

CI3715

Taller 2

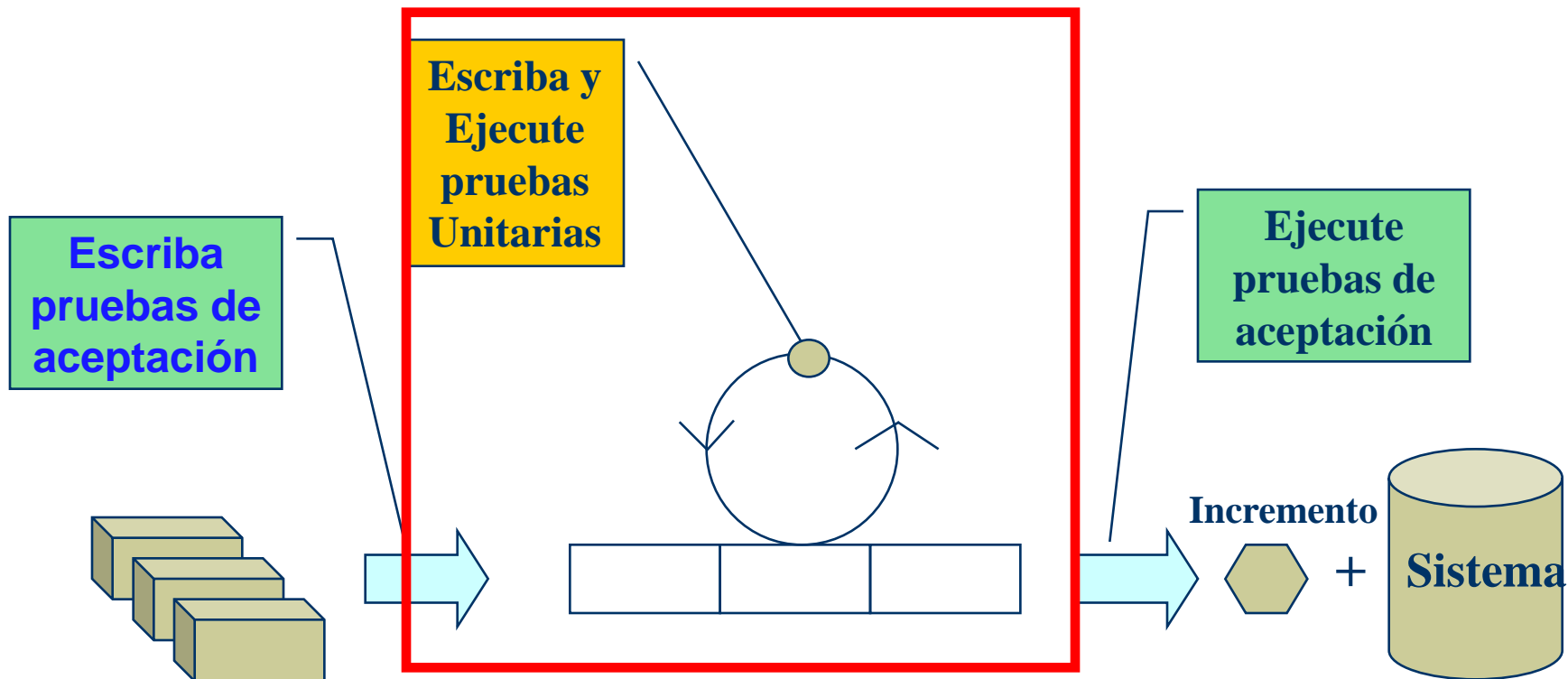
Profesor:

- Alfonso Reinoza: jareinozacg@gmail.com

Agenda

- **Verificación:**
 - **PyUnit**
- Gestión de Configuraciones y Versiones de software:
 - Introducción a *Git* básico

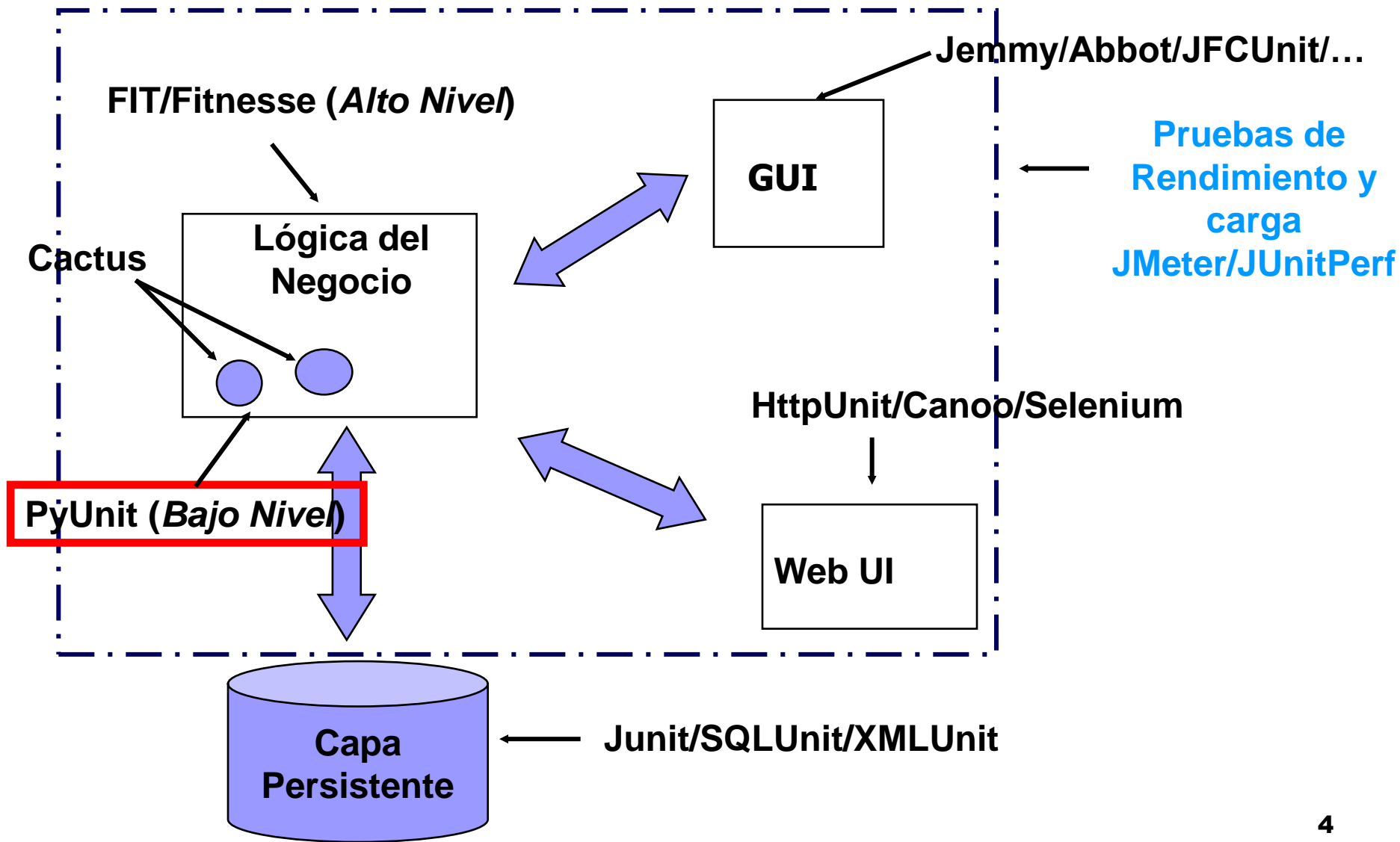
PyUnit - Desarrollo Iterativo de Software



Dar prioridad a funcionalidades

“Ejecutar después del desarrollo”

Herramientas de prueba



Pruebas con PyUnit

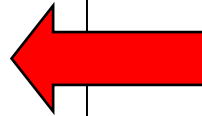
- PyUnit es un ambiente de pruebas unitarias para programas en Python creado por Steve Purcell, basándose en el JUnit desarrollado por Erich Gamma y Kent Beck.
 - Escribir casos de prueba
 - Ejecución de casos de prueba
 - Pasa/ Falla? (Resultado esperado = resultado obtenido?)
- Consiste en un marco que proporciona todas las herramientas para el análisis.
 - marco: conjunto de clases y convenciones para usarlas.
- Está integrado en Eclipse a través de un plug-in gráfico.

PyUnit

- Casos de prueba en código Python
- Marco de pruebas
 - Caso de prueba = “secuencia de operaciones + entradas + valores esperados ”

■ Production code

```
def doubleOf2()  
    #...
```



■ Test code

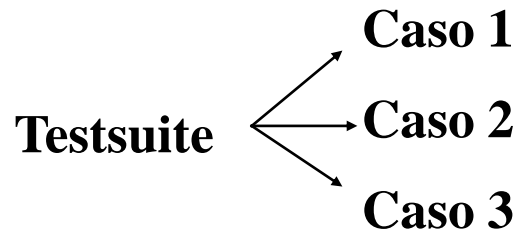
```
def testDoubleOfZero():  
    #..  
    doubleOf2()  
    #..
```

PyUnit para probar programas

- "Sustituye el programa en ejecución para comprobar el comportamiento del programa"
- Todo lo que tenemos que hacer es:
 - ☐ Importar unittest
 - ☐ escribir una subclase de unittest.TestCase
 - ☐ añadir a la misma uno o más métodos de prueba
 - ☐ if `__name__ == "__main__":`
 `unittest.main ()`

Elementos del Marco

- **TestCase**
 - Clase base para las clases que contienen pruebas
- **assert * ()**
 - Familia de Métodos para comprobar condiciones
- **TestSuite**
 - Permite agrupar varios casos de prueba



Un ejemplo

```
import unittest
from stack import *
```

```
class Stack
    def isEmpty()
        //
        ....
        //
    def push(int i)
    def pop()
```

```
class StackTester(unittest.TestCase):
```

```
def testStack(self):
    aStack = Stack()
    if not aStack.isEmpty():
        print("Stack should be empty!")
```

Debe comenzar con
"test"

```
def testStackPush(self):
    aStackPush(10)
    print("First element: " + aStack.pop())
```

Assert*()

- Son métodos públicos definidos en la clase base TestCase
- Sus nombres comienzan con “assert” y se utilizan en los métodos de prueba
 - es. assertTrue (aStack.empty (), “stack should be empty”);
- Si la condición es falsa:
 - La prueba falla
 - La ejecución salta el resto del método de prueba
 - El mensaje (si lo hay) se imprime como explicación de por qué falla
- Si la condición es verdadera:
 - La ejecución continúa normalmente

Assert*()

- Para una condicion booleana
 - **assertTrue(condition, “message for fail”);**
 - **assertFalse(condition, “message for fail”);**
- Para objeto, int, long, and valores byte
 - **assertEquals(expected_value, expression);** ← obtenido
- Para flotante y valores dobles
 - **assertEquals(expected, expression, error);**
- ...

Assert: ejemplo

```
def testStack(self):  
    aStack = Stack()  
    assertTrue(aStack.isEmpty(), "Stack should be empty!")  
  
def testStackPop(self):  
    aStack.push(4)  
    assertEquals(4, aStack.pop())
```

```
class Stack:  
    isEmpty(): ...  
    push(i): ...  
    pop(): ...  
    ...
```

Un concepto a la vez...

```
class StackTester(unittest.TestCase):  
    def testStackEmpty(self):  
        aStack = Stack()  
        assertTrue(aStack.isEmpty(), "Stack should be empty!")  
  
    def testStackNotEmpty(self):  
        aStack = Stack()  
        aStack.push(10)  
        assertTrue(not aStack.isEmpty(), "Stack should not be empty!")  
  
    def testStackOperations(self):  
        aStack = Stack()  
        aStack.push(-4)  
        assertEquals(-4, aStack.pop())
```

Regla de trabajo

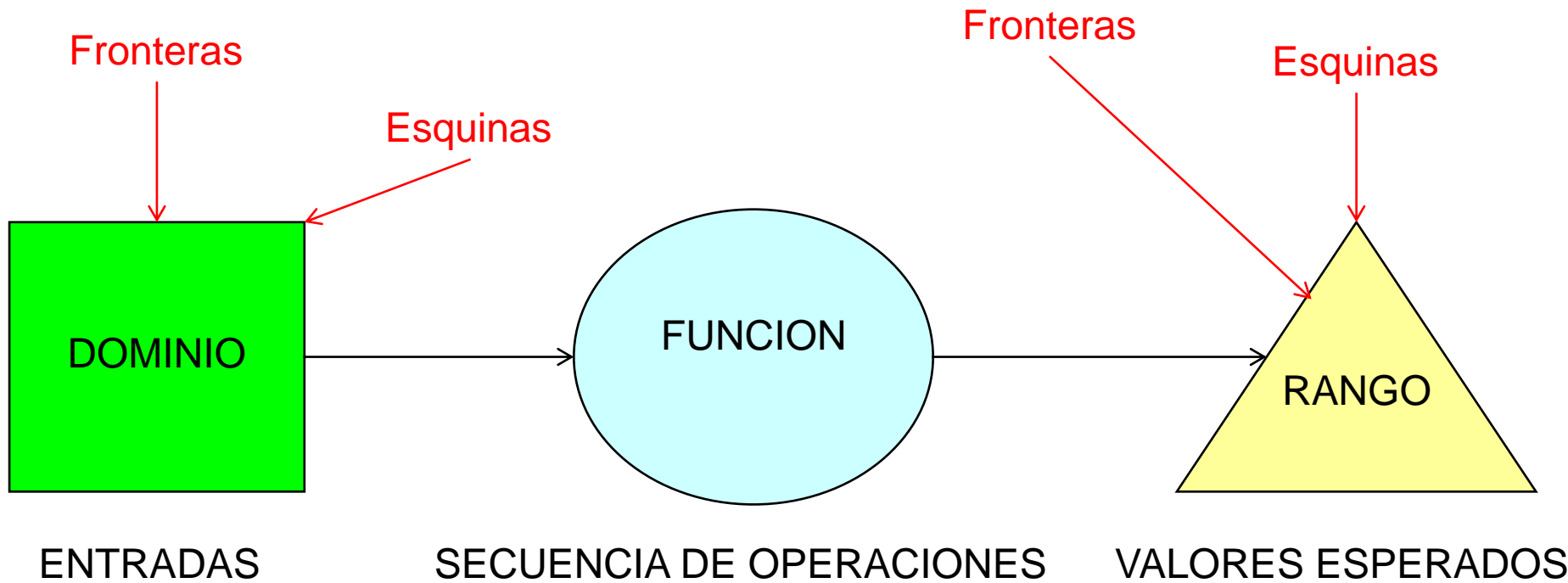
- Para cada clase de caso de prueba, PyUnit
 - ejecuta la totalidad de sus métodos de prueba públicos
 - es decir, aquellos cuyo nombre comienza con "test"
 - hace caso omiso de todo lo demás ...
- Las clases de prueba pueden contener "métodos de ayuda" a condición de que sean:
 - no públicas, o
 - cuyo nombre no comienza con "test"

SetUp() and tearDown()

- **setUp()** el método inicializa objeto(s) bajo prueba.
 - se llama antes de cada método de prueba
- **tearDown()** el método libera objeto(s) bajo prueba
 - Se llama después de cada método de casos de prueba.

```
ShoppingCart cart;  
Book book;  
  
def setUp():  
    self.cart = ShoppingCart()  
    self.book = Book("PyUnit", 29.95)  
    self.cart.addItem(book)  
  
...
```

Forma de verlo



Ejercicio

Elabore una suite de pruebas unitarias para una función que, al final de un juego de boliche, indique la puntuación de uno de los jugadores.

