

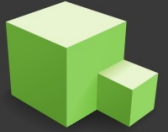
sidelab[®]
Laboratorio de Software y Entornos de Desarrollo

MSWL Development Tools

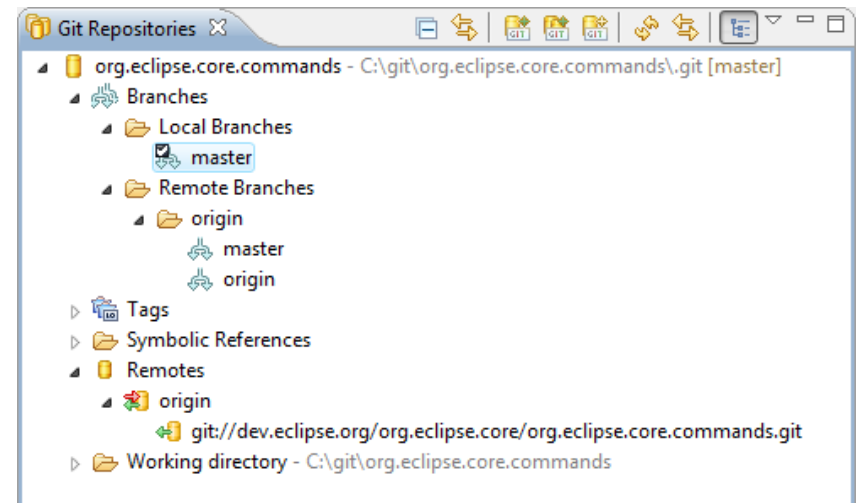
EGit

Micael Gallego
micael.gallego@urjc.es





- EGit
 - <http://www.eclipse.org/egit/>
 - EGit is an Eclipse Team provider for the Git version control system.
 - Git is a distributed SCM, which means every developer has a full copy of all history of every revision of the code, making queries against the history very fast and versatile.





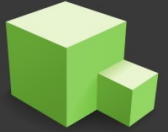
- Useful resources about Git and EGit
 - Git introduction
 - http://wiki.eclipse.org/EGit/Git_For_Eclipse_Users
 - EGit tutorial
 - http://wiki.eclipse.org/EGit/User_Guide
 - New features (tutorials can be outdated)
 - http://wiki.eclipse.org/EGit/New_and_Noteworthy



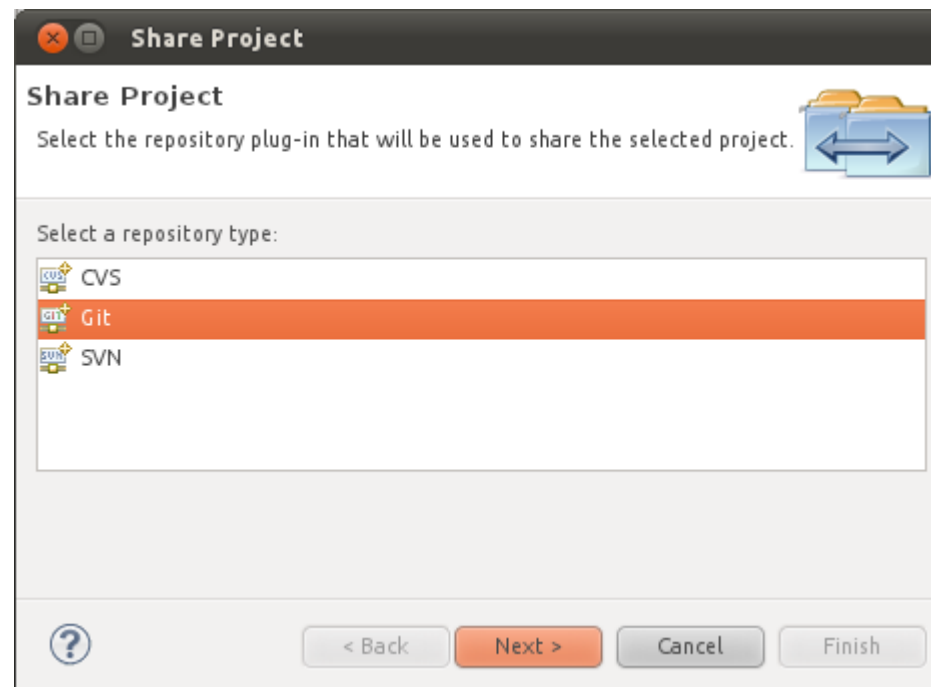
- Install EGit
 - Menu "Help > Eclipse Marketplace"
 - Type "EGit" and clic "Go"
 - Clic "Install" on EGit
 - Just "Yes", "Yes", "Continue", "Install" and so...



- Configure EGit with your credentials
 - If you have used Github from your computer (in linux), Eclipse is yet correctly configured
 - Revise credentials
 - Click Preferences > Team > Git > Configuration
 - If not available, enter the key value pairs “user.email” and “user.name”
 - Revise SSH Configuration (needed for github)
 - Click Preferences > General Connections > SSH2
 - Ensure that your SSH2 home is configured correctly (usually this is ~/.ssh) and contains your SSH2 keys

