

# Laboratorio de Computación IV

## Clase 5

*Andrés Fortier*

# ¿Consultas?

---

- Comando: top.
- Contenidos web: ruby + rspec + highline.
- Herramienta: git branches.

# Tarea para el hogar

---

- Investigar el uso de `.ruby-version` y `.ruby-gemset`.
- Investigar la gema `highline`.

# Tarea para el hogar



- Realizar los tests con rspec para el contador.
- Realizar los tests con rspec para los ejercicios 1, 6 y 7 de la Práctica 2 de Programación 3.
- Escribir en cmd.rb un programa con cuatro opciones:
  - Login → Imprime “Logueando...”
  - Logout → Imprime “Deslogueando...”
  - Estado → Imprime “Tu estado es”
  - Salir → Sale de la aplicación

# Comando del día: ssh



- SSH == Protocolo seguro de comunicación
  - Encripta los datos entre el cliente y el servidor.
- Permite
  - Terminales remotas.
  - Transferencia de archivos.
- Mucho escrito en internet, sólo veremos lo básico.

# Comando del día: ssh

- Abrir una terminal remota (requiere una cuenta)

```
$ ssh nombre_de_usuario@host
```

- Ejemplo

```
$ ssh andres@xyz.com  
andres@xyz.com's password: *****  
[andres@xyz.com ~]$
```

```
[andres@xyz.com ~]$ exit  
$
```

# Comando del día: ssh

---

- Negociación de la clave de sesión (encriptación simétrica).
- SSH sin user/password
  - Autenticación por encriptación asimétrica: public/private keys.

# Openshift



- *PAAS (Platform as a service).*
- Openshift
  - *Engines.*
  - *Applications.*
  - *Cartridges.*
- Plan gratuito con 3 *engines*.
- Lo vamos a usar para hostear nuestra aplicación web.
- Pero comencemos simplemente con SSH.



# Openshift

- Ir a <https://www.openshift.com/> y crear una cuenta (“*Sign-up*”).
- Configurar la cuenta
  - <https://developers.openshift.com/en/getting-started-debian-ubuntu.html#rhc-setup>

```
$ rvm gemset use global
```

```
$ gem install rhc
```

```
$ rhc setup
```

# Openshift

- Ir a <https://openshift.redhat.com/app/console/settings> y verificar que se encuentra la clave pública.
- Crear una aplicación (*ejemplo-ssh*)
  - <https://openshift.redhat.com/app/console/applications>
- Desde una terminal ingresar via ssh
  - Obtener **nombre@host** de la página de la app.

```
$ ssh <id>@ejemplo-ssh-<dominio>.rhcloud.com
```

# Openshift

```
$ ssh <id>@ejemplo-ssh-<dominio>.rhcloud.com  
[ejemplo-ssh-<dominio>.rhcloud.com <id>\>
```

- Evaluar

```
[ejemplo-ssh-<dominio>.rhcloud.com <id>\> ls -la
```

```
[ejemplo-ssh-<dominio>.rhcloud.com <id>\> mkdir test
```

```
[ejemplo-ssh-<dominio>.rhcloud.com <id>\> cd test
```

```
[ejemplo-ssh-<dominio>.rhcloud.com <id>\> cd ..
```

```
[ejemplo-ssh-<dominio>.rhcloud.com <id>\> rdmir test
```

```
[ejemplo-ssh-<dominio>.rhcloud.com <id>\> top
```

```
[ejemplo-ssh-<dominio>.rhcloud.com <id>\> exit
```

# Github



- Plataforma de *hosting* de repositorios git.
  - Permite compartir nuestros repositorios.
- Permite editar archivos (aunque no es lo mas recomendado).
- Soporta *pull-requests*.
- Manejo básico de *issues*.

# Github



- Crear cuenta nueva.
- Crear un repositorio nuevo.
- Crear un README.md
- Sobre el readme
  - Realizar cambios.
  - Ver el historial de commits.
  - Ver los cambios
    - Unified vs Split.