1	4	4	5	6		5			0	1	7		6			2		6
5	14	29	49	60	61	77	97	117	133									

Ejercicio: Reglas del juego

- El juego consiste en diez cuadros (*frames*).
- En cada cuadro, el jugador tiene la oportunidad de derribar hasta 10 pines en no más de dos lanzamientos (*rolls*).
- El jugador hace una chuza (*strike*) si derriba diez pines en su primer lanzamiento en un cuadro. Si esto ocurre termina el cuadro, salvo en caso que la chuza la haga en el primer lanzamiento del décimo cuadro, en cuyo caso se le otorga un bono: dos lanzamientos adicionales.
- El jugador hace una media chuza (*spare*) si derriba los diez pines en los dos lanzamientos de un cuadro. Si hace una media chuza en los primeros dos lanzamientos del décimo cuadro, se le otorga como bono un tercer lanzamiento.
- La puntuación básica de un lanzamiento es igual al número de pines derribados en él.
- Si hace media chuza [chuza], se le suma a su puntuación el número de pines derribados en el próximo lanzamiento [los próximos 2 lanzamientos.].

Ejercicio: Identificar extremos

Extremos en la salida de score:

- La puntuación mínima: 0
- La puntuación máxima: 300

ANALISIS DEL DOMINIO DE DATOS

Extremos en la entrada de score ...¿cuál es la estructura de entrada?

- Una lista de cuadros. Cada cuadro es un 1-upla, dupla o terna de lanzamientos (pines derribados).
- El décimo cuadro tiene reglas especiales
- Entre el cuadro 1 y el 9, se tiene una 1-upla si se lanzó una chuza, en caso contrario el cuadro es una dupla.
- En el décimo cuadro se tiene una terna si el primer lanzamiento fue chuza o los dos primeros lanzamientos fueron media chuza. De lo contrario el décimo cuadro es una dupla.
- Los extremos de un lanzamiento son 0, 9 y 10 pines derribados

Ejercicio: Identificar extremos

Extremos en la salida de score:

- La puntuación mínima: 0
- La puntuación máxima: 300

Extremos en la entrada de score

- Una lista que contiene el número mínimo de lanzamientos
 - ☐ ¿Cuál es la mínima puntuación que se obtiene con un mínimo de lanzamientos?
 - ☐ ¿Y el máximo? ¿Cómo se obtienen estos mínimos y máximos?
- Una lista que contiene un número máximo de lanzamientos.
 - ☐ ¿Cuál es la mínima y la máxima puntuación que se obtienen? ¿Cómo?
- Extremos en los cuadros
 - ☐ Un cuadro con chuza
 - ☐ Cuadros con media chuza y un lanzamiento en extremo: (9,1), (1,9)
 - ☐ Cuadros sin chuza ni media chuza y un lanzamiento en extremo (la suma en 9).
Note que técnicamente (0,0), (0,9) y (9,0) son cuadros esquinas –pero muy interesantes...
- Extremos en el décimo cuadro... (*queda como ejercicio*)

Ejercicio: Casos de prueba

1. Puntuación mínima (frontera sobre la salida)

`guttergame = [(0,0)]*10`

`assertEquals(score(guttergame), 0)`

2. Puntuación máxima (frontera sobre la salida)

`perfectgame = [(10)]*9 + [(10, 10, 10)]`

`assertEquals(score(perfectgame), 300)`

3. Mínimo puntaje para un mínimo número de lanzamientos

`minScoringMinRollsGame = [(10)] * 9 + [(0, 0)]`

`assertEquals(score(minScoringMinRollsGame), 30*7 + 20 + 10 +0)`

4. Máximo puntaje para un mínimo número de lanzamientos

`maxScoringMinRollsGame = [(10) * 9] + [(9, 0)]`

`assertEquals(score(maxScoringMinRollsGame), 30*7 + 29 + 19 +9)`

Ejercicio: Casos de prueba

5. Mínimo puntaje con el máximo número de lanzamientos

$\text{minScoringMaxRollsGame} = [(0,0)]*9 + [(10, 0, 0)]$

$\text{assertEquals}(\text{score}(\text{minScoringMaxRollsGame}), 10)$

6. Máximo puntaje con el máximo número de lanzamientos

$\text{maxScoringMaxRollsGame} = [(9,1)]* 9 + [(10, 10, 10)]$

$\text{assertEquals}(\text{score}(\text{maxScoringMaxRollsGame}), 19 * 8 + 20 + 30)$

7. Un juego con exactamente una chuza en un cuadro que no sea el décimo

$\text{oneStrikeNotInTenthFrame} = [(2,3)]*5 + [(10)] + [(5,4)]*4$

$\text{assertEquals}(\text{score}(\text{oneStrikeNotInTenthFrame}), 25 + 19 + 36)$

8. Un juego con exactamente media chuza que no es esquina en un cuadro que no sea el décimo

$\text{oneExtremeSpareNotInTenthFrame1} = [(3,2)]*2 + [(9, 1)] + [(4,5)] *7$

$\text{assertEquals}(\text{score}(\text{oneExtremeSpareNotInTenthFrame1}), 5*2 + 14 + 9*7)$

Ejercicio: Casos de prueba

9. Un juego con exactamente media chuza que no es esquina en un cuadro que no sea el décimo

```
oneExtremeSpareNotInTenthFrame2 = [(3,2)]*2 + [(1, 9)] + [(5,4)] *7  
assertEquals(score(oneExtremeSpareNotInTenthFrame2), 5*2 + 15 + 9*7)
```

10. Un juego con exactamente un extremo de cuadro que no sea chuza ni media chuza

```
oneExtremeNotStrikeNotSpareFrame01 = [(0,1)] + [(3,5)]* 9  
assertEquals(score(oneExtremeNotStrikeNotSpareFrame01), 1+ 72)
```

11. Un juego con exactamente un extremo de cuadro que no sea chuza ni media chuza

```
oneExtremeNotStrikeNotSpareFrame10 = [(6,2)]* 8 + [(1,0)] + [(1,3)]  
assertEquals(score(oneExtremeNotStrikeNotSpareFrame01), 64 + 1 + 4)
```

12. Un juego con exactamente un extremo de cuadro que no sea chuza ni media chuza

```
oneExtremeNotStrikeNotSpareFrame08 = [(6,1)] + [(0,8)] + [(1,3)*8]  
assertEquals(score(oneExtremeNotStrikeNotSpareFrame08), 7 + 8 + 32)
```

Ejercicio: Casos de prueba

...faltan casos de prueba para extremos del décimo cuadro

Esquinas maliciosas

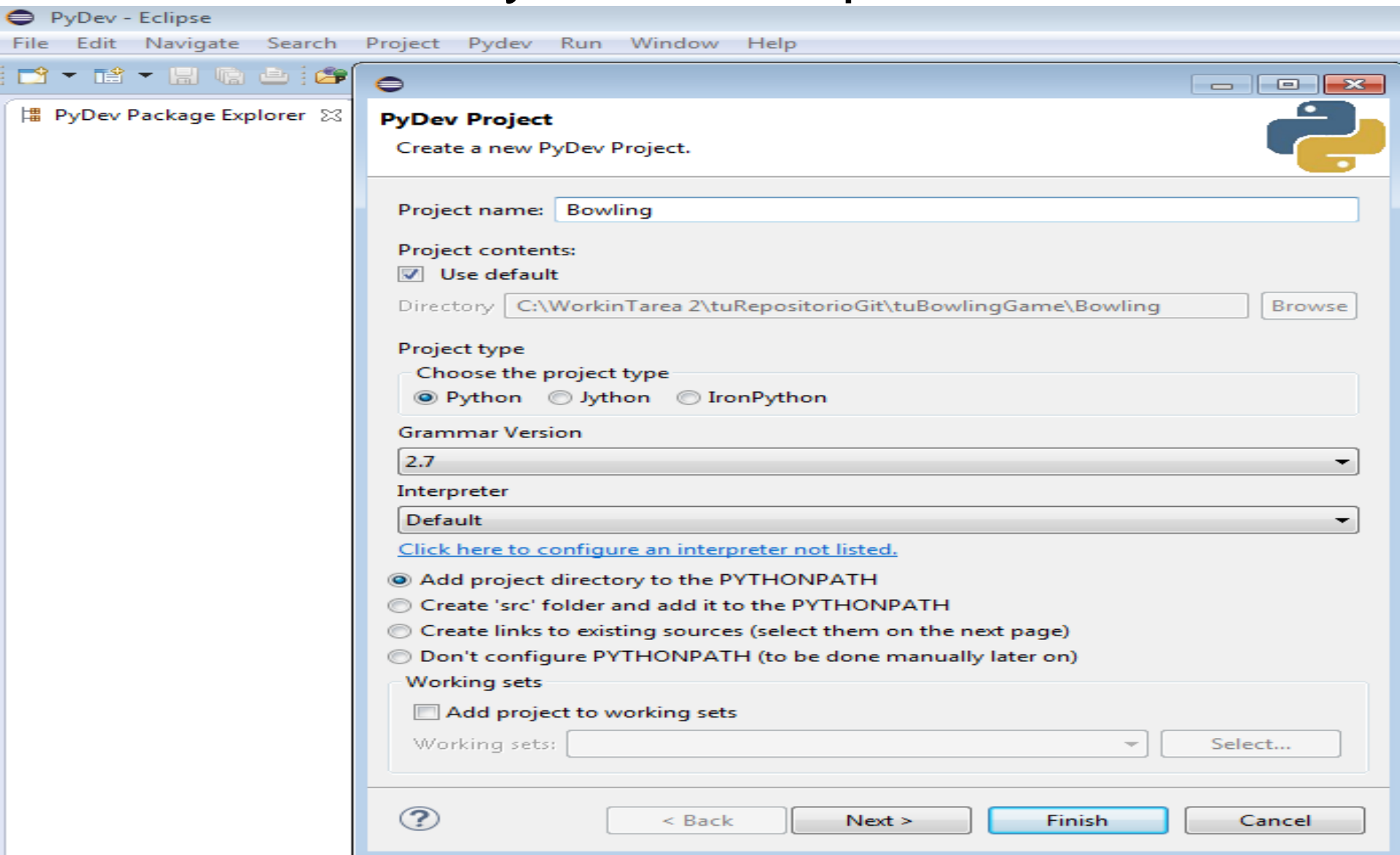
13. Un juego con exactamente una media chuza en esquina (0, 10) en un cuadro que no sea el décimo:

```
oneCornerSpareNotInTenthFrame = [(7,2)]* 8 + [(0,10)] + [(2,6)]  
assertEquals(score(oneCornerSpareNotInTenthFrame), 72 + 12 +8)
```

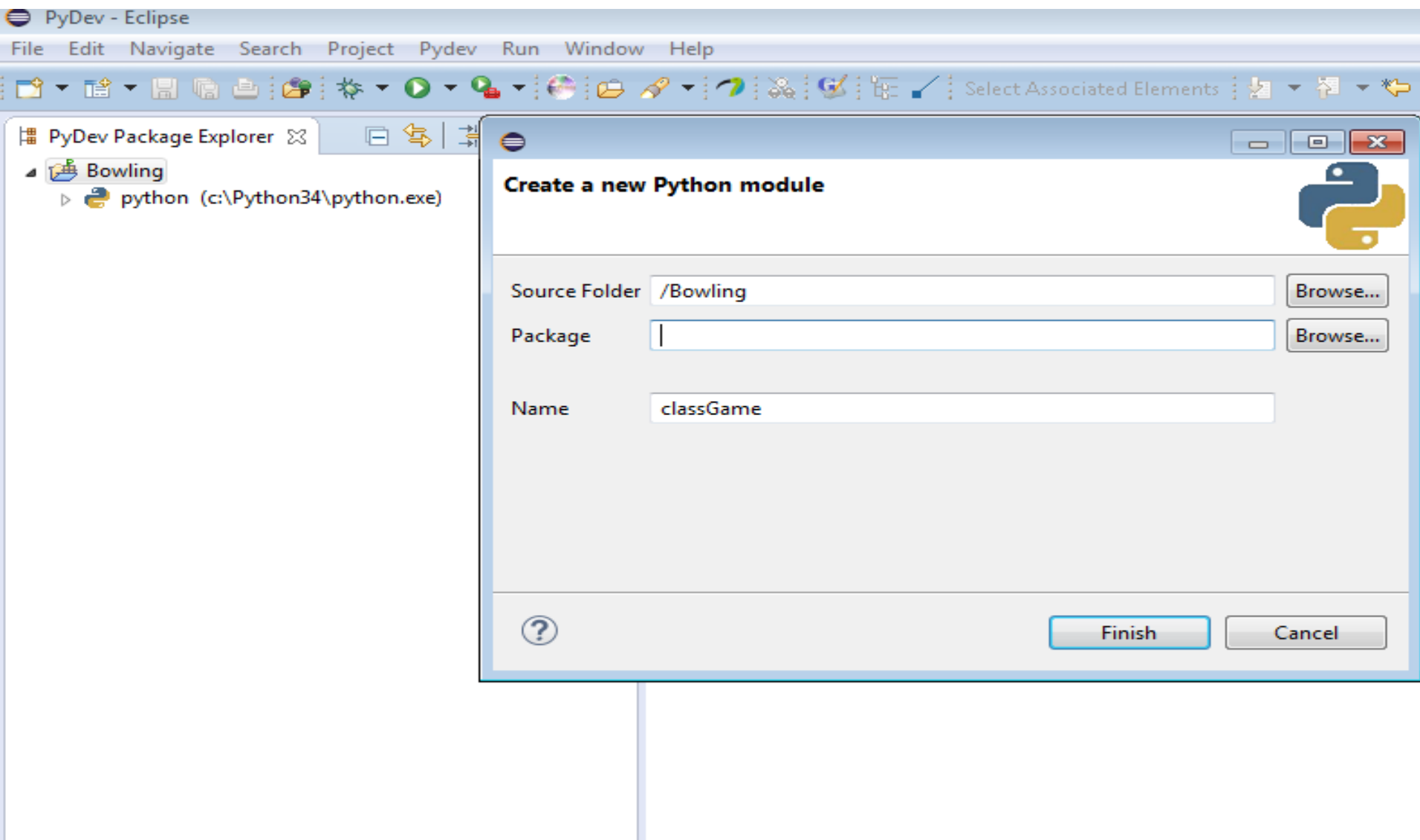
14. Un juego con un cuadro esquina (que no sea el 10°) que no contenga chuza o media chuza: un cuadro (9,0)

```
oneCornerNotStrikeNotSpareFrame= [(1,5)]* 8 + [(9,0)] + [(3,3)]  
assertEquals(score(oneCornerNotStrikeNotSpareFrame), 48 + 9+6)
```

