

Tutorial Monticello Básico

[Tutorial Monticello Básico para PDEP](#)

[¿Qué problema resuelve Monticello?](#)

[¿Qué es Monticello?](#)

[¿Paquetes?](#)

[¿Como usamos Monticello?](#)

[Descargando un Proyecto Existente](#)

[Haciendo y Grabando cambios](#)

[Usuario de SqueakSource](#)

[Creando un nuevo proyecto](#)

[¿Que otras cosas puedo hacer con Monticello?](#)

¿Qué problema resuelve Monticello?

Monticello resuelve los mismos problemas que resuelve cualquier herramienta de versionado y repositorio de código. Esas herramientas nos permiten guardar el código que escribimos en un lugar (normalmente en Internet) donde **todos** puedan acceder a la última versión. Son herramientas que sirven para los **desarrolladores** de cualquier aplicación, y también para los **usuarios** de la misma. Es decir, suelen ser usadas como herramientas para **trabajo en equipo**, aunque tienen muchas bondades que nos hacen elegir las aún si el desarrollador es uno sólo. Por ejemplo, para descargar **cualquier** versión anterior, ver modificaciones, y conocer la historia de las mismas, lo que mantiene el desarrollo más ordenado.

Monticello, entonces, es la herramienta que utiliza Pharo Smalltalk (y Squeak) con todos éstos propósitos.

¿Qué es Monticello?

Monticello es una herramienta de versionado de paquetes. Usando Monticello, se pueden guardar paquetes a repositorios de varios tipos de servidores (FTP, HTTP, etc) permitiendo la existencia de varias versiones de tu programa, fusionar versiones, volver a versiones anteriores, y explorar las diferencias entre versiones. De hecho, Monticello es un sistema distribuido de control de revisiones: permite a los programadores guardar su trabajo en distintos lugares, y no en un solo repositorio como cuando se usa CVS o Subversion. La idea es poder compartir y modificar código de una manera más rápida y accesible que, por ejemplo, haciendo file outs.

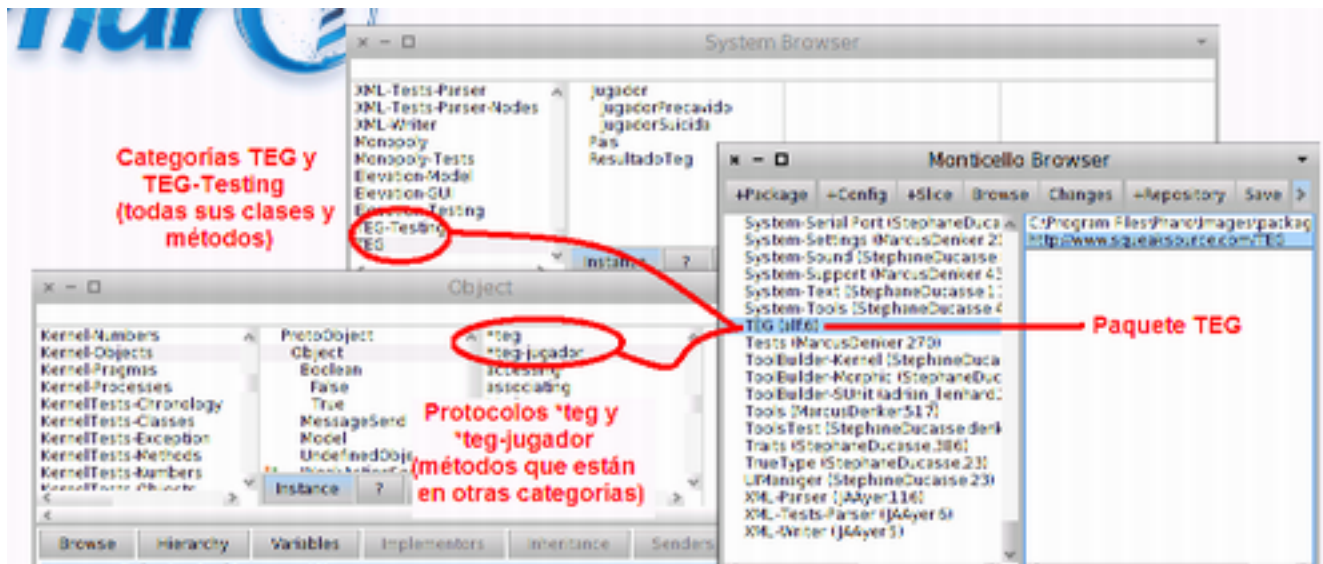
¿Paquetes?

Como Monticello maneja paquetes, seguramente se estén preguntando ¿que es un paquete? Un paquete es mas o menos equivalente a una categoría en el System Browser. El sistema de paquetes es una manera sencilla de organizar el código, usando una convención para nombrar categorías y protocolos.

Un **protocolo** es una colección de métodos relacionados, colocados en un paquete aunque el resto de los métodos de la clase a la que pertenecen esté en otro paquete.

Una **categoría** es una colección de clases relacionadas en una imagen.

Un **paquete** es una colección de clases relacionadas y *métodos de extensión* que pueden ser versionados usando Monticello.



Por convención, el nombre del paquete y de la categoría es el mismo. Las reglas de “qué va en un paquete” son:

Un paquete **Blah** contiene:

- 1) Todas las *definiciones de clases* en la categoría **Blah**, o en categorías con nombres que empiezan con **Blah-**
- 2) Todos los *métodos* en *cualquier clase* en protocolos con nombre ***Blah** o ***blah** o con nombres que empiecen con ***Blah-** o ***blah-**
- 3) todos los *métodos* en clases de la categoría **Blah**, o en una categoría cuyo nombre empiece con **Blah-**, *excepto* aquellos métodos que estén en protocolos cuyos nombres empiecen con *****.

Como consecuencia de estas reglas, cada definición de una clase pertenece *exactamente a un solo paquete*.

Los paquetes son archivos comprimidos que contienen el código completo de tu programa, tienen una extensión **.mcz**.

