

# Administración y Control de Configuraciones

Control de Versiones

Integración Continua

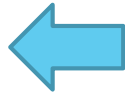
Plan de Administración y Control de Configuraciones

# Control de Versiones

Conceptos básicos

# CI (ítems bajo control de configuración)

CI



Cualquier **elemento** que esté bajo control de configuración en el **repositorio**

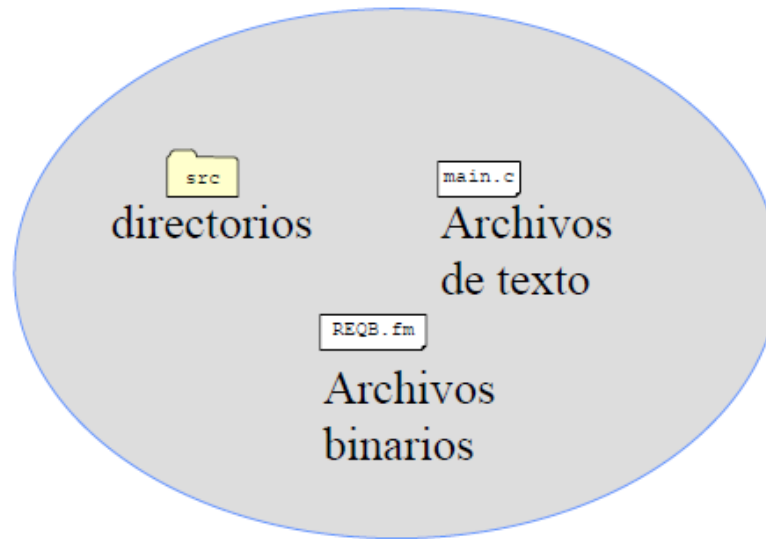
Versión

VOB

Vista

Etiqueta

Rama



# Versión

CI

Versión

VOB

Vista

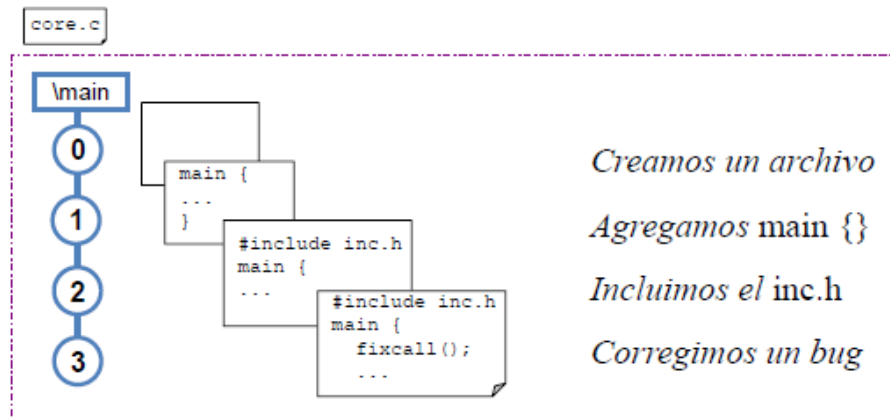
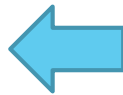
Etiqueta

Rama

Toda **evolución** de un CI (elemento)

