PREPÁRATE
PARA SER EL
MEJOR

+ ENTREMIENTO
EXPERIENCIA

BIENVENIDOS.

ENTRENAMIENTO
EXPERIENCIA

GALAXY
TRAINING

Azure DevOps

Ing. Erick Arostegui Cunza
Instructor

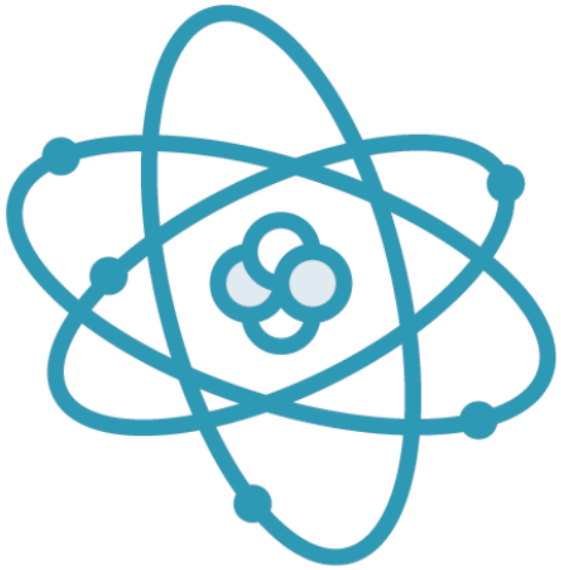
earostegui@galaxy.edu.pe

AGENDA

CI - REPOSITORIOS Y CONTROL DE CÓDIGO FUENTE

- Control de versiones (fundamentos de control de fuente y tipos de sistemas de control de fuente).
- Administrando su código fuente con Git en Azure DevOps (Git vs TFVC y Git + Azure DevOps).
- Entendiendo Git (operaciones básicas, Git Branching).
- Estrategias de ramificación (Git Hub Flow, Git Flow, Git Flow Branches).
- Configuración de CI para TFVC.
- Configuración de CI con Git.
- Políticas de Git Branch.

Fundamentos de Source Control



Realizar cambios en los archivos y mantener el historial

Capacidad para revertir cambios

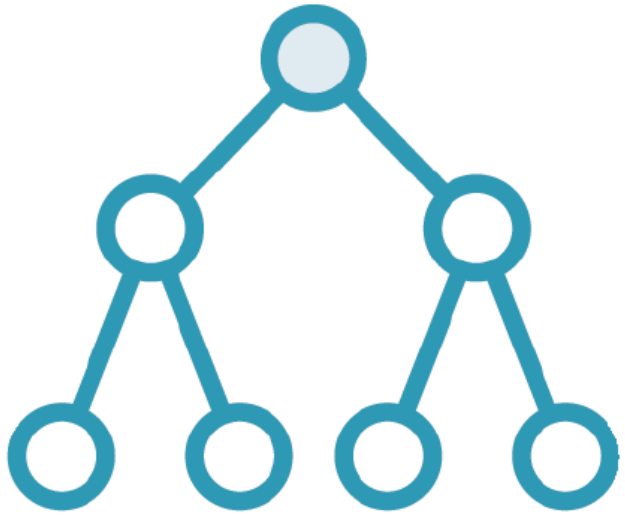
Capacidad para auditar cambios

Fundamentos para implementar entrega continua

Tipos de sistemas de control de fuente

Centralizado

Distribuido



Control de versiones (fundamentos de control de fuente y tipos de sistemas de control de fuente).

Intermedio

Tipos de sistemas de control de fuente - Centralizado

e.g. SVN, PVCS, Source Safe and Team Foundation Server

Fortalezas	Mejor uso
Escala a bases de código muy grandes	Grandes bases de código integradas
Control de permisos de nivel fino	Control y auditabilidad sobre el código fuente hasta el nivel del archivo
Permite el monitoreo de uso	
Posibilidad de bloquear archivos exclusivamente	Cuando la base de código tiene dificultades para el merge de tipos de archivos