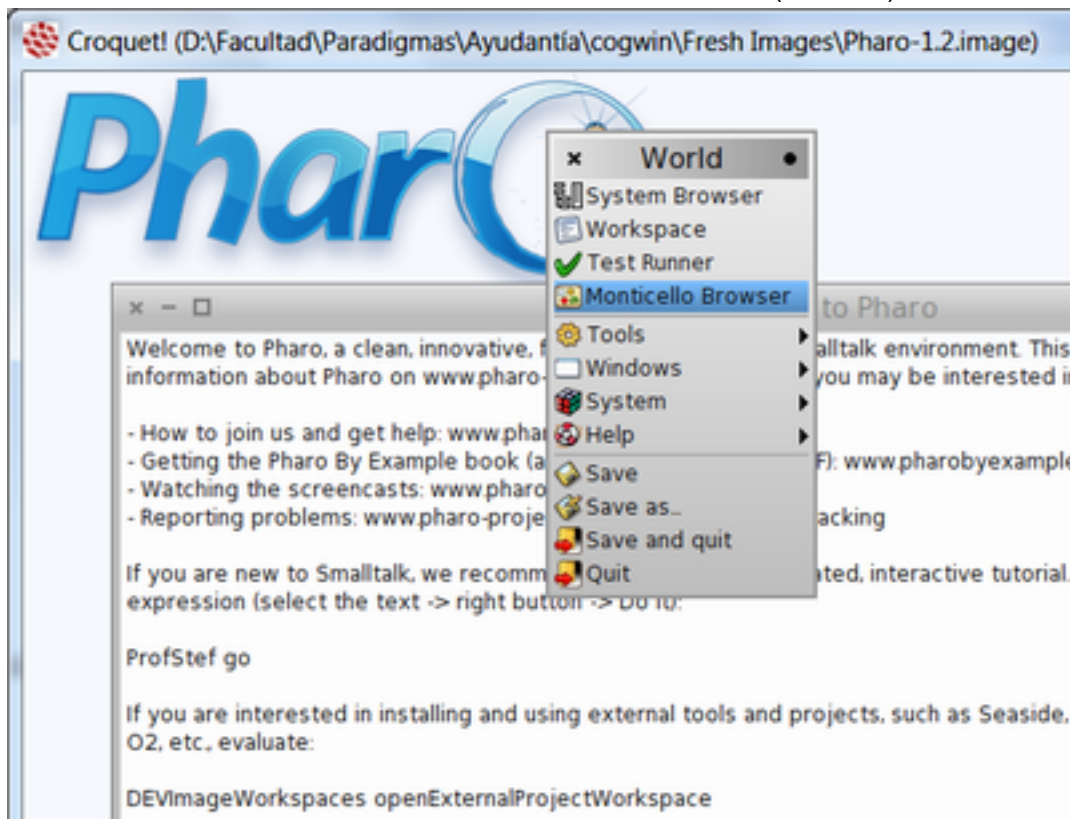
Por defecto, todos los cambios que se hacen a cualquier categoría o paquete existente se guardan en un repositorio local, el *package-cache*, lo cual es muy útil para trabajar en lugares donde no hay acceso a Internet por ejemplo, y para mantener un registro local de las versiones. El *package-cache* suele estar ubicado en el directorio en el que tenemos guardada la imagen, aunque cada usuario puede especificar en que directorio quiere que se encuentre

¿Como usamos Monticello?

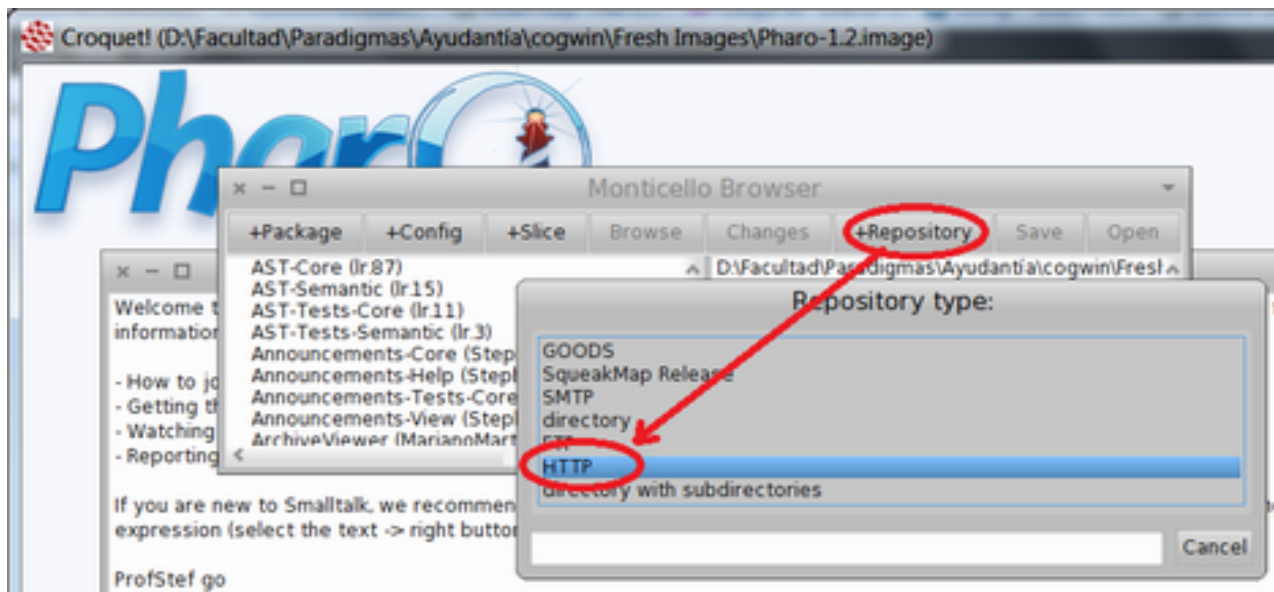
Descargando un Proyecto Existente

Mostraremos cómo se descarga un proyecto ya existente en Internet (en éste caso en www.squeaksource.com) a la imagen actual. El procedimiento es similar para descargar del repositorio local (*package-cache*).

