Grupo de Data Minning UTN-FRC

http://getyatel.com.ar

Basado en una charla de
Emilio Ramirez emilio@dot3.com.ar
De Unicode 2011
http://uniconf.com.ar/

Agenda

- Estructura de un proyecto
- Problema: Control de versionado manual
- Solución: Sistema de control de versiones
- Que hace básicamente un Sistema de control de versiones
- Sistemas Centralizados y Distribuidos
- Sistemas Distribuidos: ¿Por qué Mercurial?
- Comandos Básicos
- Ejemplos simples
- Workflows

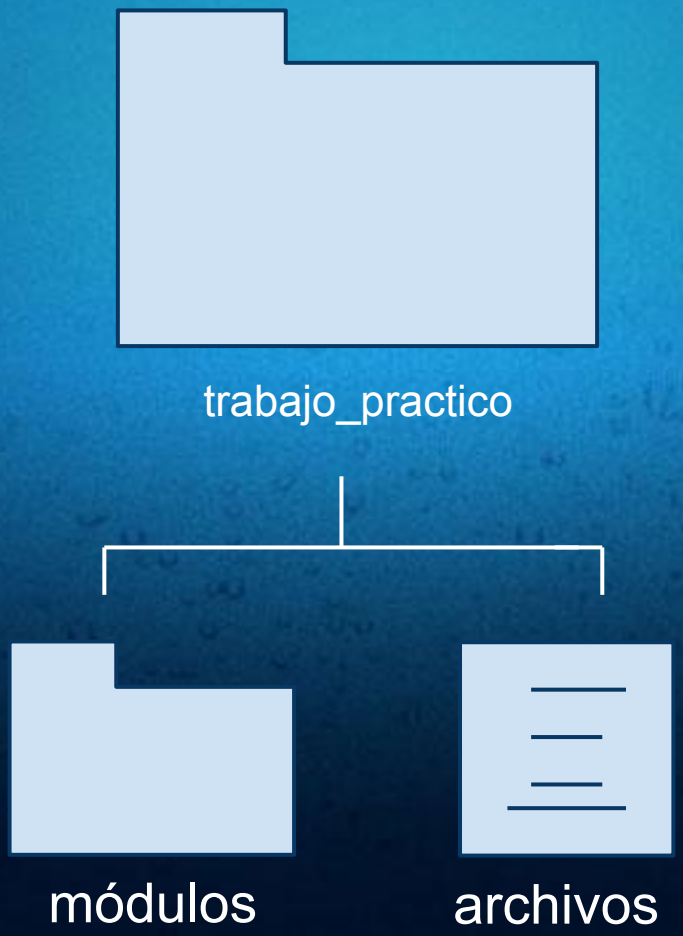
Principios fundamentales

- La trazabilidad del trabajo es una necesidad, no un capricho.
- CVS son simples (no fáciles)
- El esfuerzo de utilizarlos es infinitamente menor que el esfuerzo de versionar manualmente.
- No buscan que NO haya conflictos.
- Ordena el trabajo de personas ordenadas.
- De nuevo.. ordena trabajo, no personas.

Comenzamos un Proyecto

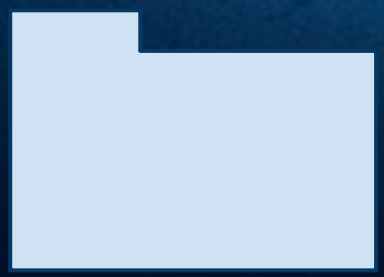
- Directorios
- Código Fuente
- Binarios
- Imagenes
- Etc...

Tiene una estructura

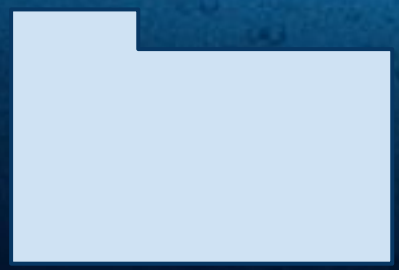


Crecimiento del proyecto

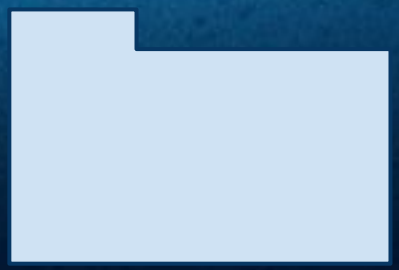
A medida que logramos código bueno, código limpio y que cumple con los objetivos. Necesitamos guardarlo.



