

Laboratorio de Computación IV

Clase 6

Andrés Fortier

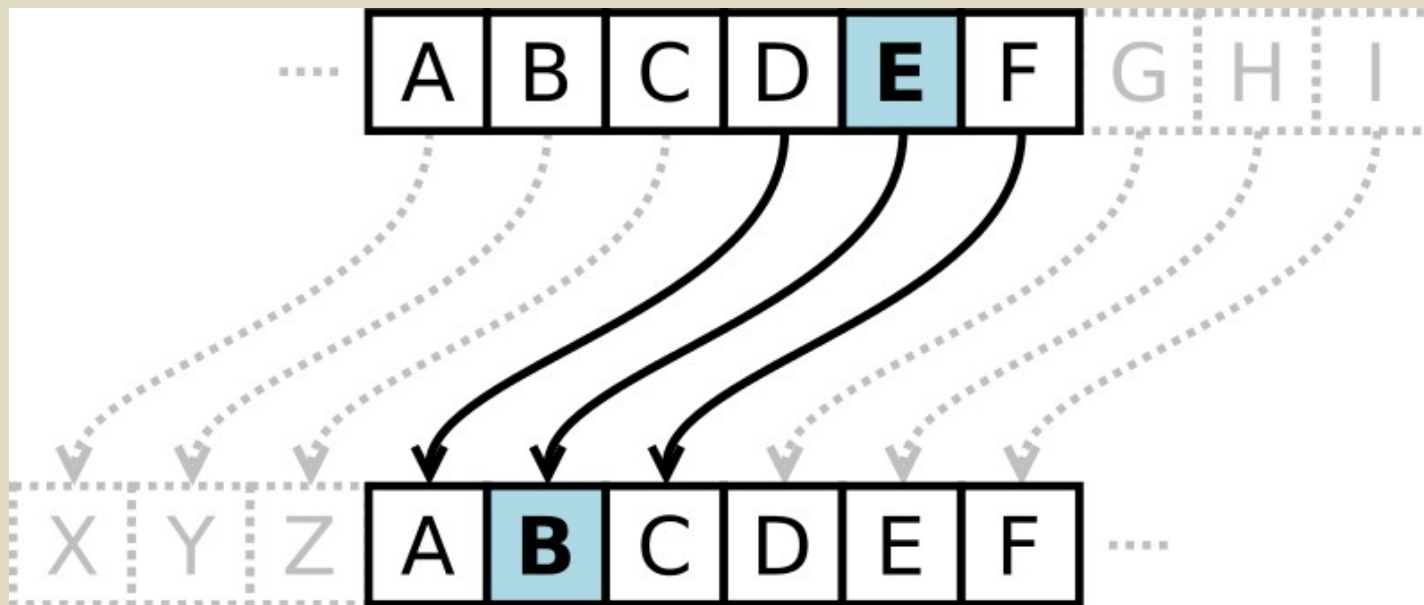
¿Consultas?

- SSH.
- Openshift.
- Seguridad en la comunicación.
- Almacenamiento de passwords
 - Texto plano.
 - Encriptación bidireccional.