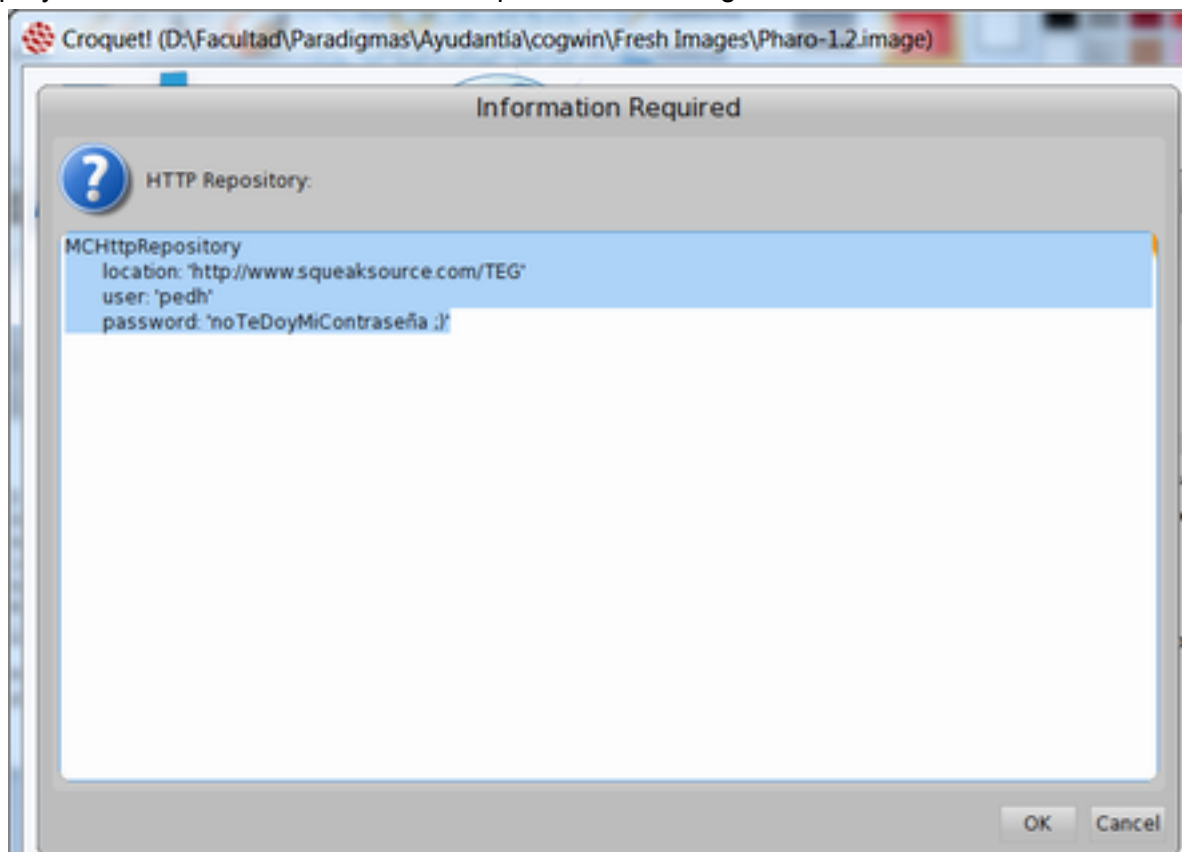
1. Abrir el Monticello Browser, haciendo click en el “World” (el fondo).



2. Agregar el repositorio del proyecto haciendo click primero en +Repository y luego en HTTP:



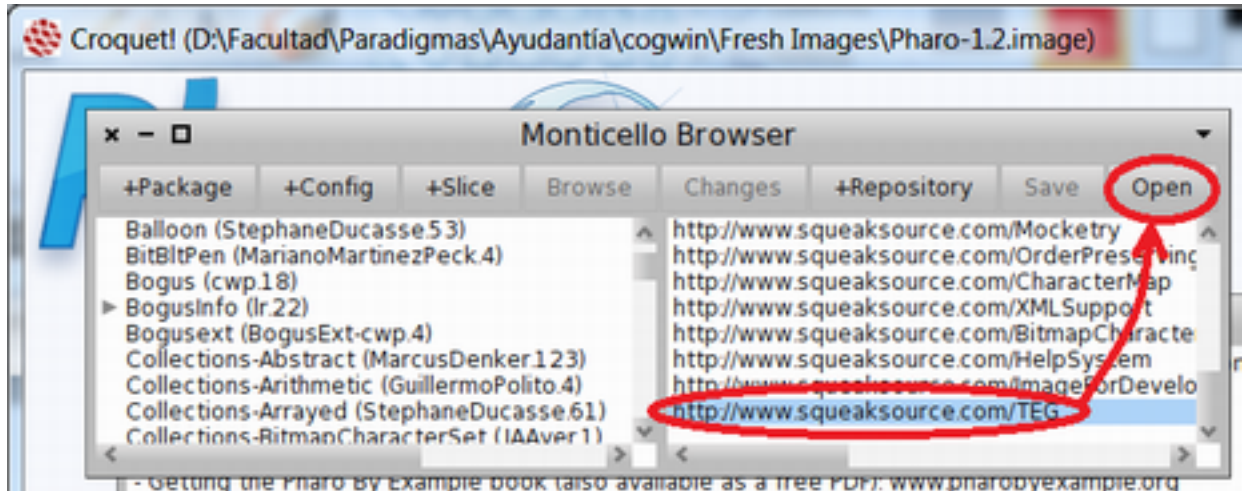
3. Ingresar los datos del repositorio.¹ En este ejemplo, se utiliza la URL de un proyecto llamado TEG hosteado en Squeaksource. Luego, hacer click en