Tipos de sistemas de control de fuente - Distribuido

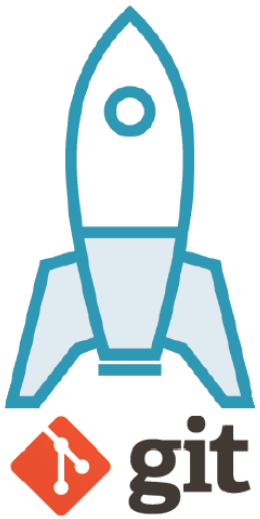
e.g. Mercurial, Git

Fortalezas	Mejor uso
Experiencia completa fuera de línea y velocidad	Bases de código modular
Repositorio completo con historial portátil	Integrando con código abierto
Soporte multiplataforma	Equipos altamente distribuidos
Uso creciente en el mercado	Bases de código portátiles entre plataformas.
Solicitudes de extracción para revisión de código	Nuevas bases de código

Tipos de sistemas de control de fuente - Distribuido

Team Foundation Version Control	Git
Creado por Microsoft	Creado por Linus Torvalds
Disponible desde comienzos de TFS	Versión inicial lanzada en Abril del 2005
Team Foundation Server 2005	Añadido a TFS en el 2013
Control de versión centralizado	Control de versión distribuida

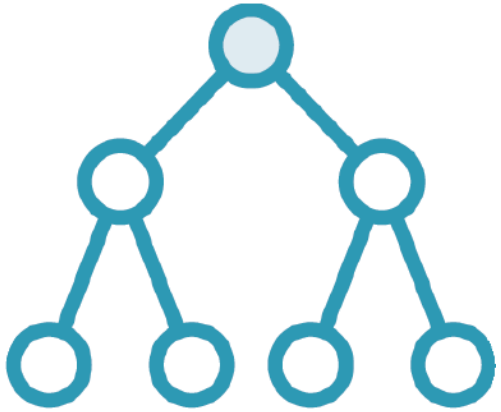
Operaciones básicas



- Clone
- Stage
- Commit
- Push
- Fetch
- Pull

Operaciones básicas

Git Branching

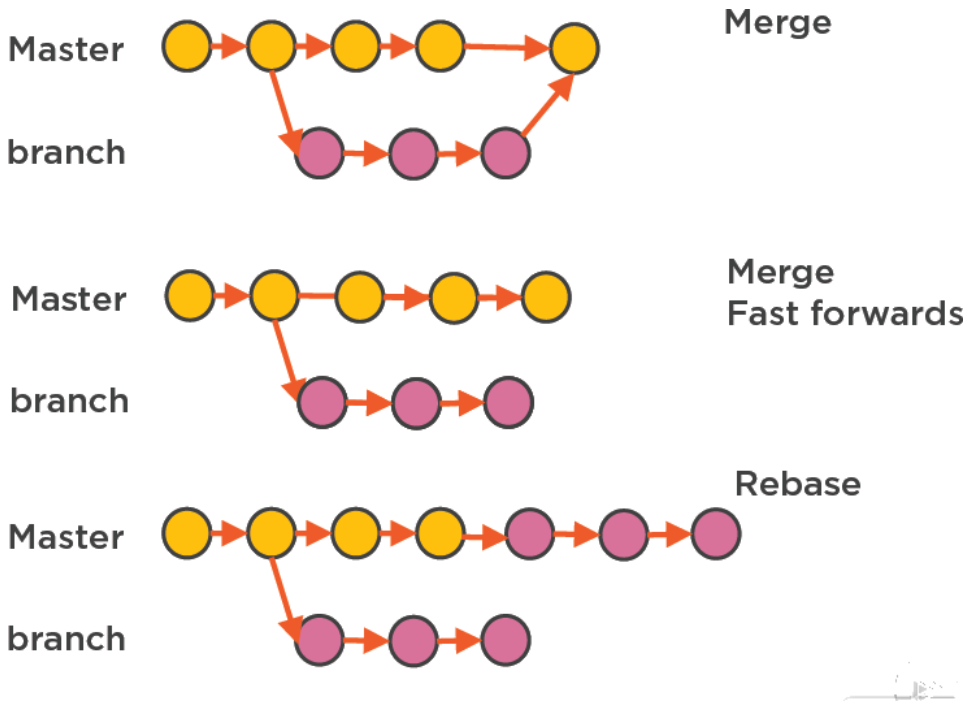


Master

Branch + Publish

Merge

Merge y Rebase



Git Branching

Estrategias de ramificación (Git Hub Flow, Git Flow, Git Flow Branches).