n

Cuando un archivo o directorio **evoluciona**, una nueva versión es generada



# VOB (Version Object Base)

CI

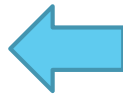
Versión

VOB

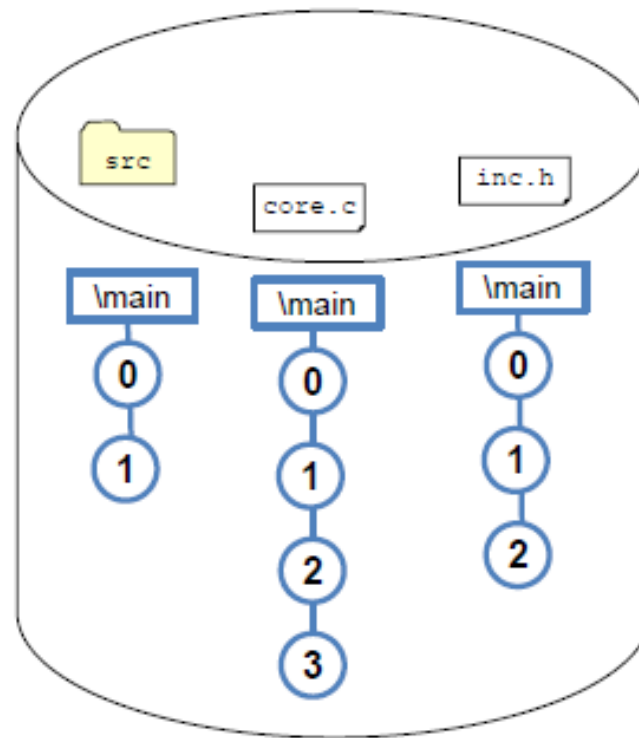
Vista

Etiqueta

Rama



Un VOB es un **repositorio** para guardar los CIs y sus versiones asociadas



# Vista

CI

Versión

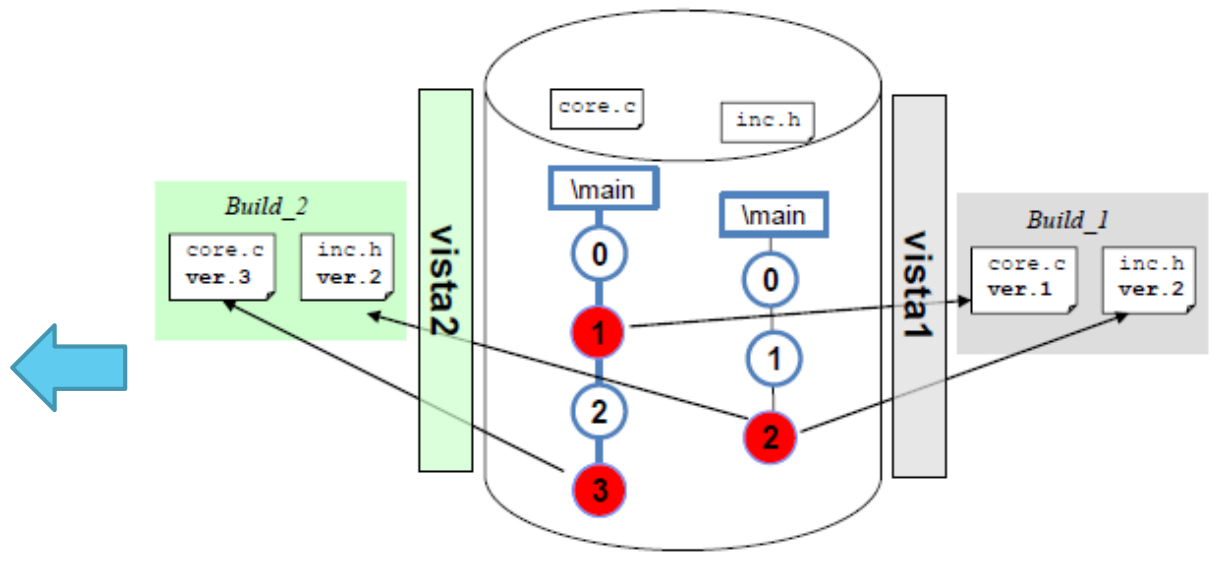
VOB

Vista

Etiqueta

Rama

Las vistas son **espacios virtuales** para trabajar con una **versión específica** de los CIs afectados



