

# Laboratorio de Computación IV

## Clase 2

*Andrés Fortier*

# ¿Consultas?

---

- Comando: `cd`.
- Contenidos web: Protocolo HTTP.

# Comando del día: ls

- ls == list directory contents.
- Ejemplos
  - `ls` → Nombre de archivos y directorios.
  - `ls -l` → Información sobre archivos y directorios.
  - `ls -lh` → Anterior pero con formateo de tamaño.
  - `ls -a` → Incluir archivos ocultos.
  - `ls -r` → Lista en orden reverso.
  - `ls -lS` → Ordena alfabéticamente.
  - `ls -d` → Sólo subdirectorios.

# Comando del día: ls

- Los modificadores se pueden combinar:
  - ``ls -lSh``
  - ``ls -lhar``
- Para ver todas las opciones ``man ls``.
- Bonus: definir sus propios comandos
  - ``alias :ls="ls -lhS"`` → ``:ls``

# HTML

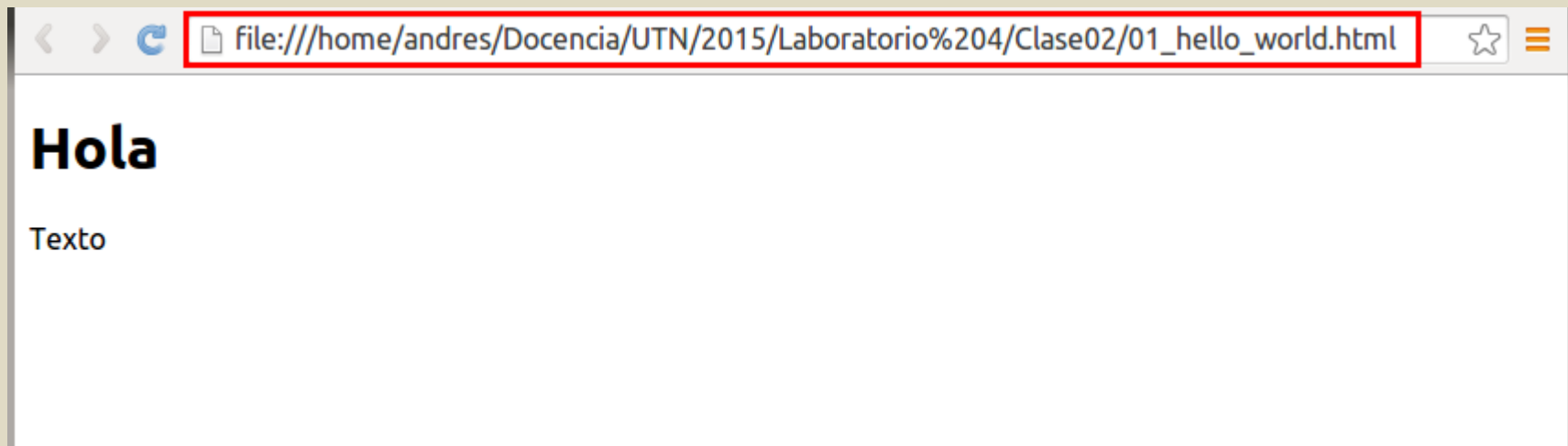
- HTML == *Hyper Text Markup Language*.
  - *Hypertext*: Texto con enlaces a otros textos.
  - *Markup*: Anotar un texto con etiquetas (*tags*).
- Ejemplo

01\_hello\_world.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Laboratorio 4</title>
  </head>
  <body>
    <h1>Hola</h1>
    <p>Texto</p>
  </body>
</html>
```

# HTML + HTTP

- Formas de verlo:
  - Abrir el archivo con un navegador



- HTTP?!

# HTTP + HTML

## – Ejecutar un servidor HTTP y servir la página

- `cd <directorio dónde se encuentra el archivo>`
- `python -m SimpleHTTPServer 8000`

```
$ python -m SimpleHTTPServer 8000  
Serving HTTP on 0.0.0.0 port 8000 ...
```

- En el navegador, ir a

[http://localhost:8000/01\\_hello\\_world.html](http://localhost:8000/01_hello_world.html)

```
Serving HTTP on 0.0.0.0 port 8000 ...  
127.0.0.1 - - [19/Mar/2015 22:22:10] "GET /  
01_hello_world.html HTTP/1.1" 200 -  
127.0.0.1 - - [19/Mar/2015 22:22:11] code 404, message  
File not found  
127.0.0.1 - - [19/Mar/2015 22:22:11] "GET /favicon.ico  
HTTP/1.1" 404 -
```

# HTTP + HTML

- `curl -v localhost:8000/01_hello_world.html`

```
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [19/Mar/2015 22:22:10] "GET /
01_hello_world.html HTTP/1.1" 200 -
127.0.0.1 - - [19/Mar/2015 22:22:11] code 404, message
File not found
127.0.0.1 - - [19/Mar/2015 22:22:11] "GET /favicon.ico
HTTP/1.1" 404 -
127.0.0.1 - - [19/Mar/2015 22:24:23] "GET /
01_hello_world.html HTTP/1.1" 200 -
```



# HTTP + HTML



