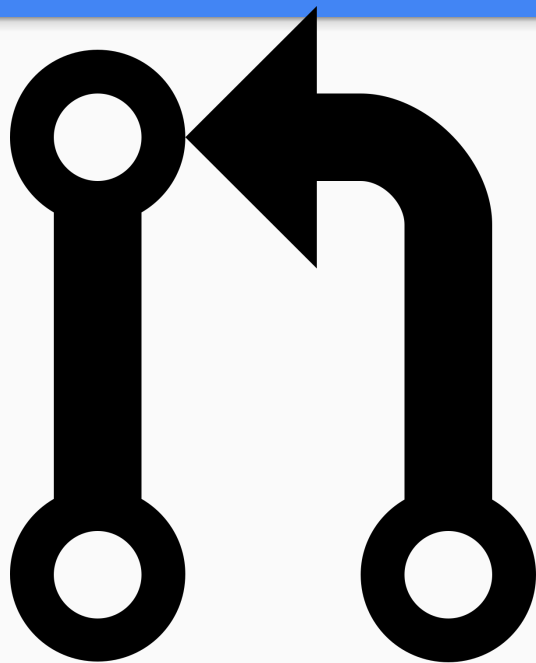
- Ramas por funcionalidad o tarea:
 - Crear una rama cada vez que se trabaje en una nueva funcionalidad o se resuelva una tarea específica
- Ramas de desarrollo y de lanzamiento:
 - Mantener una rama de desarrollo donde se integren todas las funcionalidades completadas.
 - Crear una rama de lanzamiento a partir de la rama de desarrollo antes de pasar a producción.
- Ramas de características vs. ramas de hotfix



- Pull Requests
 - Políticas de revisión de código y aprobación de fusión
- Integración continua y **pruebas automatizadas**:
 - Utilizar la integración continua para fusionar regularmente cambios en la rama principal y ejecutar pruebas automatizadas para validar la estabilidad del código.
- Seguimiento de cambios y auditoría:
 - Proporciona un seguimiento detallado de todos los cambios realizados en el repositorio, lo que facilita la identificación de quién realizó cada cambio y cuándo.

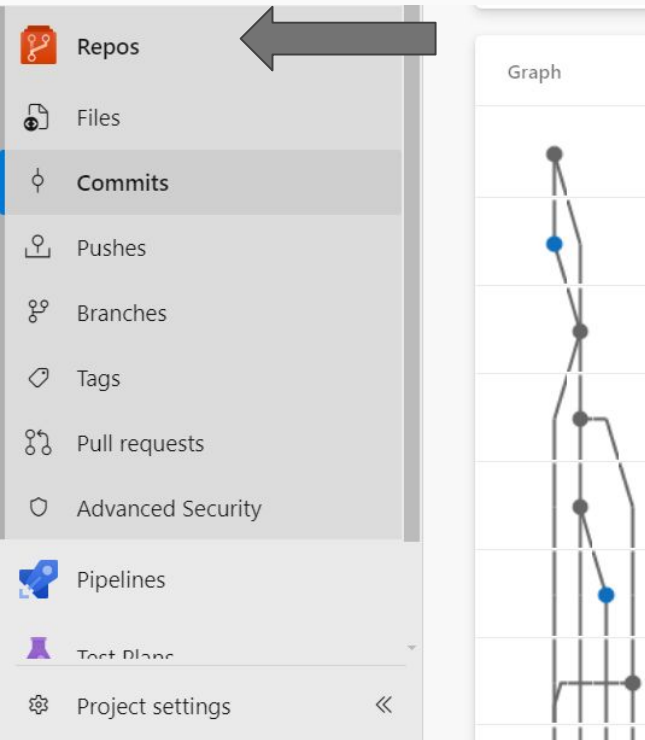


Pull Request



- Un pull request es una solicitud para revisar e integrar cambios desde una rama en otra.
- Permite configurar políticas que requieran revisiones de código y aprobaciones antes de fusionar cambios en la rama principal.
- Esto ayuda a garantizar que el código sea revisado por pares y cumpla con los estándares de calidad del equipo.