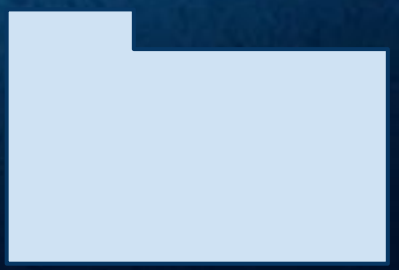
trabajo_practico



trabajo_practico_1



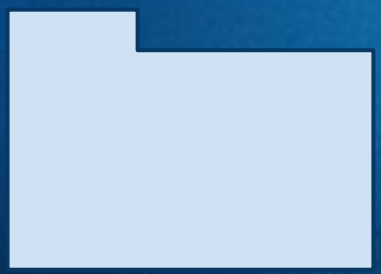
trabajo_practico_2



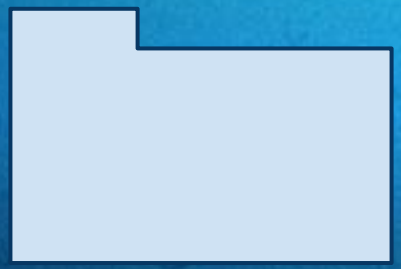
trabajo_practico_3

<sarcasmo>Se puede poner mas lindo</sarcasmo>

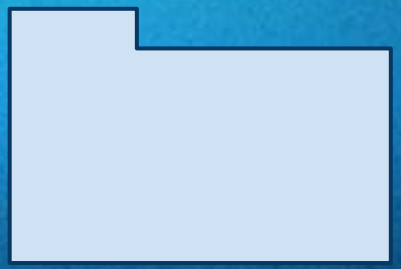
Con cosas como:



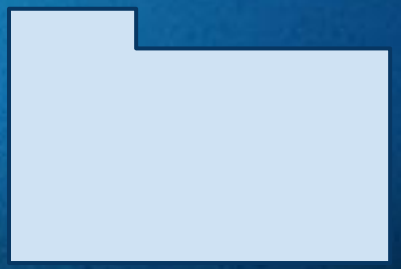
trabajo_practico



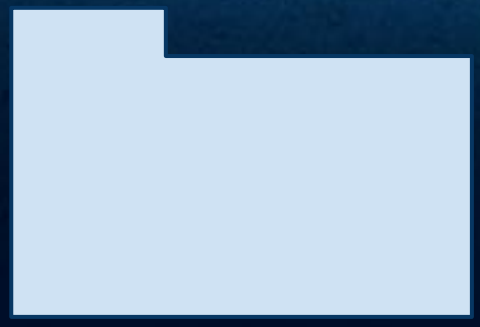
trabajo_practico_1



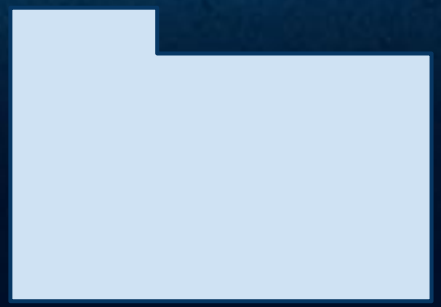
trabajo_practico_2



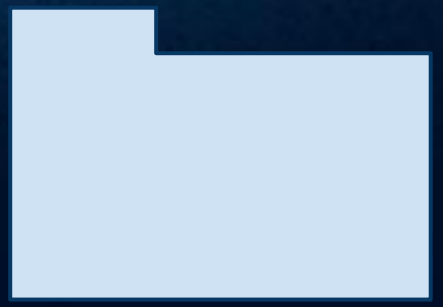
trabajo_practico_3



trabajo_practico_1 Anda



trabajo_practico_2
Funca casi todo



trabajo_practico_3 Sirve
para la entrega

¿Estamos haciendo Control de Versiones?

SÍ

Control de Versiones Manual

¿Cuál es el problema entonces?

- Repetición de código
- Espacio en disco (repetición de archivos)
- Confusión (no tengo información)
- No hay historial de cambios
- No hay recuperación por equivocaciones
- No posibilitó el Trabajo en Equipo
- Por nombrar algunas...

¿Solución?