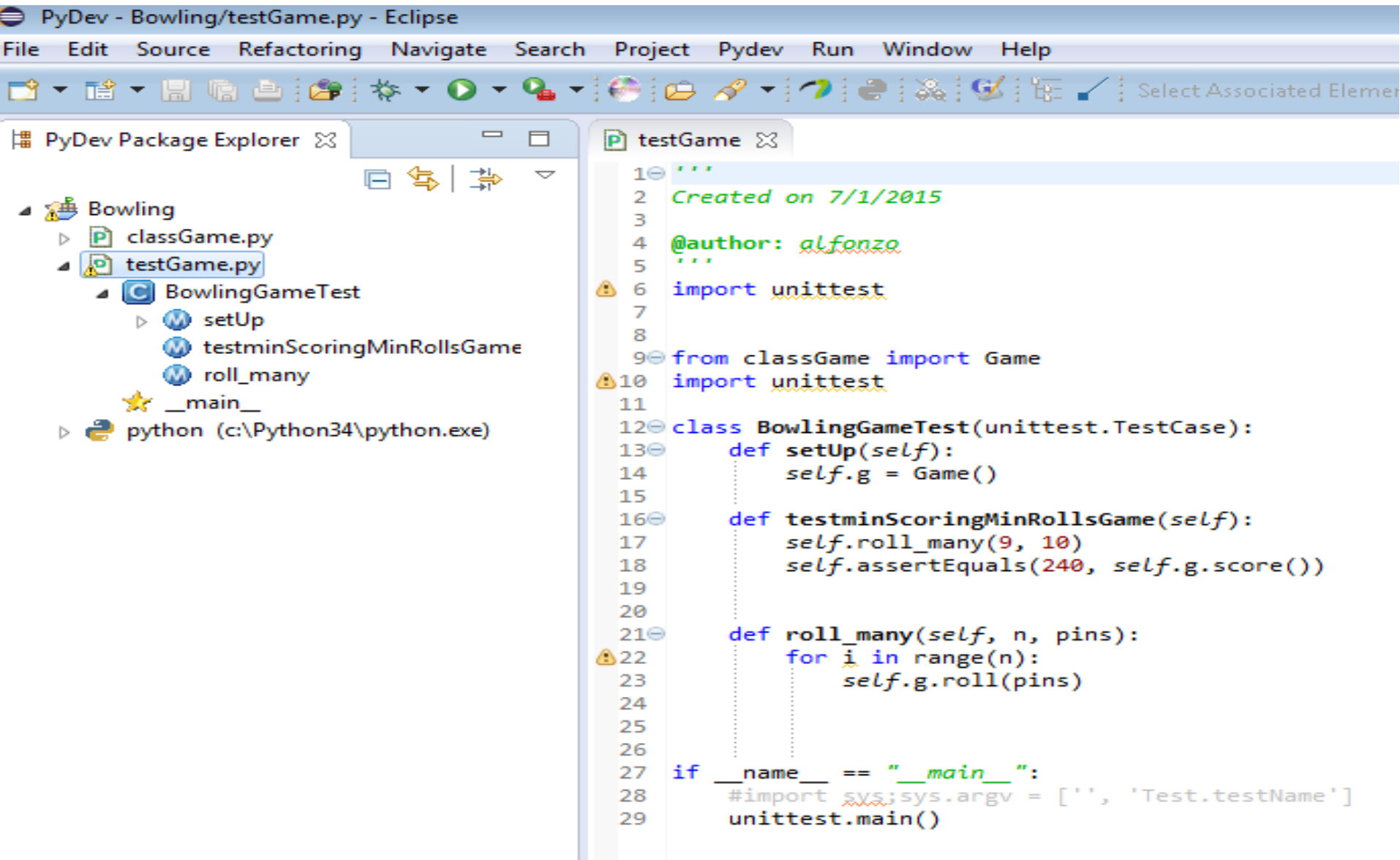

PyUnit en eclipse



PyUnit en eclipse



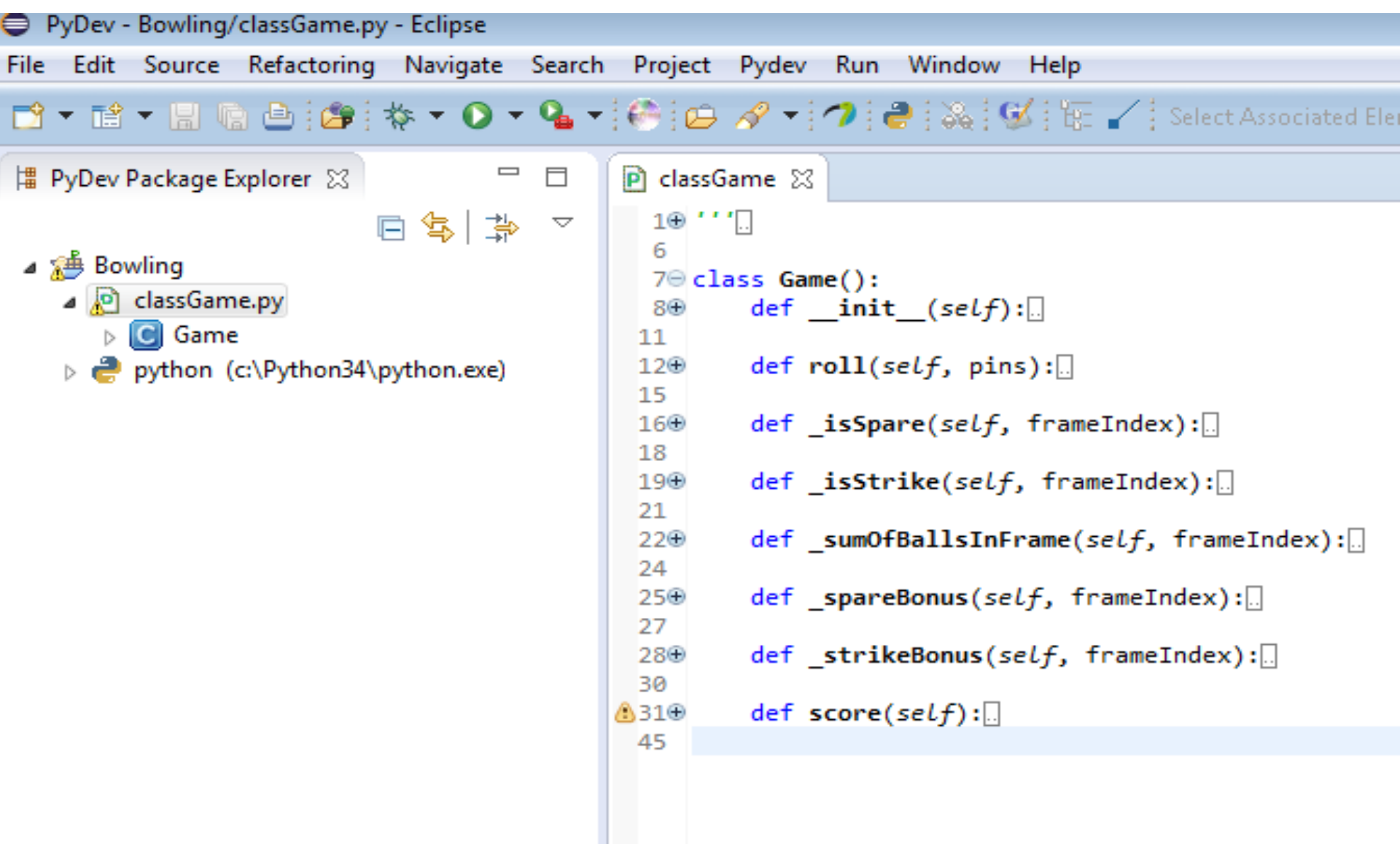
PyUnit en eclipse



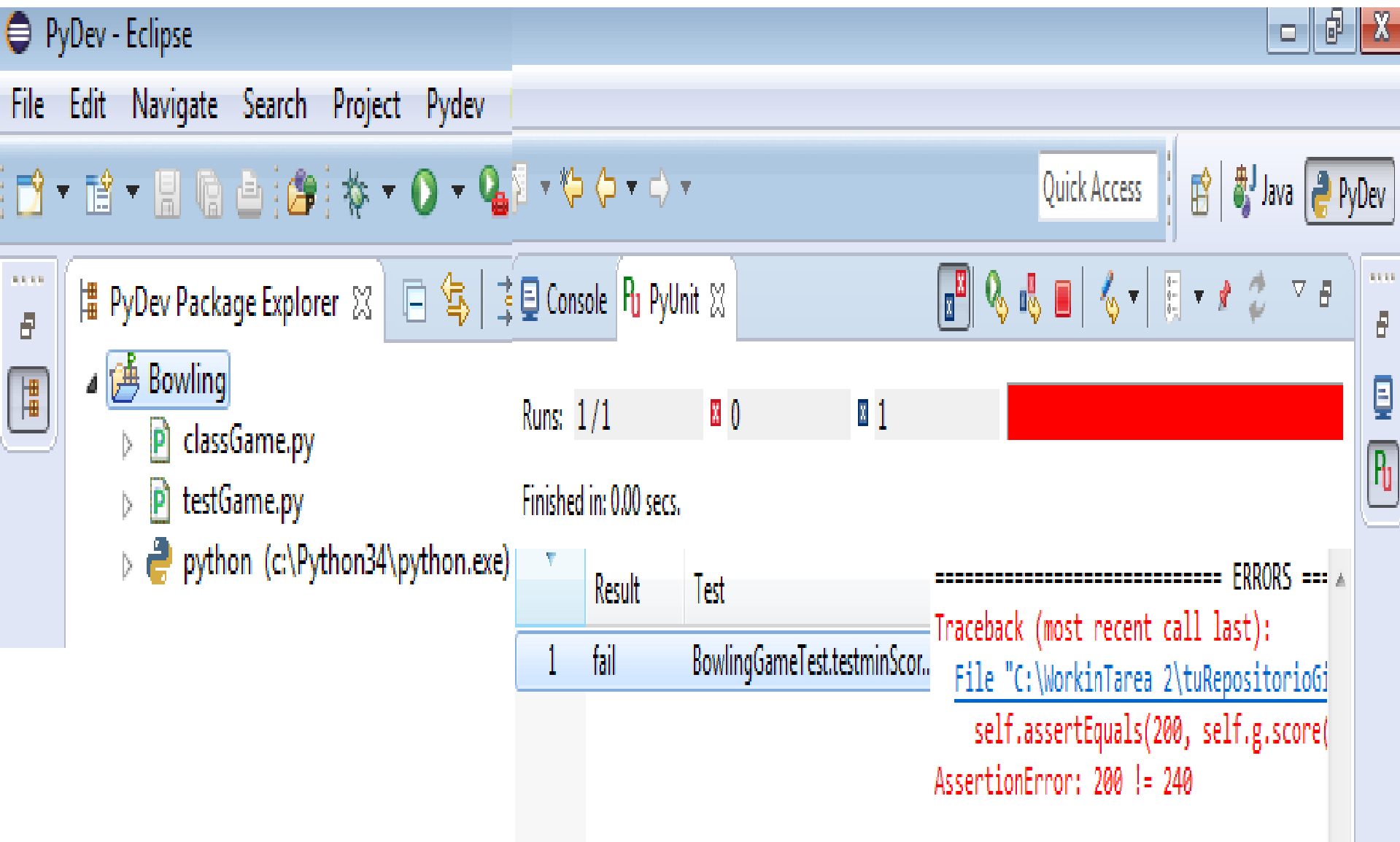
The screenshot shows the Eclipse IDE with the PyDev plugin. The left sidebar displays the 'PyDev Package Explorer' for a project named 'Bowling'. It contains two Python files: 'classGame.py' and 'testGame.py'. Under 'testGame.py', there is a class 'BowlingGameTest' with methods 'setUp', 'testminScoringMinRollsGame', and 'roll_many', along with a '__main__' block. The right pane shows the code for 'testGame.py'.

```
1  '''
2  Created on 7/1/2015
3
4  @author: alfonzo
5  '''
6  import unittest
7
8
9  from classGame import Game
10 import unittest
11
12 class BowlingGameTest(unittest.TestCase):
13     def setUp(self):
14         self.g = Game()
15
16     def testminScoringMinRollsGame(self):
17         self.roll_many(9, 10)
18         self.assertEqual(240, self.g.score())
19
20
21     def roll_many(self, n, pins):
22         for i in range(n):
23             self.g.roll(pins)
24
25
26
27 if __name__ == "__main__":
28     #import sys;sys.argv = ['', 'Test.testName']
29     unittest.main()
```

PyUnit en eclipse



PyUnit en eclipse



The screenshot shows the Eclipse IDE with the PyDev plugin. The PyDev Package Explorer on the left shows a project named 'Bowling' containing 'classGame.py', 'testGame.py', and a Python interpreter path 'python (c:\Python34\python.exe)'. The Console view shows the test results for 'BowlingGameTest.testminScore'. The PyUnit view shows a table of test results.

PyDev - Eclipse

File Edit Navigate Search Project Pydev

Quick Access

PyDev Package Explorer

Bowling

- classGame.py
- testGame.py
- python (c:\Python34\python.exe)

Console

PyUnit

Runs: 1/1 0 1

Finished in: 0.00 secs.

	Result	Test
1	fail	BowlingGameTest.testminScore

===== ERRORS =====

Traceback (most recent call last):

File "C:\WorkinTarea 2\tuRepositorioGi

self.assertEqual(200, self.g.score(

AssertionError: 200 != 240

PyUnit en eclipse

PyDev - Eclipse

File Edit Navigate Search Project Pydev Run Window

Quick Access

PyDev Package Explorer

- Bowling
 - classGame.py
 - testGame.py
 - python (c:\Python34\python.exe)

PyUnit

Runs: 1 / 1 ✗ 0 ✗ 0

Finished in: 0.00 secs.

	Result	Test

Otro análisis de dominio de datos

Basado en el clásico juego de la vieja...

- Requerimos crear una función que dado un juego de la vieja, determine quién es el ganador.
- Adicionalmente debemos realizar una Suite de Prueba para certificar nuestro algoritmo.
- ¿Cuál es el dominio de la función?
- ¿Cuál es el rango?

Análisis de dominio de datos

- Si numeramos las casillas del tablero (0 al 8)

0	1	2
3	4	5
6	7	8

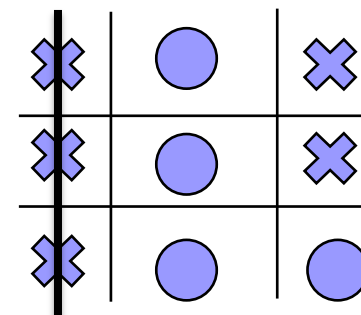
- Podemos definir un juego como un arreglo de 9 posiciones donde la posición representa el turno y el valor representa la casilla seleccionada por el jugador
 $\{ 0, 8, 2, 1, 6, \dots \}$
- Entonces el Dominio de la Función será el conjunto de arreglos de 9 posiciones donde cada número está contenido una sola vez

Análisis de dominio de datos

- ¿Cuántos valores contiene este conjunto?