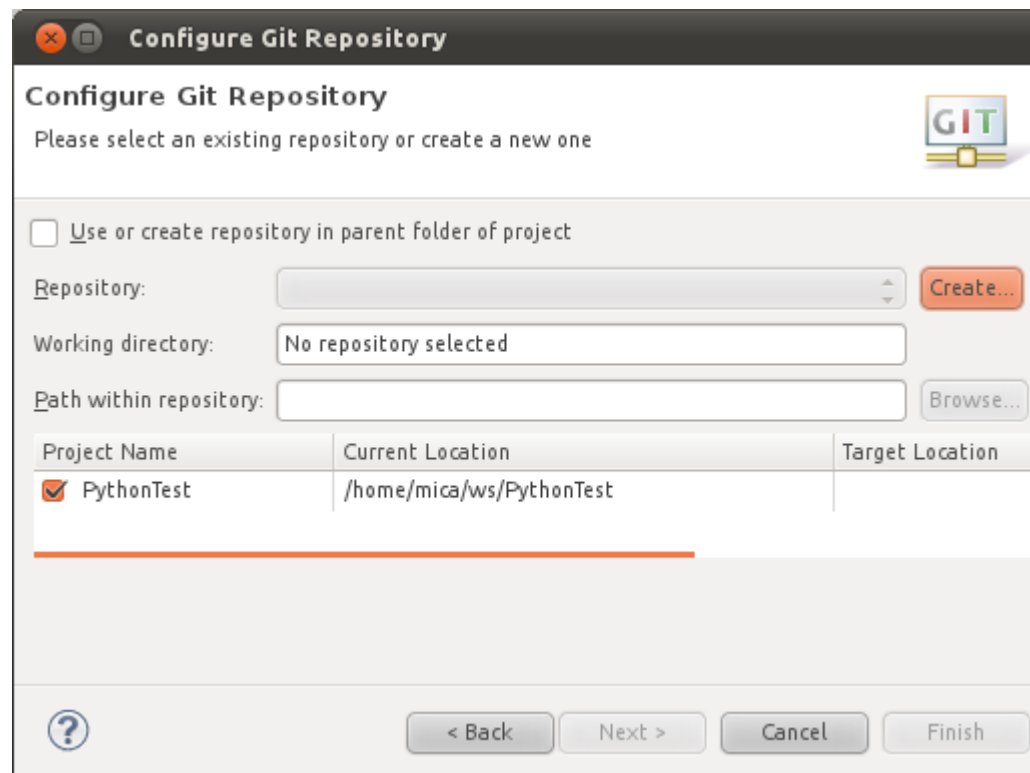


- Working with local repository
 - Create a Eclipse Project (PyDev or whatever else)
 - Select the project, click File > Team > Share Project
 - Select repository type Git and click Next



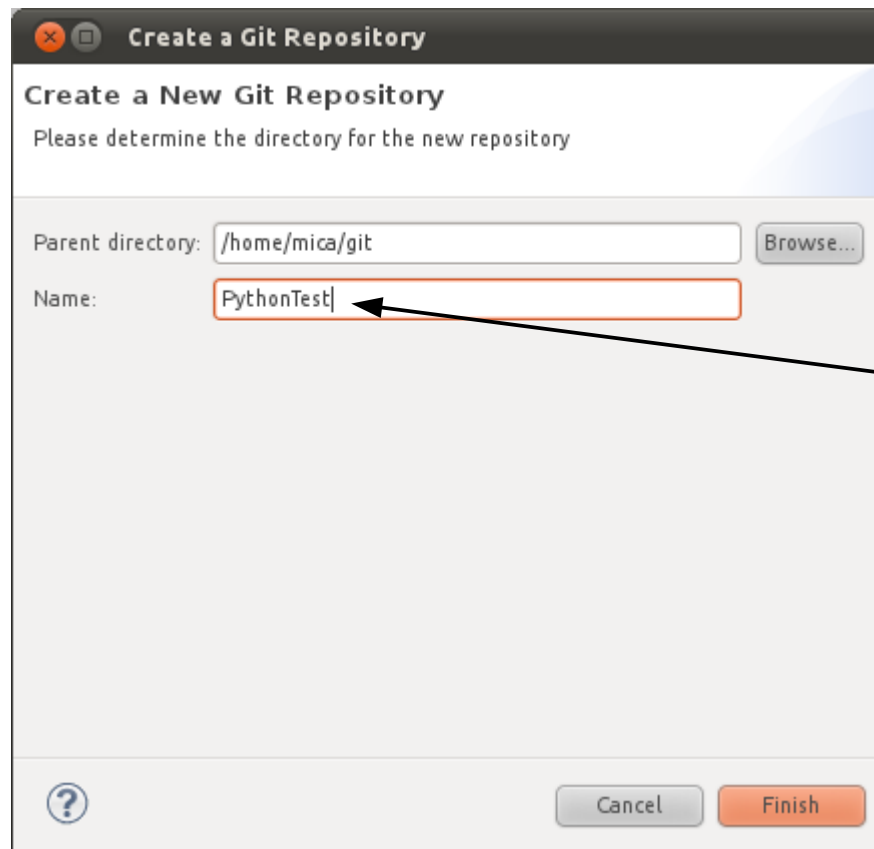


- Working with local repository (2)





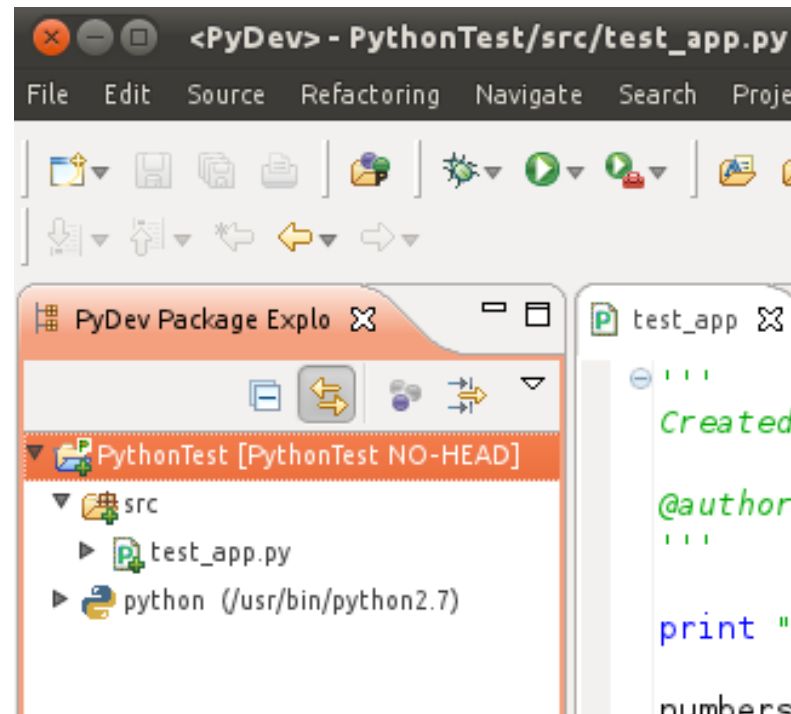
- Write the repository name and click Finish to close the wizards