¿Para qué preocuparme del trabajo de seguimiento y mantenimientos de cambios si lo puede hacer un software?

Sistemas de control de versiones

Sistema de Control de Versiones

Básicamente debe proveer:

- Mecanismo de almacenamiento de los elementos que deba gestionar. (ej. archivos de texto, imágenes, documentación...)
- Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos)
- Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto)
- Posibilitar el trabajo en equipo sobre el proyecto
- Mecanismo de vuelta atrás a un estado del proyecto

Esto que quiere decir...?

Controla y registra (lleva un historial):

- Quien hizo un cambio
- Cuando lo hizo
- Qué archivos se modificaron
- Cuales son esos cambios
- Se guarda un estado de como esta el repositorio.

Es como una fotografía del proyecto en algún momento

Aclaración

Ninguna herramienta de control de versiones puede salvar un proyecto mal administrado.

Pero la elección de herramientas puede hacer una gran diferencia en la fluidez con la cual usted puede trabajar en un proyecto.

Conceptos basicos

Repositorio: es el lugar en el que se almacenan los datos actualizados e históricos, a menudo en un servidor.

Commit: es cuando una copia de los cambios hechos a una copia local es guardada o integrada sobre repositorio.

Checkout: copia de trabajo local desde el repositorio.

Changelist/Change set: una lista de cambios identifica el conjunto de cambios hechos en un único commit.

Branch: es una rama, significa que desde ese momento en adelante, dos copias de esos ficheros puedan ser desarrolladas a diferentes velocidades o de diferentes formas, de modo independiente.

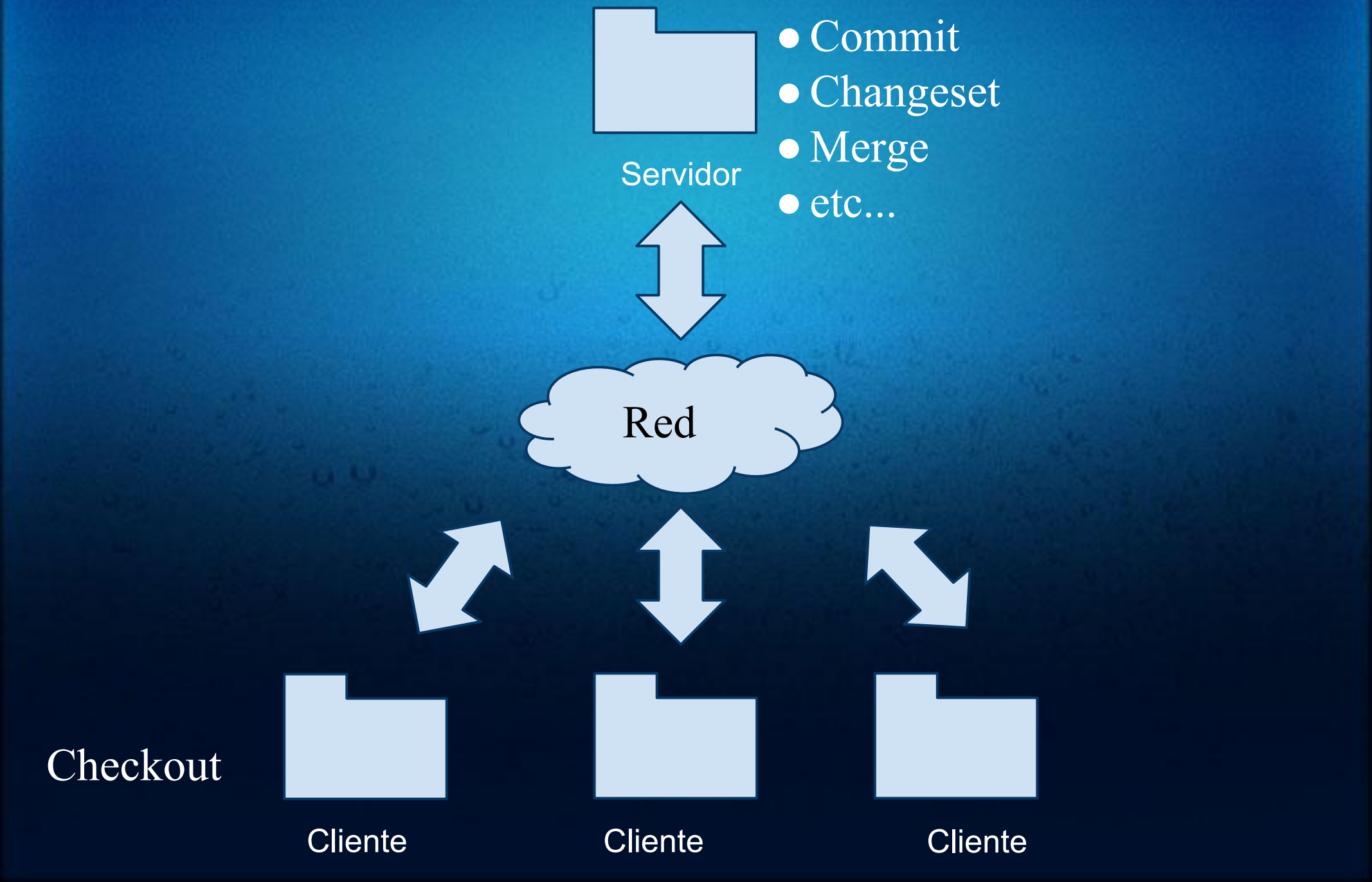
Merge: Una integración o fusión une dos conjuntos de cambios sobre un fichero o un conjunto de ficheros en una revisión unificada de dicho fichero o ficheros.