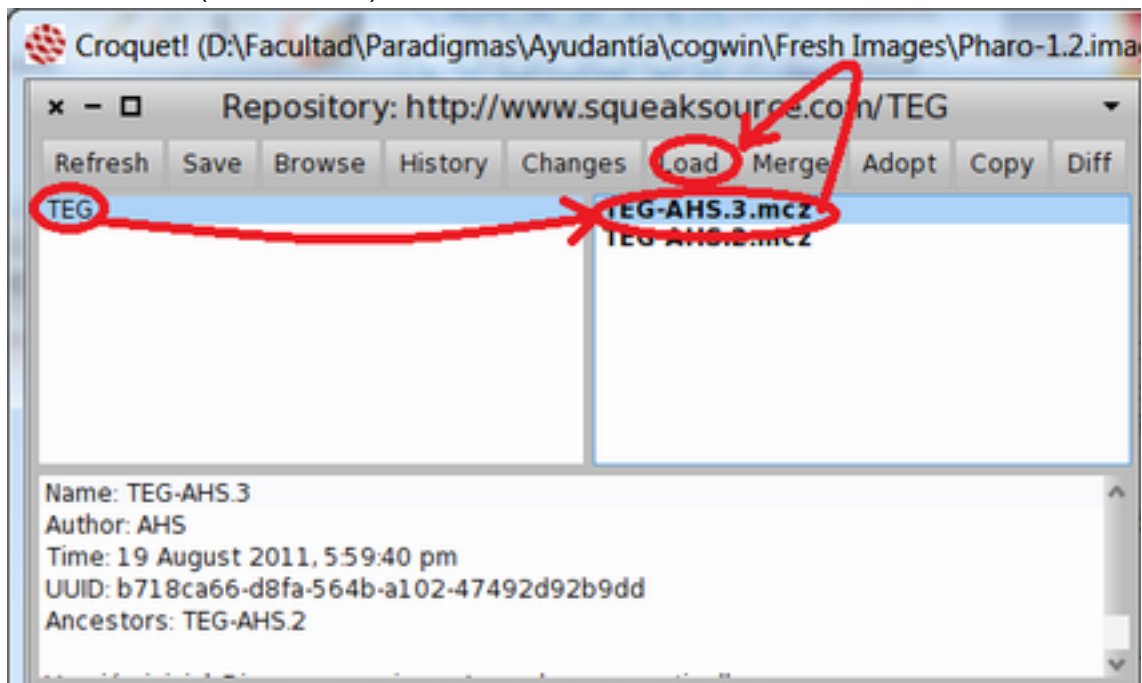
OK.

¹Si el repositorio fuera de Squeaksource, y el proyecto no fuera público, será necesario primero crear un [usuario en SqueakSource](#)

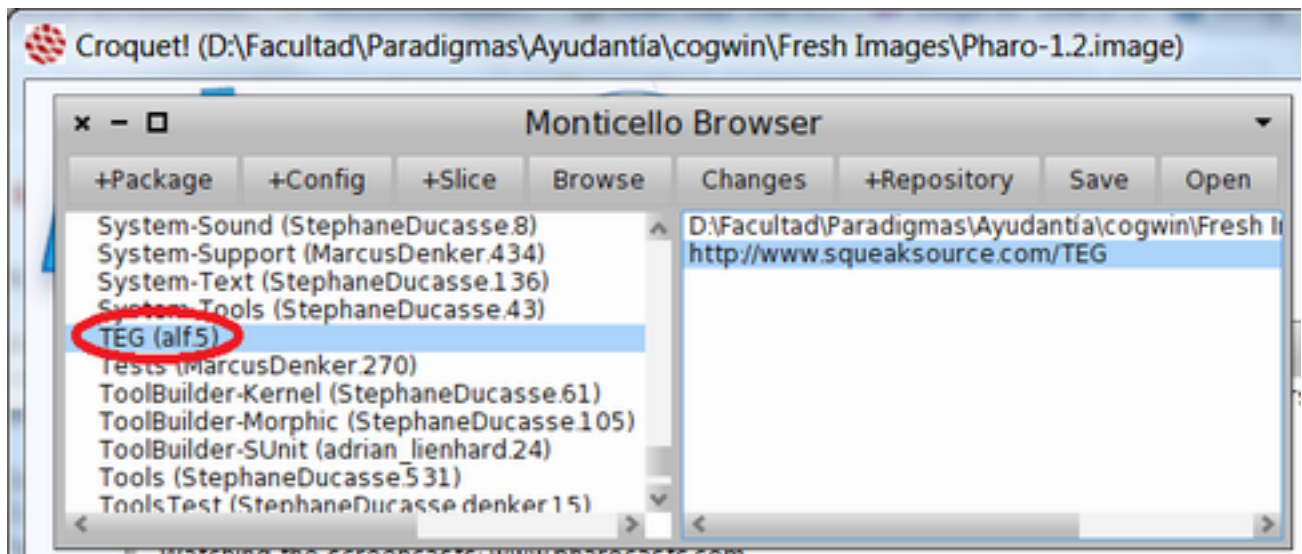
4. El repositorio debería aparecer en la lista de la derecha. Hacer click en el repositorio, y luego en "Open"



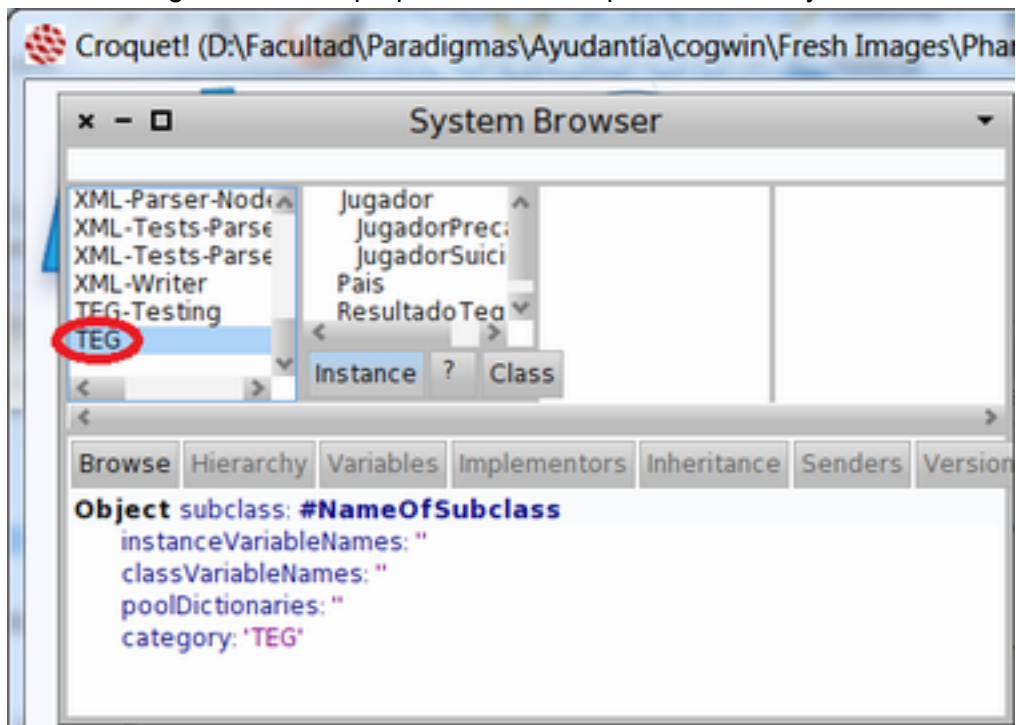
5. Seleccionar primero el paquete (a la izquierda) y luego la versión de ese paquete que se desea instalar (a la derecha). Finalmente hacer click en Load.



