

# Programando en pilas

author: Hugo Ruscitti

Esta guía resume los pasos necesarios para escribir código en el proyecto pilas.

## Introducción

Ten en cuenta que este documento es bastante extenso por dos motivos: explica algunas cosas básicas que tiene que conocer un programador para trabajar en equipo y contiene varios ejemplos con detalles paso a paso para quienes son nuevos.

Puedes leer esta guía completa, o tenerla de referencia, de cualquier modo te resultará de utilidad si quieres formar parte del equipo de pilas.

## ¿Por qué usamos un control de versiones?

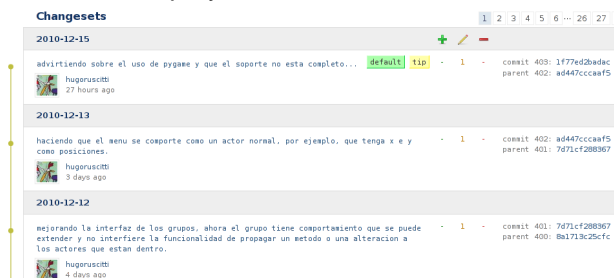
Si tienes experiencia en proyectos seguramente te parezca una pregunta innecesaria, pero es importante comenzar desde aquí para quienes son nuevos.

Pilas es un proyecto de software libre y se desarrolla de manera colaborativa, así que tenemos la necesidad de compartir el código y actualizarlo de manera rápida y confiable.

Un sistema control de versiones permite eso: actualizar el código y compartir con otros usuarios esos cambios de manera confiable, y hasta sencilla.

Básicamente el código lo tenemos en un servidor, le hacemos modificaciones en nuestros equipos y luego subimos los cambios nuevamente al servidor.

El sistema control de versiones nos facilita estas tareas y nos permite tener un historial de cambios donde podemos seguir y evaluar el avance del proyecto:



## Mercurial y bitbucket

Aquí he tomado una decisión tecnológica, y algo subjetiva: usar bitbucket y mercurial...

Existen muchos sistemas control de versiones (git, svn, bazaar etc), pero uno de los mas cercanos a los que usamos python es mercurial, así que me decanté por este sistema.

Mercurial se asemeja mucho a svn, y es uno de los mas sencillos de utilizar, la configuración por defecto es bastante buena y existen versiones para varios sistemas operativos.

En este documento comentaré algunos comandos para comenzar a utilizar mercurial si no tienes mucha experiencia con él.

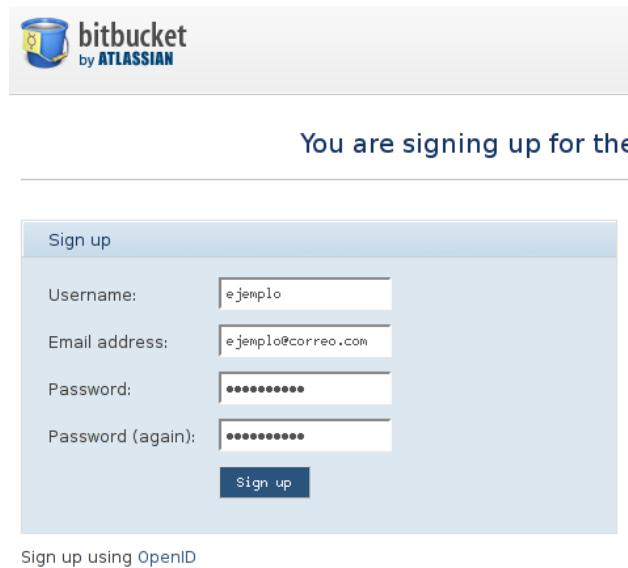
La otra herramienta que he seleccionado es bitbucket, que será nuestro proveedor de alojamiento para el código. El motivo que me impulsó a seleccionar este servicio es su facilidad de uso.

