



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

# Controlando Versiones con Git

## Clase Práctica

Daniel Gacitúa Vásquez

Comunidad GNU/Linux USACH

Enero de 2016



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

# Temas a tratar

1 Introducción

2 Instalación

3 Uso de Git

4 Conclusión



# ¿Que es Git?

- Git fue creado en 2005 por Linus Torvalds como un Sistema Distribuido de Control de Versiones

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



# ¿Que es Git?

- Git fue creado en 2005 por Linus Torvalds como un Sistema Distribuido de Control de Versiones
- Git fue concebido originalmente para versionar el Kernel de GNU/Linux

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



# ¿Que es Git?

- Git fue creado en 2005 por Linus Torvalds como un Sistema Distribuido de Control de Versiones
- Git fue concebido originalmente para versionar el Kernel de GNU/Linux
- Git sirve para tener snapshots de cada versión editada del código

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



# ¿Que es Git?

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

- Git fue creado en 2005 por Linus Torvalds como un Sistema Distribuido de Control de Versiones
- Git fue concebido originalmente para versionar el Kernel de GNU/Linux
- Git sirve para tener snapshots de cada versión editada del código
- Git permite a varios programadores colaborar de forma ordenada a la generación de código



# ¿Cómo funciona Git? [1/2]

- Git se basa en el Modelo Cliente-Servidor para alojar repositorios de código

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



# ¿Cómo funciona Git? [1/2]

- Git se basa en el Modelo Cliente-Servidor para alojar repositorios de código
- El usuario tiene un repositorio local el cual aloja (dentro de su PC) diferentes *snapshots* de su código

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión





# ¿Cómo funciona Git? [1/2]

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

- Git se basa en el Modelo Cliente-Servidor para alojar repositorios de código
- El usuario tiene un repositorio local el cual aloja (dentro de su PC) diferentes *snapshots* de su código
- El usuario sube a un repositorio remoto (servidor de Git) sus *snapshots* para ser distribuidas a otros usuarios



# ¿Cómo funciona Git? [2/2]

- Ejemplos de servidores (públicos) de Git son: *BitBucket* y *GitHub*

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



## ¿Cómo funciona Git? [2/2]

- Ejemplos de servidores (públicos) de Git son: *BitBucket* y *GitHub*
- Los *snapshots* son conocidos en git como *commits*, se identifican por una secuencia de caracteres alfanuméricos denominada *hash*

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

# Instalación Inicial

Debian / Ubuntu / Linux Mint

```
sudo apt-get install git kdiff3-qt
```

Windows

Usar el *complete installer* de GitExtensions:

<https://gitextensions.github.io/>



# Configuración en Linux [1/3]

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

## Establecer Nombre, Email y Mergetool

```
git config --global user.name "Juan Pérez"  
git config --global user.email juan.perez@usach.cl  
git config --global merge.tool kdiff3
```



# Configuración en Linux [2/3]

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

## Configurar Llave SSH [1/2]

```
ssh-keygen -t rsa -b 4096 -C "juan.perez@usach.cl"  
eval "$(ssh-agent -s)"  
ssh-add ~/.ssh/id_rsa  
cat ~/.ssh/id_rsa.pub
```



# Configuración en Linux [3/3]

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

## Configurar Llave SSH [2/2]

- Copiar la salida del comando `cat`, esta es la *llave pública*, la cual debe ser pegada en el servidor de git a usar (BitBucket, GitHub, etc)



# Esquema básico de Git

Controlando  
Versiones con  
Git

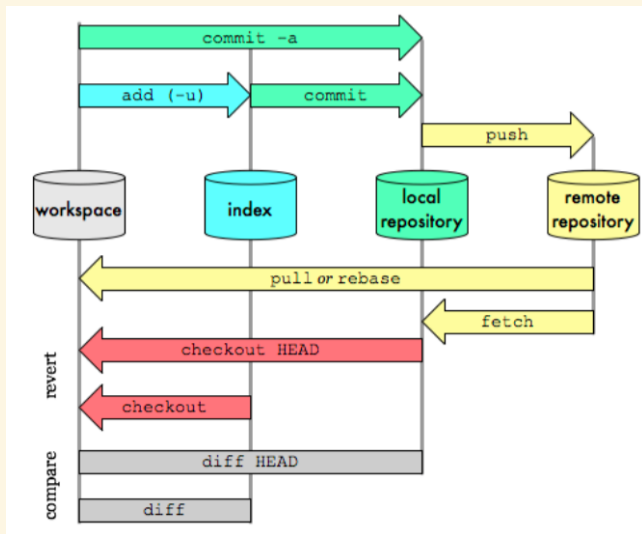
Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión







# Zonas de Trabajo

- **Workspace (o Directorio Local):** Es el directorio sobre el cual trabajamos nuestro código

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



# Zonas de Trabajo

- **Workspace (o Directorio Local):** Es el directorio sobre el cual trabajamos nuestro código
- **Index (o Índice):** Mantiene una copia observada de los archivos de código editados

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



# Zonas de Trabajo

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

- **Workspace (o Directorio Local):** Es el directorio sobre el cual trabajamos nuestro código
- **Index (o Índice):** Mantiene una copia observada de los archivos de código editados
- **Local Repository (o Repo Local):** Agrupa todas las copias observadas del Index en commits



# Zonas de Trabajo

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

- **Workspace (o Directorio Local):** Es el directorio sobre el cual trabajamos nuestro código
- **Index (o Índice):** Mantiene una copia observada de los archivos de código editados
- **Local Repository (o Repo Local):** Agrupa todas las copias observadas del Index en commits
- **Remote Repository (o Repo Remoto):** Almacena todos los commits transferidos desde el repo remoto y los distribuye a otros usuarios



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

# Comandos Básicos de Git [1/4]

## Iniciar Repo

```
git init
```

## Añadir Repo Remoto

```
git remote add origin [URL]
```

## Clonar Repo Remoto

```
git clone [URL]
```



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

# Comandos Básicos de Git [2/4]

## Tirar cambios del Repo Remoto

```
git pull [REMOTO] [RAMA]
```

## Añadir un archivo al Index

```
git add [ARCHIVO]
```

## Añadir todos los archivos al Index

```
git add --all
```



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

## Comandos Básicos de Git [3/4]

Crear un *commit* con todos los archivos del Index

```
git commit -m "[MENSAJE]"
```

Empujar todos los *commits* al Repo Remoto

```
git push [REMOTO] [RAMA]
```



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

# Comandos Básicos de Git [4/4]

## Ver el estado actual del Repo Local

```
git status
```

## Ver la lista de *commits* recientes

```
git log
```





# Comandos Avanzados de Git [1/2]

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

## Resetear el Repo Local al último *commit*

```
git reset --hard HEAD
```

## Solucionar manualmente conflictos de fusionado (*merge*)

```
git mergetool
```



Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión

## Comandos Avanzados de Git [2/2]

Crear un nuevo *commit* reversando los cambios de otro

```
git revert [HASH]
```

Reversar últimos 3 commits guardando cambios en Index

```
git revert --no-commit HEAD~3..HEAD
```



# Conclusión

- Git es una herramienta eficaz para que múltiples usuarios puedan modificar simultánea y asincrónamente un repositorio de código

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



# Gracias

## ¿Preguntas?

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión



## Más información

Recuerda bajar nuestro Manual y Cheatsheet en:  
<http://cglusach.github.io/manualgit/>

Controlando  
Versiones con  
Git

Daniel  
Gacitúa  
Vásquez

Introducción

Instalación

Uso de Git

Conclusión