```
[andres@CrazyGoat:~/Docencia/UTN/2015/Laboratorio 4/Clase02] curl -v localhost:8000/01_hello_world.html
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8000 (#0)
> GET /01_hello_world.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost:8000
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: SimpleHTTP/0.6 Python/2.7.6
< Date: Sun, 15 Mar 2015 20:50:22 GMT
< Content-type: text/html
< Content-Length: 137
< Last-Modified: Sun, 15 Mar 2015 20:25:25 GMT
<
<!DOCTYPE html>
<html>
  <head>
    <title>Laboratorio 4</title>
  </head>
  <body>
    <h1>Hola</h1>
    <p>Texto</p>
  </body>
</html>
* Closing connection 0
```

# HTTP + HTML

---

- SimpleHTTPServer es un servidor HTTP.
  - Da acceso a todo el directorio en forma recursiva.
  - No sólo sirve html (ej. txt, png, etc.).

# Etiquetas HTML

- Títulos: <h1> ... <h6>

```
<h1>Hola</h1>  
<h2>Hola</h2>  
<h3>Hola</h3>  
<h4>Hola</h4>  
<h5>Hola</h5>  
<h6>Hola</h6>
```

- Párrafos: <p>

```
<p>En un párrafo</p>
```

# Etiquetas HTML

- Líneas horizontales: `<hr />`

```
<p>Párrafo 1</p>  
<hr />  
<p>Párrafo 2</p>
```

- Preservar formato: `<pre>`

```
<pre>Prueba    de    espacios</pre>
```

- Atributos básicos: `<b>`, `<i>`, `<u>`

```
<b>Texto</b>
```

# Etiquetas HTML

- Recordemos que los elementos HTML se pueden anidar

```
<p>Texto <b>con</b> muchas <i><u>cosas</u></i></p>
```

```
<p>Texto  
  <b>con</b>  
  muchas  
  <i>  
    <u>cosas</u>  
  </i>  
</p>
```

# Etiquetas HTML

- Listas: `<ul>` y `<ol>` + `<li>`

```
<ol>  
  <li>Elemento 1</li>  
  <li>Elemento 2</li>  
  <li>Elemento 3</li>  
</ol>
```

- Listas de definciones: `<dl>`, `<dt>` y `<dd>`

```
<dl>  
  <dt>Elemento A</dt><dd>Definicion A</dd>  
  <dt>Elemento B</dt><dd>Definicion B</dd>  
</dl>
```

# Etiquetas HTML

- Tablas: <table>, <tr>, <th>, <td>

```
<table>
  <tr>
    <th>Columna 1</th>
    <th>Columna 2</th>
  </tr>
  <tr>
    <td>(1,1)</td>
    <td>(1,2)</td>
  </tr>
  <tr>
    <td>(2,1)</td>
    <td>(2,2)</td>
  </tr>
</table>
```

# Etiquetas HTML

- Los elementos HTML poseen atributos
  - Algunos son comunes a todos.
  - Otros dependen del elemento en cuestión.
- Anchors: <a>

```
<a href="http://www.google.com">Google</a>
```

- Imágenes: <img>

```

```



# Etiquetas HTML

- Forms: <form>
  - Indicar una url para que procese los datos.
  - Y el método HTTP a usar.

```
<form action="procesar.php" method="post">  
  ...  
</form>
```

- Se pueden usar distintos tipos de componentes para pedir información al usuario:
  - Textbox:

```
<input type="text" name="Nombre" value="Pepe">
```

# Etiquetas HTML

- Textarea:

```
<textarea name="descripcion" rows="5" cols="20">
  La descripción
</textarea>
```

- Radiobuttons:


```
<p>
  <input type="radio" name="edad" value="A">1-20
</p>
<p>
  <input type="radio" name="edad" value="B">21-40
</p>
<p>
  <input type="radio" name="edad" value="C">41-...
</p>
```

# Etiquetas HTML

- Select:

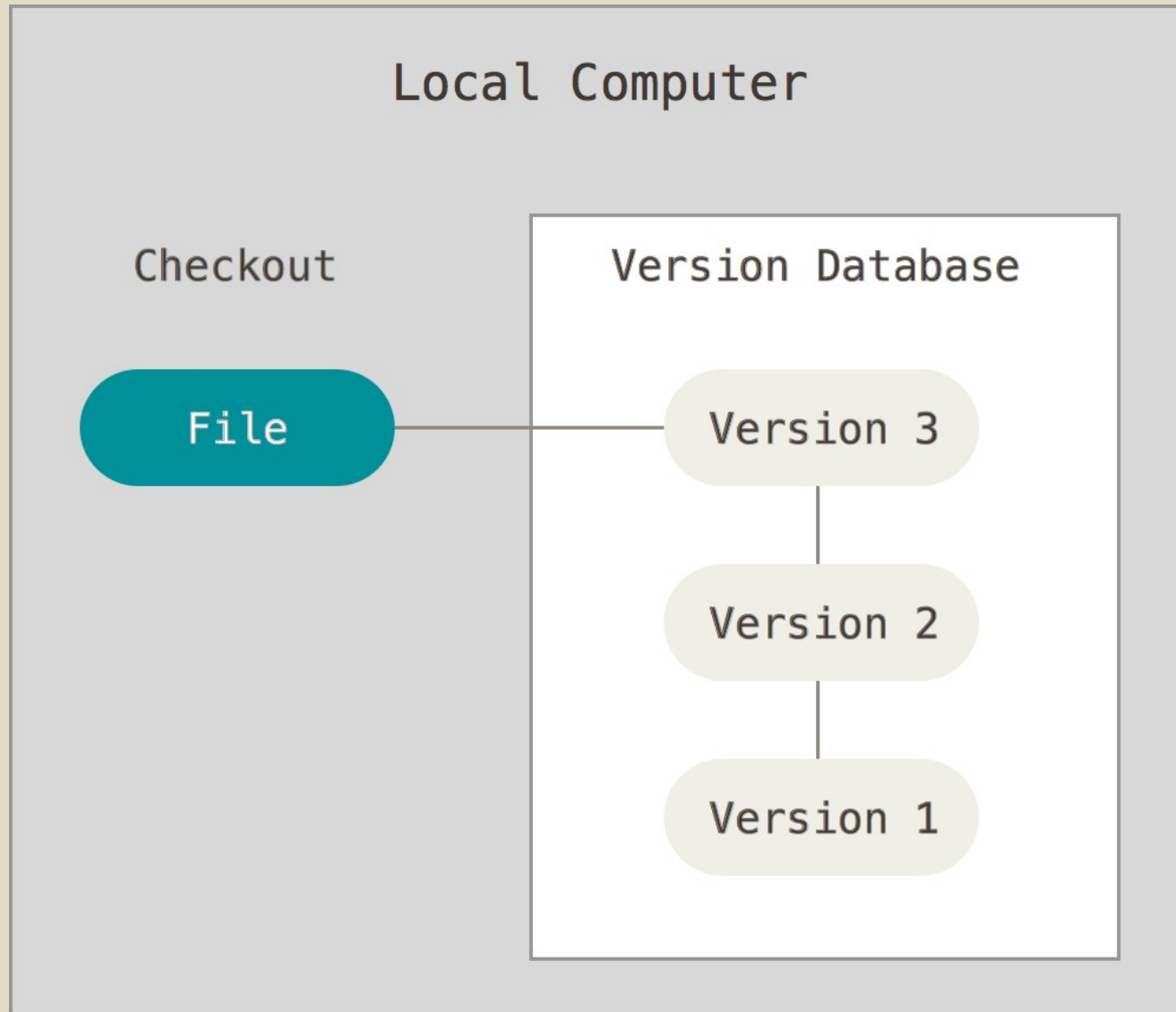
```
<select>
  <option value="1">Op 1</option>
  <option value="2" selected>Op 2</option>
  <option value="3">Op 3</option>
</select>
```