- $9! = 362.880$ Valores

- ¿Son todos ellos juegos válidos?



- Partamos del ejemplo anterior...

- $\{ 0, 8, 2, 1, 6, \dots \}$

- $\{ 0, 8, 2, 1, 6, 4, 3, \dots \}$ En este juego ya el primer jugador ha ganado...

- Después que el ganador logró la línea, es irrelevante para la función los números que vengan a continuación, por ende, todos los valores son válidos.

Hagamos el ejercicio con PyUnit...

- Tenemos la clase tictactoe
- Definamos una suite de prueba con:
 - Casos No Frontera
 - Casos Frontera
 - Casos Inválidos
 - Casos Esquina
 - Casos de Malicia

Hagamos el ejercicio con PyUnit...

- Variables que tenemos
 - ☐ Rango de la función? (0,1,2)
 - ☐ Turno de Victoria (Ganador?)
 - ☐ Tipo de Victoria (Fila, Columna, Diagonal)
 - ☐ Número de Fila, Columna o Diagonal

- Definir la Suite de Pruebas...

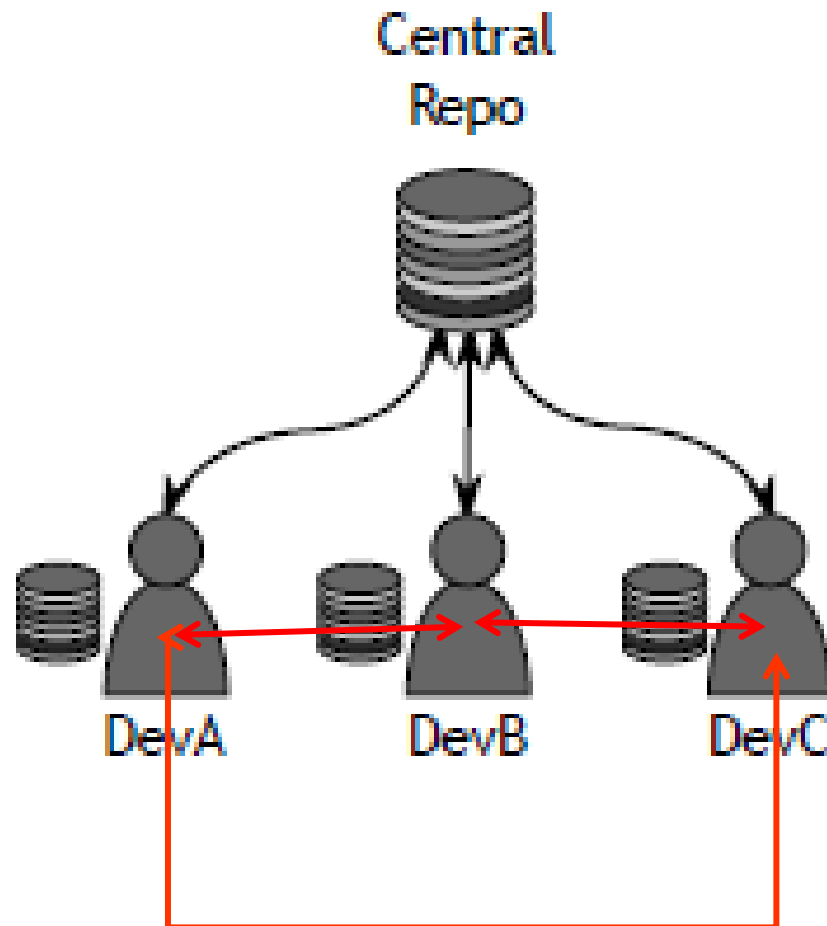
Agenda

- Verificación:
 - PyUnit
- **Gestión de Configuraciones y Versiones de software:**
 - **Introducción a *Git* básico**
- Equipo:

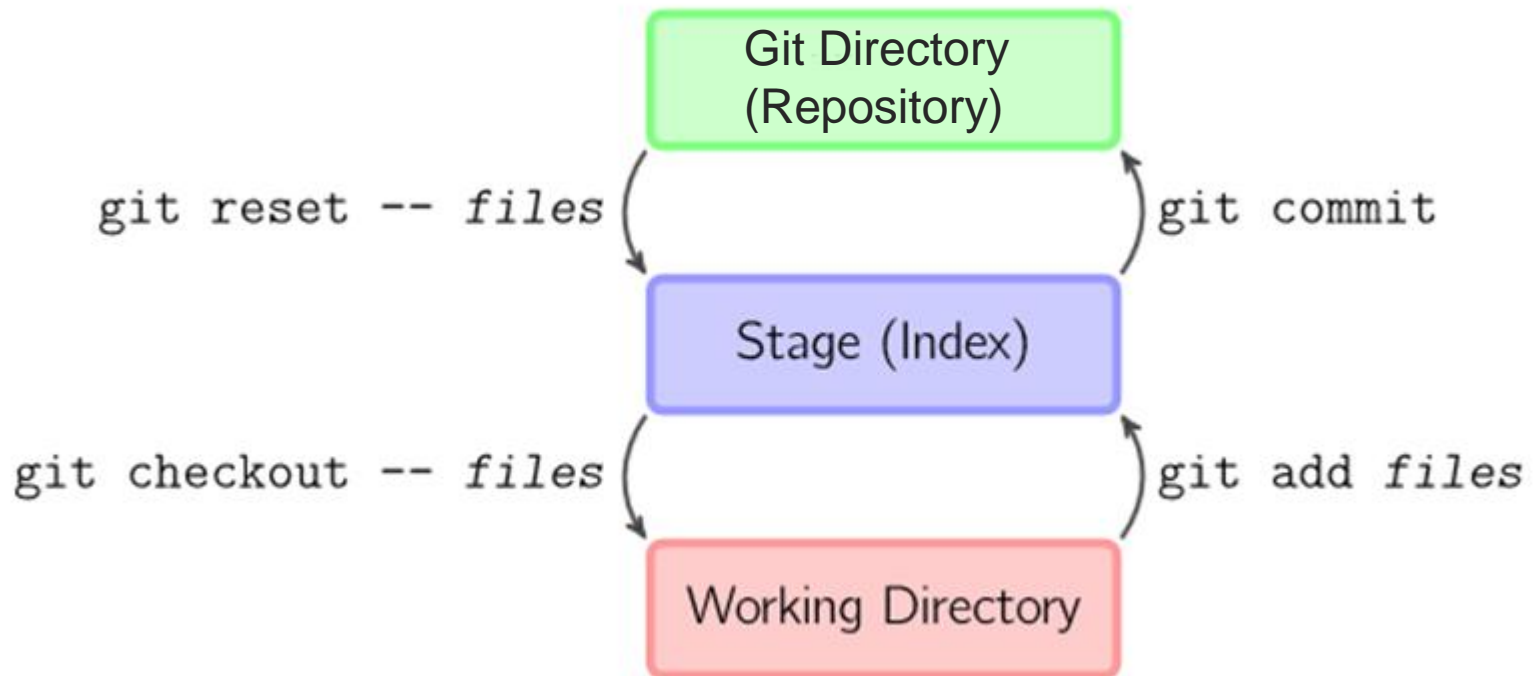
EGIT es GIT

- ¿Qué es?
- ¿Qué hace?
- ¿Cómo funciona?
- ¿Cómo usarlo?

Git - Distribuido



Git – Local

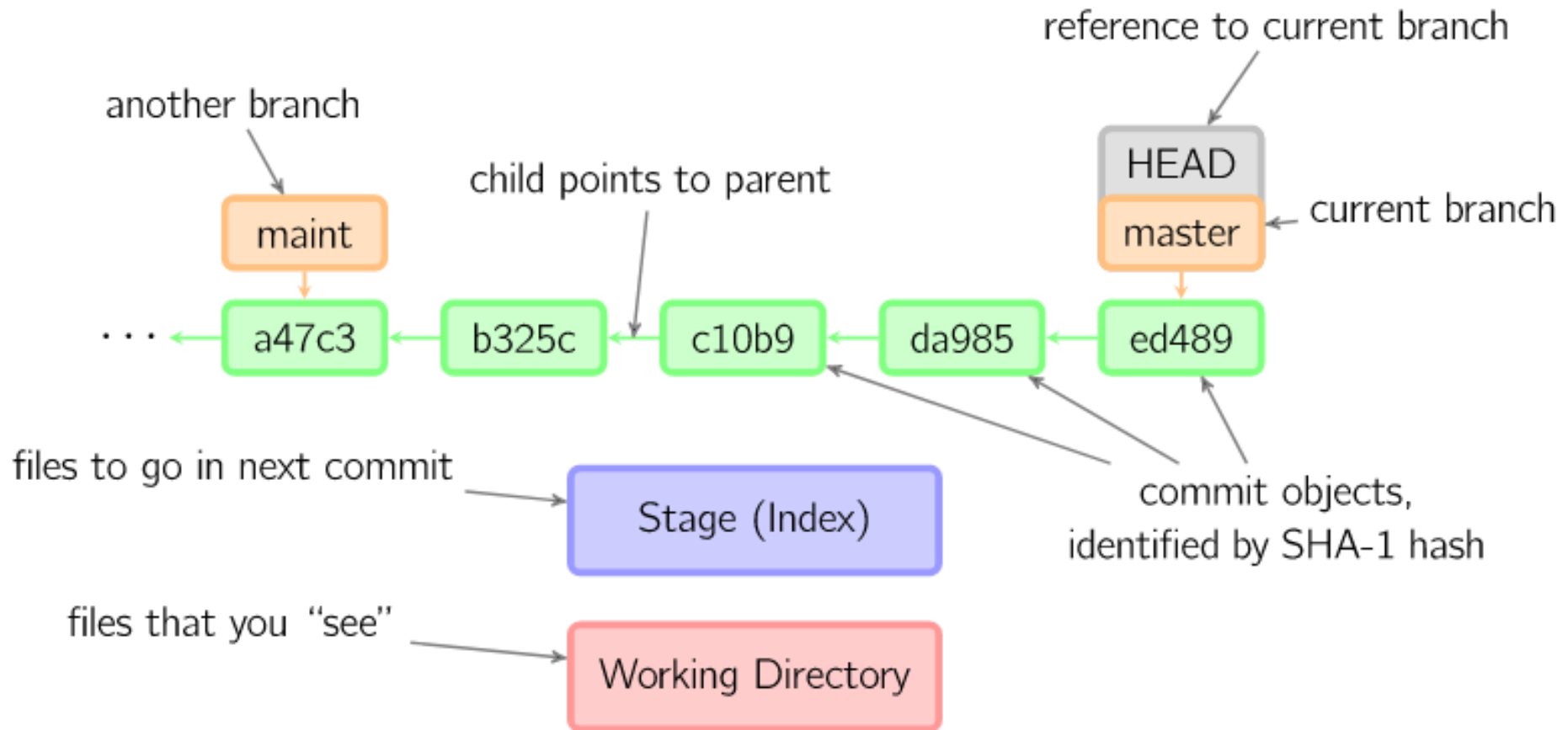


commit (en nomenclatura de la Teoría: remite y registra en repositorio local
Número de versión por *hash* (SHA-1) del **commit**:

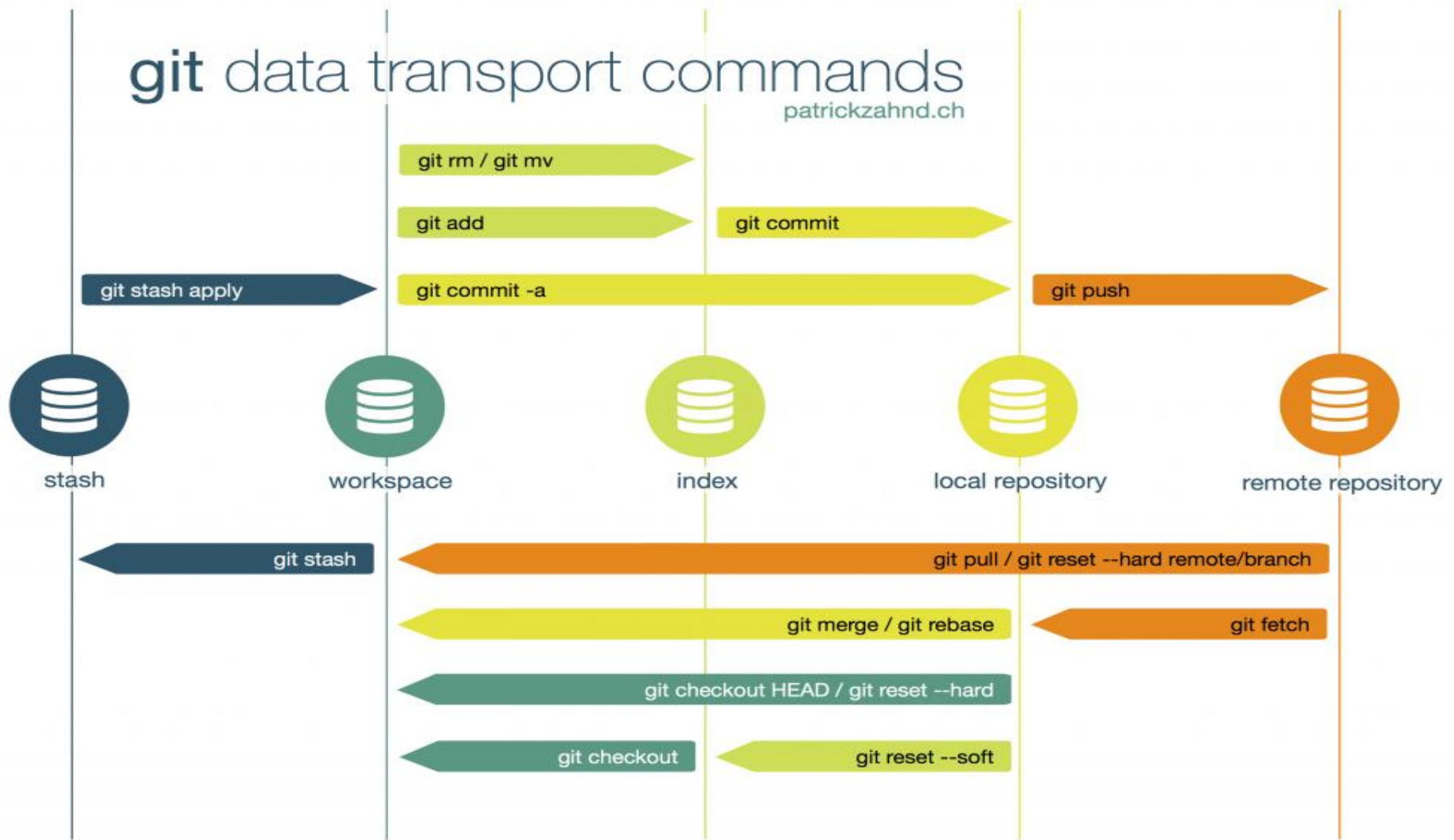
d921970aadf03b3cf0e71becdaab3147ba71cdef

