

# Desarrollo de Aplicaciones Web Empresariales

Exequiel Fuentes Lettura  
exequiel.fuentes@ucn.cl

# Información de contacto

- Exequiel Fuentes Lettura
  - Email: [exequiel.fuentes@ucn.cl](mailto:exequiel.fuentes@ucn.cl)
  - Horario de Atención: Jueves y Viernes, bloque C
- Departamento de Ingeniería de Sistemas y Computación
  - Oficina: Y1 - 329
  - <http://www.disc.ucn.cl>

# Control de versiones

- Registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo.
- Cualquier tipo de archivo puede ponerse bajo control de versiones.
- Permite:
  - Revertir archivos a un estado anterior.
  - Revertir el proyecto entero a un estado anterior.
  - Comparar cambios a lo largo del tiempo.
  - Ver quién modificó por última vez algo que puede estar causando un problema.
  - Quién introdujo un error y cuándo.
  - Recuperar archivos si fueron arruinados o perdidos.
  - Entre otras funcionalidades.

# Alternativas

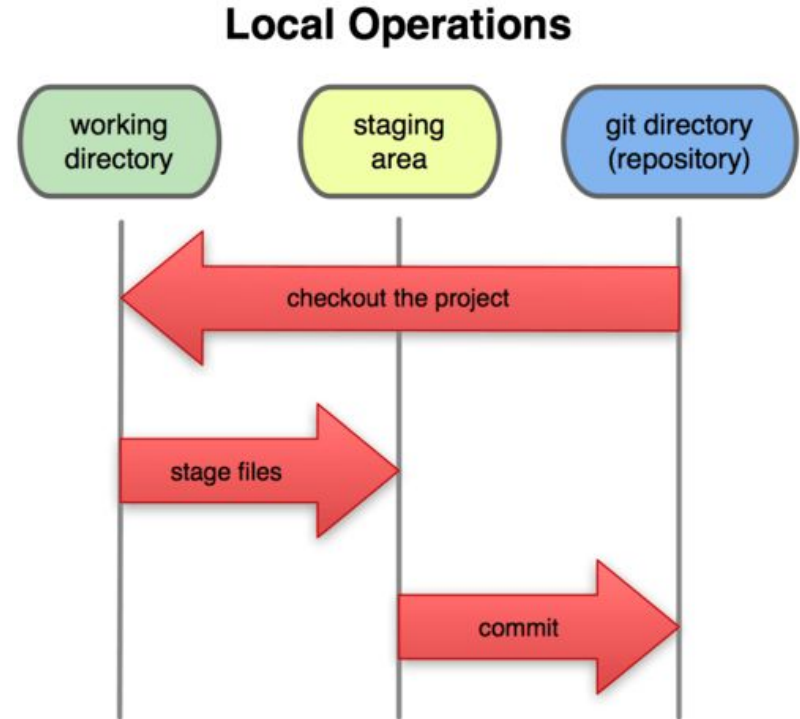
- Algunos de los sistemas de control de versiones más famosos son Subversion (también conocido como Svn), Git y Mercurial.
- Subversion (<https://subversion.apache.org/>):
  - Sistema legendario, su uso ha ido decreciendo con el paso de los años. Todavía hay muchas empresas que lo usan.
  - Usa un sistema centralizado, esquema cliente-servidor.
  - Falla cuando se quiere unir cambios para archivos que han sido renombrados.
  - No se puede subir cambios si se está desconectado.
- Mercurial (<https://www.mercurial-scm.org/>):
  - Usa un sistema distribuido.
  - Se caracteriza por ser fácil de usar, sus comandos son simples.
  - Escrito en su mayoría en Python.

# Git

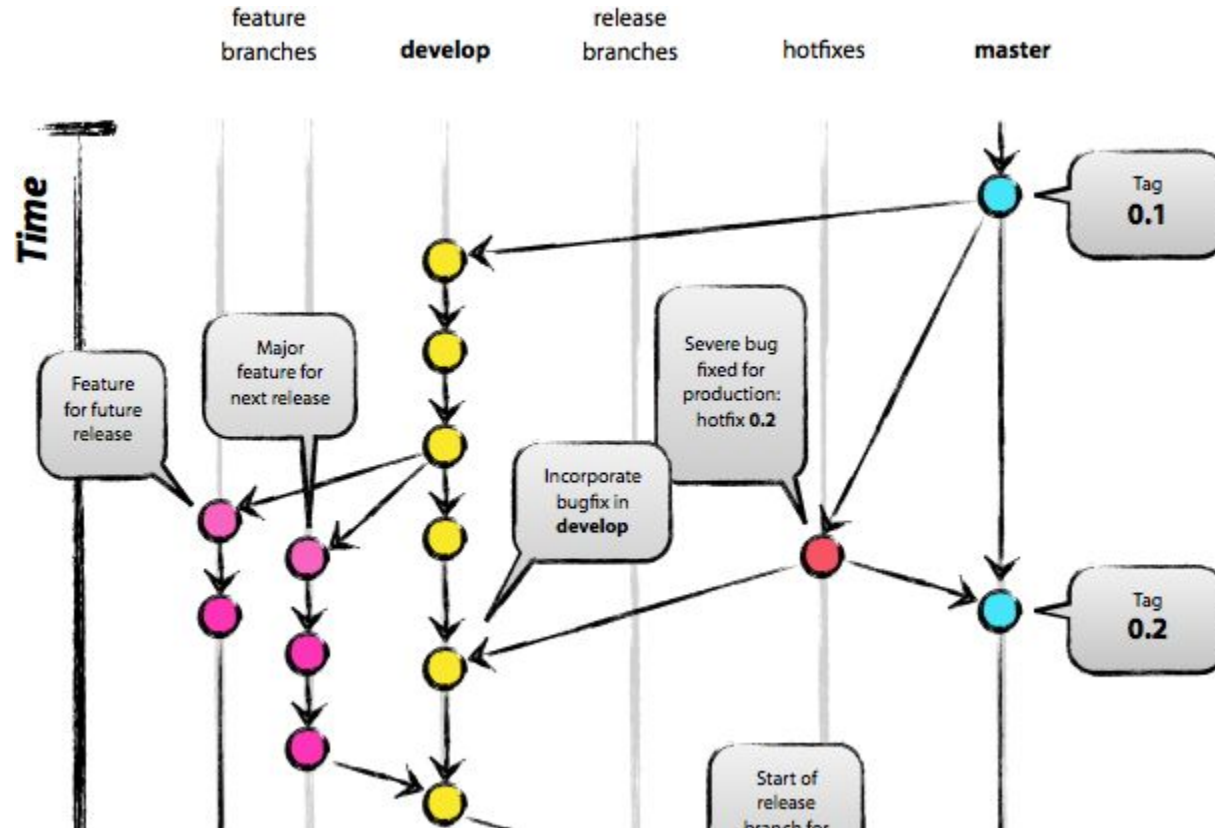
- Fue desarrollado por Linus Torvalds, el mismo padre del kernel Linux, en el año 2005.
- La mayoría de las operaciones en Git sólo necesitan archivos y recursos locales para operar.
- No pierde información durante su transmisión o sufre corrupción de archivos.
- Git guarda todo no por nombre de archivo, sino por el valor hash de sus contenidos: 24b9da6552252987aa493b52f8696cd6d3b00373
- Git tiene tres estados principales en los que se pueden encontrar los archivos: modificado (modified), preparado (staged) y confirmado (committed).

