

# 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.

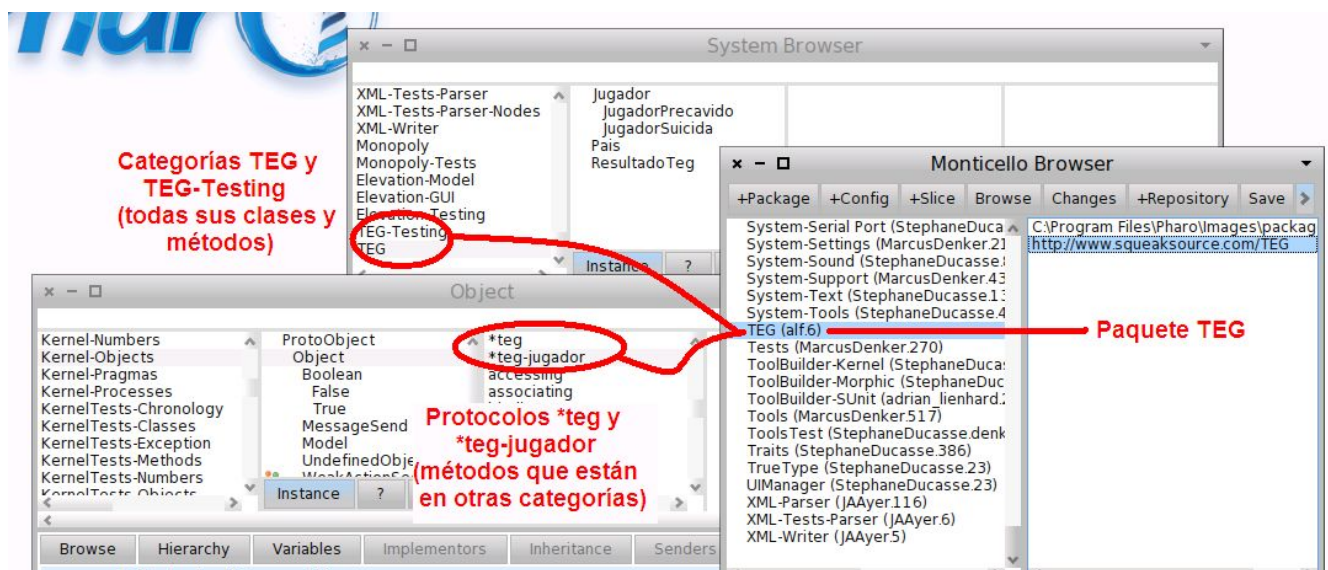
## ¿Paquetes?

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**.

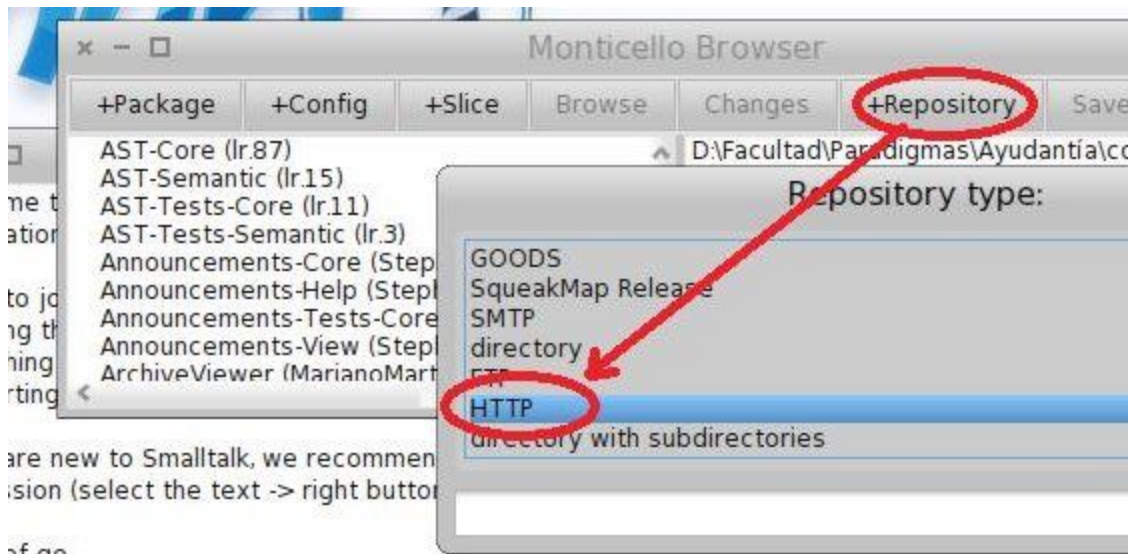
Por defecto, todos los cambios que se hacen a cualquier categoría o paquete existente se guardan en un repositorio local, el *package-cache*, lo cual es muy útil para trabajar en lugares donde no hay acceso a Internet por ejemplo, y para mantener un registro local de las versiones. El *package-cache* 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?