6. El paquete debería aparecer en el Monticello Browser, a la izquierda, con sus repositorios a la derecha



7. Y las categorías de ese paquete deberían aparecer en el System Browser.



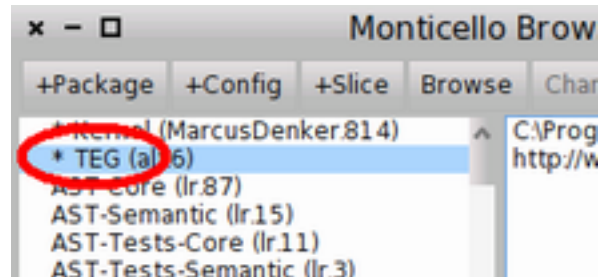
8. Con esto las clases ya estarían listas para ser usadas! Acá ven lo que les decía de las reglas de “que va en un paquete”: Cuando instalaron el paquete **TEG**, aparecieron las categorías **TEG** y **TEG-Testing** (que es el conjunto de tests de SUnit para la aplicación).

Haciendo y Grabando cambios

Los cambios se hacen normalmente programando sobre la imagen, siempre recordando que todos los métodos que escribamos fuera de nuestra categoría (y querramos que se suban al repositorio) hay que ponerlos bajo el protocolo *<nombre>, como se ejemplificó en la sección “¿Paquetes?”.

- Para grabar un cambio al repositorio local, basta entrar al Monticello Browser, hacer click en el paquete, luego en el repositorio package-cache, y darle click a Save.
- Para grabar un cambio a un repositorio en Internet (por ejemplo SqueakSource) es el mismo procedimiento, pero seleccionando, en vez de package-cache, el repositorio correspondiente. Nota: Es necesario tener un [usuario en SqueakSource](#) y permiso para modificar el proyecto.

Vale la pena notar que toda vez que un paquete se modifique, y difiera de la versión en el repositorio, ésto se indicará con un asterisco (*) antes del nombre del paquete:

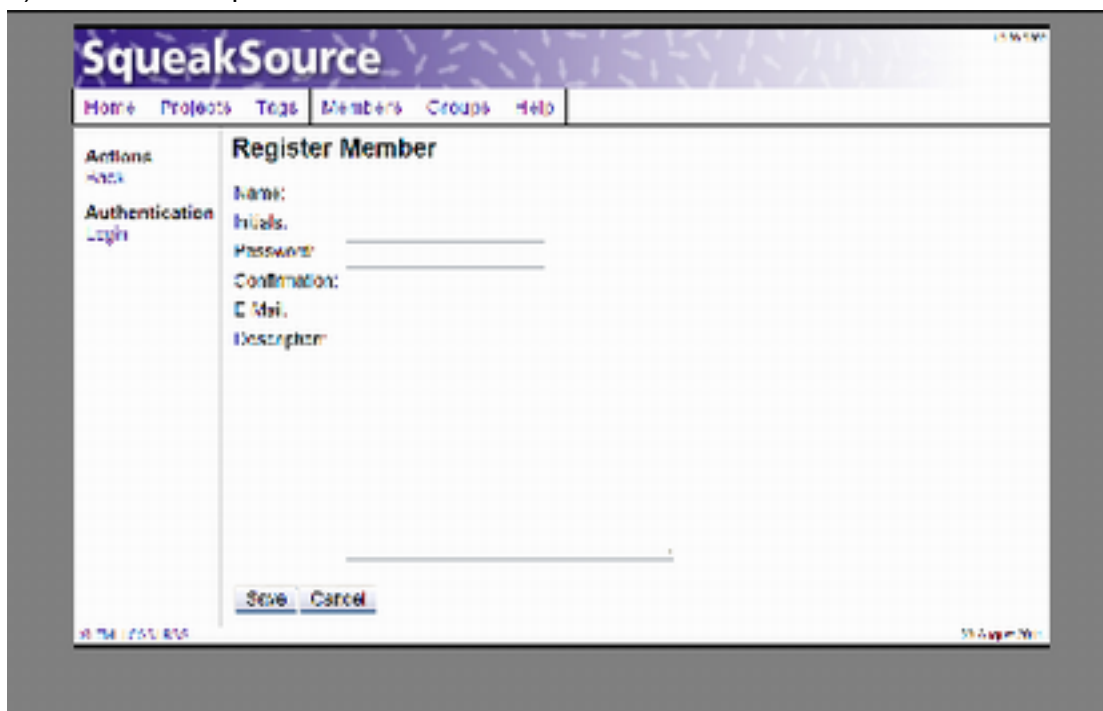


Usuario de SqueakSource / SqueakSource3 (En 3 sencillos pasos! llame ahora, llame ya!)


IMPORTANTE: Actualmente para crear proyectos en SqueakSource hay que mandar un mail al administrador, ya que debido a la inestabilidad del server se discontinuó su uso. En su lugar, deberían registrarse y crear proyectos en [SqueakSource 3](#).

1) En la página principal de SqueakSource (<http://www.squeaksource.com>) o en la de SqueakSource3 (<http://ss3.gemstone.com>) bajo el Menu **Actions** (en el panel de la izquierda) ir a **Register Member**.