# Etiqueta

CI

Versión

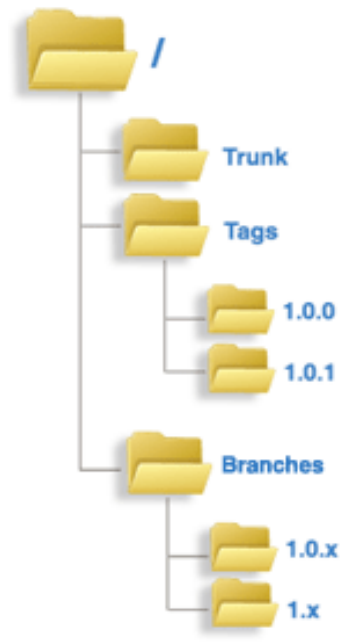
VOB

Vista

Etiqueta

Rama

Las etiquetas son **identificadores** pegados a las versiones de algunos CIs. Las etiquetas son parte del metadato del VOB, por ende deben existir antes de que se peguen a las versiones de los CIs.



# Rama

CI

Versión

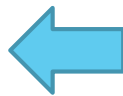
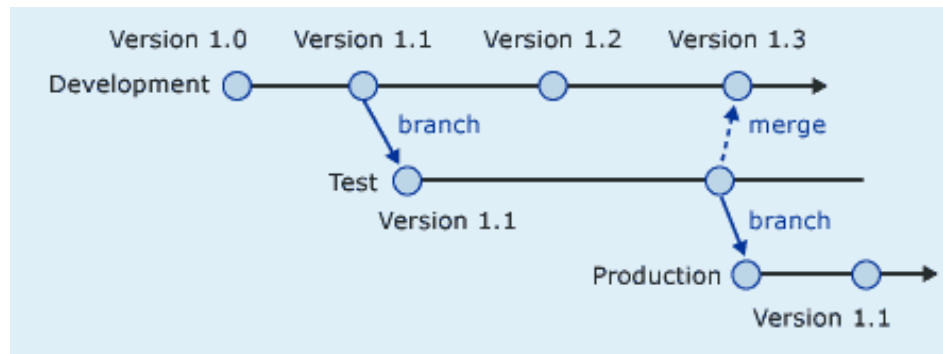
VOB

Vista

Etiqueta

Rama

Las ramas son caminos independientes de evolución de las versiones de los CIs que facilitan el desarrollo en paralelo.





# Práctica

# Acciones básicas

## Check-out

- Descargar todos los CIs de un repositorio para comenzar a trabajar.

## Check-in/Commit

- Agregar al repositorio nuevos CIs o modificaciones sobre los existentes.

## Rebase/Update

- Descargar la última versión de cada CI.