Intermedio

Git Hub Flow

El más adecuado para el despliegue continuo

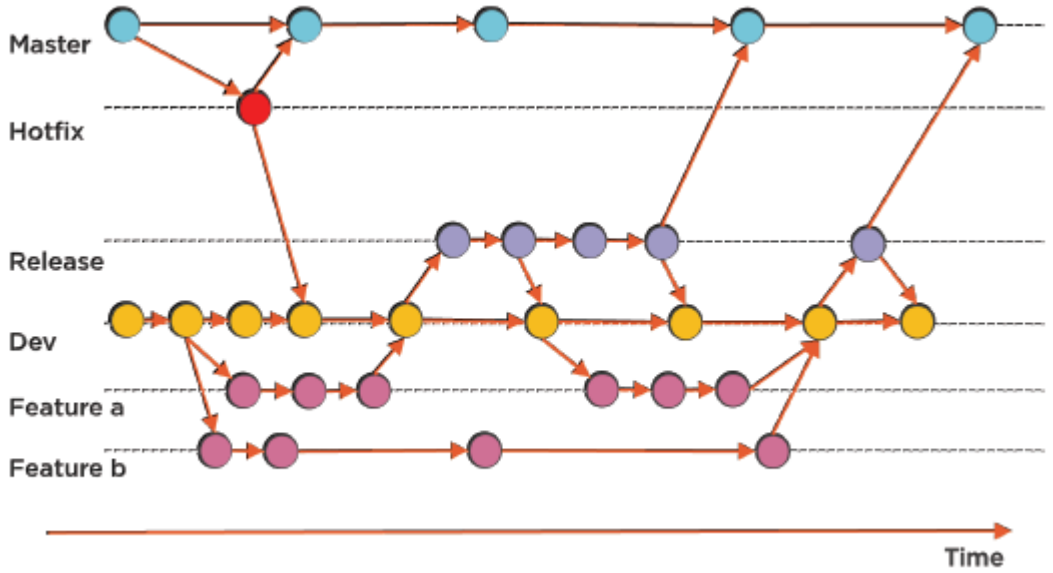
Se puede liberar varias veces al día.

Entrega de software como servicio (24x7)

Estrategias de ramificación (Git Hub Flow, Git Flow, Git Flow Branches).

Intermedio

Git Flow



Utilizado para entrega por etapas.

Libera cada iteración

Entrega de paquetes, bibliotecas, etc.

Estrategias de ramificación (Git Hub Flow, Git Flow, Git Flow Branches).

Intermedio

Git Flow

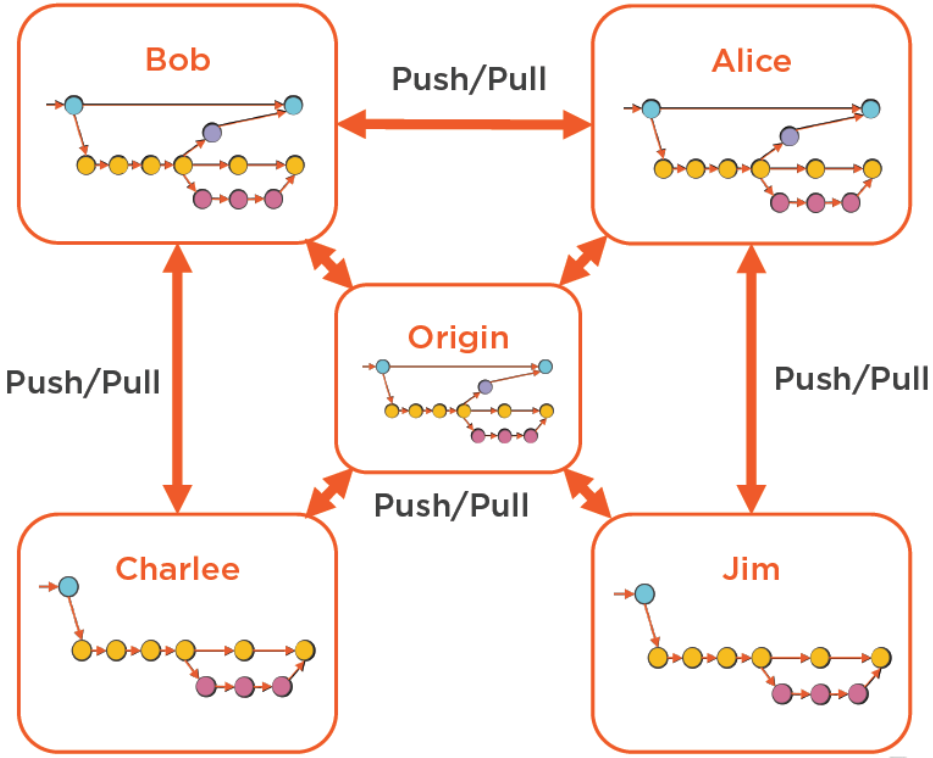
La estrategia de ramificación de Git Flow