Ejemplo de encriptación

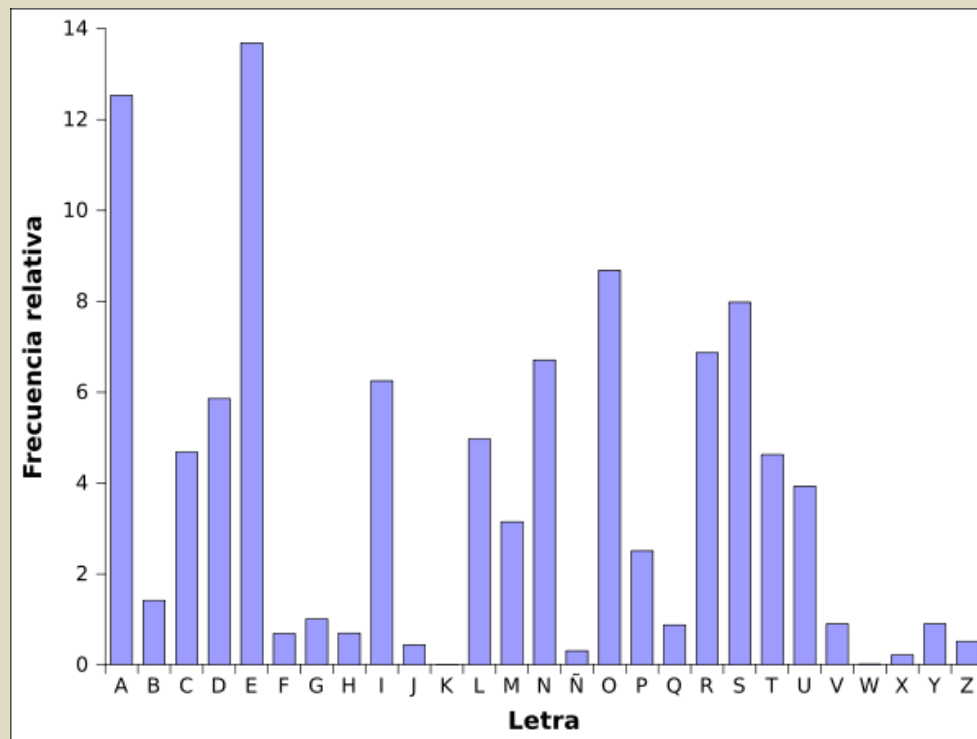
- Caesar cipher
 - Utilizado por Julio Caesar para su correspondencia.
 - “Sumaba” 3 letras.

Sin codificar: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Codificado : DEFPGHIJKLMNOPQRSTUVWXYZABC



Ejemplo de encriptación

- Fácil de quebrar
 - Fuerza bruta (probar las 27 combinaciones).
 - Análisis de frecuencia de letras.



Hashing

- Hash criptográfico (Cryptographic hash).
- Función
 - Convierte un mensaje (ej. password) en bytes (digest).

ejemplo →

```
$2a$10$H7HnvDAfNyKo1mnj.cuBoOxWOGshDpgOnmPlgXpCitIFmwo.s6b10
```

Hashing



- Hash criptográfico (Cryptographic hash).
- Función
 - Convierte un mensaje (ej. password) en bytes (digest).
 - Simple computar el hash.
 - Baja probabilidad de generar el mensaje a partir del hash (no-reversibles).
 - Baja probabilidad de modificar el mensaje sin modificar el hash.
 - Baja probabilidad de colisiones.

Hashing y fuerza bruta

- No cualquier función de hash
 - Fáciles de romper por fuerza bruta (ej. MD5 o SHA1).
 - Preferentemente adaptive hashing
 - Procesador: PBKDF2 y Bcrypt
 - Memoria: Scrypt


	Digest por segundo
NTLM	350,000,000,000
MD5	180,000,000,000
SHA1	63,000,000,000
SHA512Crypt	364,000
Bcrypt	71,000

Hashing y diccionarios



- Lista de palabras (o claves) usuales.
- Se computa y comparan los hash; si hay coincidencia, tenemos la clave.
- No garantizan encontrar la clave.
- Sensibles al “tamaño” de la clave; mas caracteres, mas combinaciones.

Hashing y lookup tables



- Pre-computar tablas (ya sean todas las combinaciones o por diccionario)
 - Ej. <clave>, bcrypt(<clave>)
- Luego se comparan los hash; si hay coincidencia, tenemos la clave.
- Sensibles al “tamaño” de la clave; tiempo vs espacio.
- Rainbow tables. Estructuras de datos que balancean tiempo vs espacio.

Hashing y salt

- Tablas pre-computadas
 - Dos usuarios con el mismo password tienen el mismo hash.
 - Permite paralelizar la búsqueda.
- Salting
 - Generar un string aleatorio por usuario

```
hash = bcrypt(password + random_salt)
```

 - Dos usuarios con mismo password, distinto hash.
 - Evita ataques por tablas (no es pre-computable).