Seguimiento de cambios y auditoría





- Utiliza el sistema de control de versiones de Azure Repos.
- Examina el historial de commits.
 - comando : git log
- Utiliza las herramientas de seguimiento de problemas.
 - work items
- Configura políticas de seguridad y acceso.
- Utiliza informes y paneles de control










Seguimiento de cambios y auditoría

Branches

Mine All Stale

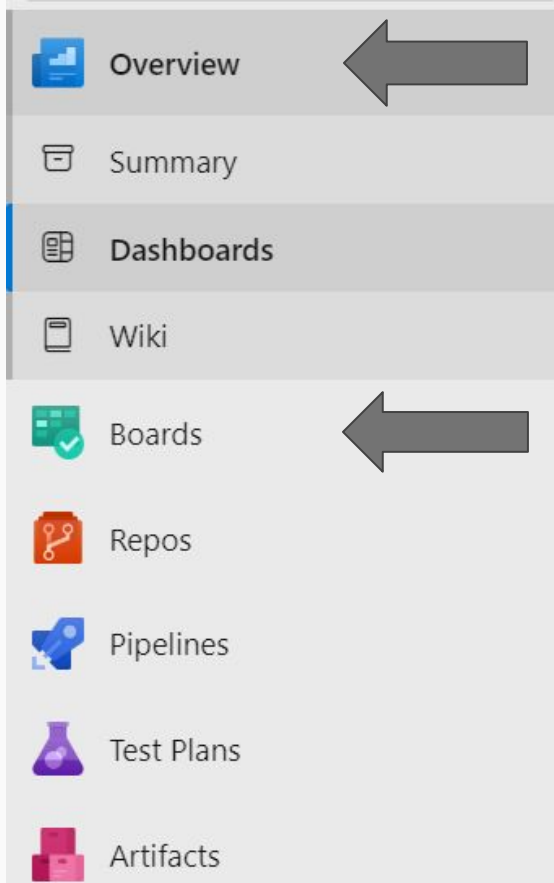
Search branch name

Branch	Co...	Author	Authore...	Behind Ahead	Sta...	Pul...
 MAIN Default Compare	cec6ef	 Álvaro Ver...	8 feb	 s		

-  New pull request
-  Delete branch
-
-  View files
-  View history
-  Compare branches
-  Set as compare branch
-
-  Lock
-  Branch policies
-  Branch security

- Configura políticas de seguridad y acceso.

Utiliza informes y paneles de control



- Utilización de informes predefinidos:
- Creación de paneles de control personalizados:
- Seguimiento de métricas de rendimiento:
- Análisis de tendencias a lo largo del tiempo:
- Comunicación transparente del progreso del proyecto:

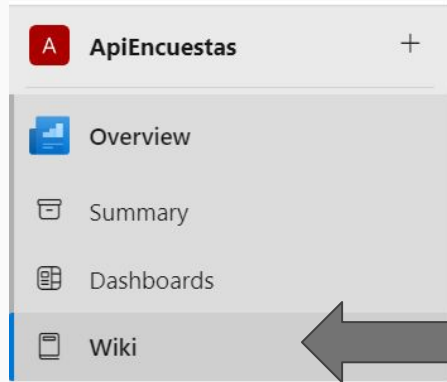
¿Cómo se identifica la rama
en uso en producción?



Identificación de la rama en uso en producción

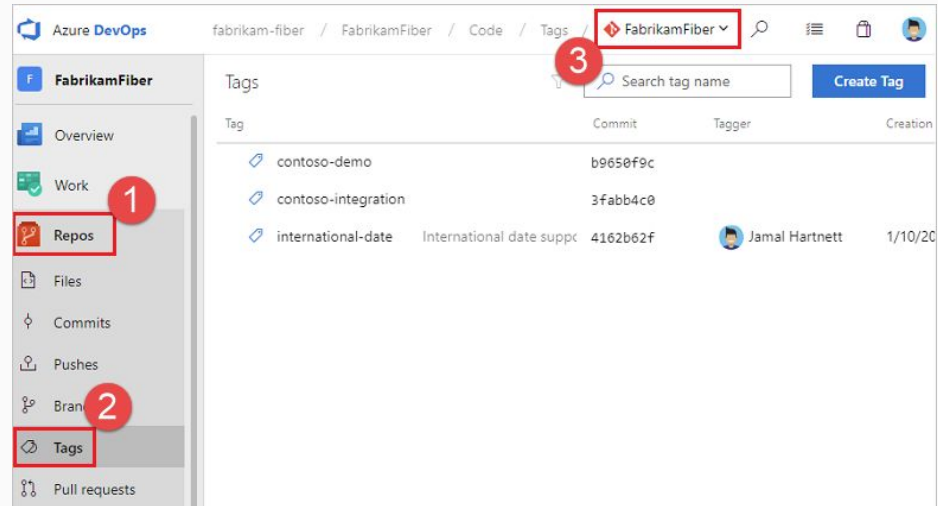
- **Documentación de versiones:**

- Mantener un documento en el proyecto de Azure DevOps que indique
 - la versión en producción
 - la rama asociada.
 - la fecha de lanzamiento,
 - los cambios implementados