Introducido en 2010 por Vincent Driessen

Use un modelo de servidor central, llamado Origen

Cada miembro del equipo trabaja en un clon

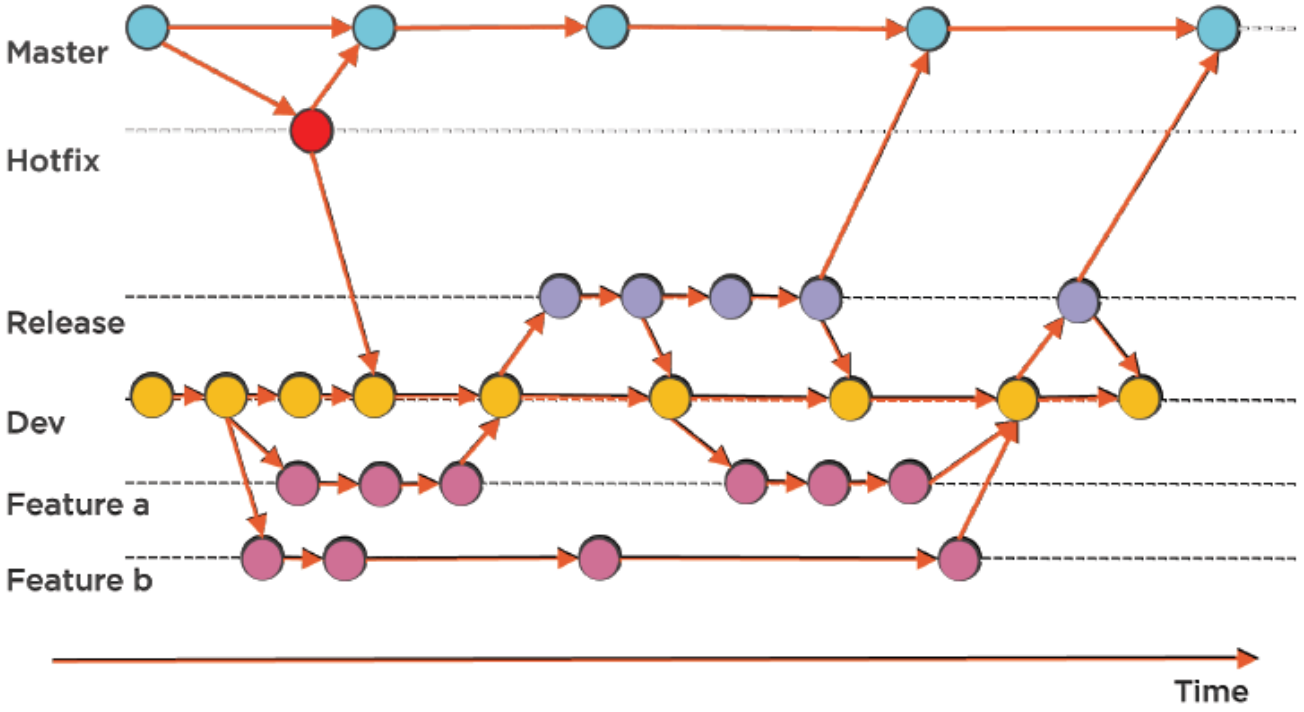
Los miembros del equipo pueden realizar push / pull de los cambios del equipo



Estrategias de ramificación (Git Hub Flow, Git Flow, Git Flow Branches).

Intermedio

Git Flow Branches



Definición de diferentes ramas y orientación clara sobre cuándo usar qué tipo de rama

Estrategias de ramificación (Git Hub Flow, Git Flow, Git Flow Branches).

Intermedio

Demo

Configuración de CI para TFVC.

"Continuous integration (CI) is the practice, in software engineering, of merging all developer working copies to a shared mainline several times a day"

Grady Booch (1991)

Configuración de CI para TFVC.

Configuración de CI con Git.

Políticas de Git Branch.


GALAXY
TRAINING