2) Van a ver una pantalla como esta:



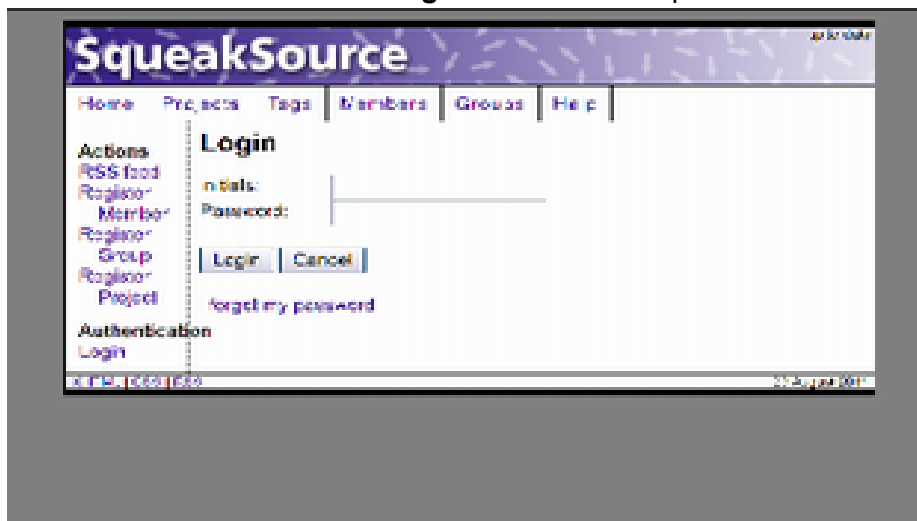
Y una como esta en SqueakSource3:



The screenshot shows the 'Register Member' page on SqueakSource3. The page has a header with the SqueakSource3 logo and a 'Not logged in' status. A navigation bar includes links for Home, Projects, Tags, Members, Groups, and Help. On the left, there is a sidebar with 'Actions' (Back, RSS Feed), 'Authentication' (Login), and 'Register Member'. The main content area is titled 'Register Member' and contains a form with fields for Name, Login, Password, Email address, and Description. There are 'Save' and 'Cancel' buttons at the bottom of the form.

Completan con sus datos y le dan **Save**. Recuerden que el campo **Initials (o Login)** es su nombre de usuario en SqueakSource.

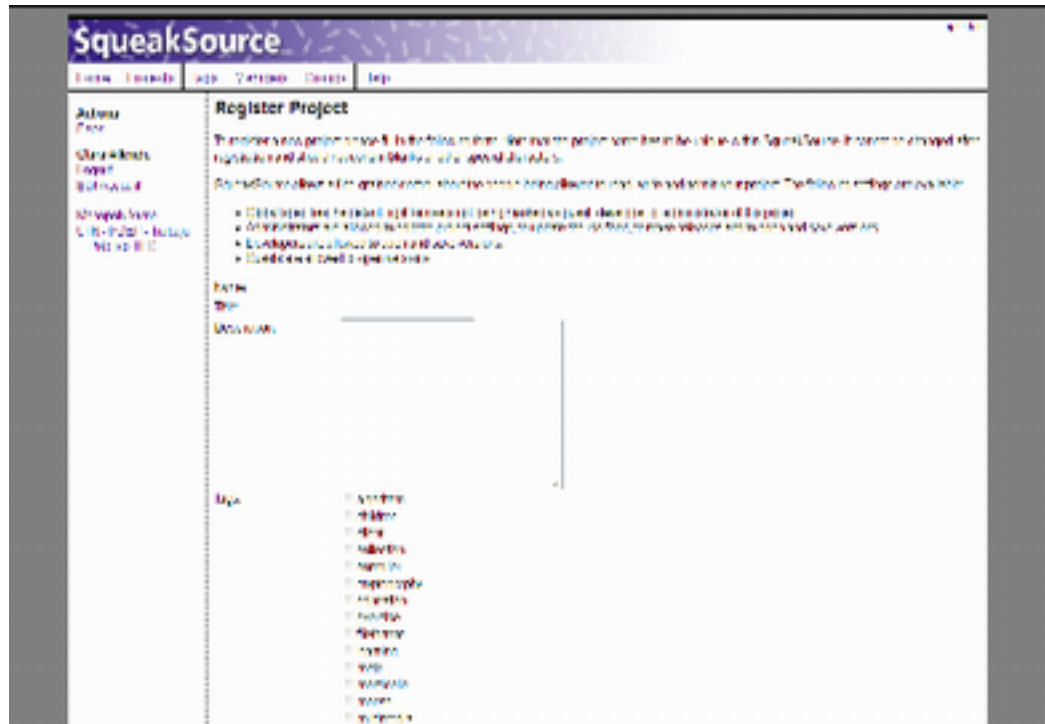
3) Una vez que guardaron los cambios, van a volver a la pantalla principal, y ya van a estar logueados :D De ahora en más, cada vez que entren a SqueakSource deberan loguearse, para eso van a **Authentication**→ **Login**. Van a ver una pantalla asi:



The screenshot shows the 'Login' page on SqueakSource. The page has a purple header with the SqueakSource logo and a 'Not logged in' status. A navigation bar includes links for Home, Projects, Tags, Members, Groups, and Help. On the left, there is a sidebar with 'Actions' (RSS feed, Register, Member, Register, Group, Register, Project), 'Authentication' (Login), and 'Login'. The main content area is titled 'Login' and contains a form with fields for Initials and Password. There are 'Login' and 'Cancel' buttons, and a link for 'forgot my password'.



The screenshot shows the 'Login' page on SqueakSource. The page has a white header with the SqueakSource3 logo and a 'Not logged in' status. A navigation bar includes links for Home, Projects, Tags, Members, Groups, and Help. On the left, there is a sidebar with 'Actions' (RSS Feed, Register Member), 'Authentication' (Login), and 'Login'. The main content area is titled 'Login' and contains a form with fields for Login and Password. There are 'Login' and 'Cancel' buttons, and a link for 'I forgot my password'.



Recomiendo leer la explicación que se da sobre la creación de un proyecto. Completen los campos, en particular, además del nombre y título del proyecto (uno es el nombre con el que se guarda en SqueakSource y el otro es el que queda en la URL del repositorio) y la descripción, nos interesan los campos del final:

Acá en SqueakSource3 cambió un poquito la UI, pero es básicamente lo mismo.

License: MIT

Enable Blessing: ☐ Enable Blessing

Enable Issue Tracking: ☒ Enable Issue Tracking

Global Right: Read

Administrators:

Available:	Selected:
Admin Brown	Chris Alford
Adrian Barlow	
Alexander Vurch	
Alexander Bergel	
Andrius Koles	
Angel H. Nunez	

Developers:

Available:	Selected:
Admin Brown	
Adrian Barlow	
Alexander Vurch	
Alexander Bergel	
Andrius Koles	
Angel H. Nunez	

Default move target:

Parent Project:

Guests:

Available:	Selected:
Admin Brown	
Adrian Barlow	
Alexander Vurch	
Alexander Bergel	
Andrius Koles	
Angel H. Nunez	

Who can move versions: Developers

Subscriptions: ☐ Email Subscription

Also send Emails to: Address
The input is empty.