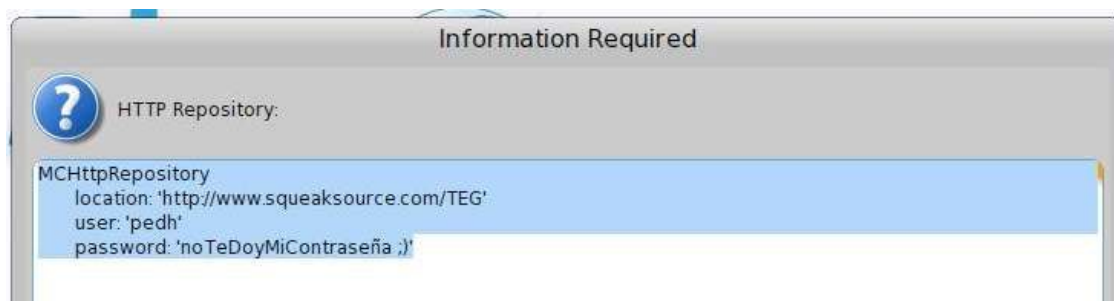
### Descargando un Proyecto Existente

Mostraremos cómo se descarga un proyecto ya existente en Internet (en éste caso en [www.squeaksource.com](http://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), y luego en “Monticello Browser”.
2. Agregar el repositorio del proyecto haciendo click primero en +Repository y luego en HTTP:

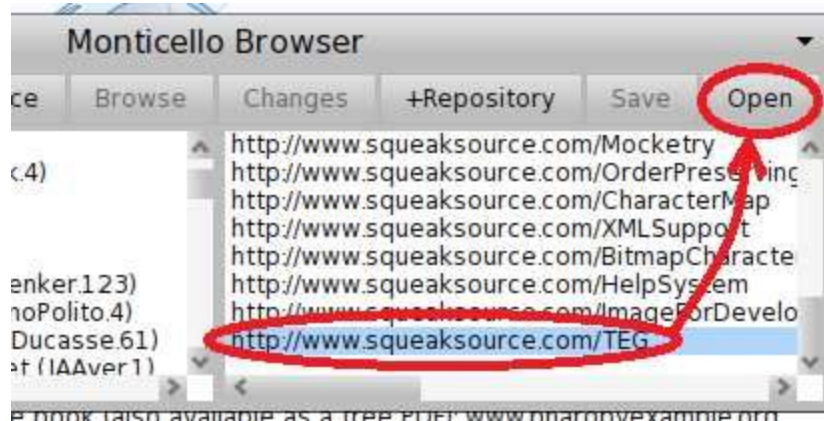


3. Ingresar los datos del repositorio.<sup>1</sup> En este ejemplo, se utiliza la URL de un proyecto llamado TEG hospedado en Squeaksource. Luego, hacer click en OK.