# Git

- **Modified:** archivo modificado, pero todavía no confirmado en la base de datos.
- **Staged:** marcar un archivo modificado para que vaya a la próxima confirmación.
- **Committed:** datos almacenados en la base de datos local.



# Git



# Git

- Inicializando un repositorio en un directorio existente:

```
1 | $> git init
```

- Controlando versiones de archivos existentes:

```
1 | $> git add *.c  
2 | $> git add README  
3 | $> git commit -m 'Version inicial del proyecto'
```

- Clonando un repositorio remoto:

```
1 | $> git clone https://gitlab.com/efuleet/arbol_binario.git
```



# Git

- Comprobando el estado de los archivos:

```
1 | $> git status
```

- Deshaciendo la modificación de un archivo:

```
1 | $> git checkout -- README.md
```

- Subiendo cambios al repositorio remoto:

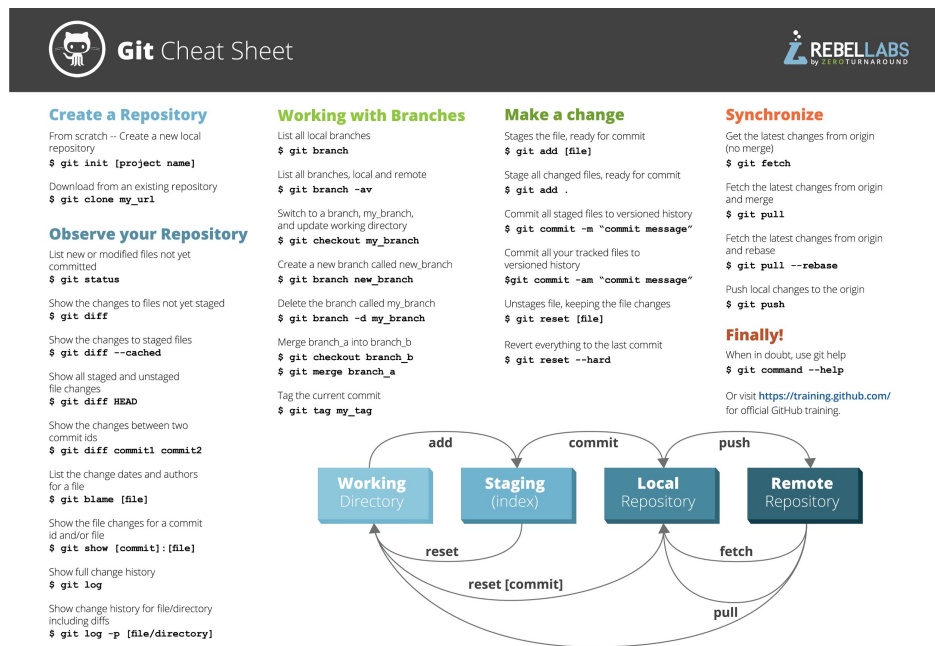
```
1 | $> git push origin master
```

- Viendo el histórico:

```
1 | $> git log
```

# Git

- Más información acerca de Git: <https://git-scm.com/book/es/v1>



# Git

- En este curso se utilizará Bitbucket o GitLab dependiendo de la cantidad de alumnos (buscamos un repositorio gratuito).
- En cualquier caso, se debe crear una cuenta (si no la tiene) y enviar los datos al profesor para ser agregado al proyecto.



# Ejemplo simple

- Historia de usuario: Árbol binario.
  - Como desarrollador, quiero implementar un árbol binario. Condiciones de aceptación:
    - Un árbol puede estar vacío.
    - Un nodo puede tener un hijo izquierdo y un hijo derecho.
    - Un nodo padre no puede tener más de dos hijos.
    - Acepta sólo enteros numéricos.
    - Se puede recorrer en profundidad: preorden, postorden y inorden.
- Realizar:
  - Descomponer la historia de usuario en tareas.
  - Asignar tareas a grupos de trabajo.
  - Diseñar e implementar la solución.
  - Subir los cambios a un repositorio.
  - Integrar y verificar solución.

# ¿Preguntas?