Send Email to: ☐ admins ☐ developers ☐ guests

Reply-To Address:

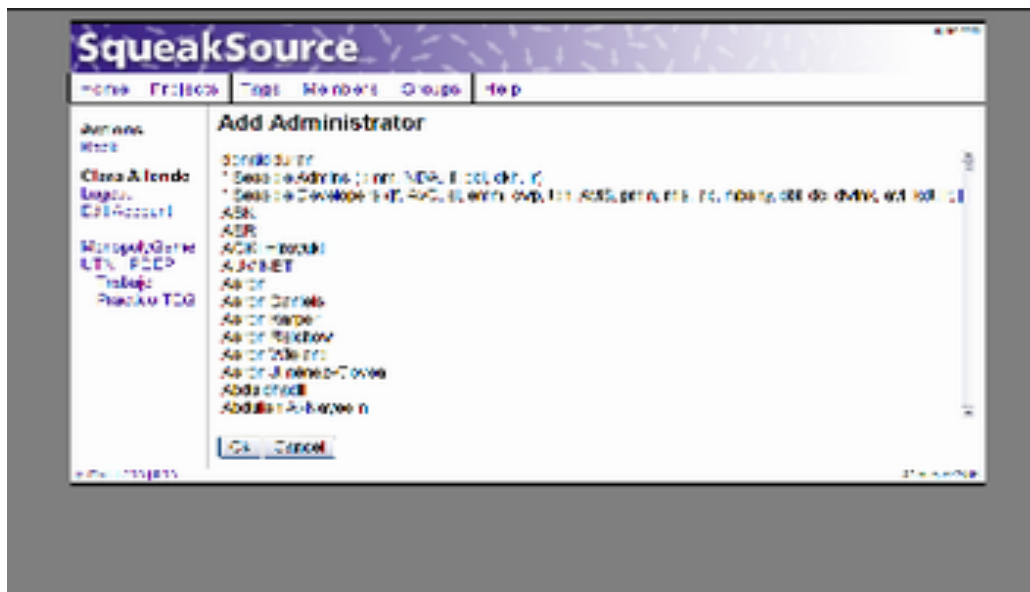
Save Cancel

En **License** van a encontrar varias opciones, pongan **MIT** (que es la licencia que tiene Pharo).

En **Global (o Global Right)** es donde ustedes definen los permisos que tienen las personas (TODAS las personas) que descarguen su proyecto (**Read, Read/Write, No Access**). Por defecto, la opción seleccionada es Read, es decir, permisos de lectura.

Además, pueden especificar una dirección de correo a la cual recibir notificaciones cuando se publica una nueva versión al repositorio (es decir, cuando hay un *commit*), activando el checkbox en **Subscriptions**.

En **Administrators, Developers y Guests**, si hacen click en el link **add** van a ver algo así:

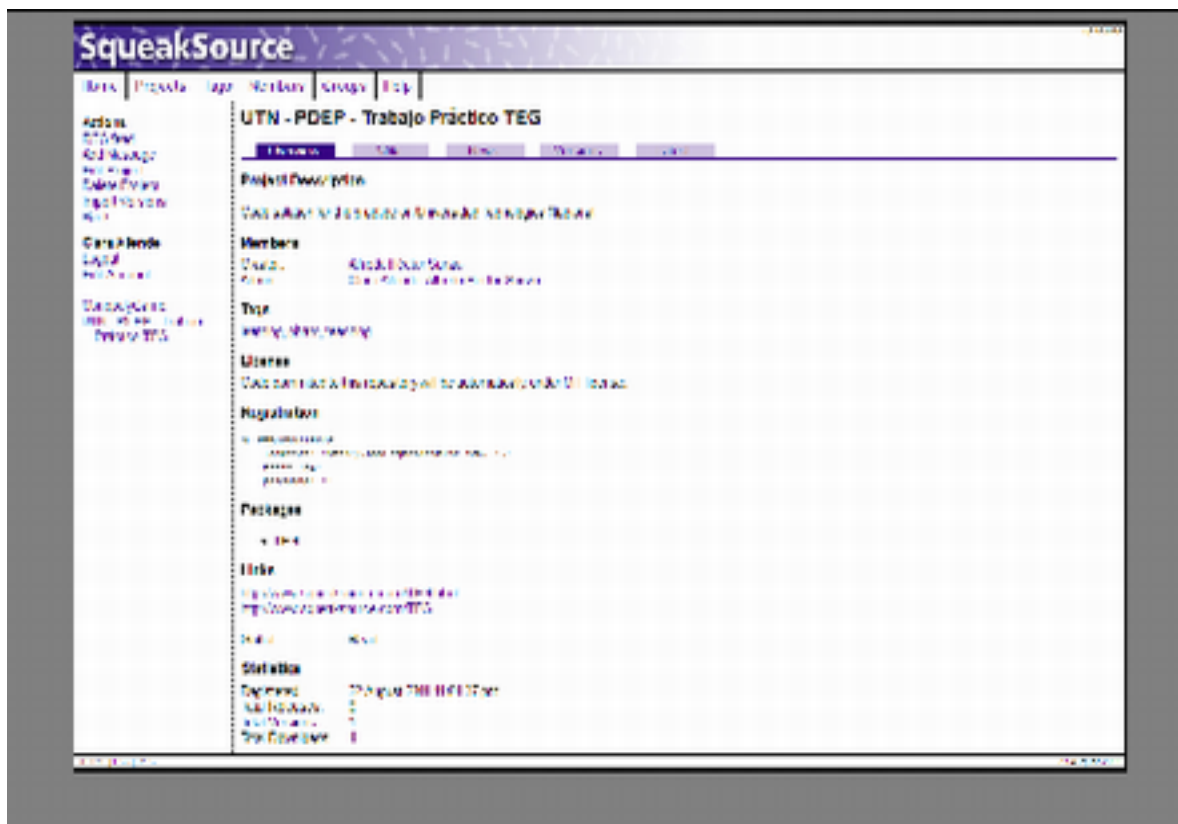


Que no es más que una lista de todos los usuarios registrados en SqueakSource :) De ahí seleccionan a aquellos a los que quieran dar permisos de administrador, desarrollador o invitado (según corresponda) y le dan **OK**.