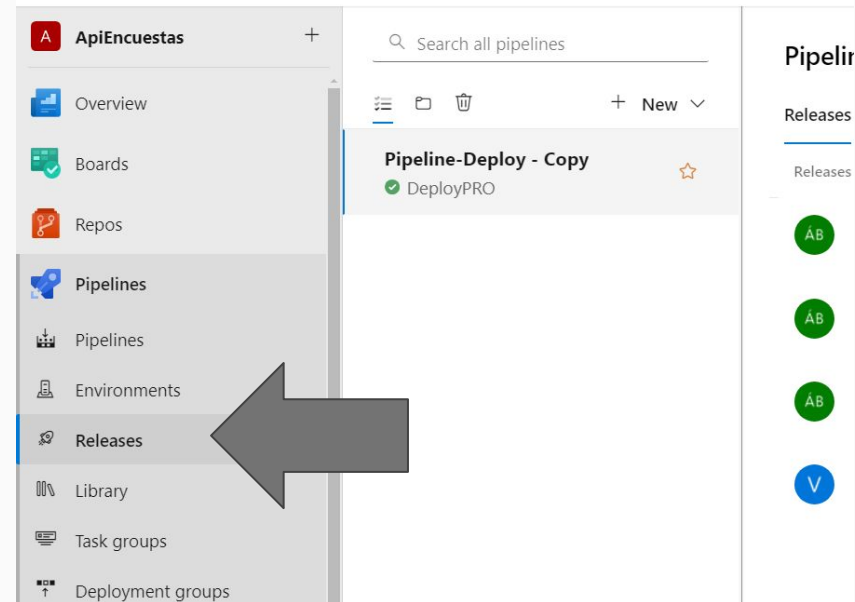
- **Etiquetado de versiones:**

- Puedes etiquetar versiones utilizando Git.



Identificación de la rama en uso en producción

- **Seguimiento de despliegues:**
 - Utilizar la funcionalidad de "Releases" (versiones) para llevar un registro de los despliegues realizados en cada entorno
 - Configurar pipelines de lanzamiento que automatizan el proceso de implementación en diferentes entornos y registran los despliegues realizados.



Puedes configurar integraciones con herramientas de gestión de despliegues externas para registrar y rastrear los despliegues realizados fuera de Azure DevOps.

¿Cómo se gestiona el
alineamiento antes nuevas
funciones?



Mantenimiento del alineamiento/desalineamiento antes de nuevas funcionalidades:

Utiliza ramas separadas para dev

Crea ramas separadas para desarrollar nuevas funcionalidades o realizar correcciones de errores. Esto te permite trabajar de manera aislada sin afectar la rama principal (generalmente la rama de producción).

Pruebas en entornos previos

Antes de implementar una nueva funcionalidad en producción, realiza pruebas exhaustivas en entornos previos, como entornos de desarrollo, pruebas y preproducción.

Esto te permite identificar problemas y corregirlos antes de que afecten a los usuarios finales.

Mantenimiento del alineamiento/desalineamiento antes de nuevas funcionalidades:

Revisión de código y pull request

Configura una política que requiera revisión de código y aprobación de fusión antes de fusionar cambios en la rama principal. Esto ayuda a garantizar que todas las nuevas funcionalidades se revisen y aprueben antes de ser implementadas en producción.

Despliegues escalonados:

En lugar de implementar todas las nuevas funcionalidades de una vez, considera la posibilidad de realizar despliegues escalonados,

Esto te permite detectar problemas de manera temprana y mitigar el impacto en caso de errores.

¿Cuál es el procedimiento
para forzar el despliegue?



Despliegue manual

Configurar un pipeline de lanzamiento (release pipeline) para desplegar tu aplicación en diferentes entornos, incluido el de producción.

Crear una nueva versión del pipeline de release asociada a una rama de código y forzar su despliegue en producción

Automatización del despliegue

Puedes configurar triggers para que el pipeline de lanzamiento se active automáticamente cuando se realicen cambios en una determinada rama de código.

De esta manera, los despliegues se realizarán automáticamente tan pronto como se publiquen nuevos cambios en la rama.

¿Que hago en caso de
necesitar revertir la última
versión del código?