# *Version Control Systems (VCSs)*

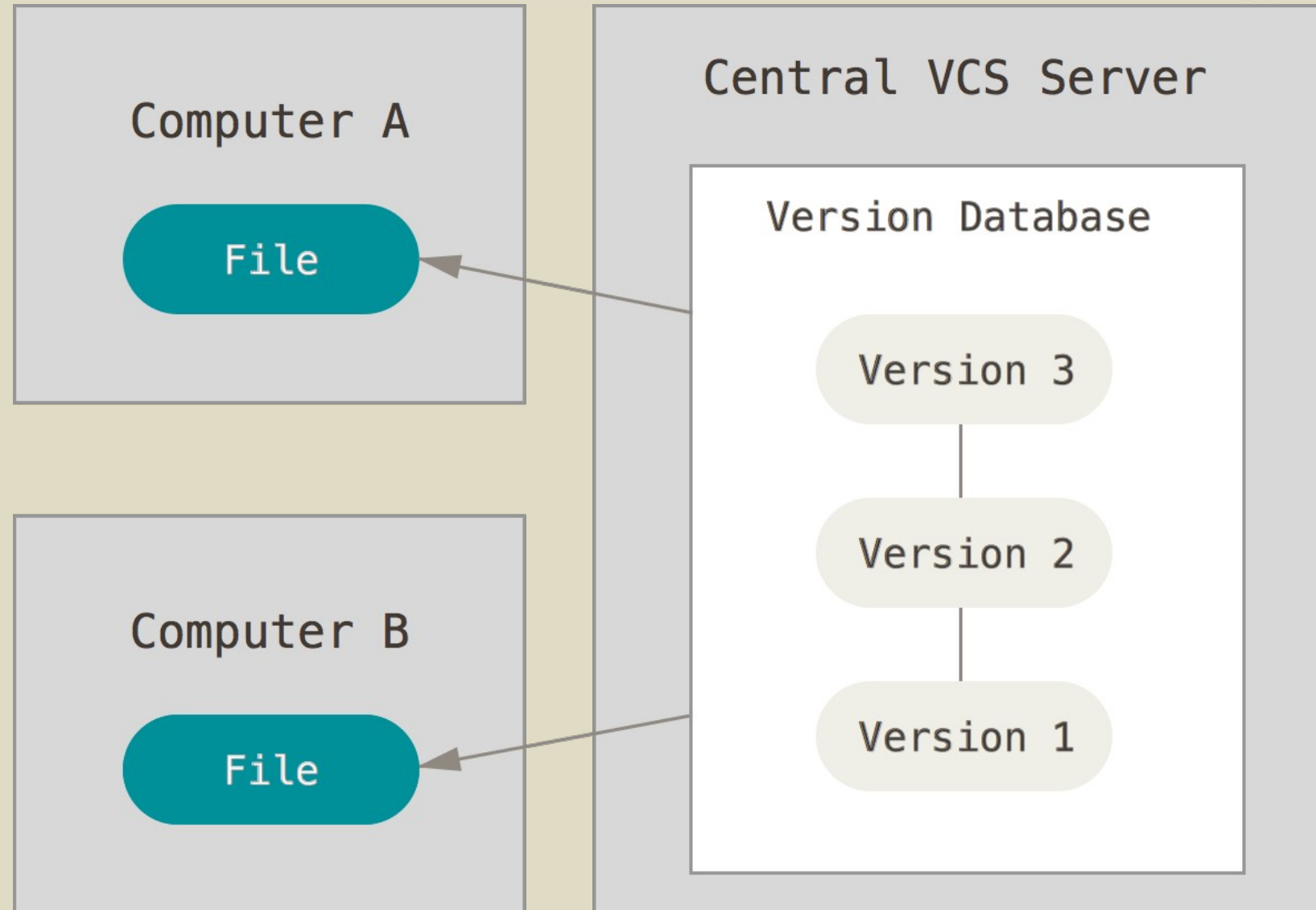


- Registra los cambios en uno o mas archivos a medida que pasa el tiempo.
- Permite
  - Revertir a un estado anterior.
  - Comparar cambios.
  - Determinar quién realizo un determinado cambio.

# Version Control Systems (VCSs)



# Version Control Systems (VCSs)



# Version Control Systems (VCSs)

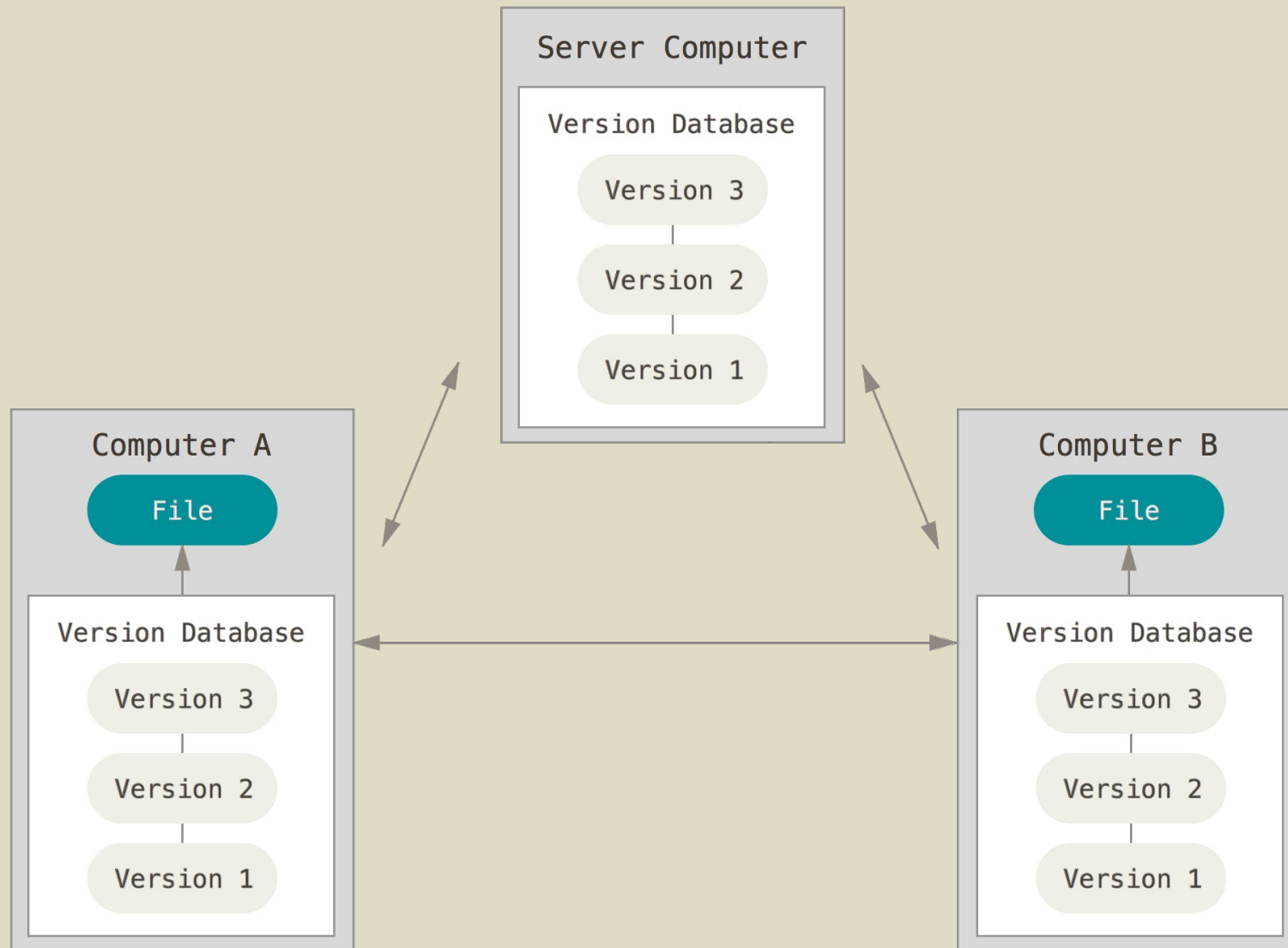