En SqueakSource3 tienen los menús tipo combo box donde pueden elegir un usuario y cambiarlo con las flechitas (>> y <<)

Una vez que completaron todos los campos, le dan **Save**.

3) Ya tienen su proyecto creado! Ahora van a estar en la página de su proyecto, que se ve más o menos así:



SqueakSource ³ Release 3.10.0

Clara Allende • Logout

Home Projects Tags Versions Groups Map

PharoInbox

Actions
— Back
RSS Feed
Project Feed
Import versions
Submit issue
Add to favorites

Clara Allende
Logout
Edit Account

My Projects
mainio:TADP

Favorites

Overview Wiki News Issues Versions Latest Statistics

Project Description

inbox for pharo

Members

Creator: ducasse
Admin: Marcus Denker, Damien Cassou, ducasse

Tags

This project doesn't yet have tags.

License

Code committed to this repository will be automatically under MIT license.

Getting the Code

Monticello Registration Code Installer

Links

<https://s3.gemstone.comiss/PharoInbox.html>
<http://s3.gemstone.comiss/PharoInbox>

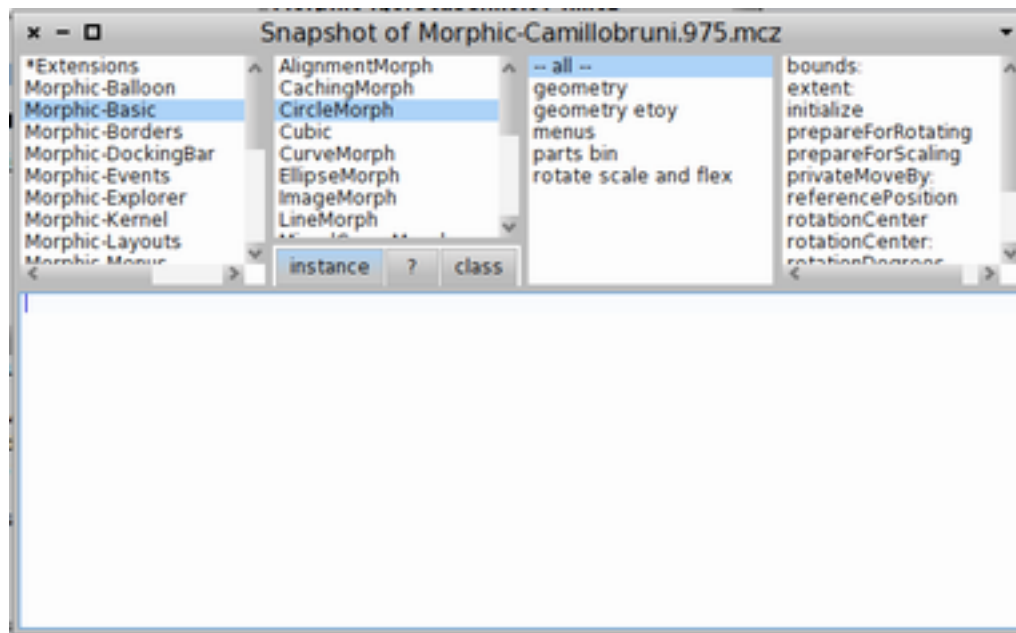
Permissions

Global: Read And Write

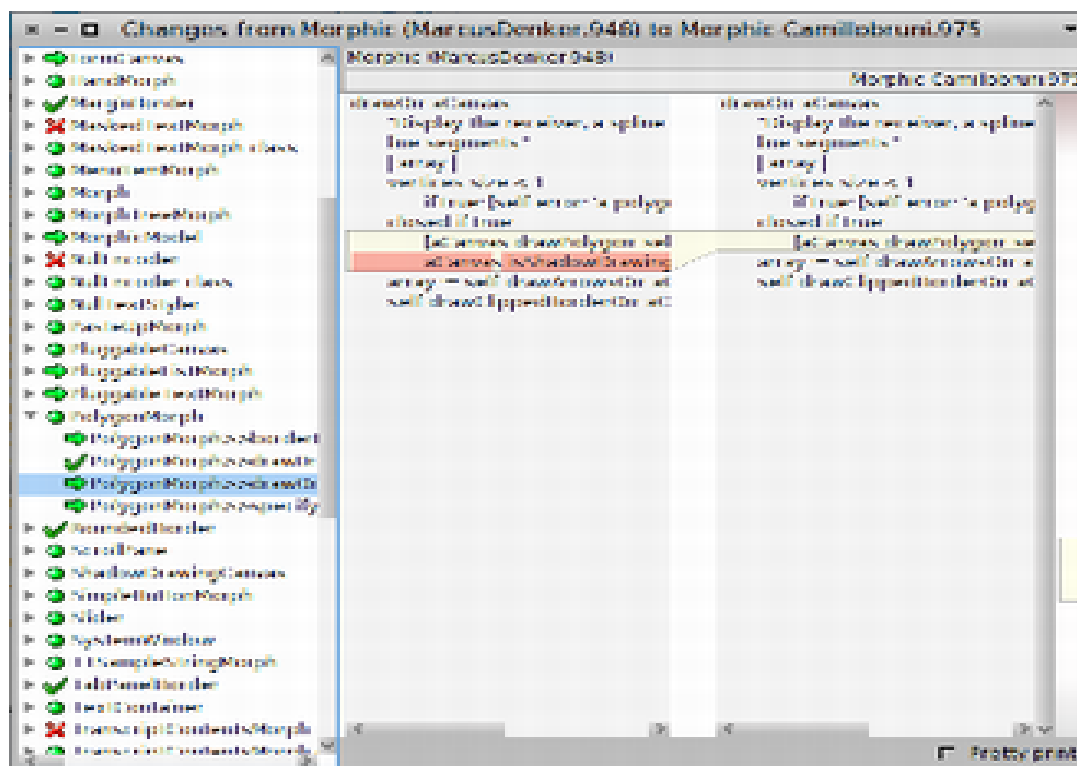
¿Que otras cosas puedo hacer con Monticello?

Con una versión seleccionada:

Browse permite ver la aplicación sin cargarla a la imagen.



Changes permite ver los cambios que se harán en la imagen al cargar la versión.



Copy permite hacer una copia de una versión, que después se puede poner en otro repositorio.

Refresh actualiza la vista del repositorio.

Merge carga solo los cambios que introduce la versión seleccionada respecto a la versión actual.