Write your preferred name

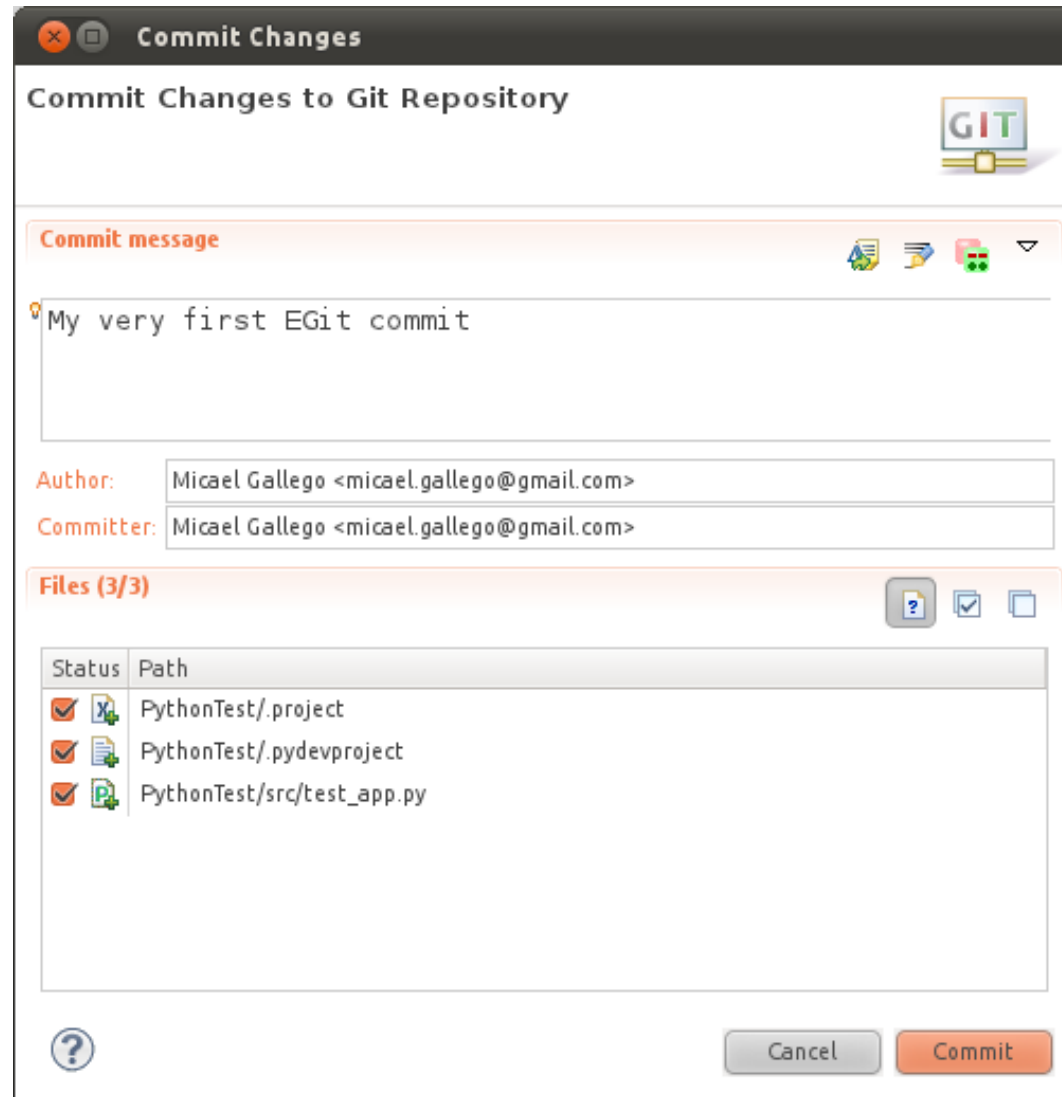


- Track changes
 - In the Click Team > Add on the project node
 - The + decorators show that now the project's files have been added to version control



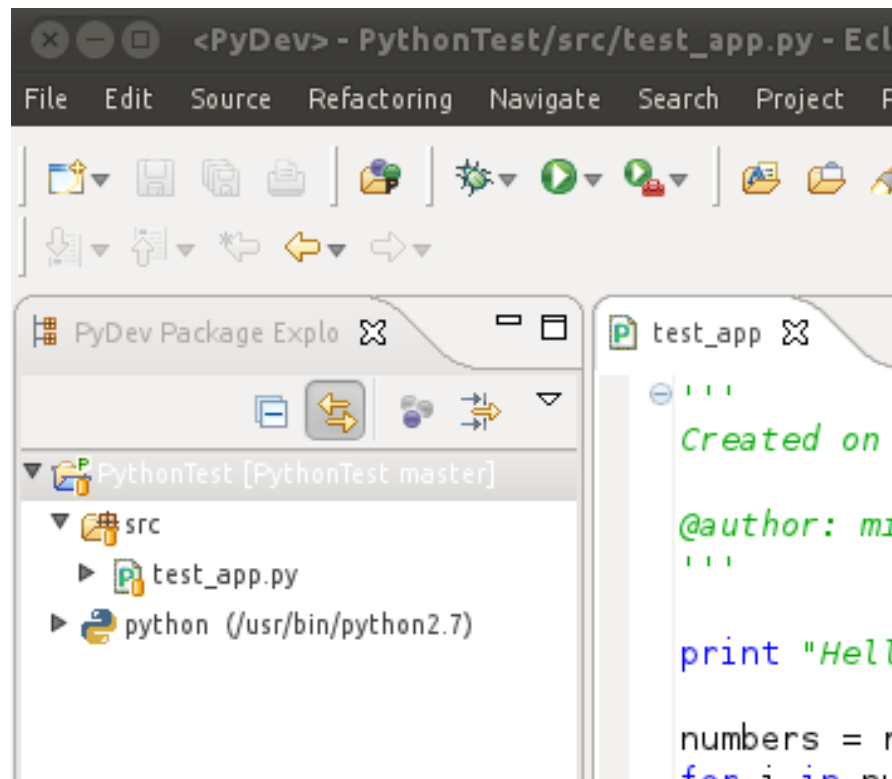


- Committing changes to repository
 - Click Team > Commit in the project context menu
 - Enter a commit message explaining your change, the first line (followed by an empty line) will become the short log for this commit.
 - Click Commit to commit your first change.