# Github

- En un proyecto real no vamos a editar los archivos desde Github.
- Git: repositorios distribuidos
  - Clonar un repositorio

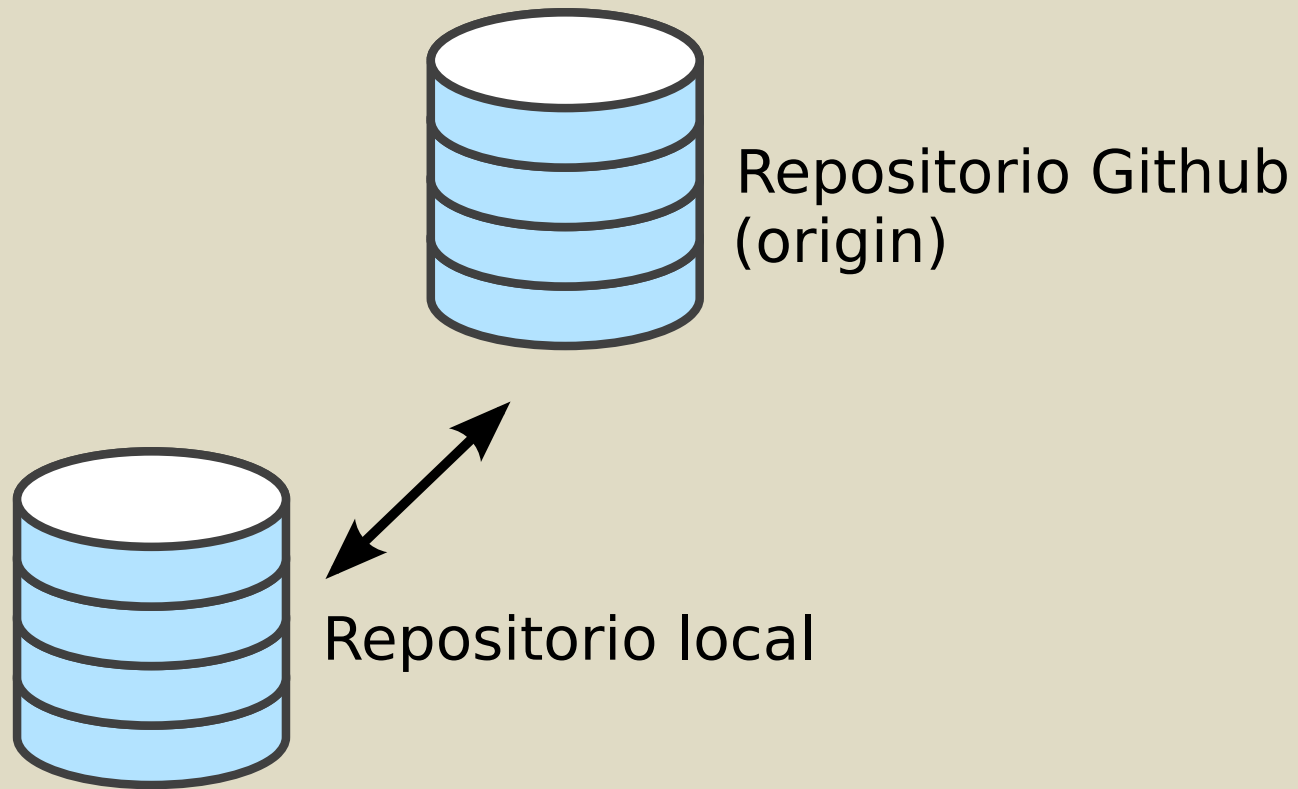
```
$ git clone https://github.com/<usuario>/<repo>.git
```

```
$ cd <repo>
```

- Ver el *remote* “origin”

```
$ git remote -v  
origin https://github.com/<usuario>/<repo>.git (fetch)  
origin https://github.com/<usuario>/<repo>.git (push)
```

# Github



# Github

- Veamos el contenido

```
$ ls -la
total 24
drwxrwxr-x 3 andres andres 4096 abr  4 13:30 .
drwxrwxr-x 4 andres andres 4096 abr  4 16:01 ..
drwxrwxr-x 8 andres andres 4096 abr  4 16:04 .git
-rw-rw-r-- 1 andres andres  17 abr  4 13:30 README.md
```