Snapshot de la infraestructura

Si tu aplicación se ejecuta en una infraestructura en la nube crear snapshots de tus recursos (máquinas virtuales, bases de datos) antes de realizar un despliegue importante.

Estos snapshots actúan como copias de seguridad que pueden ser restauradas en caso de necesidad.

Revertir cambios en el repo

Si necesitas revertir la publicación de la última versión de código, puedes hacerlo utilizando herramientas de control de versiones como Git.

Puedes revertir los commits específicos que introdujeron los cambios problemáticos y volver al estado anterior del código.

Respaldo de datos críticos:

Si tu aplicación utiliza una base de datos u otros datos críticos, asegúrate de tener un plan de respaldo y restauración en su lugar.

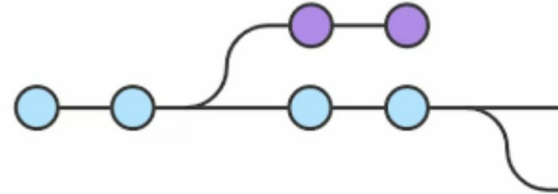
Puedes realizar copias de seguridad regulares de tus bases de datos y almacenarlas en un lugar seguro. En caso de necesidad, puedes restaurar una copia de seguridad anterior para revertir los cambios no deseados.



Estrategias de Branching

Estrategias de Branching

- Git Flow
- GitHub Flow
- GitLab Flow
- Trunk-Based Development
- Estrategia de ramas de equipo



Git Flow





- Ingreso al portal de Devops
 - <https://aex.dev.azure.com>
- Creemos un proyecto en Devops
- Creemos un proyecto local con Vs o VSCode
- Subamos el proyecto a Devops
- Realicemos un branch
- Actualizamos el branch con un pull request



Azure Pipelines

Azure Pipelines

Azure Pipelines



Azure Pipelines

Automatización de compilación y despliegue.

Creación de pipelines de compilación y despliegue.

Ejecutando pipelines en Devops vs Ejecutando pipelines Localmente

Configuración de pipelines para diferentes tipos de aplicaciones

1. Aplicaciones web
2. Contenedores
3. Funciones de Azure

Uso de plantillas y variables para una configuración más flexible.

Azure Pipeline



Azure Pipelines

Un pipeline es una serie de pasos automatizados para construir, probar y desplegar aplicaciones, y generalmente se escribe en YAML para definir la secuencia y configuración de estos procesos.

Ejemplo De Pipeline

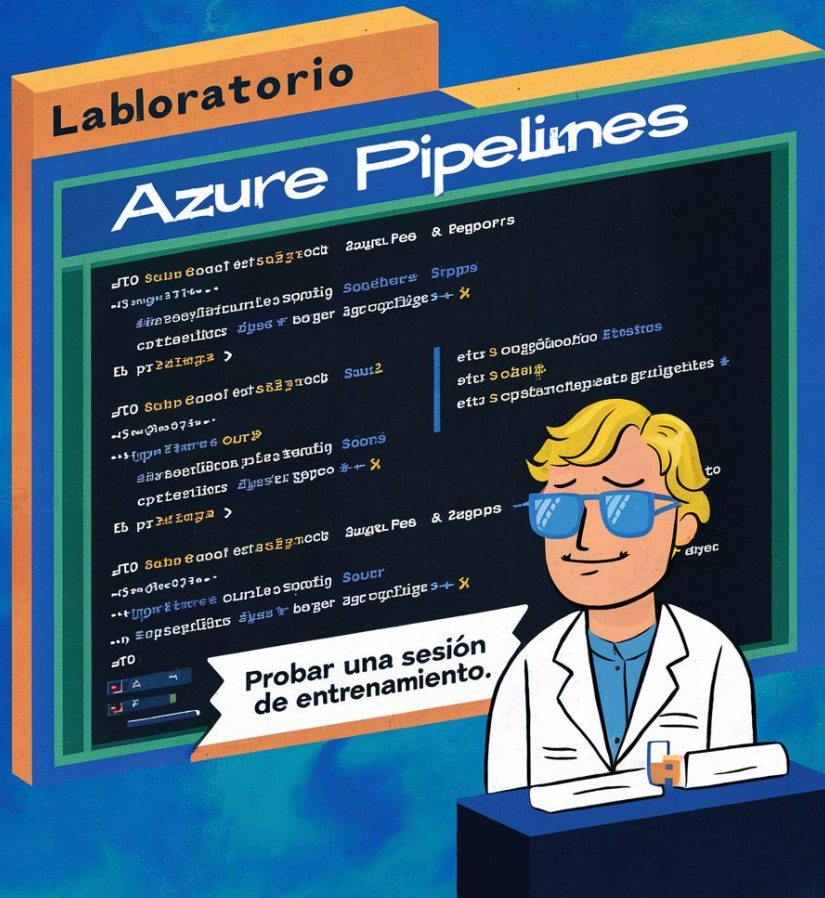
estebancalabria / demo-sonarqube / Pipelines