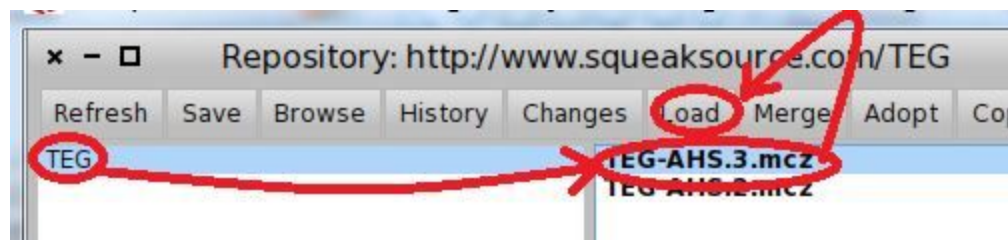


<sup>1</sup>Si el repositorio fuera de Squeaksource, y el proyecto no fuera público, será necesario primero crear un [usuario en SqueakSource](#)

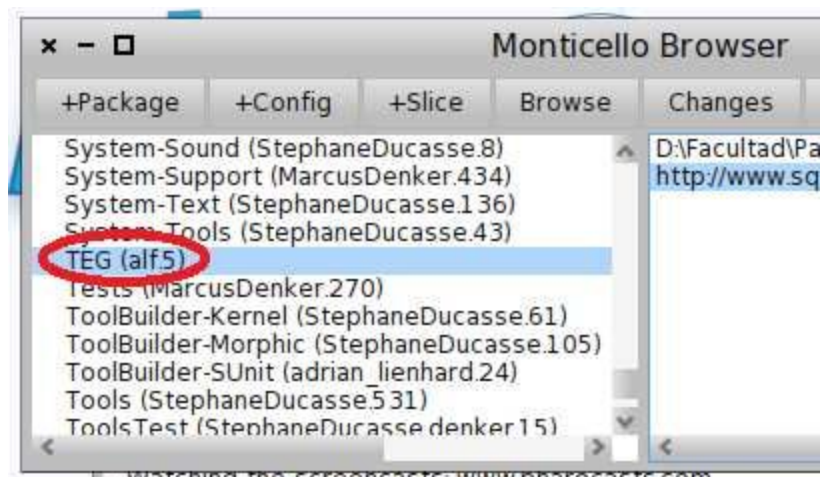
4. El repositorio debería aparecer en la lista de la derecha. Hacer click en el repositorio, y luego en "Open"



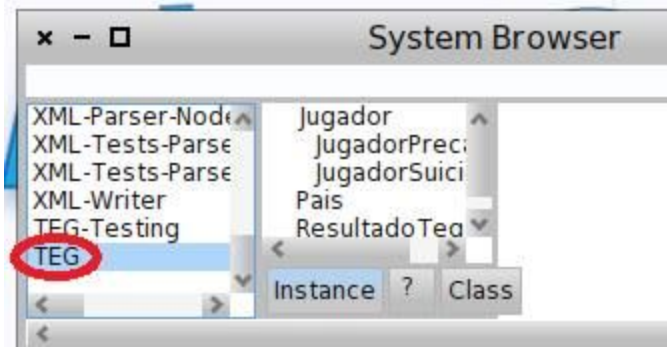
5. Seleccionar primero el paquete (a la izquierda) y luego la versión de ese paquete que se desea instalar (a la derecha). Finalmente hacer click en Load.



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.



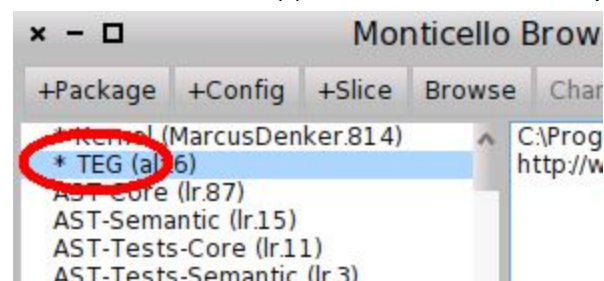
8. Con esto las clases ya estarían listas para ser usadas! Acá ven lo que les decía de las reglas de “que va en un paquete”: Cuando instalaron el paquete **TEG**, aparecieron las categorías **TEG** y **TEG-Testing** (que es el conjunto de tests de SUnit para la aplicación).

## 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 descontinuó su uso. En su lugar, deberían registrarse y crear proyectos en [SqueakSource 3](http://www.squeaksource.com), o en el más nuevo [SmalltalkHub](http://ss3.gemstone.com).