- Veamos un log

```
$ git log
```



# Github

- Editemos el contenido de README.md

```
# Título de mi proyecto  
Hola!
```

- Creemos una versión

```
$ git add README.md  
$ git ci -m 'Primer commit local'  
[master 20c6ea9] Primer commit local  
1 file changed, 2 insertions(+), 2 deletions(-)
```

# Github (*git push*)

- Sincronicemos con el repositorio remoto

```
$ git push origin master
Username for 'https://github.com': andres-fortier
Password for 'https://andres-fortier@github.com':
Counting objects: 5, done.
Writing objects: 100% (3/3), 286 bytes | 0 bytes/s,
done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/andres-fortier/utn-test.git
    ed00eec..20c6ea9  master -> master
```

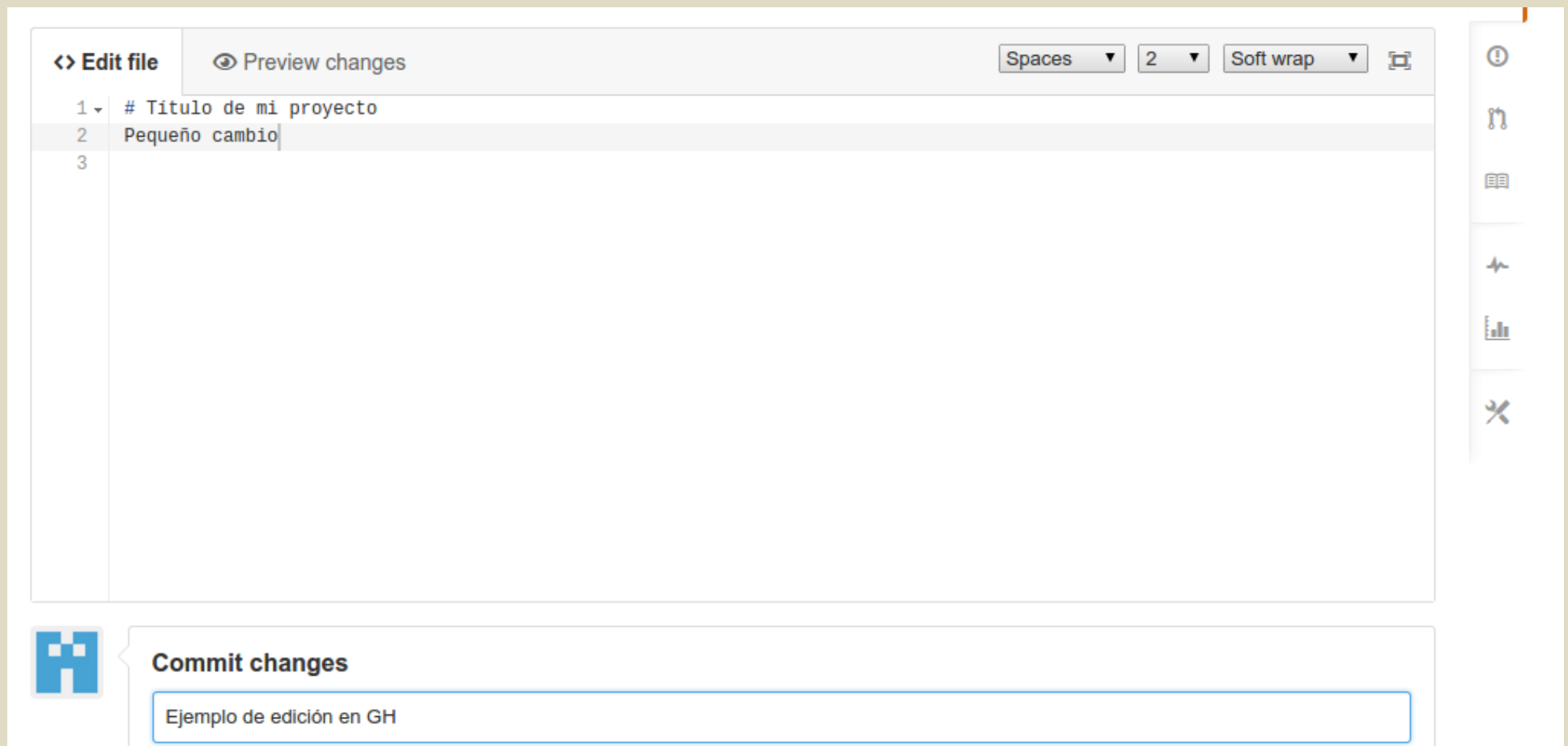
# Github



- Vayan a la página de su repo en Github
  - El readme debería haber cambiado.
  - Deberían ver un nuevo commit en el log.
  - Vean el diff.

# Github

- Veamos ahora como traer cambios desde el repositorio remoto.



# Github (*git pull*)

- Ver el log en github.
- Traer los cambios del repositorio remoto al local

```
$ git pull origin master
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-
reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/andres-fortier/utn-test
* branch                master      -> FETCH_HEAD
   20c6ea9..b89ab41      master      -> origin/master