## Darse de alta como usuario

Para comenzar a utilizar bitbucket tienes que ir a la siguiente dirección:

- <http://bitbucket.org/plans>

y dar de alta una cuenta usando alguna de las ofertas. La gratuita es bastante buena:



The image shows the Bitbucket sign-up page. At the top, it says "You are signing up for the". Below this is a "Sign up" form with the following fields: Username (ejemplo), Email address (ejemplo@correo.com), Password (masked with dots), and Password (again) (masked with dots). There is a "Sign up" button at the bottom of the form. Below the form, there is a link that says "Sign up using OpenID".

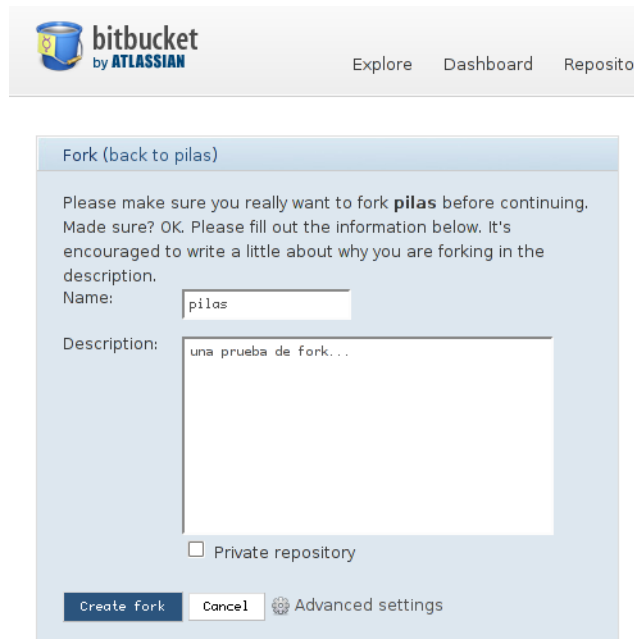
En la imagen de arriba se muestran algunos nombres de ejemplos. Puedes ingresar los datos para tu nueva cuenta ahí, o simplemente pulsar el botón OpenID para ingresar usando la cuenta de otro servicio como gmail.

## Haciendo un fork

Una vez que ingresas en tu cuenta, busca el proyecto pilas que está dentro de la cuenta del usuario hugoruscitti y pulsa el botón fork this repository:



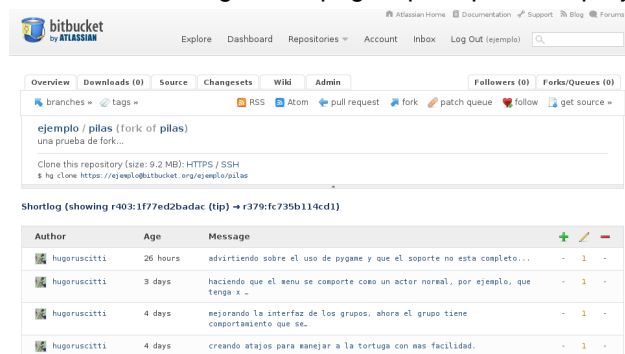
Se te solicitarán datos de este nuevo fork:



Una vez que terminas pulsa el botón `fork`.

## Clonando para comenzar a programar

Ahora que tienes el fork, bitbucket te dirigirá a la página principal de tu proyecto a partir de pilas.



Author	Age	Message	
huguruscitti	26 hours	advirtiendo sobre el uso de pygame y que el soporte no esta completo...	- 1 -
huguruscitti	3 days	haciendo que el menu se comporte como un actor normal, por ejemplo, que tenga x -	- 1 -
huguruscitti	4 days	mejorando la interfaz de los grupos, ahora el grupo tiene comportamiento que se.	- 1 -
huguruscitti	4 days	creando atajos para manejar a la tortuga con mas facilidad.	- 1 -

Lo que ahora tienes que hacer es invocar al comando de mercurial que aparece en pantalla. En el caso de este ejemplo es:

```
hg clone https://ejemplo@bitbucket.org/ejemplo/pilas
```

Luego de ejecutar el comando tienes que ver en la pantalla algo como esto:

```
Destination directory: pilas
requesting all changes
adding changesets
adding manifests
adding file changes
added 404 changesets with 1641 changes to 394 files (+1 heads)
updating to branch default
resolving manifests
getting .hgignore
getting .project
getting .pydevproj
```

```
[...]
333 files updated, 0 files merged, 0 files removed, 0 files unresolved
```

Ahora puedes ingresar al directorio `pilas` y ¡comenzar a programar !.