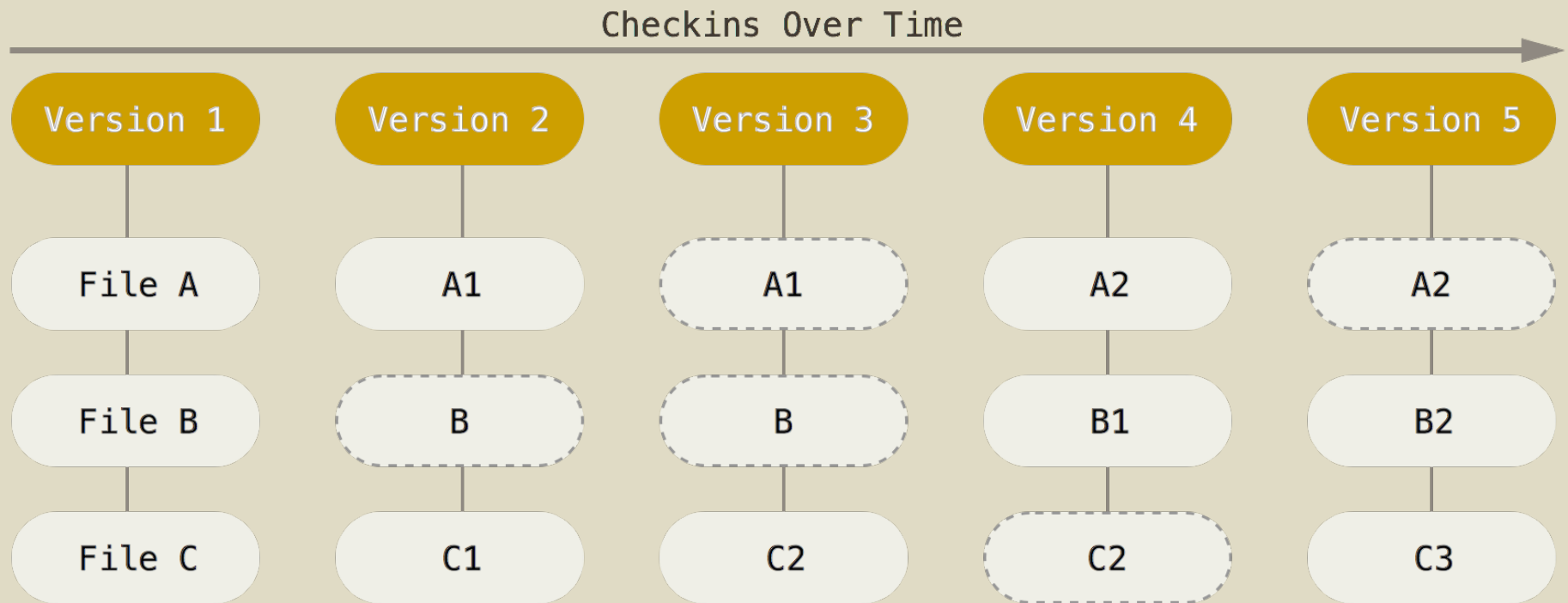


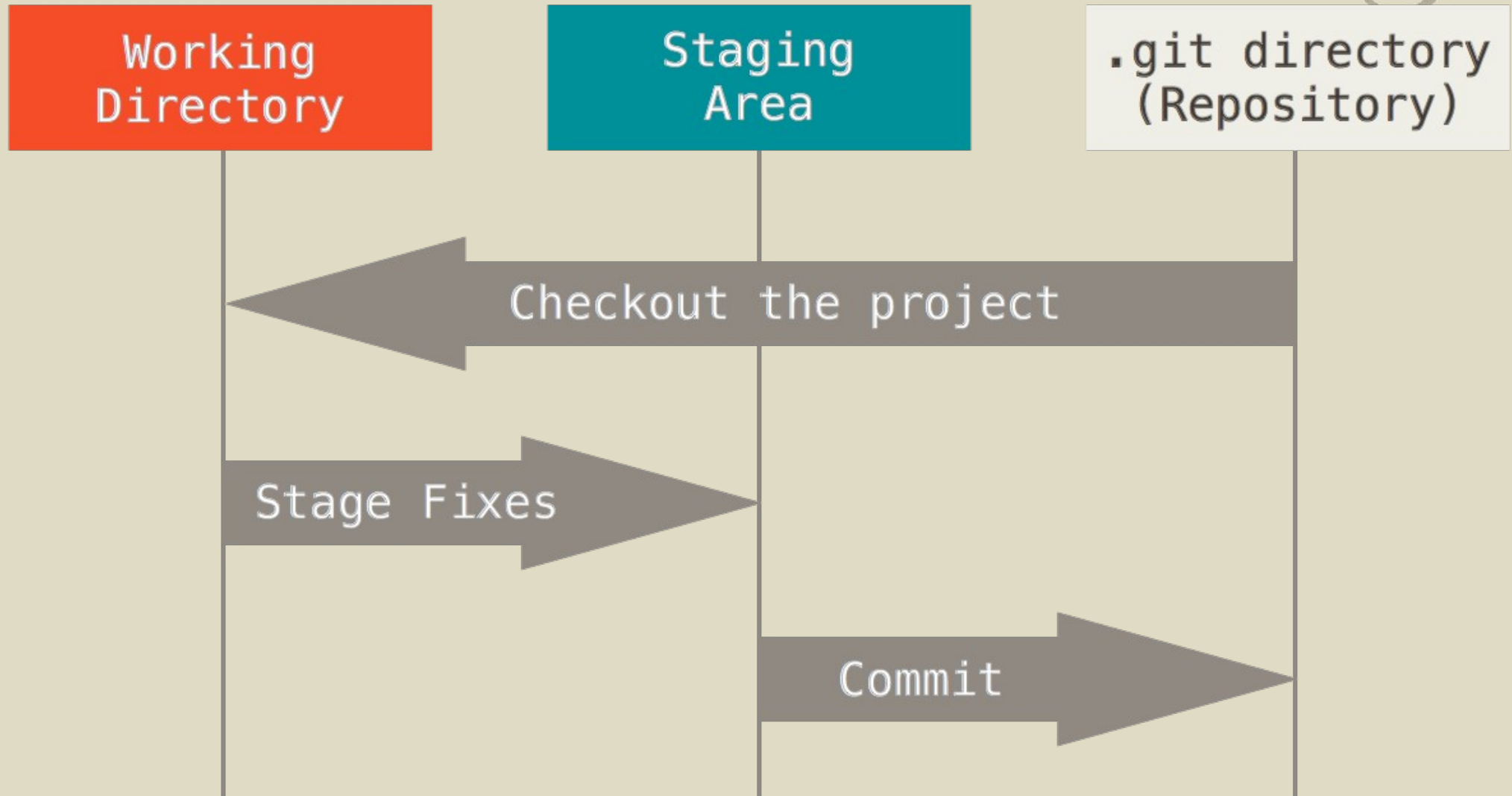
Imagen modificada de "Pro Git" <http://git-scm.com/book/en/v2/>, escrito por Scott Chacon y Ben Straub.  
Licenciada bajo Creative Commons Attribution Non Commercial Share Alike 3.0 license.

# Versiones en git





# Areas en git



# Vayamos a la práctica

- Verificar si git está instalado