Updating 20c6ea9..b89ab41
Fast-forward
 README.md | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

# Github



- Veamos ahora como publicar un repositorio existente (entrega1)
  - Crear un nuevo repositorio en github (entrega1).
  - Agregarlo como *remote* a nuestro repositorio local.
  - Hacer un *push* para sincronizarlos.

# Github

- Creen el repo en github.


**Quick setup — if you've done this kind of thing before**

or HTTPS SSH `git@github.com:andres-fortier/entrega1.git` 

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


**...or create a new repository on the command line**

```
echo "# entrega1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:andres-fortier/entrega1.git
git push -u origin master
```



**...or push an existing repository from the command line**

```
git remote add origin git@github.com:andres-fortier/entrega1.git
git push -u origin master
```



**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

# Github

- Chequeemos los *remotes* en nuestro repo local

```
$ git remote -v  
$
```

- Agreguemos el nuevo *remote*

```
$ git remote add origin git@github.com:<usr>/entregal.git
```

```
$ git remote -v  
origin git@github.com:<usr>/entregal.git (fetch)  
origin git@github.com:<usr>/entregal.git (push)
```



# Github

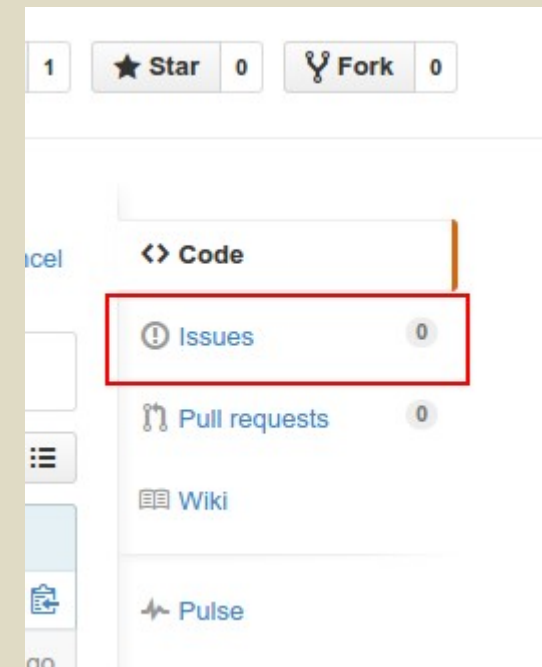
- Hagamos un *push*