## ¿Que tenemos hasta ahora?

Si has llegado a este paso realizando cada uno de los comandos de mas arriba, tendrás un directorio llamado `pilas` con tu propio `fork` de `pilas`.

Es momento de comenzar algunas cosas de lo que hemos hecho hasta ahora.

Un `fork` es similar a una copia de código completa, tu directorio `pilas` se puede modificar y alterar como quieras, y cualquier persona podrá colaborar y modificarlo solo si tú le das permiso.

Ese directorio tendrá todo el historial del proyecto, y aceptará tus cambios. Puedes usar comandos como `hg log` o `hg view` para inspeccionar los cambios.

Lo que vamos a hacer ahora es realizar cambios en el código, notificarlos con el comando `commit` y luego subirlos. En la siguiente sección vamos a ver un ejemplo de esto.

## Haciendo cambios

Dentro del directorio `pilas` puedes abrir el archivo que te interesa modificar y aplicar los cambios que creas necesarios.

Vamos a tomar un archivo de ejemplo: "README.rst" y lo vamos a modificar un poco.

Una vez modificado el archivo, puedes consultar el estado de los archivos con el comando `hg st`:

```
>> hg st
M README.rst
```

Y nos mostrará que el archivo `README.rst` ahora está en estado modificado.

Además, si queremos ver exactamente que ha cambiado del archivo podríamos ejecutar el comando `hg diff`:

```
>> hg diff

diff -r 1f77ed2badac README.rst
--- a/README.rst      Tue Dec 14 23:12:44 2010 -0300
+++ b/README.rst      Thu Dec 16 02:04:57 2010 -0300
@@ -4,6 +4,9 @@
     Un motor de videojuegos que presenta una manera sencilla (y algo experimental)
     de hacer videojuegos.

+Pilas está orientado a docentes y estudiantes para que resulte
+mas sencillo crear juegos y hacer proyectos didácticos.
+
Recursos en la web
=====
```

Esto nos indica que se han agregado unas líneas en el archivo y que aún no se han subido al repositorio, solo están en nuestra copia local.

Si queremos notificar el cambio y agregarlo al repositorio tenemos que ejecutar el comando `commit` indicando el cambio que hemos hecho:

```
>> hg ci -m "agregando parrafo mas descriptivo al archivo README."
```

Con eso, mercurial tomará tu cambio y lo procesará para que luego se pueda integrar a la versión oficial.

La dinámica de trabajo con mercurial suele ser así: cada cambio que vamos realizando emite un commit pequeño pero que mejora algo. No es buena idea dispersar esto, los commits pequeños ayudan a tener mas control sobre los cambios y poder identificarlos.

Luego de hacer uno, o varios commits, tenemos que subir nuestras modificaciones a bitbucket. El comando que te permite hacer esto es `push`:

```
>> hg push
```

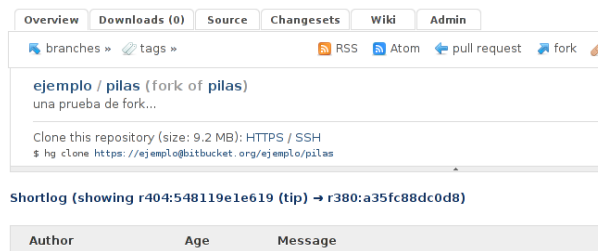
y se te solicitará la contraseña de acceso a bitbucket (al final del artículo comentaré una forma de evitar que siempre la solicite).