```
$ git --version
```

- Si no, instalarlo

```
$ sudo apt-get install git
```

- Configurar lo básico y verificarlo

```
$ git config --global user.name "José"  
$ git config --global user.email jose@example.com
```

```
$ git config --list
```

# Crear un nuevo repositorio



- Crear un directorio

```
$ mkdir pruebas
```

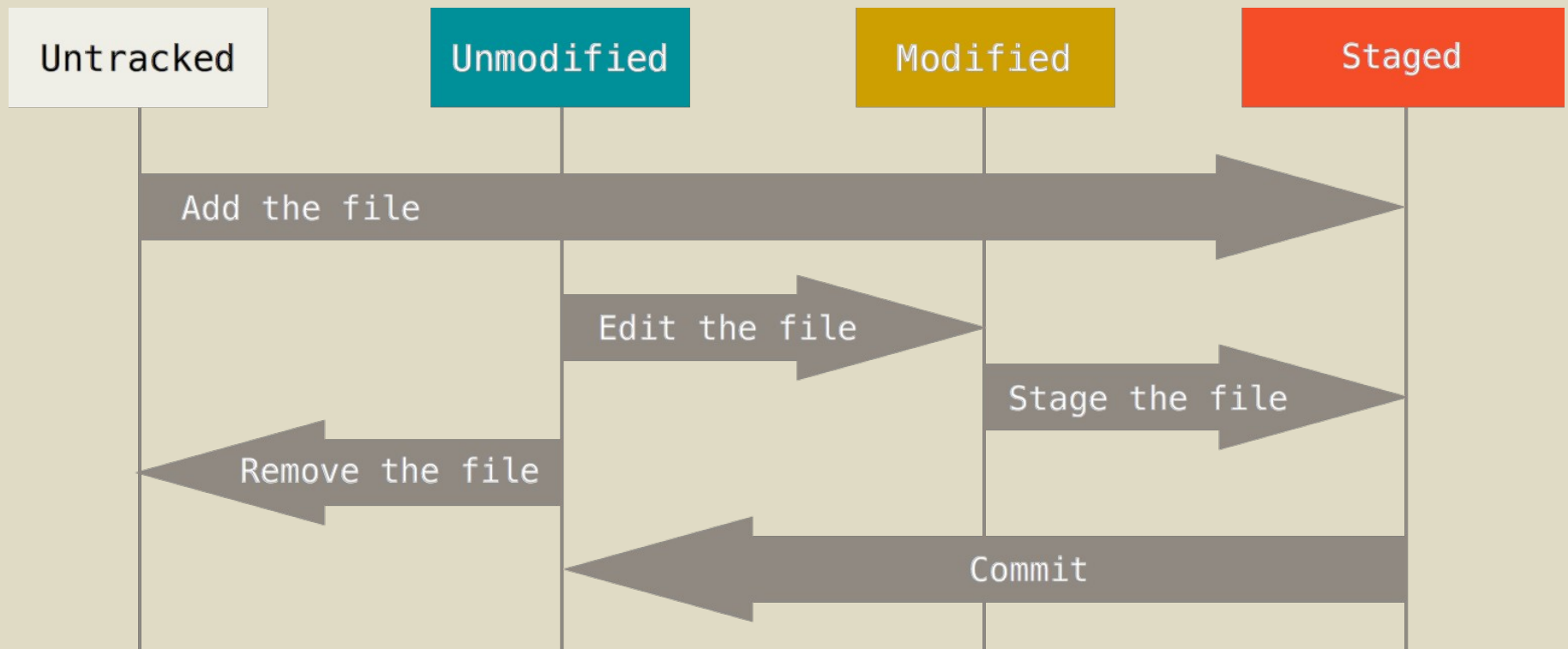
- Acceder al mismo