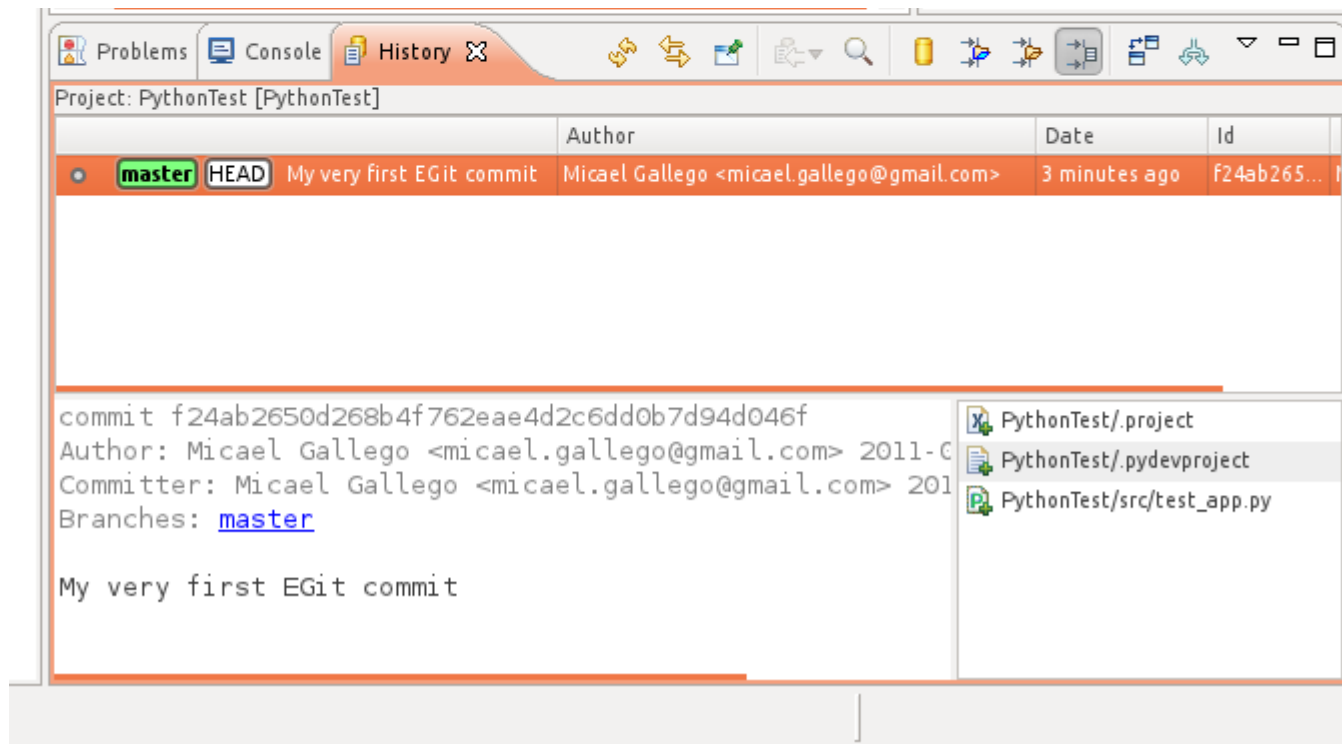


- Note that the decorators of the committed files changed





- Inspect History
 - Click Team > Show in History from the context menu to inspect the history of a resource





- Change the test_app.py file and commit to repository

The screenshot shows the EGit interface for a project named 'PythonTest'. The 'History' tab is active, displaying a table of commits. The current commit is 'Just another EGit commit' on the 'master' branch, committed 14 seconds ago. Below the table, the console shows the commit details for the previous commit, 'My very first EGit commit', including the commit hash, author, committer, and parent commit.

	Author	Date	Id
master HEAD Just another EGit commit	Micael Gallego <micael.gallego@gmail.com>	14 seconds ago	548e35d1...
My very first EGit commit	Micael Gallego <micael.gallego@gmail.com>	7 minutes ago	f24ab265...

```
commit f24ab2650d268b4f762eae4d2c6dd0b7d94d046f
Author: Micael Gallego <micael.gallego@gmail.com> 2011-0
Committer: Micael Gallego <micael.gallego@gmail.com> 201
Child: 548e35d1db45c2ecaf0a82a07c973de49e0fdbf6 (Just an
Branches: master

My very first EGit commit
```

PythonTest/.project
PythonTest/.pydevproject
PythonTest/src/test_app.py



- Comparing changes between commits
 - Select the two commits you want to compare
 - Right click > Compare with Each Other