Hashing - iteraciones

- Iterar sobre el password generado

```
hash = bcrypt(password + random_salt)
for (i = 0; i < 100; i++) {
  hash = bcrypt(password + random_salt + hash)
}
```

- Toma mas tiempo.
- No se encuentran en diccionarios.

Hashing - recuperar password



- Un hash criptográfico no es reversible.
 - No se puede recuperar el password.
- Sitio web
 - Link “perdí mi password”.
 - Mail al usuario con un token
 - <http://misitio.com?recover=SDCACJVJERVNV...>
 - El token expira en un lapso de tiempo predefinido.
 - El token es válido para un solo ingreso.

Entrega 1



- Modificar el programa en ``cmd.rb`` para que autentique usuarios.
- Implementar las tres estrategias vistas
 - Texto plano.
 - Caesar cipher.
 - Bcrypt.
 - <https://rubygems.org/gems/bcrypt/versions/3.1.11>
- Algunos casos de uso

Entrega 1

- Salir del programa

```
> Seleccione una accion:  
1-Login  
2-Logout  
3-Estado  
4-Salir  
? 4  
> Adios, vuelva pronto!  
$
```

Entrega 1

- Estado de persona no logueada

```
> Seleccione una accion:  
1-Login  
2-Logout  
3-Estado  
4-Salir  
? 3  
> Usted no se encuentra logueado  
> Seleccione una accion:  
...
```

Entrega 1

- Estado de persona logueada

```
> Seleccione una accion:  
1-Login  
2-Logout  
3-Estado  
4-Salir  
? 3  
> Usted está logueado como "Pepe"  
> Seleccione una accion:  
...
```


Entrega 1

- Log in exitoso

```
> Seleccione una accion:
1-Login
2-Logout
3-Estado
4-Salir
? 1
> Usuario: "Pepe"
> Password: *****
> Usted se ha logueado exitosamente!
> Seleccione una accion:
1-Login
2-Logout
3-Estado
4-Salir
? 3
> Usted está logueado como "Pepe"
> Seleccione una accion:
...
```

Entrega 1

- Log in fallido

```
> Seleccione una accion:
1-Login
2-Logout
3-Estado
4-Salir
? 1
> Usuario: "Pepe"
> Password: *****
> Nombre de usuario o contraseña incorrecta
> Seleccione una accion:
1-Login
2-Logout
3-Estado
4-Salir
? 3
> Usted no se encuentra logueado
> Seleccione una accion:
...
```

Entrega 1

- Logout de un usuario logueado

```
> Seleccione una accion:  
1-Login  
2-Logout  
3-Estado  
4-Salir  
? 2  
> Usted se ha deslogueado en forma exitosa  
> Seleccione una accion:  
...
```

Entrega 1

- Logout de un usuario no logueado

```
> Seleccione una accion:  
1-Login  
2-Logout  
3-Estado  
4-Salir  
? 2  
> Usted no se encuentra logueado  
> Seleccione una accion:  
...
```

Requerimientos mínimos



- Nombre de usuario y password válido predefinido.
- Seleccionar la estrategia a usar en forma sencilla
 - Por medio de ``require`` (o ``require_relative``).
 - Comentando/des-comentando alguna línea de código.

Evaluación

- Recuerden que es individual.
- Correcto funcionamiento.
- Diseño
 - Separar el modelo de la vista/controlador.
 - POLIMORFISMO.
- Tests.
- Uso de git.