El *hash* corto: **d92197**

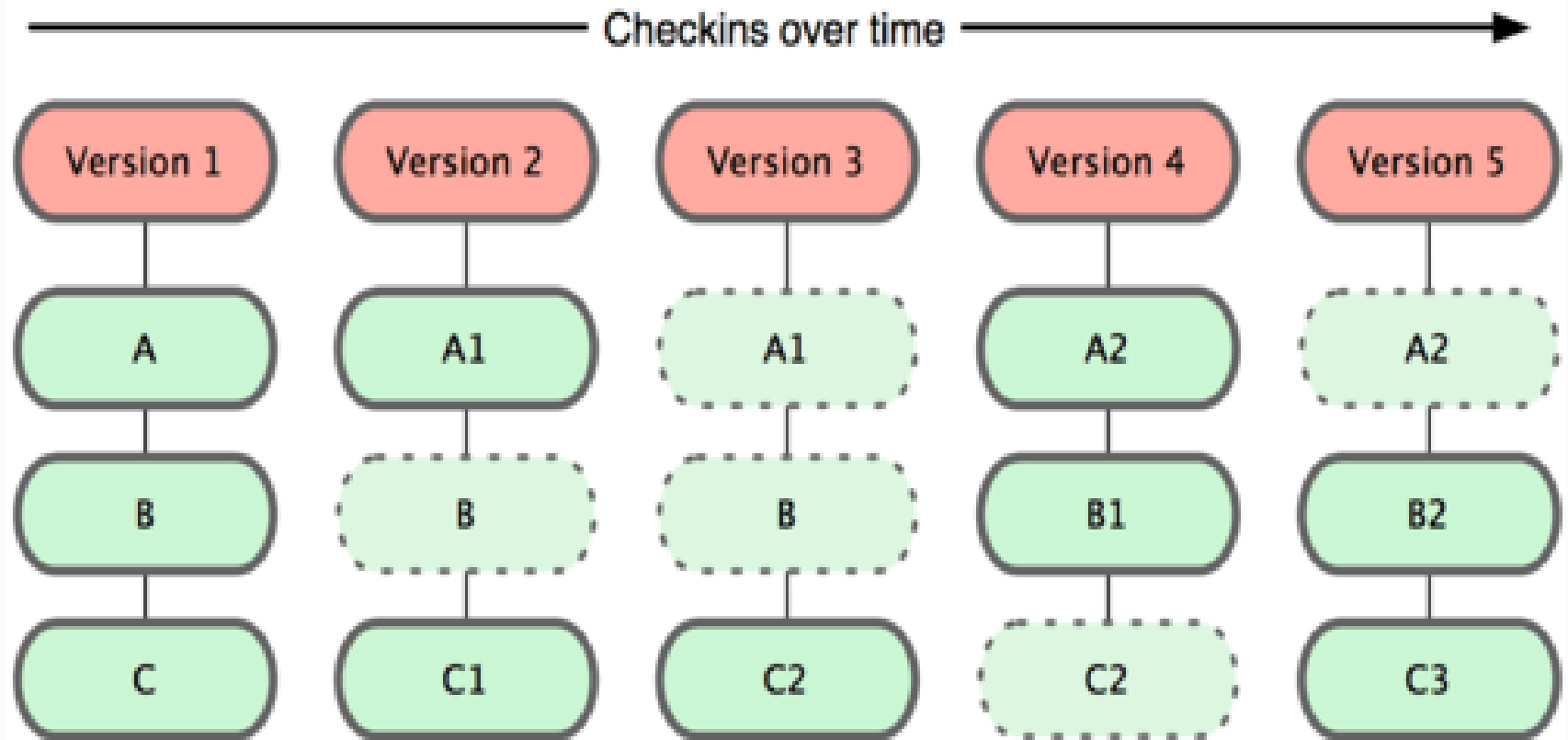
Git – Repositorio



Git – comandos de transporte de datos

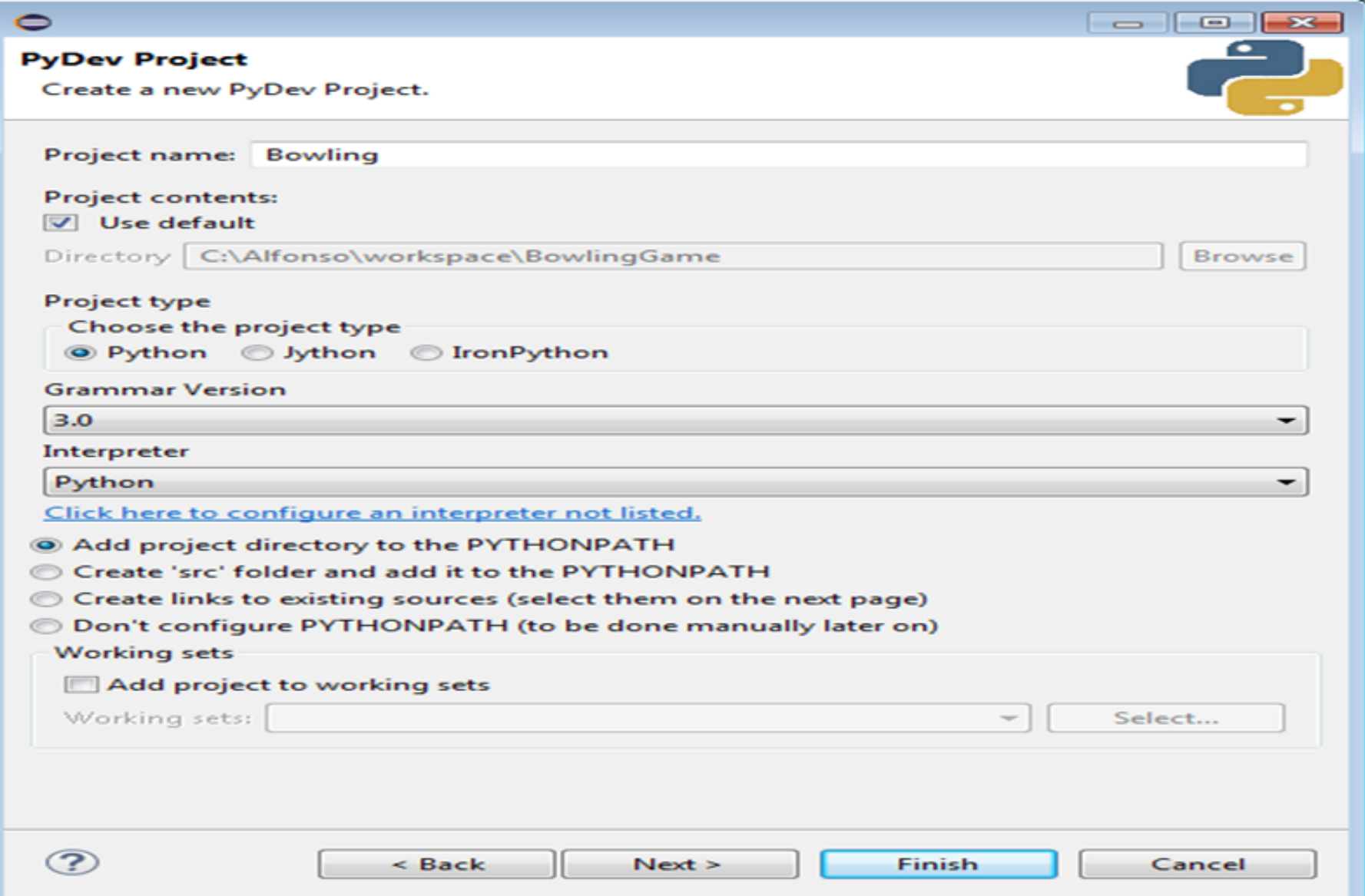


Git - Versiones



ProGit, by Scot Chacon

Crear el proyecto Bowling



PyDev Project
Create a new PyDev Project.

Project name:

Project contents:
☒ Use default
 Directory:

Project type
 Choose the project type
☒ Python ☐ Jython ☐ IronPython

Grammar Version

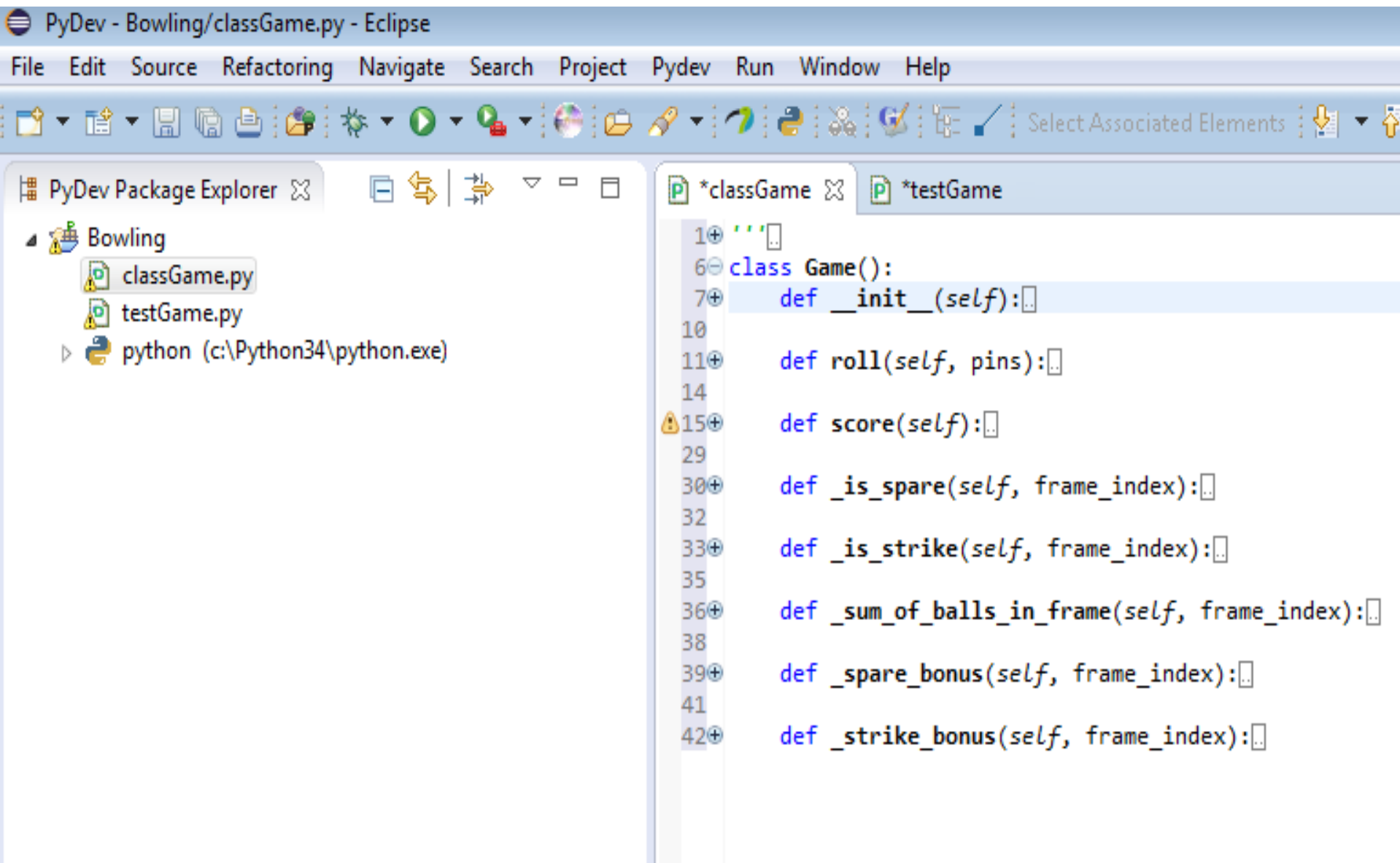
Interpreter

[Click here to configure an interpreter not listed.](#)

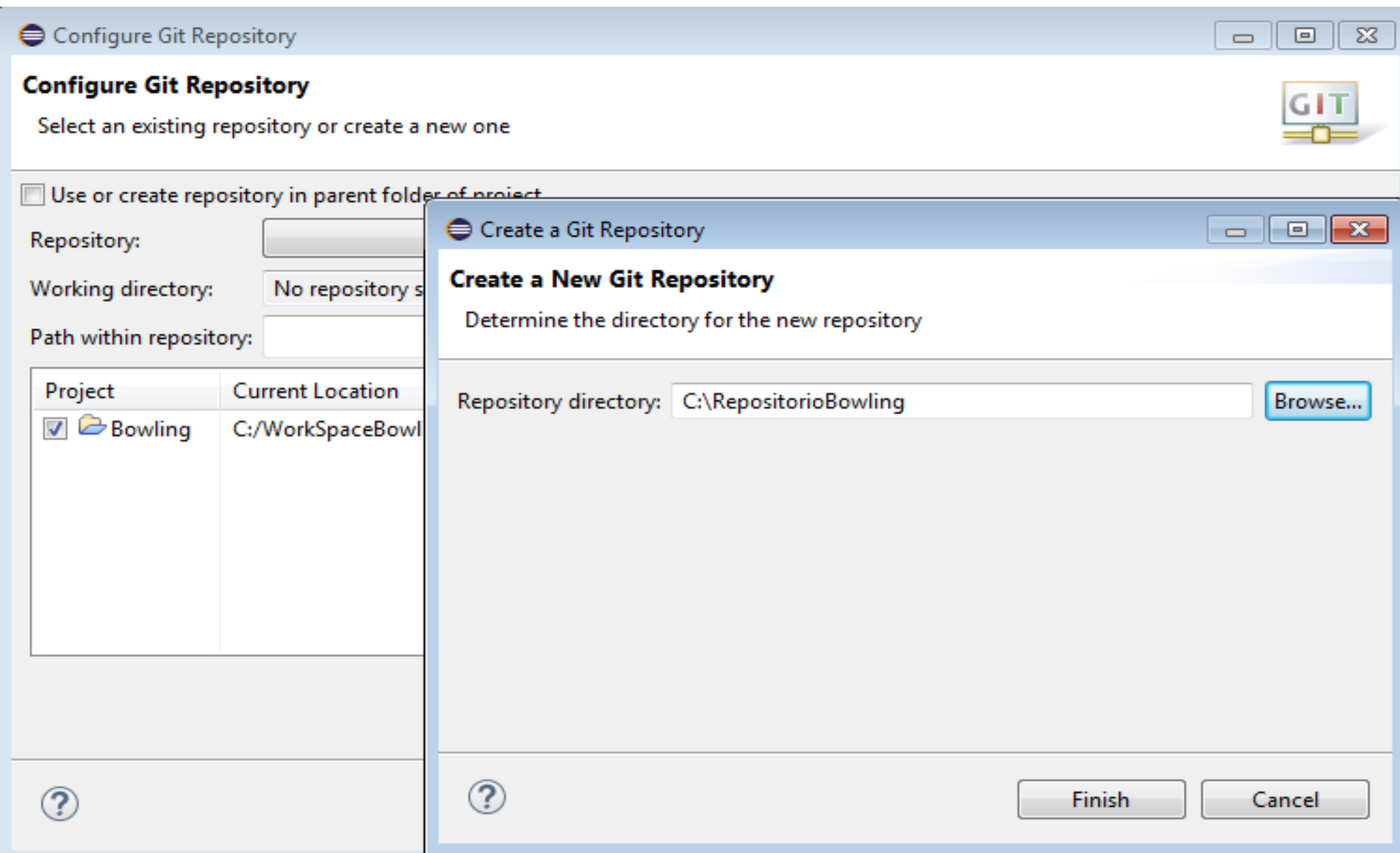
☒ Add project directory to the PYTHONPATH
☐ Create 'src' folder and add it to the PYTHONPATH
☐ Create links to existing sources (select them on the next page)
☐ Don't configure PYTHONPATH (to be done manually later on)