```
$ cd pruebas
```

- Inicializar el repositorio

```
$ git init
```

# Ciclo de vida de un archivo



# Trabajando con archivos

- Verificar el estado del repositorio

```
$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add"
to track)
```

- Crear un archivo

```
$ gedit archivo1.txt
```

```
Hola
```

# Trabajando con archivos

- Verificar el estado del repositorio

```
$ git status  
On branch master
```

```
Initial commit
```

**Untracked files:**

(use "git add <file>..." to include in what will be committed)

**archivo1.txt**

**nothing added to commit but untracked files present**  
(use "git add" to track)

# Trabajando con archivos

- Agregar el archivo y verificar el estado del repositorio

```
$ git add archivo1.txt
```

```
$ git status  
On branch master
```

```
Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:    archivo1.txt
```

# Trabajando con archivos

- Realizar un *commit* para persistir el cambio.

```
$ git commit -m 'El primer commit'  
[master (root-commit) 879f655] El primer commit  
1 file changed, 1 insertion(+)  
create mode 100644 archivo1.txt
```

```
$ git status  
On branch master  
nothing to commit, working directory clean
```



# Trabajando con archivos

- Modificar el archivo

```
$ gedit archivo1.txt
```

```
Hola mundo
```

# Trabajando con archivos

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be
  committed)
  (use "git checkout -- <file>..." to discard
  changes in working directory)

    modified:   archivo1.txt

Untracked files:
  (use "git add <file>..." to include in what will be
  committed)

  archivo1.txt~

no changes added to commit (use "git add" and/or "git
commit -a")
```

# Trabajando con archivos

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be
  committed)
  (use "git checkout -- <file>..." to discard changes
  in working directory)
```

```
    modified:   archivo1.txt
```

**Untracked files:**

```
  (use "git add <file>..." to include in what will
  be committed)
```

```
    archivo1.txt~
```

```
no changes added to commit (use "git add" and/or "git
commit -a")
```

# Trabajando con archivos

- Eliminemos el archivo secundario

```
$ rm archivo1.txt~
```

```
$ git status
```

```
On branch master
```

```
Changes not staged for commit:
```

```
  (use "git add <file>..." to update what will be  
committed)
```

```
  (use "git checkout -- <file>..." to discard  
changes in working directory)
```

```
    modified:   archivo1.txt
```

```
no changes added to commit (use "git add" and/or "git  
commit -a")
```

# Trabajando con archivos

- Descartemos los cambios

```
$ git checkout -- archivo1.txt
```

```
$ git status  
On branch master  
nothing to commit, working directory clean
```

# Trabajando con archivos

- Modifiquemos nuevamente el archivo

```
$ gedit archivo1.txt
```

```
Adios
```

```
$ rm archivo1.txt~
```

```
$ git status
```

```
On branch master
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be  
committed)
```

```
(use "git checkout -- <file>..." to discard  
changes in working directory)
```

```
modified:    archivo1.txt
```

```
no changes added to commit (use "git add" and/or "git  
commit -a")
```

# Trabajando con archivos

```
$ git add archivo1.txt
```

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   archivo1.txt
```

```
$ git commit -m 'Modificamos archivo1.txt'
[master dad7ae1] Modificamos archivo1.txt
1 file changed, 1 insertion(+), 1 deletion(-)
```