Bonus track



- Sólo mostrar “Logout” si el usuario está logueado.
- Mecanismo de autenticación
 - Agregar una opción para cambiarlo en tiempo de ejecución.
 - Pasar un parámetro al programa

```
$ ruby cmd.rb auth=caesar
```

- Ver por ejemplo <http://www.sitepoint.com/ruby-command-line-interface-gems/>

Bonus track



- Agregar la opción de registrar usuarios (dar de alta usuario y password)
 - Sin persistir, sólo en memoria.
- Coverage de los tests
 - <https://github.com/colszowka/simplecov>
 - https://shvets.github.io/blog/2013/10/19/configure_simplecov.html

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

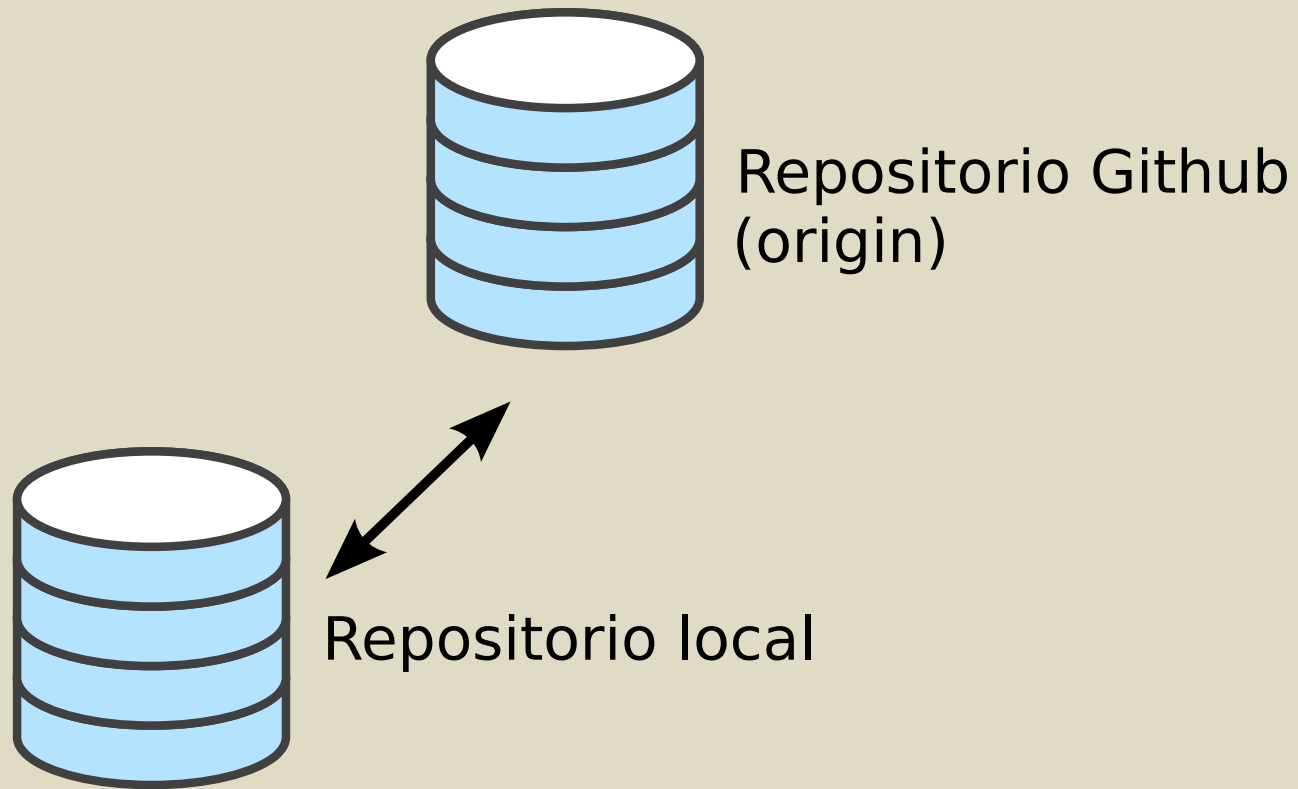
```
$ git clone https://github.com/<usuario>/<repo>.git
```

```
$ 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
```