Existen dos arquitecturas

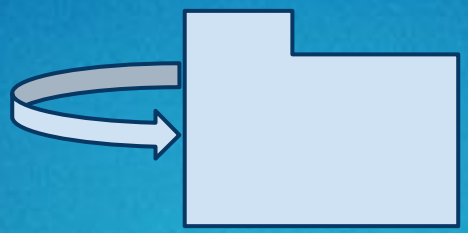
Sistemas centralizados

Sistemas distribuidos

Sistema Centralizado



Sistema Distribuido



Servidor

- Commit
- Update
- Merge
- Changeset

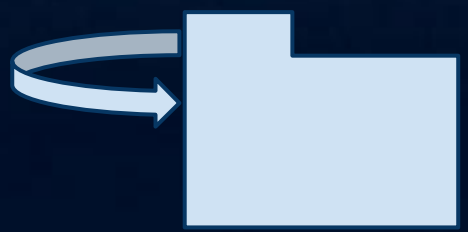
push

pull

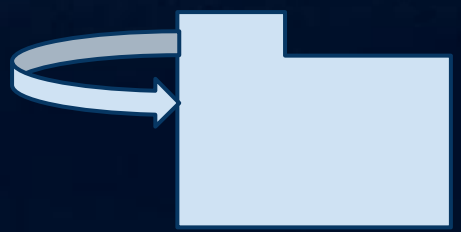
No solo una red

Clone

Servidor



Servidor



Servidor

- Commit
- Update
- Merge
- Changeset

Ventajas de un Sistema Distribuido

- Desarrollo individual.
- Fácil compartición.
- Confiabilidad (no es dependiente de una red en sentido estricto).
- Los metadatos están en todos los clones..
- Es rápido (se interactúa mucho con el repositorio).
- Replicación (cada clon es una copia completa del repo).

¿Porque Mercurial?

- Porque es distribuido
- Es fácil de aprender
- Es fácil de usar
- Es liviano
- Escala de forma excelente
- Es fácil de acondicionar
- Están saliendo varias publicaciones de porque es lo mejor que hay.
- Esta hecho en Python