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, en SqueakSource3:

SqueakSource3 <sup>Public Alpha</sup>

Home Projects Tags Members

**Register Member**

Actions  
← Back  
RSS Feed

Authentication  
Login

Name:

Login:

Password:

Email address:

Description:

Save Cancel

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 deberán loguearse, para eso van a **Authentication**→ **Login**. Van a ver una pantalla así:

SqueakSource3 <sup>Public Alpha</sup>

Home Projects Tags Members Groups Help

**Login**

Actions  
RSS Feed  
Register Member

Authentication  
Login

Login:

Password:

Login Cancel

[I forgot my password](#)

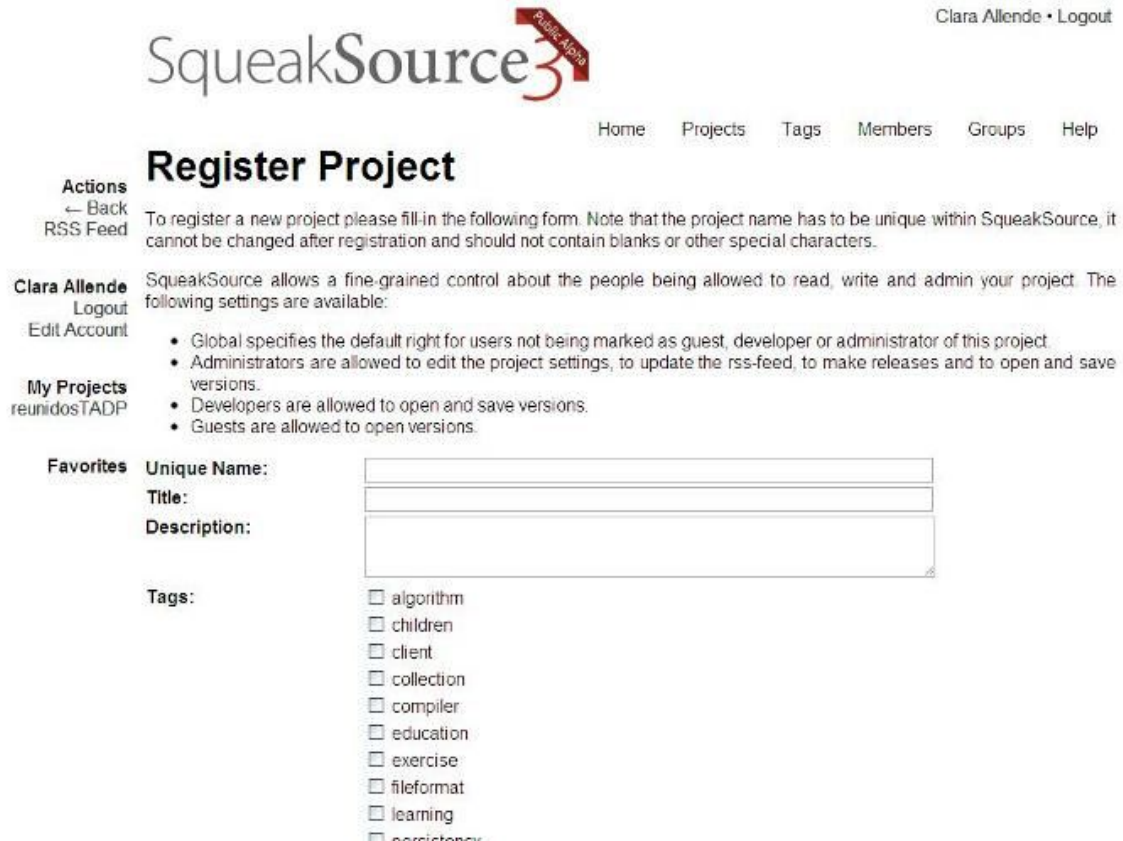
Completan con su usuario y contraseña y *voilà!* ya pueden crear un proyecto, ver los proyectos hospedados, etc. Una vez que tengan un proyecto (o alguien mas les de permisos de escritura o administracion de un proyecto), además van a poder ver un link a ese proyecto :)

## Creando un nuevo proyecto

Genial, ya tengo usuario, ahora quiero un proyecto para empezar a compartir código...¿Y cómo hago? Es simple, es claro (?) y es muy similar a crear un usuario:

1) En la página principal, bajo el Menu **Actions** seleccionar **Register Project**.