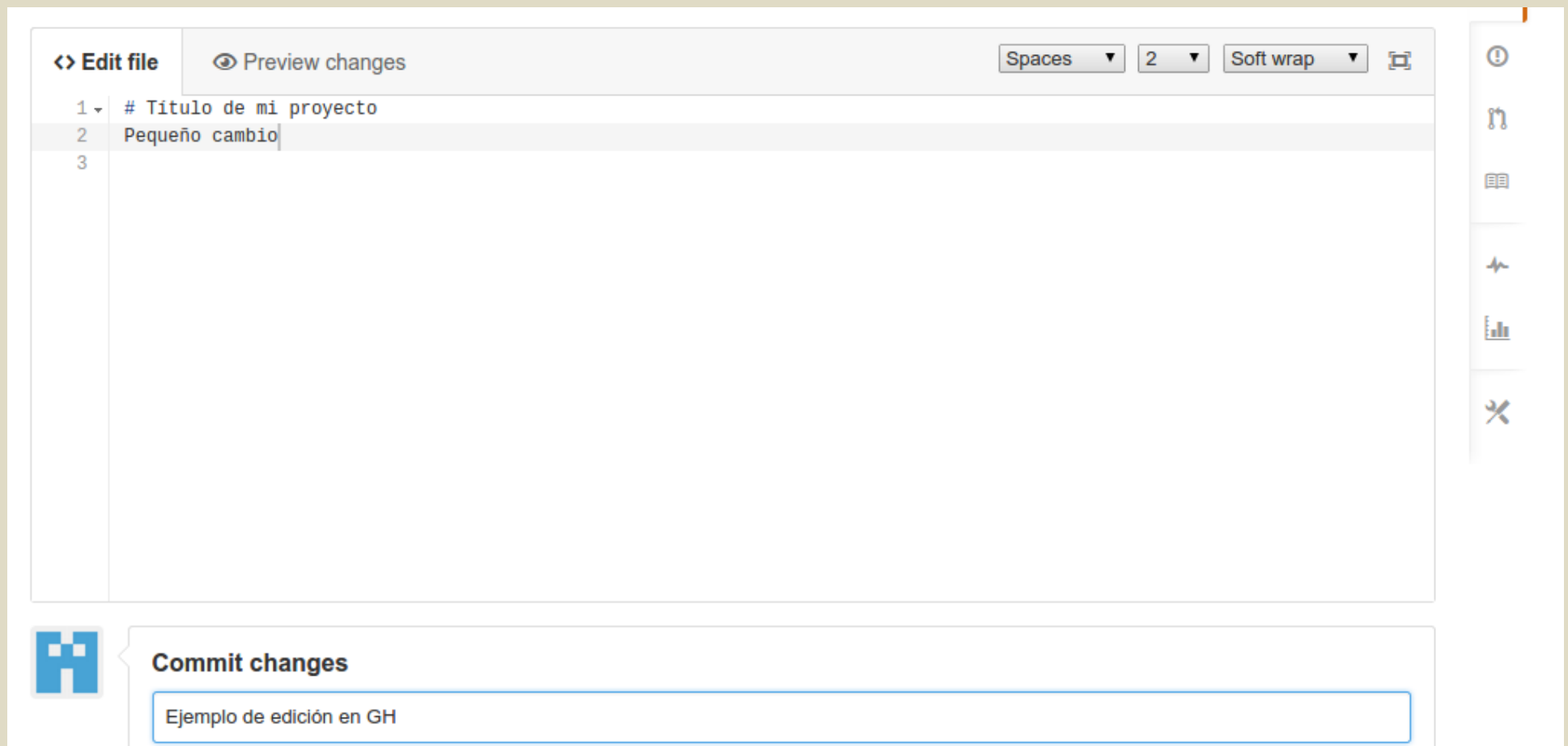
Github



- 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

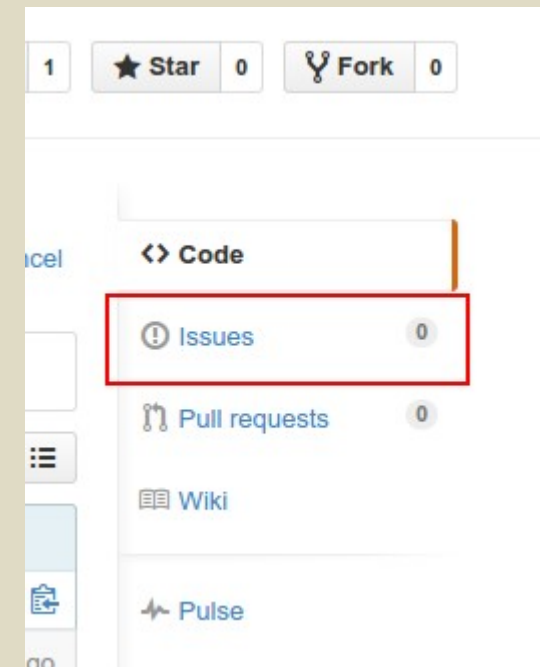
- Hagamos un *push*

```
$ 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 bar 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' dropdown 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'. To the right of the text area, there are links for 'Markdown supported' and 'Edit in fullscreen'. At the bottom of the text area, there is a note: 'Attach images by dragging & dropping, selecting them, or pasting from the clipboard.' A green 'Submit new Issue' button is located at the bottom right of the form.

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

Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

Submit new Issue

Labels: None yet
Milestone: No milestone
Assignee: No one—assign yourself


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 issue page for the repository 'andres-fortier / utn-test'. The issue is titled 'Ejemplo de issue #1' and is marked as 'Open'. It was opened by 'andres-fortier' just now and has 0 comments. 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' (self-assigned). The right sidebar shows the issue's metadata: Labels (None yet), Milestone (No milestone), Assignee (andres-fortier), and Notifications (Unsubscribe). The URL bar at the top shows the issue's link.

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

The screenshot displays the GitHub interface for the repository 'andres-fortier / utn-test'. The address bar at the top shows the URL 'https://github.com/andres-fortier/utn-test/issues'. The repository header includes the GitHub logo, a search bar, and navigation links for 'Explore', 'Gist', 'Blog', and 'Help'. The repository name 'andres-fortier / utn-test' is prominently