Working sets
☐ Add project to working sets
 Working sets:

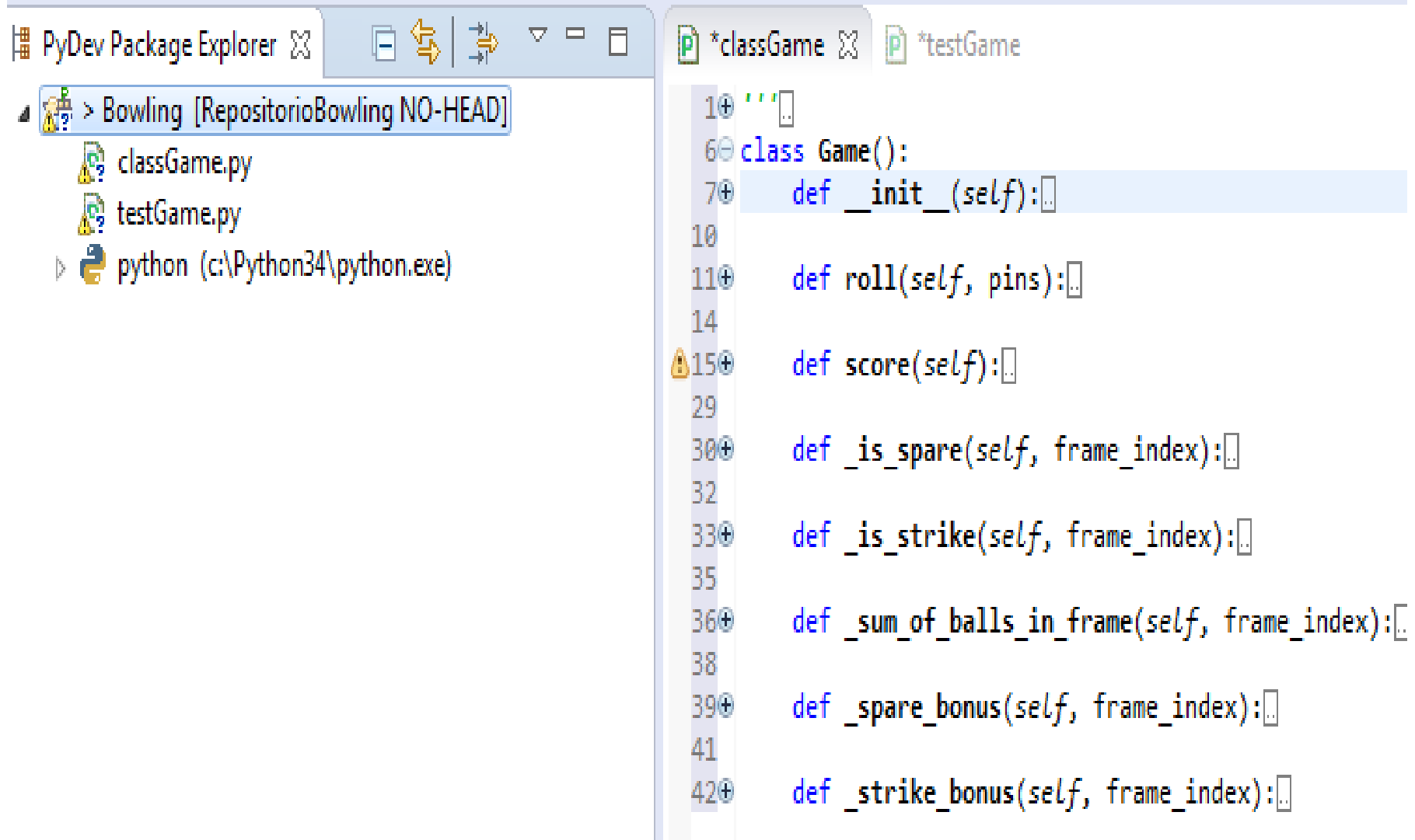
Proyecto Bowling



Crear un repositorio con EGit



Conectados localmente-RepositoryBowling



PyDev Package Explorer

- Bowling [RepositoryBowling NO-HEAD]
 - classGame.py
 - testGame.py
 - python (c:\Python34\python.exe)

*classGame

```
1  ...  
6  class Game():  
7      def __init__(self):  
10  
11     def roll(self, pins):  
14  
15     def score(self):  
29  
30     def _is_spare(self, frame_index):  
32  
33     def _is_strike(self, frame_index):  
35  
36     def _sum_of_balls_in_frame(self, frame_index):  
38  
39     def _spare_bonus(self, frame_index):  
41  
42     def _strike_bonus(self, frame_index):
```

*testGame

Perspectiva Git - Unstaged

The screenshot displays the Git GUI interface. On the left, a sidebar shows the repository structure under 'RepositorioBowling [NO-HEAD]'. The 'Working Directory - C:\Repo' is expanded, showing files: .project, .pydevproject, classGame.py, and testGame.py. The main window has tabs for Properties, History, Synchronize, Git Staging (selected), and Git Reflog. Below the tabs, the repository name 'RepositorioBowling' is shown. The 'Unstaged Changes (4)' section lists four files: .project - Bowling, .pydevproject - Bowling, classGame.py - Bowling, and testGame.py - Bowling. The 'Staged Changes (0)' section is empty. The 'Commit Message' section is also empty. At the bottom right, the 'Author' and 'Committer' fields are both set to 'Alfonso Reinoza <jareinozacg@>'. Below these fields are two buttons: 'Commit and Push' and 'Commit'.

Git Repositories

- RepositorioBowling [NO-HEAD]
 - Branches
 - Tags
 - References
 - Remotes
 - Working Directory - C:\Repo
 - .git
 - Bowling
 - .project
 - .pydevproject
 - classGame.py
 - testGame.py

Properties **History** **Synchronize** **Git Staging** **Git Reflog**

Filter files

RepositorioBowling

Unstaged Changes (4)

- .project - Bowling
- .pydevproject - Bowling
- classGame.py - Bowling
- testGame.py - Bowling

Staged Changes (0)

Commit Message

Author: Alfonso Reinoza <jareinozacg@>

Committer: Alfonso Reinoza <jareinozacg@>

Commit and Push Commit

Perspectiva Git - Staged

The screenshot displays the Git GUI interface with the 'Git Staging' tab selected. The left sidebar shows the repository structure, including the 'Working Directory' at 'C:\Reposi'. The main area is divided into three sections: 'Unstaged Changes (0)', 'Staged Changes (4)', and 'Commit Message'. The 'Staged Changes' section lists four files: '.project - Bowling', '.pydevproject - Bowling', 'classGame.py - Bowling', and 'testGame.py - Bowling'. The 'Commit Message' section has a text area for the message and fields for 'Author' and 'Committer', both set to 'Alfonso Reinoza <jareinozacg@>'. At the bottom right, there are buttons for 'Commit and Push' and 'Commit'.

Git Repositories

- RepositorioBowling [NO-HEAD] -
 - Branches
 - Tags
 - References
 - Remotes
 - Working Directory - C:\Reposi
 - .git
 - Bowling
 - .project
 - .pydevproject
 - classGame.py
 - testGame.py

Properties **History** **Synchronize** **Git Staging** **Git Reflog**

Filter files

RepositorioBowling

Unstaged Changes (0)

Staged Changes (4)

- .project - Bowling
- .pydevproject - Bowling
- classGame.py - Bowling
- testGame.py - Bowling

Commit Message

Author: Alfonso Reinoza <jareinozacg@>

Committer: Alfonso Reinoza <jareinozacg@>

Commit and Push Commit

El primer commit con EGit

The screenshot shows the EGit application window. At the top, there are tabs for 'Git Staging' and 'Git Reflog', a 'Filter files' search bar, and icons for refreshing, committing, and window management. Below the tabs is a header bar for the 'Repositorio Bowling'. The main area is divided into two columns. The left column has two sections: 'Unstaged Changes (0)' with an empty text box, and 'Staged Changes (4)' with a list of files: '.pydevproject - Bowling', 'classGame.py - Bowling', and 'testGame.py - Bowling'. The right column has a 'Commit Message' section with a text box containing 'Mi primer commit', and fields for 'Author' and 'Committer', both set to 'Alfonso Reinoza <jareinozacg@'. At the bottom right, there are two buttons: 'Commit and Push' and 'Commit'.

Git Staging Git Reflog

Repositorio Bowling

Unstaged Changes (0)

Staged Changes (4)

- .pydevproject - Bowling
- classGame.py - Bowling
- testGame.py - Bowling

Commit Message















Mi primer commit

Author:

Committer:

Committed

History




















Project: Bowling [RepositorioBowling]

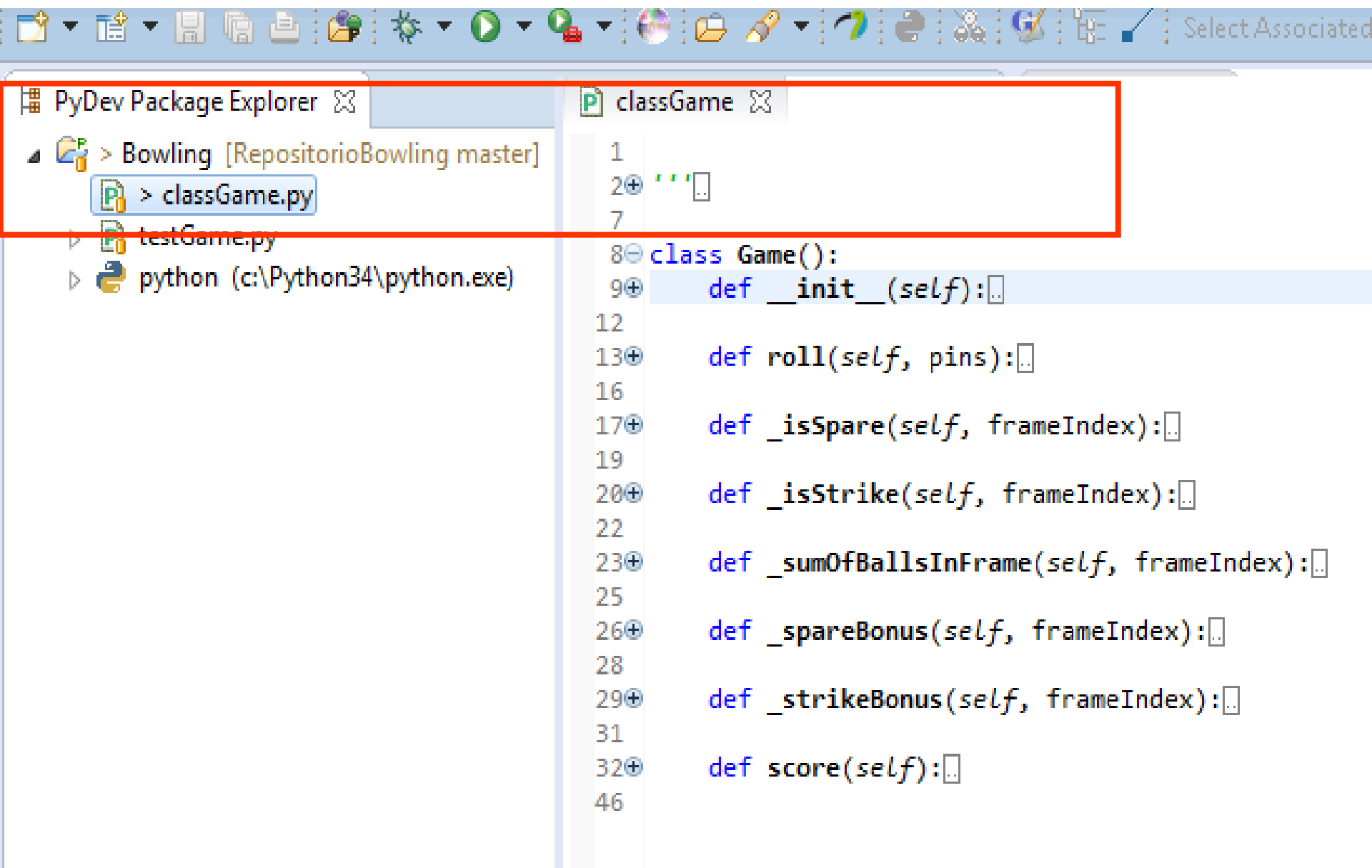
Id	Message	Author	Auth...	Com...	Com...
da32884	master (HEAD) Mi primer Com	Alfonso R	18 minut	Alfonso R	18 minut

```
commit da32884900fb610d6bf1a7e6fd9369ec2a01a0c4
Author: Alfonso Reinoza <jareinozacg@gmail.com>
2015-04-15 06:58:37
Committer: Alfonso Reinoza <jareinozacg@gmail.com>
2015-04-15 06:58:37
Branches: master

Mi primer Commit
```

 Bowling/.project
 Bowling/.pydevproject
 Bowling/classGame.py
 Bowling/testGame.py

Cambio



Cambio - Unstaged

Repository: Bowling

Unstaged Changes (1)

> classGame.py - Bowling

Staged Changes (0)

Commit Message

Author: Alfonso Riossa <jarinosag@gmail.com>

Committer: Alfonso Riossa <jarinosag@gmail.com>

Commit and Push

Commit

Cambio – Mi segundo commit

Package Explorer | Properties | History | Synchronize | Git Staging | Git Reflog

Bowling [RepositorioBowling master]

- classGame.py
- testGame.py

Project: Bowling [RepositorioBowling]

Id	Message	Author	Authore...	Commi...	Commi...
ec9b798	master HEAD Mi segundo commit	Alfonso Reir	73 seconds	Alfonso Reir	73 seconds
201c998		Alfonso Reir	4 minutes	Alfonso Reir	4 minutes

- Open in Commit Viewer
- Checkout**
- Push Commit...
- Create Branch...
- Delete Branch
- Rename Branch...