2) Van a ver una pantalla así:



Clara Allende • Logout

# SqueakSource<sup>3</sup>

Home Projects Tags Members Groups Help

## Register Project

To register a new project please fill-in the following form. Note that the project name has to be unique within SqueakSource, it cannot be changed after registration and should not contain blanks or other special characters.

SqueakSource allows a fine-grained control about the people being allowed to read, write and admin your project. The following settings are available:

- Global specifies the default right for users not being marked as guest, developer or administrator of this project.
- Administrators are allowed to edit the project settings, to update the rss-feed, to make releases and to open and save versions.
- Developers are allowed to open and save versions.
- Guests are allowed to open versions.

**Unique Name:**

**Title:**

**Description:**

**Tags:**

- ☐ algorithm
- ☐ children
- ☐ client
- ☐ collection
- ☐ compiler
- ☐ education
- ☐ exercise
- ☐ fileformat
- ☐ learning
- ☐ ...

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:

The screenshot shows a web form for creating a project in SqueakSource. The form is organized into several sections:

- License:** A dropdown menu set to "MIT".
- Enable Blessing:** A checkbox that is unchecked.
- Enable Issue Tracking:** A checkbox that is checked.
- Global Right:** A dropdown menu set to "Read".
- Administrators:** A section with an "Available:" list (Administrator, Aldo Bertero, Alexander Munich, Alexandre Bergel, Andy Kellens, Angel Nunez) and a "Selected:" list (Clara Allende). Navigation buttons (>>, <<) are between the lists.
- Developers:** A section with an "Available:" list (Administrator, Aldo Bertero, Alexander Munich, Alexandre Bergel, Andy Kellens, Angel Nunez) and an empty "Selected:" list. Navigation buttons (>>, <<) are between the lists.
- Default move target:** A dropdown menu.
- Parent Project:** A dropdown menu.
- Guests:** A section with an "Available:" list (Administrator, Aldo Bertero, Alexander Munich, Alexandre Bergel, Andy Kellens, Angel Nunez) and an empty "Selected:" list. Navigation buttons (>>, <<) are between the lists.
- Who can move versions:** A dropdown menu set to "Developers".
- Subscriptions:** A checkbox for "Email Subscription" that is unchecked.
- Also send Emails to:** A section with a text input field containing "The report is empty." and an "Add" button.
- Send Email to:** A section with checkboxes for "admins", "developers", and "guests", all of which are unchecked.
- Reply-To Address:** A text input field.

At the bottom of the form are "Save" and "Cancel" buttons.

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 version al repositorio (es decir, cuando hay un *commit*), activando el checkbox en **Subscriptions**.

En **Administrators, Developers y Guests**, tienen los menús tipo combo box donde pueden elegir un usuario y cambiarlo con las flechitas (>> y <<) para añadirlo o sacarlo del correspondiente grupo.

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 mas o menos así:

SqueakSource<sup>3</sup>

Public Alpha

Clara Allende • Logout

HomeProjectsTagsMembersGroupsHelp

PharoInbox

Actions

- ← Back
- RSS Feed
- Project Feed
- Import versions
- Submit Issue
- Add to favorites

Clara Allende

- Logout
- Edit Account

My Projects

- reunidosTADP

Favorites

OverviewWikiNewsIssuesVersionsLatestStatistics

### 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 RegistrationGoferInstaller

### Links

<http://ss3.gemstone.com/ss/PharoInbox.html>  
<http://ss3.gemstone.com/ss/PharoInbox>