# Checkout

Repositorio  
Local



```
git clone /path/to/repository
```

Repositorio  
Remoto

```
git clone username@host:/path/to/repository
```

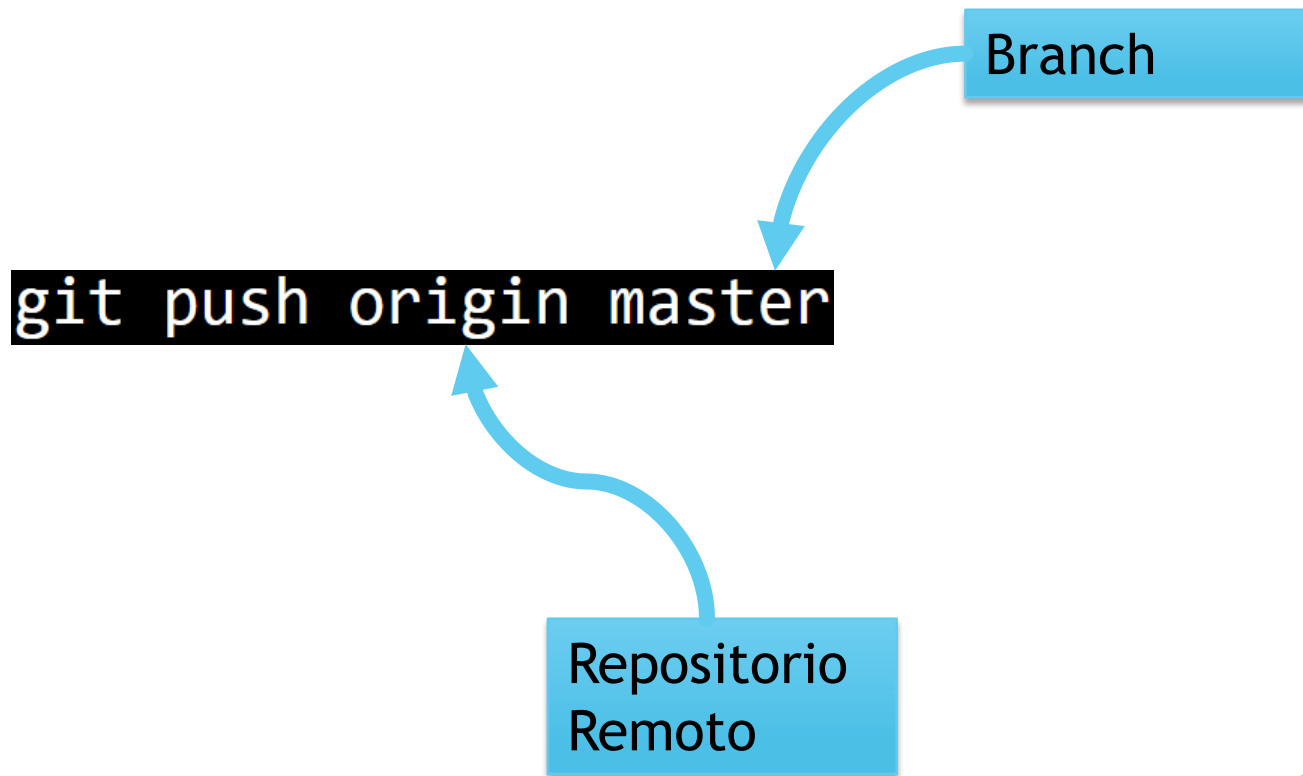
```
git remote add origin <server>
```

# Añadir archivos al repositorio local

```