← demo-sonarqube

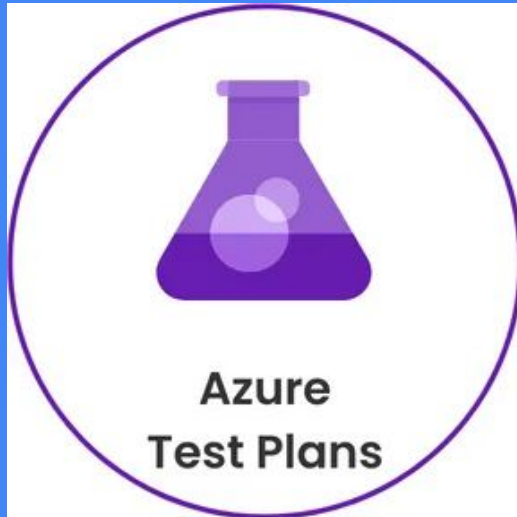
master ▼ demo-sonarqube / azure-pipelines.yml

```
1 trigger:
2   - master
3
4 pool:
5   - vmImage: vs2017-win2016
6
7 steps:
8   - script: echo "Iniciando Build"
9     displayName: 'Iniciando Build'
10
11   - checkout: self
12     clean: true
13     displayName: "Bajando los fuentes"
14
15 Settings
16   - task: SonarQubePrepare@4
17     inputs:
18       - SonarQube: Devops-Sonar
19       - scannerMode: MSBuild
20       - projectKey: demo-sonarqube_demo-sonarqube
21
22 Settings
23   - task: DotNetCoreCLI@2
24     displayName: "Construyendo Solucion"
25
26 Settings
27   - task: SonarQubeAnalyze@4
28
29 Settings
30   - task: SonarQubePublish@4
31     inputs:
32       - pollingTimeoutSec: 300
33
34 Settings
35   - task: sonar-buildbreaker@8
```

Creando nuestro primer pipeline para desplegar nuestra app en Azure

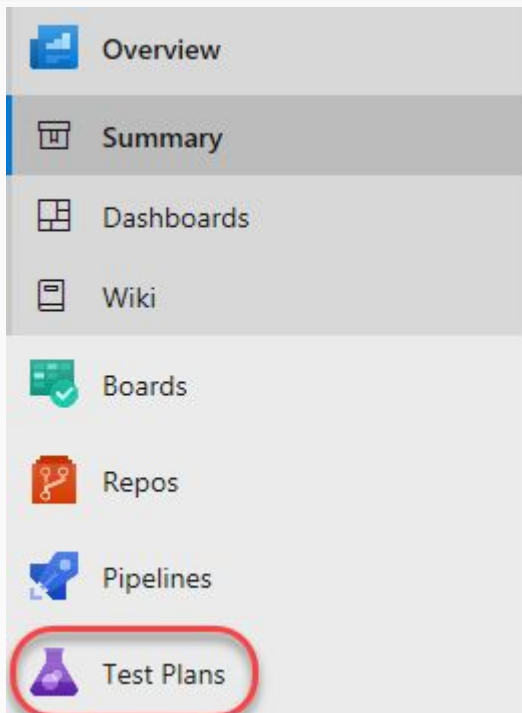


Veamos cómo
desplegar nuestra
app en Azure
utilizando Azure
Pipelines



Azure Azure Test Plans

Azure Test Plans



En esta sesión, explicaremos cómo Azure DevOps facilita la gestión del ciclo de vida de las pruebas en proyectos de software."

Veremos cómo planificar pruebas, organizar casos de prueba y supervisar el progreso de las pruebas."



- Organización de pruebas.
- Ejecución de pruebas
- Seguimiento de resultados.
- Integración con otros servicios.
- Integración con pipelines para una colaboración efectiva entre equipos de desarrollo y operaciones.