The screenshot shows the EGit interface with a 'Text Compare' window. The window title is '"PythonTest": Comparing "548e35d1db45c2ecaf0a82a07c973de49e0fdbf6" with "f24ab2650d268b4f762eae4d2c6dd0b7d94d046f"'. The interface is split into two panes. The left pane shows the commit hash '548e35d1db45c2ecaf0a82a07c973de49e0fdbf6' and the code for 'test_app.py'. The right pane shows the commit hash 'f24ab2650d268b4f762eae4d2c6dd0b7d94d046f' and the code for 'test_app.py'. The code in both panes is identical, except for a change in the right pane where a new line 'print "Just a change for git"' has been added. The 'Structure Compare' pane at the top shows the file structure with 'src' and 'test_app.py'.

```
'''
Created on 22/09/2011

@author: mica
'''

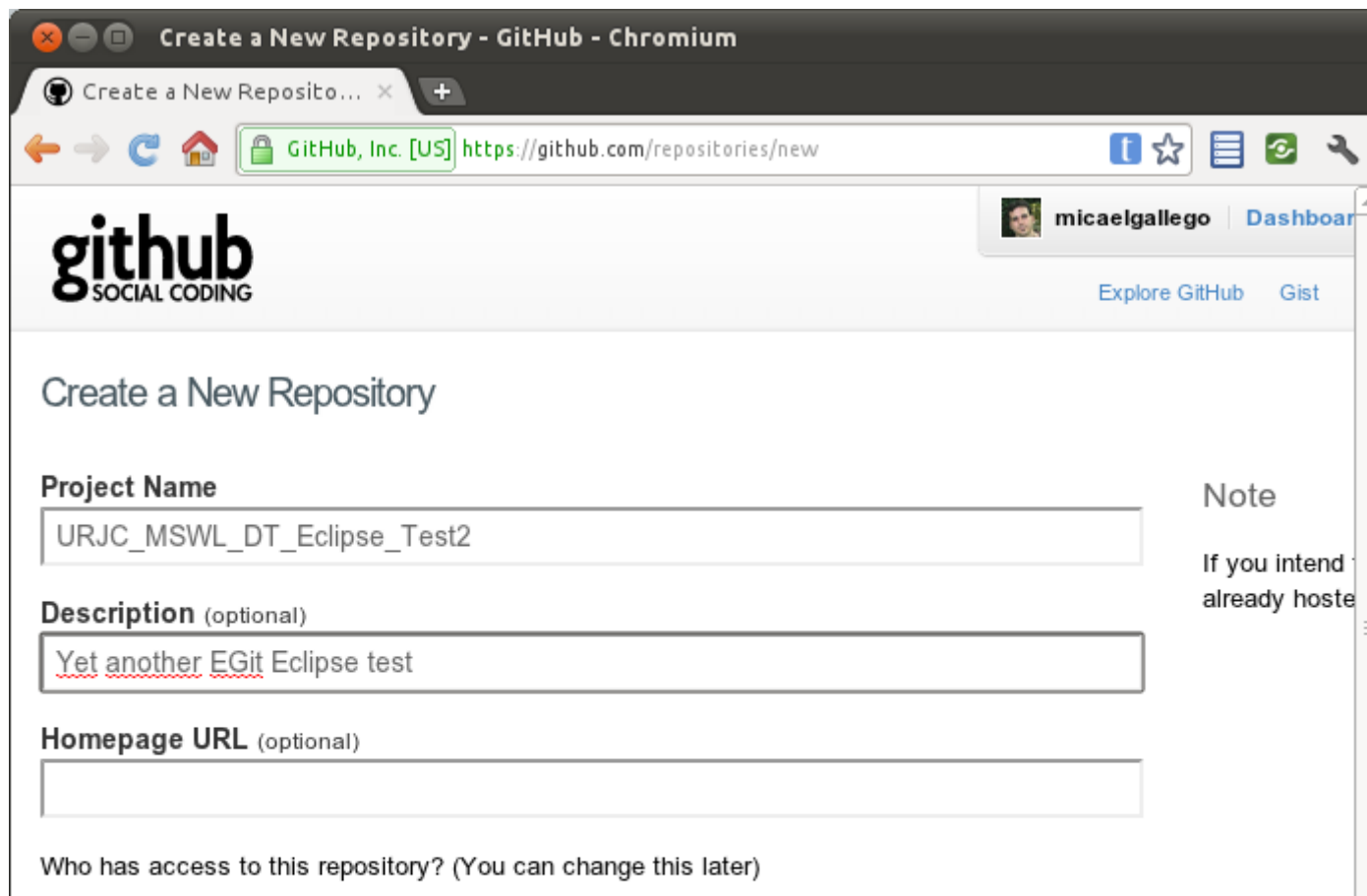
print "Hello World!"

numbers = range(10)
for i in numbers:
    print i

print "numbers length :{0}".format(len(numbers))
```



- Pushing a repository to github
 - Create a new github repo





- Push Upstream
 - Click Team > Remote > Push... and copy and paste the https URL of your new github repository
 - Enter the passphrase too
 - Click Next



Push to Another Repository

Destination Git Repository
Enter the location of the destination repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

Store in Secure Store



- On the next wizard page click Add all branches spec to map your local branch names 1:1 to the same branch names in the destination repository



Push to: https://michaelgallego@github.com/michaelgallego/URJC_MSWL_D

Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref: Destination ref:

Add delete ref specification

Remote ref to delete:

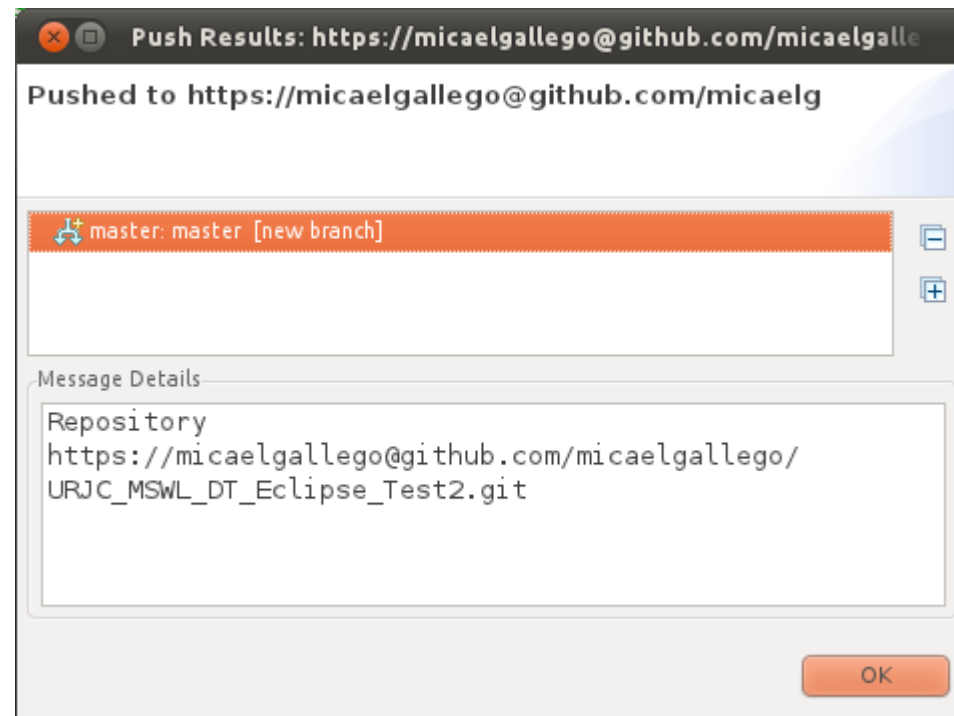
Add predefined specification

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
+ Update	refs/heads/*	refs/heads/*	<input type="checkbox"/>	<input type="button" value="Remove"/>