git add <filename>    git add *
```

```
git commit -m "Commit message"
```

# Subiendo los cambios al repositorio remoto



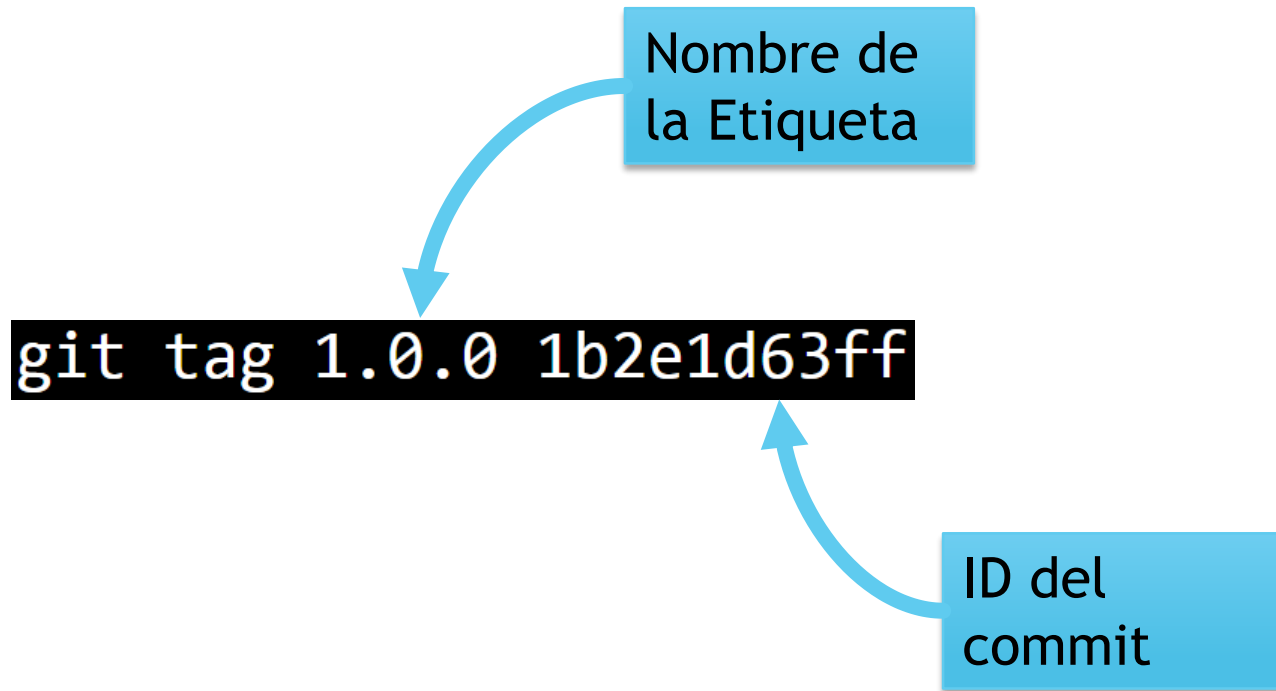
# Actualizando el repositorio

```
git fetch origin
```

```
git pull
```

```
git merge <branch>
```