```
$ git push -u origin master
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 586 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To git@github.com:andres-fortier/entrega1.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from
origin.
```

- Vayan a Github y chequeen los commits.

# Github issues

- Cualquier proyecto no-trivial debe(ría) utilizar un sistema de tickets o similar.
  - Planificar el trabajo.
  - Registrar hallazgos.
  - Reportar bugs de usuarios.
- Github provee uno llamado *issues*.
- Nos da un manejador por repo.



# Github issues

- Crear un issue

The screenshot displays the GitHub interface for creating a new issue. At the top, there are tabs for 'Issues', 'Pull requests', 'Labels', and 'Milestones'. The 'Issues' tab is active. Below the tabs, there is a header area with the GitHub logo and a title field containing 'Ejemplo de issue'. To the right of the title field, there are settings for 'Labels' (None yet), 'Milestone' (No milestone), and 'Assignee' (No one—assign yourself). The 'Assignee' section is highlighted with a red box. Below the title field, there is a 'Write' tab and a 'Preview' tab. The 'Write' tab is active, showing a text area with the following content: '## Los issues se pueden editar con markdown', 'Texto descriptivo', 'Lista de elementos:', '\* Uno', and '\* Dos'. Below the text area, there is a footer that says 'Attach images by dragging & dropping, selecting them, or pasting from the clipboard.' At the bottom right, there is a green button labeled 'Submit new Issue'.


# Github issues

Issues

Pull requests

Labels

Milestones



Ejemplo de issue

Write

Preview

Markdown supported Edit in fullscreen

## Los issues se pueden editar con markdown

Texto descriptivo

Lista de elementos:

- Uno
- Dos

Submit new Issue

Labels

None yet

Milestone

No milestone

Assignee


No one—assign yourself

<>

!

# Github issues

IssuesPull requestsLabelsMilestones



Ejemplo de issue

WritePreview

Markdown supportedEdit in fullscreen

## Los issues se pueden editar con markdown

Texto descriptivo

Lista de elementos:

- \* Uno
- \* Dos

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Submit new Issue


Labels

None yet

Milestone

No milestone

Assignee

 andres-fortier

# Github issues

<https://github.com/andres-fortier/utn-test/issues/1>

The screenshot shows a GitHub repository page for 'andres-fortier / utn-test'. The issue title is 'Ejemplo de issue #1', which is highlighted with a red box. The issue is marked as 'Open' and was opened by 'andres-fortier' just now. The issue body contains the text 'Los issues se pueden editar con markdown' and a list of items: 'Uno' and 'Dos'. The issue is assigned to 'andres-fortier', which is also highlighted with a red box. The assignee's name is 'andres-fortier'. The issue was self-assigned by 'andres-fortier' just now, which is also highlighted with a red box. The right sidebar shows the issue's metadata: Labels (None yet), Milestone (No milestone), Assignee (andres-fortier), and Notifications (Unsubscribe).

andres-fortier / utn-test

Unwatch 1 Star 0 Fork 0

Ejemplo de issue #1

Open andres-fortier opened this issue just now · 0 comments

andres-fortier commented just now

Owner

Los issues se pueden editar con markdown

Texto descriptivo  
Lista de elementos:

- Uno
- Dos

andres-fortier self-assigned this just now

Labels  
None yet

Milestone  
No milestone

Assignee  
andres-fortier

Notifications  
Unsubscribe

You're receiving notifications

# Github issues

<https://github.com/andres-fortier/utn-test/issues>

This repository Search Explore Gist Blog Help andres-fortier

andres-fortier / utn-test Unwatch 1 Star 0 Fork 0

Issues Pull requests Labels Milestones Filters is:issue is:open New Issue

1 Open 0 Closed Author Labels Milestones Assignee Sort

**Ejemplo de issue**  
#1 opened 17 seconds ago by andres-fortier

**ProTip!** Exclude everything labeled `bug` with `-label:bug`.

# Tarea



- Jugar un poco con openshift
  - Ahora pueden romper tranquilos (crear/borrar aplicaciones, repos git, etc).
  - <https://developers.openshift.com/en/getting-started-overview.html>
- Configurar su *key ssh* para github.
- Leer un poco sobre markdown.
- Leer sobre github.



# Tarea



- <https://guides.github.com/activities/hello-world/>
- <http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1>
- <http://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes>
- <https://www.atlassian.com/git/tutorials/syncing>
- <https://guides.github.com/features/issues/>