- Click “Finish” and you will see a information dialog if all goes ok





Push to: https://michaelgallego@github.com/michaelgallego/URJC_MSWL_D

Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref: Destination ref:

Add delete ref specification

Remote ref to delete:

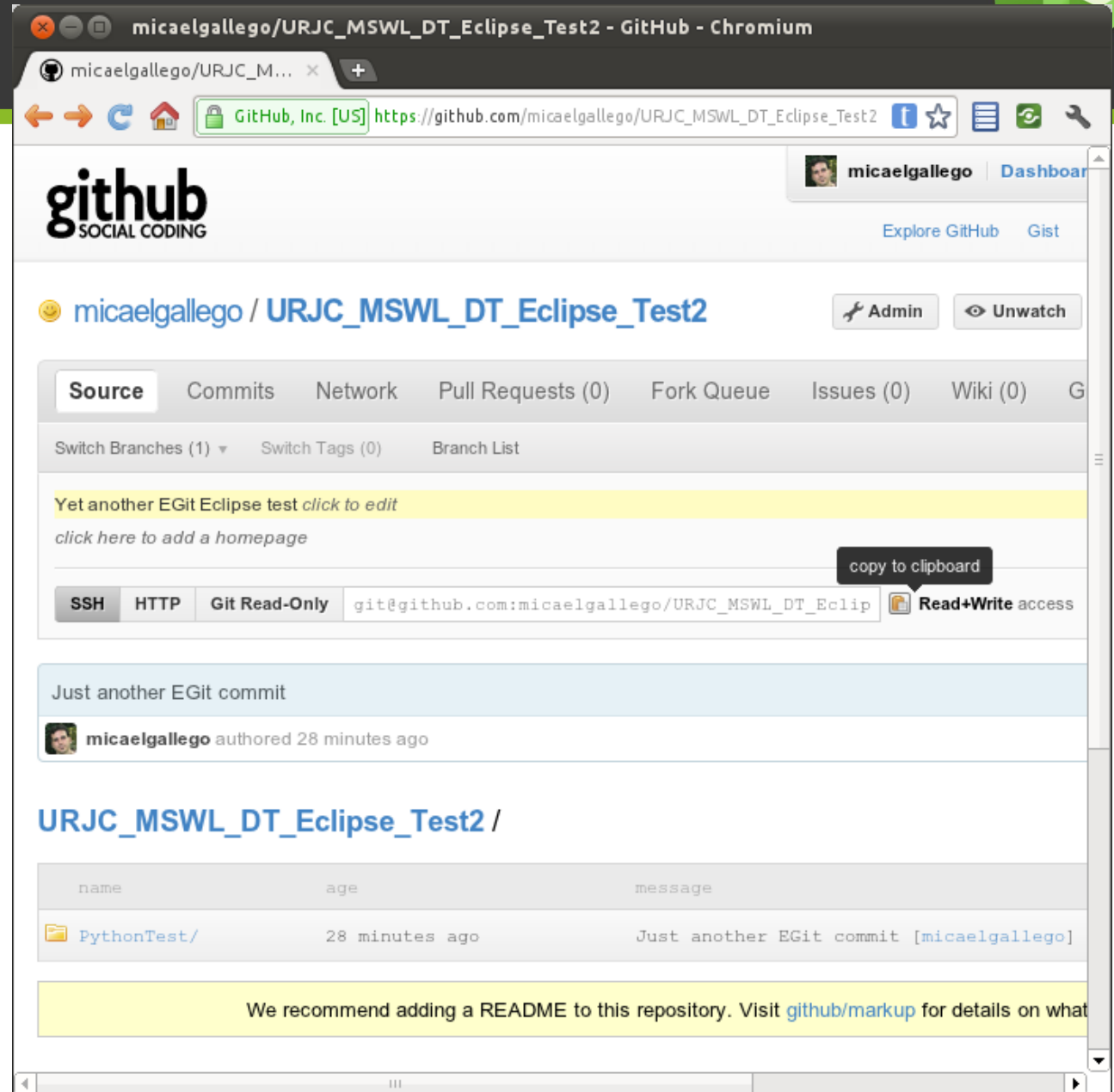
Add predefined specification

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
+ Update	refs/heads/*	refs/heads/*		

EGit

- Point your browser at your GitHub repository to see that your new repository content has arrived



The screenshot shows a web browser window displaying a GitHub repository page. The browser's address bar shows the URL `https://github.com/micaelgallego/URJC_MSWL_DT_Eclipse_Test2`. The repository name is `URJC_MSWL_DT_Eclipse_Test2` by user `micaelgallego`. The page shows a commit message "Just another EGit commit" by `micaelgallego` from 28 minutes ago. Below the commit, a table lists the files in the commit:

name	age	message
<code>PythonTest/</code>	28 minutes ago	Just another EGit commit [<code>micaelgallego</code>]

At the bottom of the page, there is a yellow banner with the text: "We recommend adding a README to this repository. Visit [github/markup](https://github.com/markup) for details on what".