Efecto del Checkout

Properties History Synchronize Git Staging Git Reflog

Project: Bowling [RepositorioBowling]

Id	Message	Author	Authore...	Commi...	Commi...
ec9b798	master Mi segundo commit	Alfonso Reir	2 minutes a	Alfonso Reir	2 minutes a
201c998	HEAD Mi primer comit	Alfonso Reir	5 minutes a	Alfonso Reir	5 minutes a

Creamos una rama

The screenshot shows an IDE interface with the 'Team' menu open. The 'Package Explorer' on the left shows a project named 'Bowling' with files 'classGame' and 'testGame.p'. The 'Team' menu is open, showing options like 'Go Into', 'Open in New Window', 'Show In', 'Copy', 'Copy Qualified Name', 'Paste', 'Delete', 'Remove from Context', 'Build Path', 'Refactor', 'Import...', 'Export...', 'Refresh', 'Close Project', 'Assign Working Sets...', 'Validate', 'Profile As', 'Debug As', 'Run As', 'Team', and 'Compare With'. The 'Team' menu is highlighted, and the 'New Branch...' dialog is open, showing the 'master' branch and an 'Other...' option.

Package Explorer

Bowling [Repository]

classGame

testGame.p

Go Into

Open in New Window

Show In Alt+Shift+W

Copy Ctrl+C

Copy Qualified Name

Paste Ctrl+V

Delete Delete

Remove from Context Ctrl+Alt+Shift+Down

Build Path

Refactor Alt+Shift+T

Import...

Export...

Refresh F5

Close Project

Assign Working Sets...

Validate

Profile As

Debug As

Run As

Team

Compare With

Fetch from Upstream

Push HEAD...

Remote

Switch To

Advanced

Pull

Synchronize Workspace

Merge Tool

Merge...

Rebase...

Reset...

Create Patch...

Apply Patch...

Add to Index

Remove from Index

Ignore

Show in History

Show in Repositories View

Disconnect

New Branch...

master

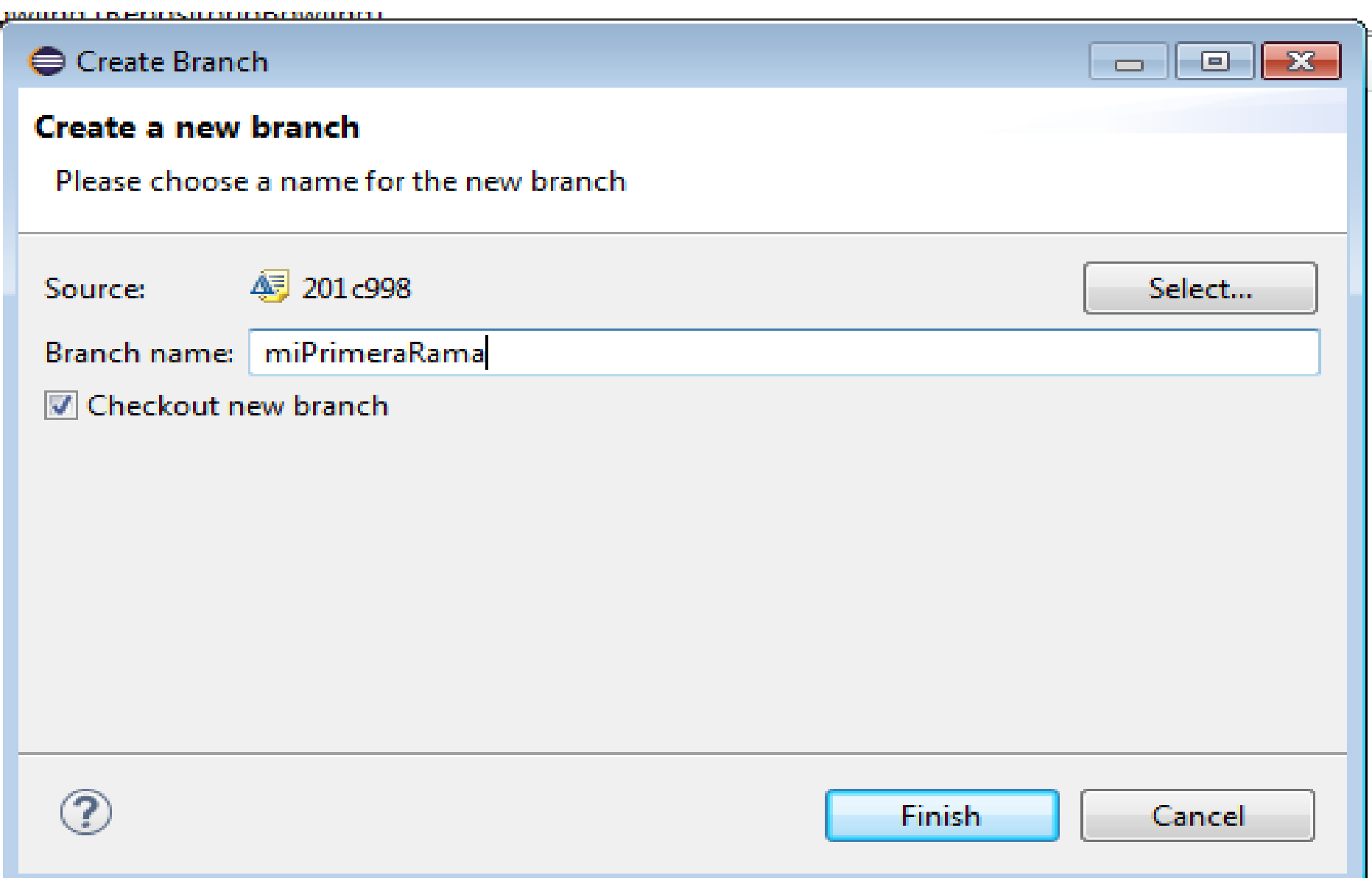
Other...

4-15 20:16:20

5-04-15 20:16:20

gundo commit)

Creamos una rama




The image shows a 'Create Branch' dialog box with a title bar containing a globe icon and the text 'Create Branch'. The dialog has standard window controls (minimize, maximize, close) in the top right. The main content area is titled 'Create a new branch' and includes the instruction 'Please choose a name for the new branch'. Below this, there is a 'Source:' label followed by a file icon and the text '201c998', and a 'Select...' button to its right. A 'Branch name:' label is followed by a text input field containing 'miPrimeraRama'. Below the input field is a checked checkbox labeled 'Checkout new branch'. At the bottom left is a help icon (question mark in a circle). At the bottom right are 'Finish' and 'Cancel' buttons.

Create Branch

Create a new branch

Please choose a name for the new branch

Source:  201c998 Select...

Branch name:

☒ Checkout new branch

? Finish Cancel

Creamos una rama




Commit una rama

Properties History Synchronize Git Staging Git Reflog Filter files

RepositorioBowling

Unstaged Changes (0)

Staged Changes (1)

 classGame.py - Bowling

Commit Message

Commit a la rama

Author: Alfonso Reinoza <jareinozacg@gmail.com>

Committer: Alfonso Reinoza <jareinozacg@gmail.com>

Commit and Push Commit

Grafo Git

Project: Bowling [RepositorioBowling]

Id	Message	Author	Authore...	Commi...	Commi...
829d411	miPrimeraRama HEAD Commit a la ran	Alfonso Reir	14 seconds	Alfonso Reir	14 seconds
ec9b798	master Mi segundo commit	Alfonso Reir	9 minutes	Alfonso Reir	9 minutes
201c998	Mi primer comit	Alfonso Reir	12 minutes	Alfonso Reir	12 minutes

```

commit 201c998cb128f3611cb971a666e5b1671e507071
Author: Alfonso Reinoza <jareinozacg@gmail.com> 2015-04-15 20:16:20
Committer: Alfonso Reinoza <jareinozacg@gmail.com> 2015-04-15 20:16:20
Child: 829d411339965453c21adff7c6bc29a65e233a5a (Commit a la rama)
Child: ec9b79815d5e3773c9d8e3dc92dfd376892328c3 (Mi segundo commit)
Branches: master, miPrimeraRama
  
```

Checkout de la rama al master







Properties History Synchronize Git Staging History





Project: Bowling [RepositorioBowling]

Id	Message	Author	Authored ...	Co
1c9be12	miPrimeraRama HEAD Mi segundo Commit	Alfonso Reinoz	10 minutes ago	Alfo
3ad5928	Commit en la rama	Alfonso Reinoz	81 minutes ago	Alfo
da32884	master Mi primer Commit	Alfonso Reinoz	3 hours ago	Alfo






- Open in Commit Viewer
- Checkout
- Push Commit...
- Create Branch...

Checkout de la rama al master

 Properties  History  Synchronize  Git Staging  History 

Project: Bowling [RepositorioBowling]

Id	Message	Author	Authored ...
1c9be12	  miPrimeraRama Mi segundo Commit	Alfonso Reinoz12 minutes ago	
3ad5928	 Commit en la rama	Alfonso Reinoz83 minutes ago	
da32884	  master [HEAD] Mi primer Commit	Alfonso Reinoz3 hours ago	

Commit al master

History History

Project: Bowling [RepositorioBowling]

Id	Message	Author
214ad87	master (HEAD) Segundo Commit master	Alfonso Reir
1c9be12	miPrimeraRama Mi segundo Commit	Alfonso Reir
3ad5928	Commit en la rama	Alfonso Reir
da32884	Mi primer Commit	Alfonso Reir

```
commit 214ad87533f09164a2503d93135b20fdc210bd91
Author: Alfonso Reinoza <jareinozacg@gmail.com> 2015-04-15 10:03:42
Committer: Alfonso Reinoza <jareinozacg@gmail.com> 2015-04-15 10:03:42
Parent: da32884900fb610d6bf1a7e6fd9369ec2a01a0c4 (Mi primer Commit)
Branches: master
```

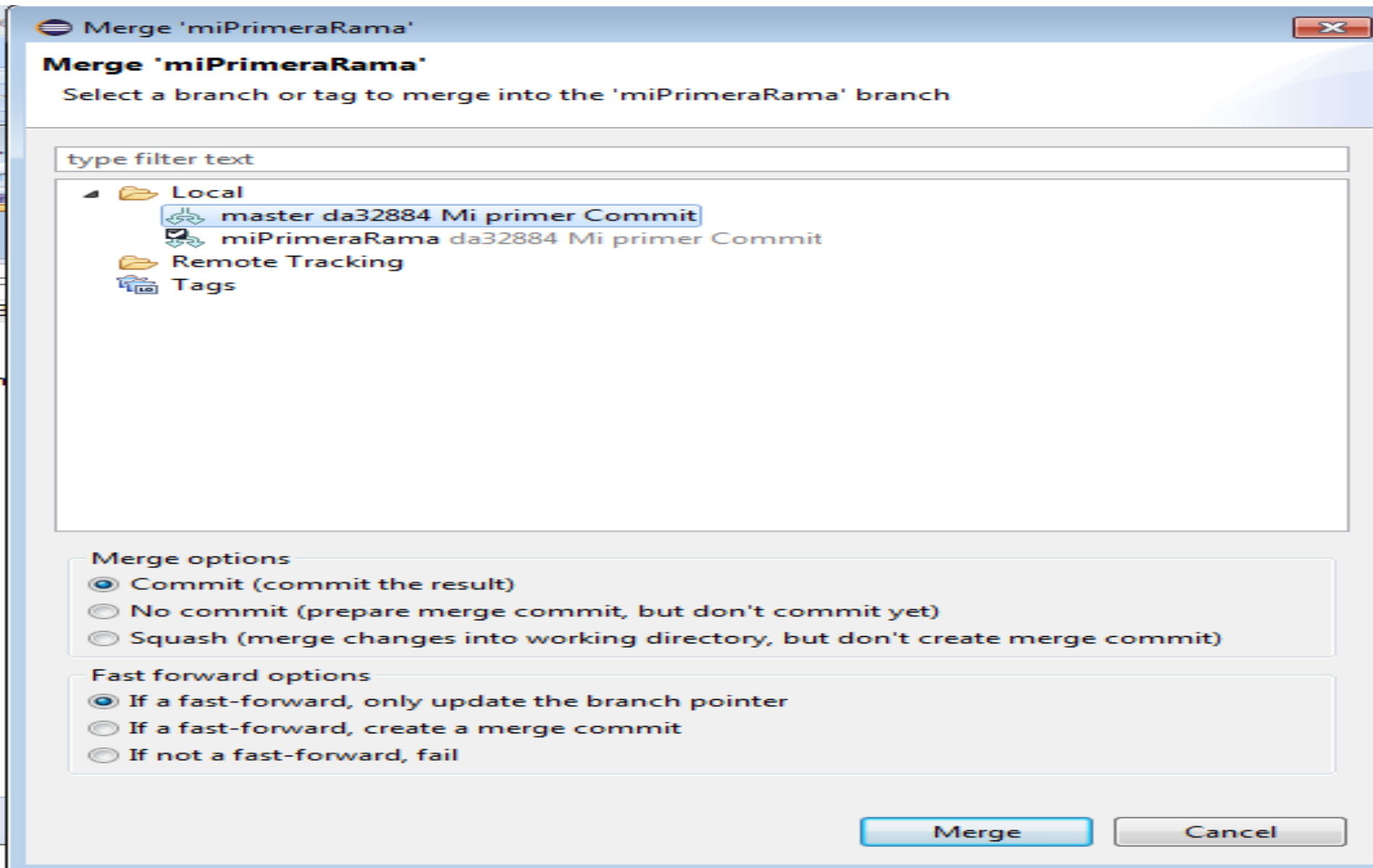
Segundo Commit master

Merge

The screenshot shows an IDE interface with a project explorer on the left and a context menu open. The project explorer shows a project named 'PyDev Package E...' and a sub-project 'Bowling [Repository]'. Inside 'Bowling', there are files 'classGame.py', 'testGame.py', and a 'python' interpreter path '(c:\Py...'. The context menu is open over the 'Bowling' project, showing various actions. The 'Merge...' option is highlighted at the top of the menu.

Action	Shortcut
Rename...	
Remove from Context	Ctrl+Alt+Shift+ Down
Import...	
Export...	
Refresh	F5
Close Project	
Close Unrelated Projects	
Profile As	
Debug As	
Run As	
Validate	
Team	
Merge...	
Rebase...	
Reset...	
Create Patch...	
Apply Patch...	
Add to Index	
Remove from Index	
Ignore	
Show in History	
Show in Repositories View	
Disconnect	

Merge rama - master



Aviso de conflicto

The screenshot shows the PyDev Package Explorer on the left with the following structure:

- Bowling [RepositorioBowling|Conflicts master]
 - classGame.py
 - Game
 - testGame.py
 - BowlingGameTest
 - __main__
 - python (c:\Python34\python.exe)

On the right, a "Merge Result" dialog box is open, showing a conflict. The dialog has a title bar with a close button (X). The content area displays:

Result: Conflicting

Merge input

214ad875: Segundo Commit master (Alfonso Reinoza on 15/04/2015 10:03 AM)

1c9be12f: Mi segundo Commit (Alfonso Reinoza on 15/04/2015 09:23 AM)

At the bottom of the dialog is an "OK" button.