Comandos básicos

hg init

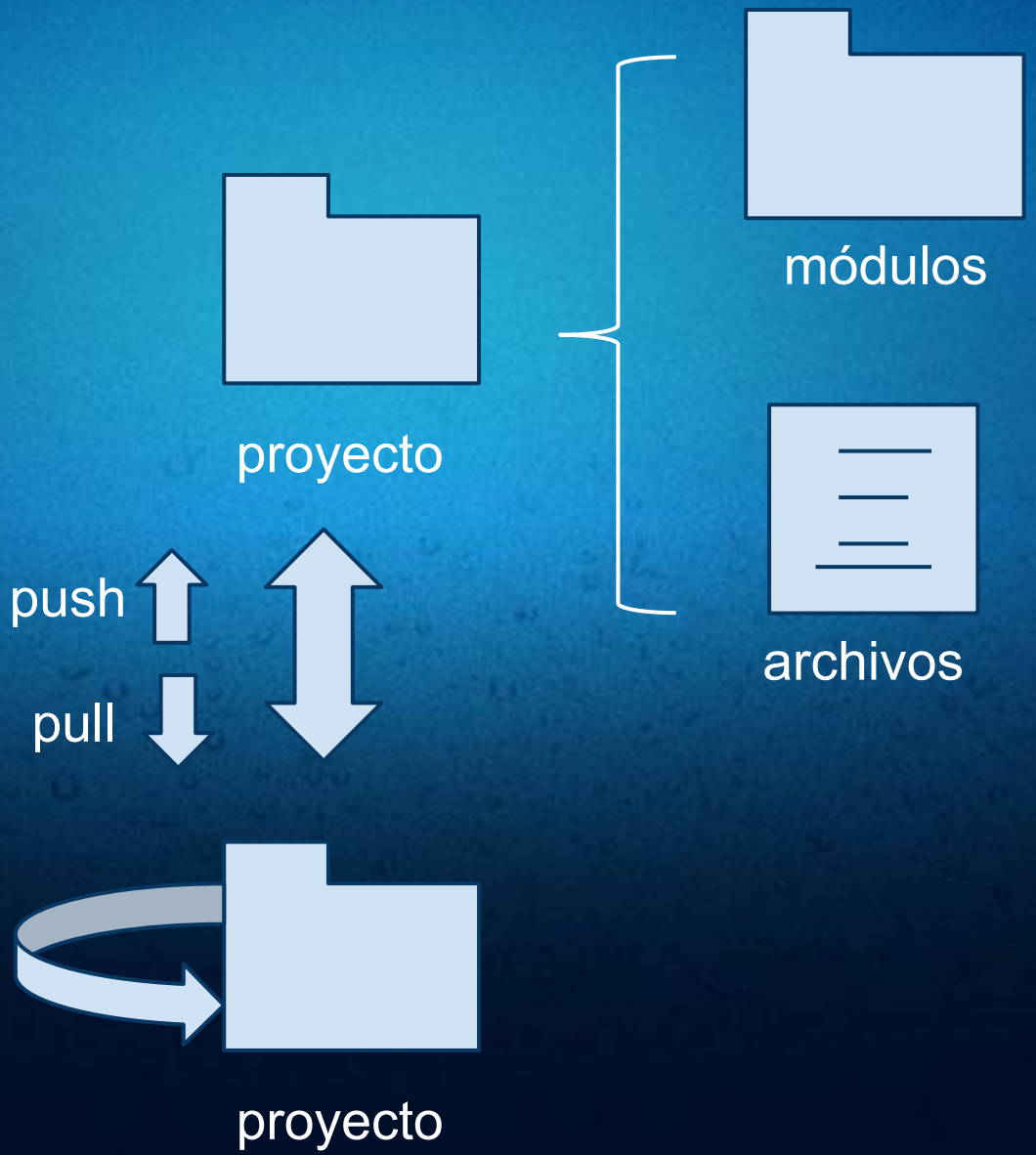
hg add

hg clone

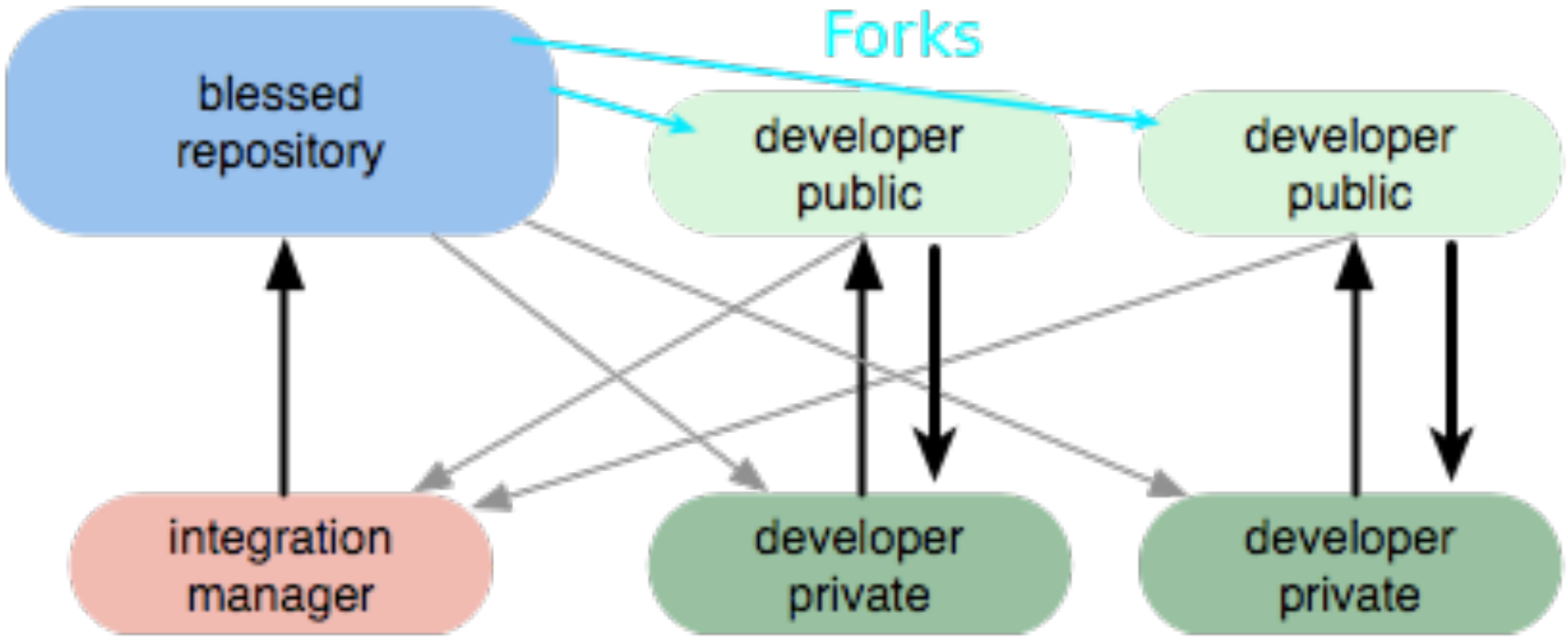
hg commit

hg pull

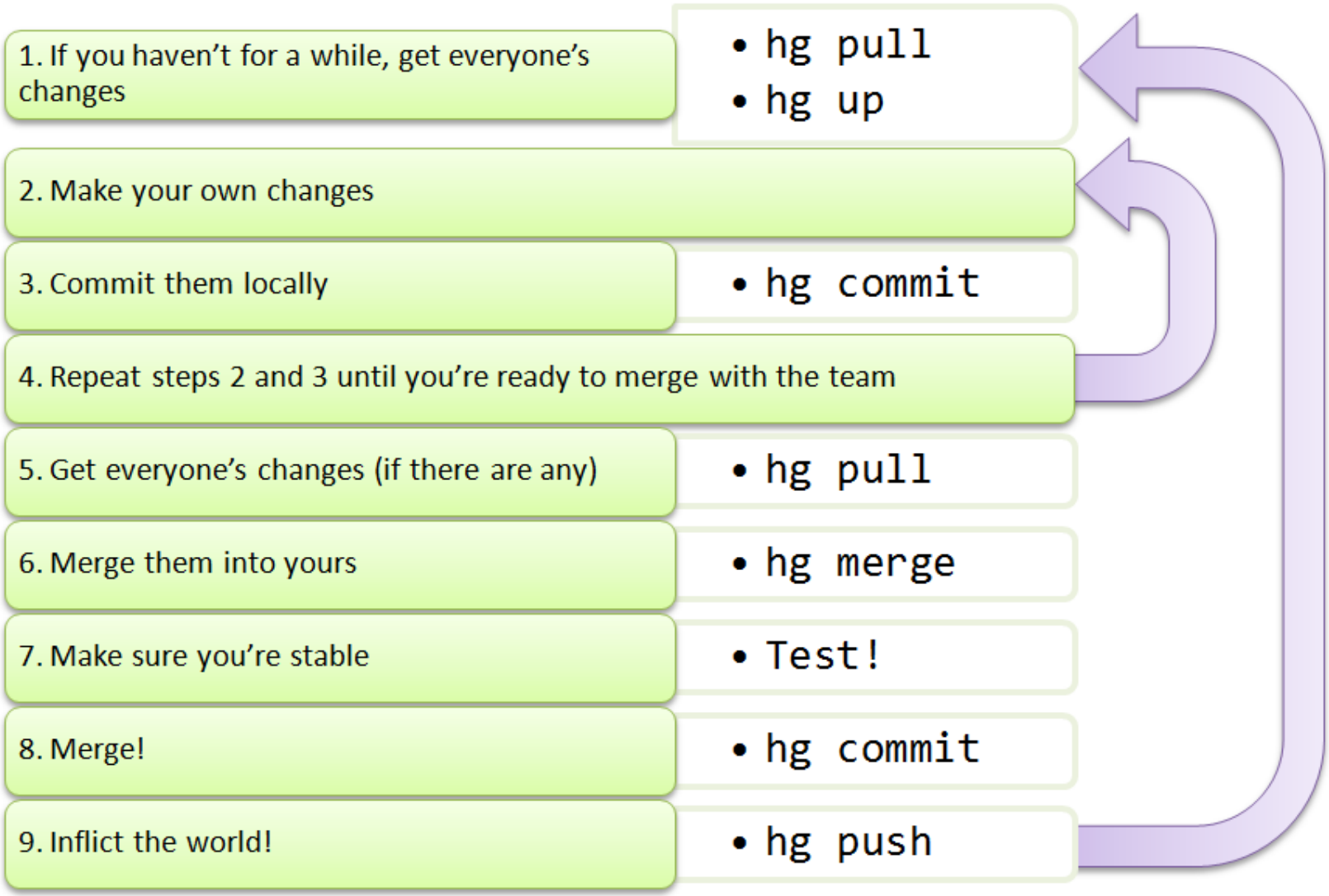
hg push



Workflow Grupal



Workflow Personal



derived from joel spolsky's 'hginit' tutorial, see http://hginit.com/02.html

Lista de comandos

add **add the specified files on the next commit**
addremove add all new files, delete all missing files
annotate show changeset information by line for each file
archive create an unversioned archive of a repository revision
backout reverse effect of earlier changeset
bisect subdivision search of changesets
branch set or show the current branch name
branches list repository named branches
bundle create a changegroup file
cat output the current or given revision of files
clone **make a copy of an existing repository**
commit **commit the specified files or all outstanding changes**
copy mark files as copied for the next commit
diff diff repository (or selected files)
export dump the header and diffs for one or more changesets
forget forget the specified files on the next commit
grep search for a pattern in specified files and revisions