- Import and Clone
 - You can add git projects to Eclipse
 - The project can be on disk or you can clone using Eclipse
- Cloning from github
 - Import > Git > Projects from Git
 - Click “Clone...” button
 - Enter github https URL and passphrase



Clone Git Repository

Source Git Repository
Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

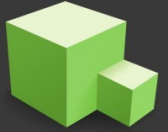
Port:

Authentication

User:

Password:

Store in Secure Store

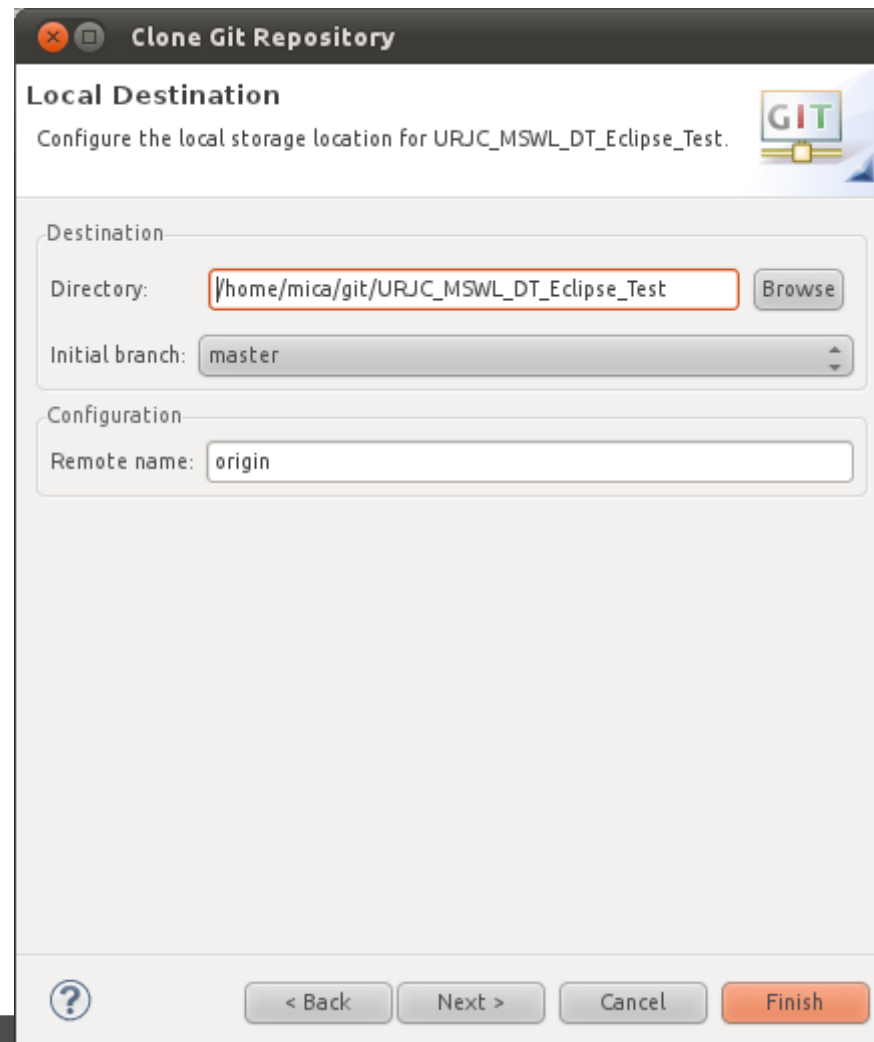


- Select branches you want to clone



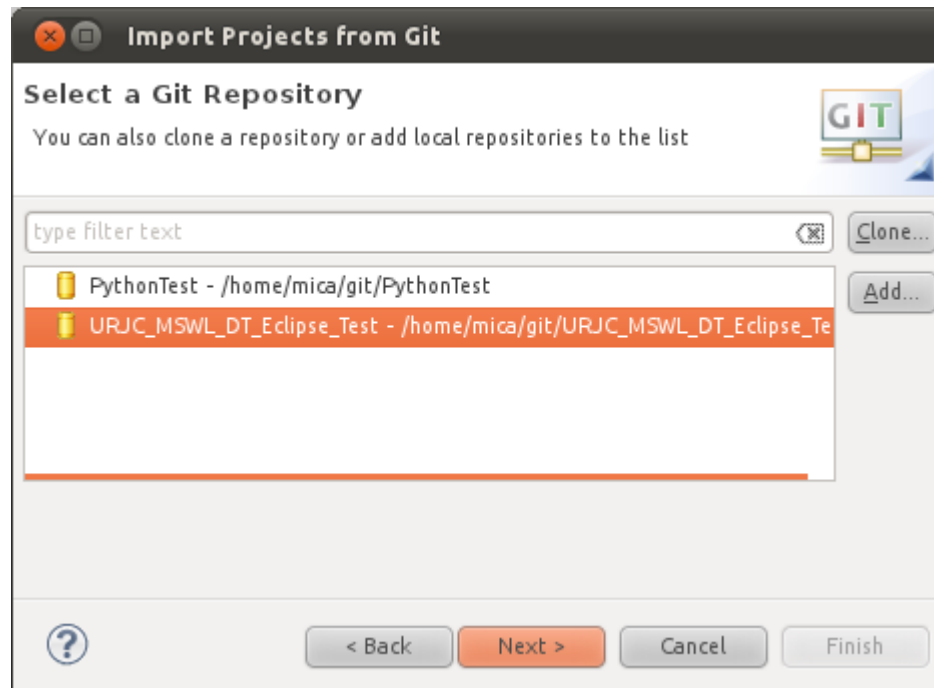


- Select where to store your local repo



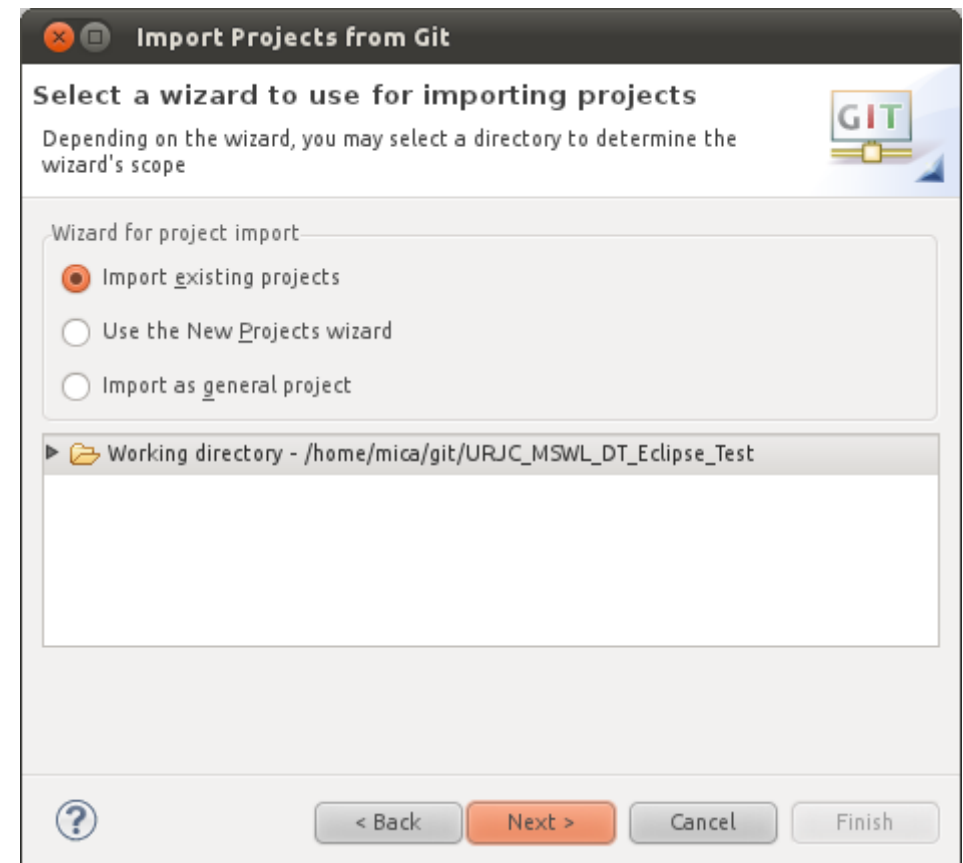


- The newly cloned repo is available in repositories view
- You can now select a repository and click Next. On the following wizard page, you will decide how to import projects.





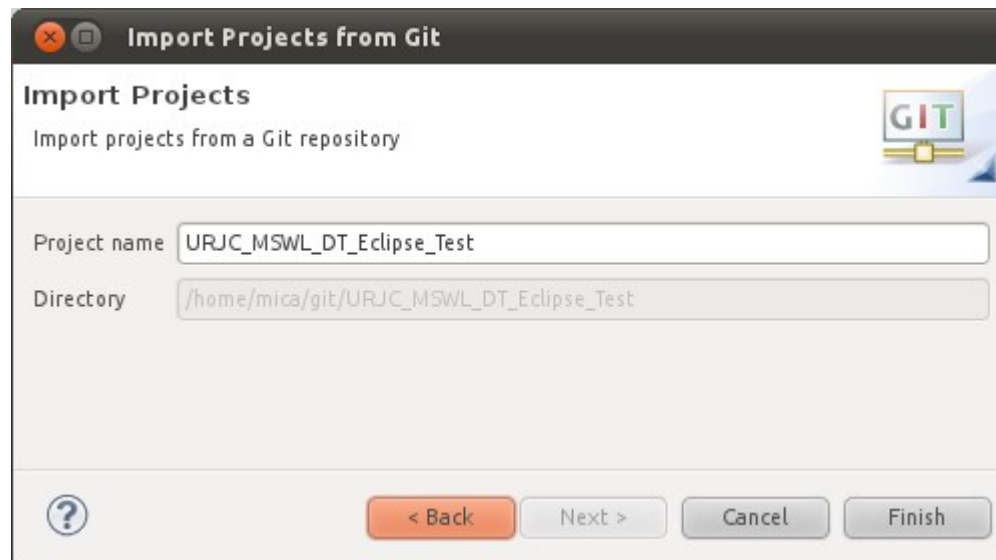
- **Import existing projects**
 - When the project in git was created with Eclipse
- **Use the New Projects wizard**
 - When the project in git was created without Eclipse and you want to “convert” it
- **Import as general project**
 - When the project in git was created without Eclipse and you don't want to “convert” it



We will use “Import as general project” because “Use the New Projects wizard” doesn't work
https://bugs.eclipse.org/bugs/show_bug.cgi?id=324145



- Enter the project name and click “Finish”





<PyDev> - URJC_MSWL_DT_Eclipse_Test/README - Eclipse

File Edit Navigate Search Project Pydev Run Window Help

PyDev Package Explor

- PythonTest [PythonTest master]