displayed, along with statistics: 1 Unwatch, 0 Stars, and 0 Forks. Below the repository name, there are tabs for 'Issues', 'Pull requests', 'Labels', and 'Milestones'. The 'Issues' tab is active, showing a search filter 'is:issue is:open' and a 'New Issue' button. The issue list shows 1 Open issue and 0 Closed issues. The first issue, titled 'Ejemplo de issue', is highlighted with a red box. It is marked as a bug (green bug icon) and was opened 17 seconds ago by 'andres-fortier'. A 'ProTip!' at the bottom suggests excluding everything labeled 'bug' with the filter '-label:bug'.

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

andres-fortier / **utn-test**

Issues Pull requests Labels Milestones

Filters is:issue is:open New Issue

1 Open 0 Closed

Ejemplo de issue

#1 opened 17 seconds ago by andres-fortier

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

Cerrando

- No diseñen sus protocolos de seguridad.
 - Estén al día con las implementaciones.
 - Usen librerías conocidas y probadas.

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/>

Links



- <http://security.stackexchange.com/questions/211/how-to-securely-hash-passwords>
- <http://codahale.com/how-to-safely-store-a-password/>
- <http://chargin.matasano.com/chargin/2015/3/26/enough-with-the-salts-updates-on-secure-password-schemes.html>
- <http://chargin.matasano.com/chargin/2007/9/7/enough-with-the-rainbow-tables-what-you-need-to-know-about-secure-password-schemes.html>

Links



- <http://blog.codinghorror.com/speed-hashing/>
- <https://crackstation.net/hashing-security.htm>
- <http://blog.moertel.com/posts/2007-02-09-dont-let-password-recovery-keep-you-from-protecting-your-users.html>
- <http://blog.codinghorror.com/youre-probably-storing-passwords-incorrectly/>
- <http://plaintextoffenders.com/faq/devs>