# Historial de cambios

```
$ git log  
commit dad7ae163558d32b90207f5081dfa839111c050a  
Author: Andrés Fortier <andres@creativa77.com>  
Date: Thu Mar 19 21:43:41 2015 -0300
```

Modificamos archivo1.txt

```
commit 879f655705648182914372b5de9566c2783b1047  
Author: Andrés Fortier <andres@creativa77.com>  
Date: Thu Mar 19 21:25:52 2015 -0300
```

El primer commit



# Historial de cambios

```
$ git diff 879f655..dad7ae1
diff --git a/archivo1.txt b/archivo1.txt
index a19abfe..3ec006e 100644
--- a/archivo1.txt
+++ b/archivo1.txt
@@ -1,1 @@
-Hola
+Adios
```

# Trabajando con archivos

- Algunos cambios mas

```
$ gedit archivo1.txt
```

```
Adios  
Amigos
```

```
$ rm archivo1.txt~
```

```
$ git status  
On branch master  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be  
committed)  
  (use "git checkout -- <file>..." to discard changes  
in working directory)  
  
    modified:   archivo1.txt  
  
no changes added to commit (use "git add" and/or "git  
commit -a")
```

# Trabajando con archivos

```
$ git add archivo1.txt
```

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   archivo1.txt
```

```
$ git reset HEAD archivo1.txt
Unstaged changes after reset:
M   archivo1.txt
```

# Trabajando con archivos

- Vuelvo al mismo estado anterior a hacer el *add*

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be
  committed)
  (use "git checkout -- <file>..." to discard changes
  in working directory)

    modified:   archivo1.txt

no changes added to commit (use "git add" and/or "git
commit -a")1
```

# Trabajando con archivos

- Continuemos

```
$ git add archivo1.txt
```

```
$ git commit -m 'Agregamos linea en archivo'  
[master 0839195] Agregamos linea en archivo  
1 file changed, 1 insertion(+)
```

```
$ git status  
On branch master  
nothing to commit, working directory clean
```

# Historial de cambios

```
$ git log  
commit 0839195770a487adeace0b8214ce43762d8fbbb4  
Author: Andrés Fortier <andres@creativa77.com>  
Date: Thu Mar 19 21:58:41 2015 -0300
```

Agregamos linea en archivo

```
commit dad7ae163558d32b90207f5081dfa839111c050a  
Author: Andrés Fortier <andres@creativa77.com>  
Date: Thu Mar 19 21:43:41 2015 -0300
```

Modificamos archivo1.txt

```
commit 879f655705648182914372b5de9566c2783b1047  
Author: Andrés Fortier <andres@creativa77.com>  
Date: Thu Mar 19 21:25:52 2015 -0300
```

El primer commit

# Historial de cambios

```
$ git diff dad7ae1..0839195
diff --git a/archivo1.txt b/archivo1.txt
index 3ec006e..6ed6d9b 100644
--- a/archivo1.txt
+++ b/archivo1.txt
@@ -1 +1,2 @@
  Adios
+Amigos
```

```
$ git diff 879f655..0839195
diff --git a/archivo1.txt b/archivo1.txt
index a19abfe..6ed6d9b 100644
--- a/archivo1.txt
+++ b/archivo1.txt
@@ -1 +1,2 @@
-Hola
+Adios
+Amigos
```

# Trabajando con archivos

- Uso del “.”

```
$ touch archivo2.txt
```

```
$ touch archivo3.txt
```

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be
  committed)
```

```
    archivo2.txt
```

```
    archivo3.txt
```

```
nothing added to commit but untracked files present
(use "git add" to track)
```



# Trabajando con archivos



```
$ git add .
```

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   archivo2.txt
    new file:   archivo3.txt
```

# Git - Links



- <http://git-scm.com/book/en/v2/>  
<http://git-scm.com/book/es/v2/>
- <http://rogerdudler.github.io/git-guide/index.html>  
<http://rogerdudler.github.io/git-guide/index.es.html>
- <http://gitimmersion.com/>
- <https://try.github.io/>

# Tarea para el hogar



- Recuerden que nos vemos en una semana.
- Crear un nuevo repositorio de git.
- Crear en el repositorio las siguientes páginas HTML y servir las usando el servidor HTTP python
  - Árbol genealógico partiendo de ustedes usando listas anidadas. Utilizar `<h1>` para un título y `<p>` para descripción.

# Tarea para el hogar

- Una página con una tabla con los mismos datos que aparecen en [http://indicadores.datospublicos.gob.ar/reporte/grafico\\_demo/?apps0%5B%5D=CABA&apps0%5B%5D=Corrientes&apps0%5B%5D=Mendoza&tabla\\_id=253](http://indicadores.datospublicos.gob.ar/reporte/grafico_demo/?apps0%5B%5D=CABA&apps0%5B%5D=Corrientes&apps0%5B%5D=Mendoza&tabla_id=253)
- Una página que utilice una imagen pública en internet y una local.
- Para ejercitar git
  - Hagan *commits* periódicos y vean los *diffs*.
  - Pierdan el miedo a mover un archivo entre el *working directory* y *staging area*.

# Tarea para el hogar

---

- Bonus - investigar sobre alias en linux:
  - Google “linux crear alias”