## Llevando los cambios a la versión oficial

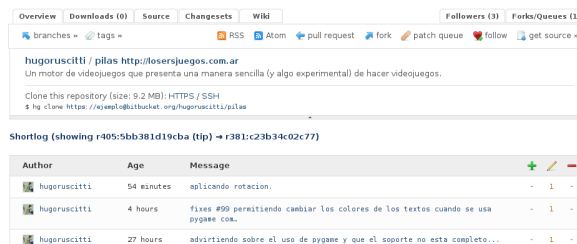
Si te gusta el cambio que has realizado, y quieres que se publique en la versión oficial de pilas tienes que seguir unos pocos pasos.

Primero tienes que asegurarte de haber ejecutado los comandos `commit` y `push` para que tus cambios estén en bitbucket.

Luego tienes que ir a la interface de bitbucket de tu proyecto, hacer click en donde dice pilas (fork of):



y luego pulsar el botón `pull request`.



Esto abrirá una ventana para que le puedas notificar al encargado del proyecto pilas que puede ver tus cambios y subirlos al repositorio oficial:

Send pull request (back to pilas)

You're about to send a pull request to the owners of the repository **pilas**.

Message:

Hola, realicé cambios en el archivo README y quisiera que los suban al repositorio de pilas.

Instead of manually specifying a location above, you can choose one of your repositories on bitbucket:

pilas (2 in, 1 out)

**Notify these users:**

☒ hugoruscitti

☐ hectorsanchez

☐ josejorge

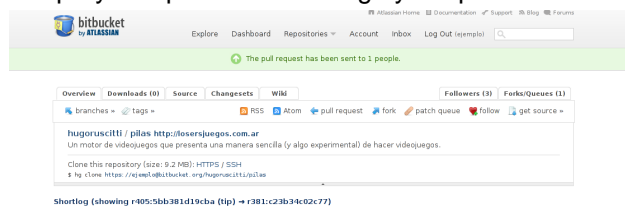
Send request Cancel

Ten en cuenta que tienes que tildar el nombre de la persona a quien quieres avisarle del cambio, posiblemente a hugoruscitti.

Si tus cambios están interesantes las vamos a subir en poco tiempo al repositorio original, sino te avisamos que se puede cambiar...

## Esperar la respuesta

Listo, cuando veas el cuadro que indica que has notificado el cambio, solo queda esperar (o insistir por correo), hasta que alguien del proyecto pilas vea tu código y lo apruebe.



Ten en cuenta que tu repositorio de fork de pilas puedes configurarlo como público también, así te recomendamos que avises en el foro de mensaje y así mas personas pueden colaborar y ver qué avance tiene el proyecto.

## Anexo: Manteniendo tu fork actualizado

Si quieres tener los últimos cambios de la versión oficial de pilas, tienes que hacer un pull desde el repositorio de hugo. Ejecuta el siguiente comando:

```
hg pull http://bitbucket.org/hugoruscitti/pilas
```

Eso traerá los cambios mas recientes a tu fork.

Puede que tengas que escribir algo cómo:

```
hg heads .
```

y luego identificar lo que quieres fusionar haciendo algo cómo esto:

```
hg merge 400
```

Pero no es tan importante, a veces no ocurre y ya.

## Conclusión

Mercurial y bitbucket son dos de las herramientas que usamos en pilas para desarrollar el código y la documentación del proyecto.

Te invito a que colabores con el proyecto, ya sea programando o sugiriendo mejoras, avisando a otras personas que existe pilas, mejorando este documento o escribiendo juegos.

Hay mucho por hacer, y necesitamos tu ayuda.

## Anexo: ¿como puedo guardar mi contraseña de acceso?

Mercurial te solicitará la clave de usuario en bitbucket cada vez que realices el comando `push`. Esto puede ser un poco molesto si lo haces con frecuencia.

Para evitar que bitbucket te pida la contraseña cada vez, puedes editar el archivo `".hg/hgrc"`, y colocar tu contraseña antes el arroba.

Ejemplo sin contraseña:

```
[paths]
default = https://ejemplo:@bitbucket.org/ejemplo/pilas
```

Ejemplo con contraseña:

```
[paths]
default = https://ejemplo:micontraseña@bitbucket.org/ejemplo/pilas
```

En donde `micontraseña` tienes que reemplazarla por la contraseña de tú usuario bitbucket.