Requisitos

SERVICIOS INDIVIDUALES

Azure Pipelines



1 CI/CD hospedado por Microsoft gratuito
1 CI/CD autohospedado gratuito

Empiece gratis

- 1 trabajo hospedado por Microsoft con 1,800 minutos al mes para CI/CD y 1 trabajo autohospedado con minutos ilimitados al mes
- €36,99 por trabajo paralelo extra en una canalización de CI/CD hospedada por Microsoft y €13,88 por trabajo paralelo extra en una canalización de CI/CD autohospedada con minutos ilimitados

Azure Artifacts



2 GiB gratis;
después, a partir de €1,85 por GiB

Empiece gratis

- Servidor NuGet líder del sector
- Compatibilidad con paquetes de Maven, npm y Python
- Orígenes ascendentes para ayudar a proteger las dependencias de código abierto
- Integrado con Azure Pipelines
- Controles de acceso sofisticados

LICENCIAS DE USUARIO

Plan Básico



Los 5 primeros usuarios gratis;
después, €5,55 por usuario al mes

Empiece gratis

- **Azure Pipelines:** Incluye la oferta gratis de SERVICIOS INDIVIDUALES
- **Azure Boards:** seguimiento de elementos de trabajo y paneles kanban
- **Azure Repos:** repositorios GIT privados ilimitados
- **Azure Artifacts:** 2 GiB gratis por organización

Básica + Test Plans



€48,09 por usuario
al mes

Evaluación gratuita de 30 días

- Incluye todas las características del plan Básico
- Planificación, seguimiento y ejecución de pruebas
- Pruebas basadas en explorador con anotación
- Ejecución de pruebas de cliente mejorado
- Pruebas de aceptación del usuario
- Informes centralizados

Visual Studio subscription	Azure DevOps plan
Visual Studio Professional	Basic
Visual Studio Enterprise	Basic + Test
Visual Studio Test Professional	Basic + Test
MSDN Platforms	Basic + Test

Requisitos



Azure DevOps

serviciosmin-adi / Settings / Users

Search



Organization Settin...

serviciosmin-adi

Search Settings

General

Overview

Projects

Users

Billing

Global notifications

Usage

Extensions

Microsoft Entra

All users

Filter users

Access Level

Microsoft Entra User Type



Total 46

Summary

Name ↑

Access Level

Date Added

Last Accessed



Aarón Hernández
aaron.hernandez@alten.es

Visual Studio Professional subscription

4/5/2022

27/5/2024



Adrian Palma Lima
adrian.palma@izertis.com

Visual Studio Enterprise subscription

21/7/2023

16/8/2023



Ana.masa
Ana.masa@alten.es

Visual Studio Professional subscription

12/9/2022

4/6/2024



Carlos Codesal Garcia
ccodesal@izertis.com

Visual Studio Enterprise subscription

31/5/2024

4/6/2024

Azure Test Plans



Test Plans



Test plans



Progress report



Parameters

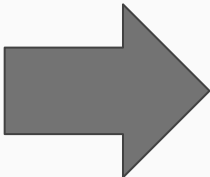


Configurations



Runs

- **Gestión de Casos de Prueba**
 - Permite gestionar y organizar casos de prueba manuales
 - Garantizar una cobertura de pruebas adecuada.
- **Ejecución de Pruebas**
 - Facilita la ejecución de pruebas manuales y automatizadas
 - verificar que las aplicaciones cumplen con los requisitos y funcionan correctamente.
- **Seguimiento de Defectos**
 - Integración con Azure Boards
- **Informes y Análisis**
 - Capacidades de informes y análisis para visualizar los resultados de las prueba
 - Métricas clave como la tasa de éxito de las pruebas
 - Progreso de las pruebas a lo largo del tiempo.
- **Integración Continua**
 - Integración con las pipelines de Azure DevOps
 - Ejecución automática de pruebas en cada build



- Overview
- Summary
- Dashboards
- Wiki
- Boards
- Repos
- Pipelines
- Test Plans**

Mine All + New Test Plan

Title

My Favorites

No favorites yet! Favorite ★ a test plan to quickly access it here.

Parts Unlimited Team

Parts Unlimited_TestPlan1

Ventajas Azure Test Plans

- Integración Completa
- Flexibilidad en la Planificación
- Organización Eficiente
- Seguimiento y Análisis
- Colaboración en Equipo



Conclusiones y preguntas

Preguntas



Recursos adicionales y siguientes pasos para profundizar en Azure DevOps.

Microsoft Learn

<https://learn.microsoft.com/es-es/credentials/certifications/devops-engineer/>

Laboratorios Az-400

<https://microsoftlearning.github.io/AZ400-DesigningandImplementingMicrosoftDevOpsSolutions/>

Gracias