- URJC_MSWL_DT_Eclipse_Test [URJC_M...]
 - README

README

This is only a fake README with useless in

Outline

An outline is not available.

Problems Console History

0 items

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

URJC_MSWL_DT_Eclipse_Test/README



- You can convert your “General project” in a PyDev Project with a hack
 - Open the .project file in a text editor
 - Write `<nature>org.python.pydev.pythonNature</nature>`
 - Close and reopen Eclipse project

A screenshot of a text editor window titled ".project (~/.git/URJC_MSWL_DT_Eclipse_Test) - gedit". The window shows an XML file with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<projectDescription>
  <name>URJC_MSWL_DT_Eclipse_Test</name>
  <comment></comment>
  <projects>
  </projects>
  <buildSpec>
  </buildSpec>
  <natures>
    <nature>org.python.pydev.pythonNature</nature>
  </natures>
</projectDescription>
```

The status bar at the bottom indicates "XML", "Ancho de la tabulación: 8", "Ln 10, Col 63", and "INS".



<PyDev> - URJC_MSWL_DT_Eclipse_Test/src/app_test.py - Eclipse

File Edit Source Refactoring Navigate Search Project Pydev Run Window Help

PyDev Package Explo

- PythonTest [PythonTest master]
- URJC_MSWL_DT_Eclipse_Test [URJC_M...]
 - src
 - app_test.py
 - README
 - python (/usr/bin/python2.7)

```
'''  
Created on 23/09/2011  
  
@author: mica  
'''  
  
print "Hello... I'm here again"
```

Outline

type filter text

Problems Console History

<terminated> /home/mica/git/URJC_MSWL_DT_Eclipse_Test/src/app_test.py

Hello... I'm here again

1 items selected