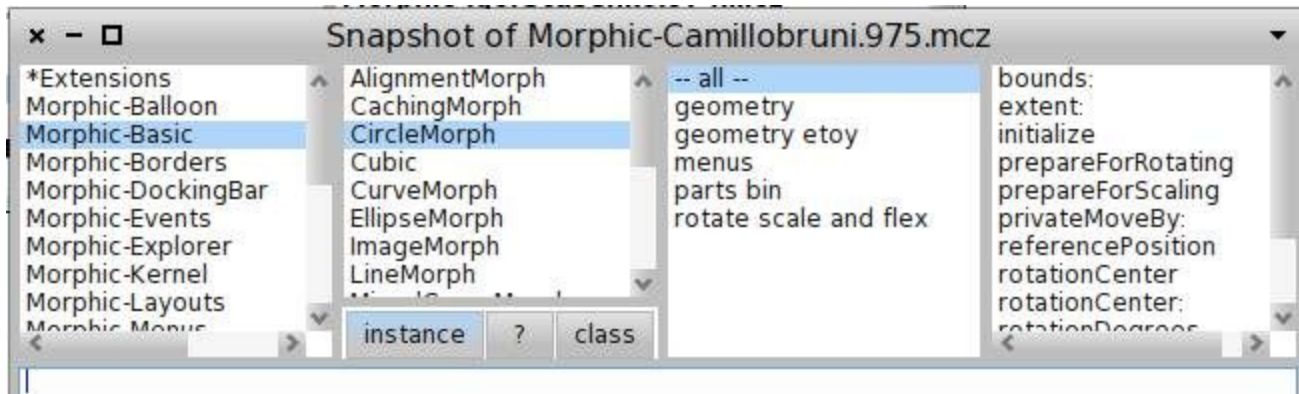
### Permissions

**Global:** Read And Write

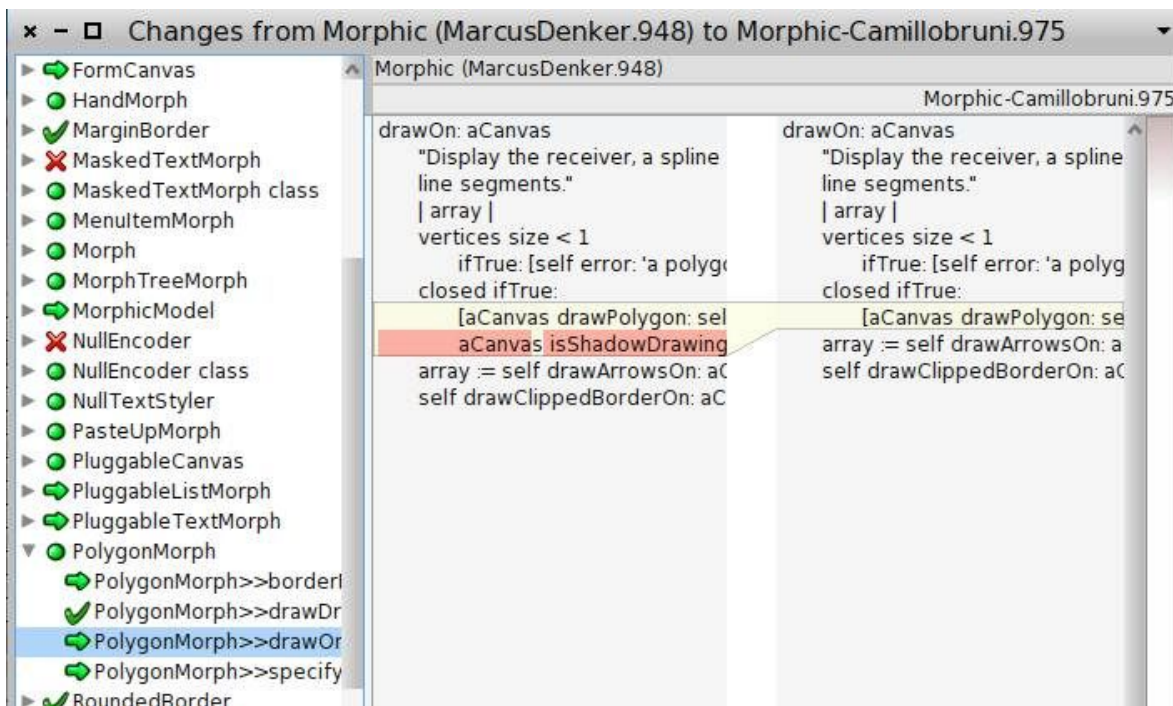
## ¿Que otras cosas puedo hacer con Monticello?

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.