Lista de comandos

heads show current repository heads or show branch heads
help show help for a given topic or a help overview
identify identify the working copy or specified revision
import import an ordered set of patches
incoming show new changesets found in source
init create a new repository in the given directory
locate locate files matching specific patterns
log show revision history of entire repository or files
manifest output the current or given revision of the project manifest
merge merge working directory with another revision
outgoing show changesets not found in the destination
parents show the parents of the working directory or revision
paths show aliases for remote repositories
pull pull changes from the specified source
push push changes to the specified destination
recover roll back an interrupted transaction
remove remove the specified files on the next commit
rename rename files; equivalent of copy + remove
resolve redo merges or set/view the merge status of files

Lista de comandos

revert restore individual files or directories to an earlier state
rollback roll back the last transaction (dangerous)
root print the root (top) of the current working directory
serve start stand-alone webserver
showconfig show combined config settings from all hgrc files
status show changed files in the working directory
summary summarize working directory state
tag add one or more tags for the current or given revision
tags list repository tags
tip show the tip revision
unbundle apply one or more changegroup files
update update working directory (or switch revisions)
verify verify the integrity of the repository
version output version and copyright information

¿Hay mas?

- Se puede extender por plugins
- Se puede importar desde otro controlador de versiones (ej. subversion)
- Tiene administración web
- Hay interfaces gráficas como TortoiseHG
- Es multiplataforma
- Para los Giteros: tiene stash (shelve), rebase y cherry-pick (transplant y record). Además los plugins no son ciudadanos de segunda.

En resumen

- Sistema de versionado te hace la vida más fácil (la palabra clave es más)
- Las diferencias entre Centralizado y Distribuido
- Elegimos Distribuido por su independencia
- Elegimos Mercurial por su simplicidad
- La curva de aprendizaje es baja

¿Alguna pregunta?

¡Muchas Gracias!

Juan B Cabral
jbc.develop@gmail.com

http://jbcabral.com