# Creando etiquetas



# Integración Continua

The background of the slide features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side and bottom of the frame, creating a modern, dynamic aesthetic.



# Enfoque tradicional



## El lenguaje del poder: la cultura y la negociación de conflictos

# Problemas Enfoque Tradicional

Los problemas que surgen al momento de integración no son detectados por el testing de los módulos antes de la integración.

Los errores son detectados semanas o meses después de que son introducidos en el sistema.

Cuando se encuentra un error es muy difícil determinar el módulo que lo causa.

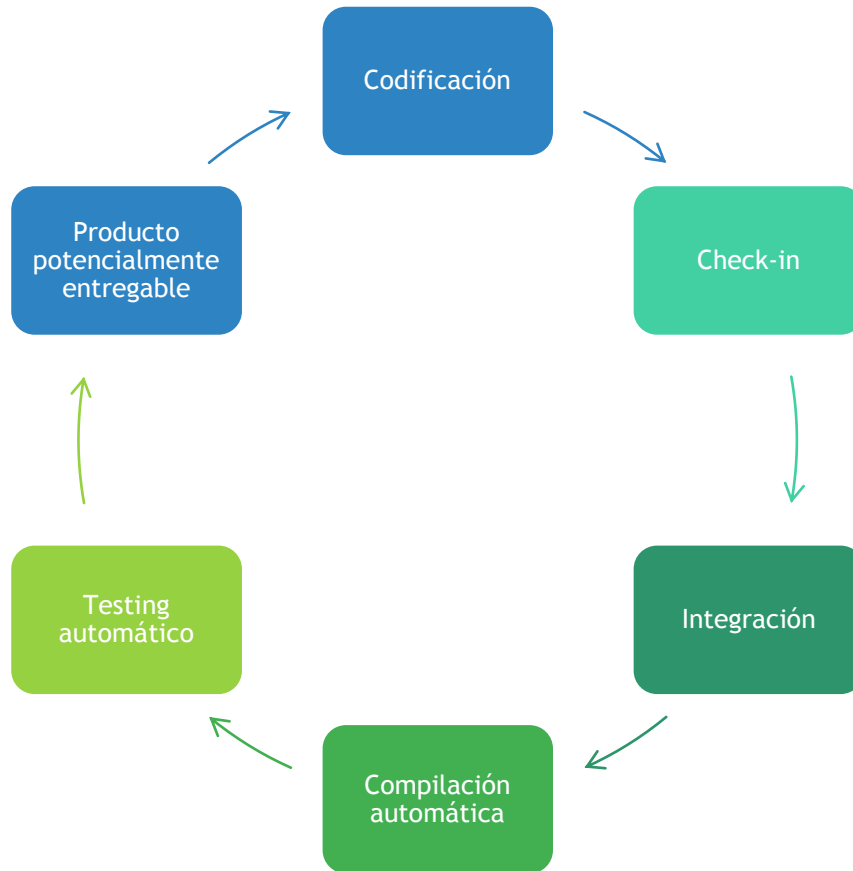
# Problemas Enfoque Tradicional

El proceso de build es poco frecuente y usualmente lleva a un infierno de integración, existe gran cantidad de conflictos y nadie conoce la causa.

La integración tardía lleva al descubrimiento de errores cerca de la fecha de release.

No es posible tener un producto potencialmente entregable antes de la etapa de integración.

# Integración Continua



# Fundamentos de la Integración Continua

Scripts de compilación automatizados.

Código auto-testeable (tests automatizados)

Check-Ins frecuentes.

Único punto de acceso al código fuente.

# Integración Continua

---

## Beneficios

Los desarrolladores integran su código una vez al día o aún más frecuentemente.

---

Cada integración es automáticamente compilada y testeada.

---

El resultado de la compilación y de la ejecución de los tests es notificado a todos automáticamente.

---

El estado del código fuente siempre es conocido.

---

Los conflictos de integración son simples de resolver debido al poco tiempo entre integraciones.

---

Los desarrolladores están permanentemente al tanto del código integrado por los demás.

---

# Software Configuration Management Plan

# Contenido Básico del Plan de SCM

## ► Introducción

### ► Propósito del Plan

- ¿Por qué hacemos este plan?
- ¿Para quién hacemos el plan?

### ► Alcance

- Descripción del proyecto.
- Ítems de software que estarán bajo control de configuración.
- Otro software que podría tener relación con el plan.
- Limitaciones/Restricciones.
- Supuestos.



# Contenido Básico del Plan de SCM

## ► Gestión de SCM

### ► Organización

- Áreas de la organización que participan en las actividades del proyecto.
- Rol de cada área de la organización en la estructura del proyecto.
- Relación entre las áreas de la organización.

### ► Responsabilidades

- Asignar las tareas/actividades del plan de SCM al área funcional correspondiente.

# Contenido Básico del Plan de SCM

## ▶ Actividades de SCM

### ▶ Identificar los Configuration Items.

- ▶ ¿Qué estructura se usará?
- ▶ ¿Nombre de los elementos?
- ▶ ¿Dónde se ubicarán físicamente los elementos?

### ▶ Control de la configuración (Change Control Board).

- ▶ ¿Quiénes participan en la CCB?
- ▶ ¿Cómo solicitar cambios en el proyecto?
- ▶ Proceso de evaluación de los cambios solicitados.
- ▶ Aprobación o desaprobación de los cambios solicitados.
- ▶ Implementación de los cambios solicitados.

# Contenido Básico del Plan de SCM

- ▶ Registro de las actividades de SCM
  - ▶ ¿Qué se debe registrar?
  - ▶ ¿Cómo se consigue la información a registrar?
  - ▶ ¿Dónde guardar la información registrada?
  - ▶ ¿Qué reportes se deben hacer de la información registrada?
    - ▶ ¿En qué formato?
    - ▶ ¿Con que frecuencia?
  - ▶ Herramientas que automaticen el registro de las actividades.

# Contenido Básico del Plan de SCM

## ▶ Administración del código fuente

- ▶ Rama principal (trunk).
- ▶ Branches.
  - ▶ Esquema.
  - ▶ Criterios de calidad.
- ▶ Especificación de las políticas de merging.
- ▶ Tags.
  - ▶ ¿Cuándo y de qué hacer tags?
- ▶ Build Management
  - ▶ Tipos de builds.
  - ▶ Integración continua.
- ▶ Release Management
  - ▶ ¿Cómo hacer el build para release?
  - ▶ ¿Qué compone el build de release?