Below the dialog, a portion of a Python file is visible, showing a conflict in the `testMinRollsGame` method:

```

14
15 <<<<<<< HEAD
16     def testMinRollsGame(self):
17         self.roll_many(9, 10)
18         self.assertEqual(90, self.g.score())
19     =====

```

Ejecutamos el merge

Package Explorer

Bowling [RepositorioBowling]

- classGame.py
- testGame.py

CONFLICTO

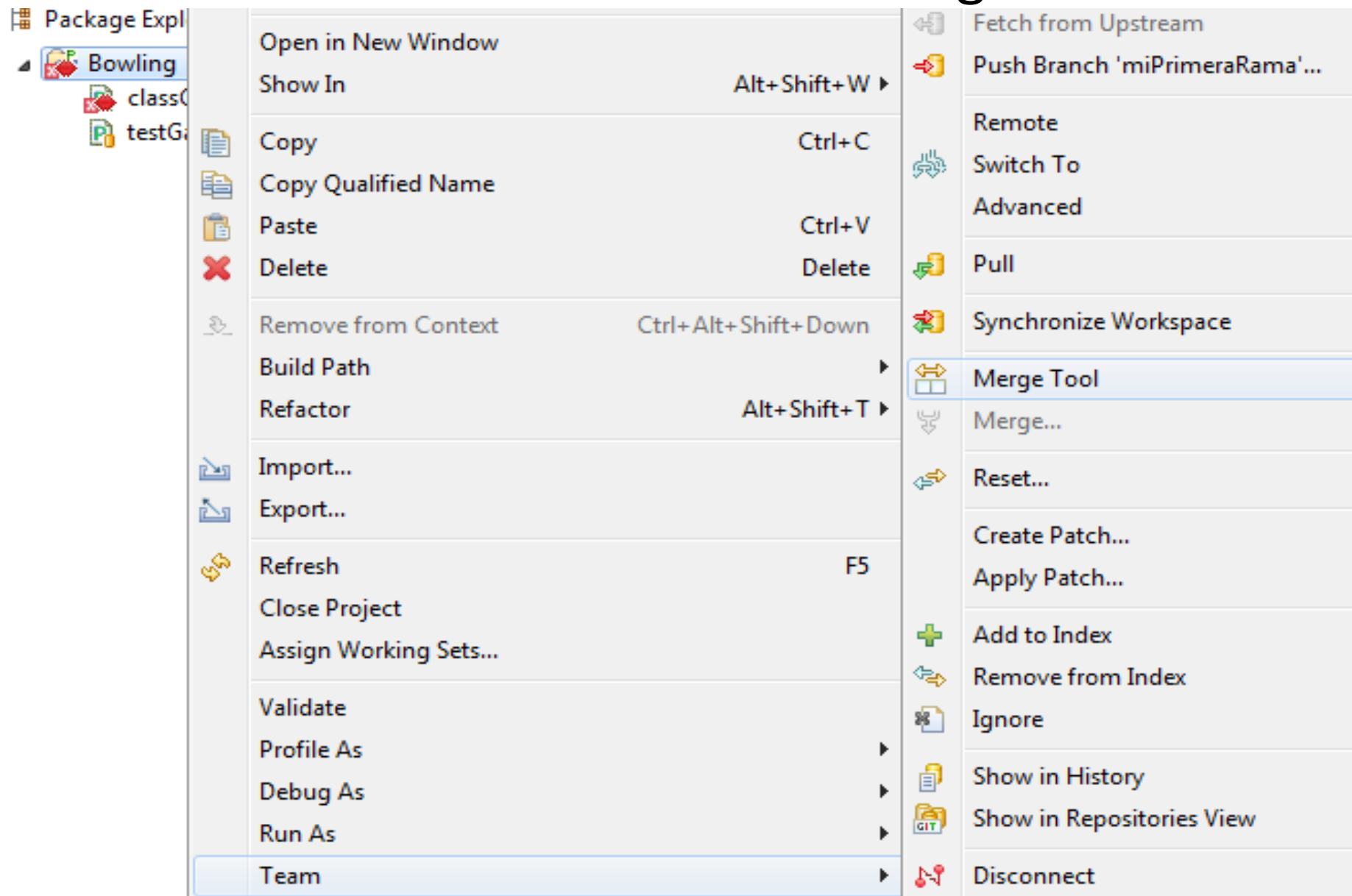
```
1
2 <<<<<<< HEAD
3
4 =====
5 >>>>>>> refs/heads/master
6 ...
7 Created on 7/1/2015
8
9 @author: alfonzo
10 ...
11
12 class Game():
13     def __init__(self):
14         self._rolls = [0] * 21
15         self._currentRoll = 0
16
17     def roll(self, pins):
18         self._rolls[self._currentRoll] = pins
```

Properties History Synchronize Git Staging Git Reflog

Project: Bowling [RepositorioBowling]

Id	Message	Author	Authore...	Commi...	Commi...
829d411	miPrimeraRama HEAD Commit a la rama	Alfonso Reir	4 minutes a	Alfonso Reir	4 minutes a
ec9b798	master Mi segundo commit	Alfonso Reir	12 minutes a	Alfonso Reir	12 minutes a
201c998	Mi primer comit	Alfonso Reir	15 minutes a	Alfonso Reir	15 minutes a

En conflicto tenemos un merge tool



The screenshot shows the Eclipse IDE's Package Explorer on the left with a project named 'Bowling'. A context menu is open over the 'Bowling' project, listing various actions. The 'Team' option at the bottom of the menu is highlighted, and its submenu is displayed on the right. In this submenu, the 'Merge Tool' option is highlighted.

Menu Item	Shortcut
Open in New Window	
Show In	Alt+Shift+W
Copy	Ctrl+C
Copy Qualified Name	
Paste	Ctrl+V
Delete	Delete
Remove from Context	Ctrl+Alt+Shift+Down
Build Path	
Refactor	Alt+Shift+T
Import...	
Export...	
Refresh	F5
Close Project	
Assign Working Sets...	
Validate	
Profile As	
Debug As	
Run As	
Team	

Team Submenu Item
Fetch from Upstream
Push Branch 'miPrimeraRama'...
Remote
Switch To
Advanced
Pull
Synchronize Workspace
Merge Tool
Merge...
Reset...
Create Patch...
Apply Patch...
Add to Index
Remove from Index
Ignore
Show in History
Show in Repositories View
Disconnect

El merge tool

Text Compare

Commit a la rama - 829d411

```
1  
2  
3  
4 Created on 7/1/2015  
5  
6 @author: alfonzo  
7  
8  
9 class Game():  
10     def __init__(self):  
11         self._rolls = [0] * 21  
12         self._currentRoll = 0  
13  
14     def roll(self, pins):  
15         self._rolls[self._currentRoll] = pins  
16         self._currentRoll += 1  
17  
18     def _isSpare(self, frameIndex):  
19         return self._rolls[frameIndex] + self._rolls[frameIndex]
```

Mi segundo commit - ec9b798

```
1  
2  
3 Created on 7/1/2015  
4  
5 @author: alfonzo  
6  
7  
8 class Game():  
9     def __init__(self):  
10         self._rolls = [0] * 21  
11         self._currentRoll = 0  
12  
13     def roll(self, pins):  
14         self._rolls[self._currentRoll] = pins  
15         self._currentRoll += 1  
16  
17     def _isSpare(self, frameIndex):  
18         return self._rolls[frameIndex] + self._rolls[frameIndex]
```

Merge tool

Text Compare ▼

Commit a la rama - 829d411

```
1  
2 '''  
3 Created on 7/1/2015  
4  
5 @author: alfonzo  
6 '''  
7  
8 class Game():  
9     def __init__(self):  
10         self._rolls = [0] * 21  
11         self._currentRoll = 0  
12  
13     def roll(self, pins):  
14         self._rolls[self._currentRoll] = pins  
15         self._currentRoll += 1  
16  
17     def _isSpare(self, frameIndex):  
18         return self._rolls[frameIndex] + self._rolls[frameIndex]  
19  
20     def _isStrike(self, frameIndex):
```

Mi segundo commit - ec9b798

```
1  
2 '''  
3 Created on 7/1/2015  
4  
5 @author: alfonzo  
6 '''  
7  
8 class Game():  
9     def __init__(self):  
10         self._rolls = [0] * 21  
11         self._currentRoll = 0  
12  
13     def roll(self, pins):  
14         self._rolls[self._currentRoll] = pins  
15         self._currentRoll += 1  
16  
17     def _isSpare(self, frameIndex):  
18         return self._rolls[frameIndex] + self._rolls[frameIn  
19  
20     def _isStrike(self, frameIndex):
```

CONFLICTO RESUELTO

Merge tool



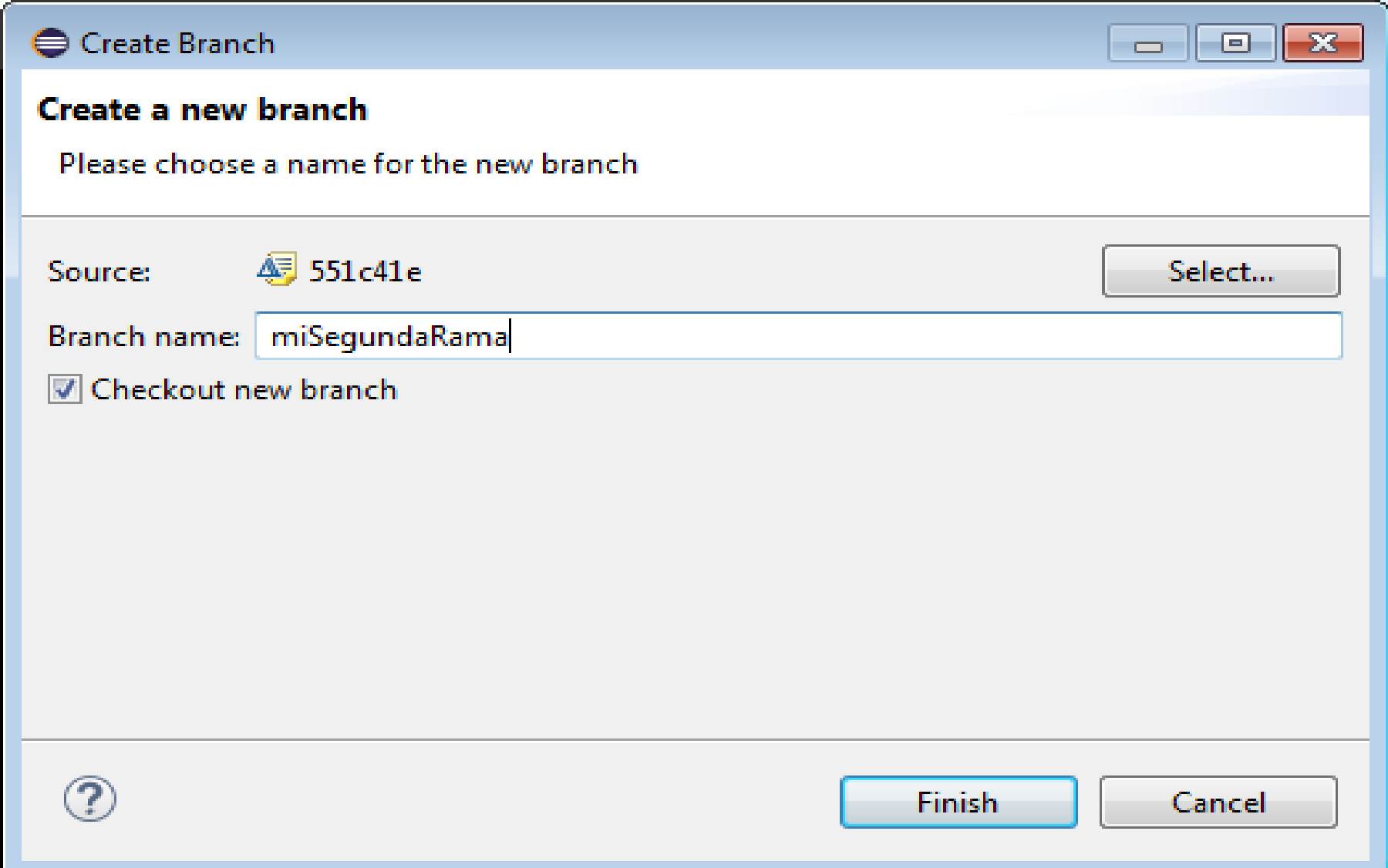
The screenshot shows the PyDev Package Explorer on the left and the testGame.py file editor on the right. The Package Explorer shows a project named 'Bowling' with a sub-project 'RepositorioBowling miPrime'. Inside 'RepositorioBowling miPrime', there are two files: 'classGame.py' and 'testGame.py'. The 'testGame.py' file is selected. The code in 'testGame.py' is as follows:

```

1  ...
2  Created on 7/1/2015
3
4  @author: alfonzo
5  ...
6
7  import unittest
8
9
10 from classGame import Game
11 import unittest
12
13 class BowlingGameTest(unittest.TestCase):
14     def setUp(self):
15         self.g = Game()
16
17     def testminScoringMinRollsGame(self):
18         self.roll_many(9, 10)
19         self.assertEqual(240, self.g.score())
20
21
22     def roll_many(self, n, pins):
23         for i in range(n):
24             self.g.roll(pins)
25
26
27
28 if __name__ == "__main__":
29     #import sys;sys.argv = ['', 'Test.testName']
30     unittest.main()

```


Creamos segunda rama



Create Branch

Create a new branch

Please choose a name for the new branch

Source:  551c41e Select...

Branch name:

☒ Checkout new branch

? Finish Cancel

Checkout – Commit

PyDev - Bowling/classGame.py

File Edit Source Refactoring

PyDev Package Explorer

- Bowling [RepositorioBowli]
 - classGame.py
 - testGame.py
 - python (c:\Python34\p

Commit Changes to Git Repository

Commit message

Commit a la segunda rama

Author: Alfonso Reinoza <jareinozacg@gmail.com>

Committer: Alfonso Reinoza <jareinozacg@gmail.com>

Files (1/1)

type filter text

Status	Path
<input checked="" type="checkbox"/>	Bowling/classGame.py

Open [Git Staging](#) view

Commit and Push Commit Cancel

Merge

Problems
@ Javadoc
Declaration
History

Project: Bowling [Bowling-1]

Id	Message	Author	Authored Date	Committer	Committed ...
4e6fe84	<div> <div>master</div> <div>origin/master</div> <div>HEAD</div> </div> Merge branch 'miSegunda	Alfonso Reinoza	2 hours ago	Alfonso Reinoza	2 hours ago
2bb88ef	Commit a la segunda rama	Alfonso Reinoza	2 hours ago	Alfonso Reinoza	2 hours ago
00fbaf1	Mi segundo commit	Alfonso Reinoza	2 hours ago	Alfonso Reinoza	2 hours ago
551c41e	Mi primer commit	Alfonso Reinoza	2 hours ago	Alfonso Reinoza	2 hours ago

Checkout al Master – todo ok

The screenshot shows the PyDev Package Explorer interface. The 'Git Repositories' tab is active, displaying the structure of the 'Bowling' repository. The 'Local' branch list shows 'master' (00fbaf1) and 'miPrimeraRama' (72adc03). A context menu is open over the 'miSegundaRama' branch, with the 'Checkout' option selected. The menu also includes options like 'Push Branch...', 'Create Branch...', 'Rename Branch...', 'Configure Branch...', 'Delete Branch', 'Merge', 'Rebase on', 'Synchronize with Workspace', 'Reset...', 'Show In', and 'Paste Repository Path or URI'.

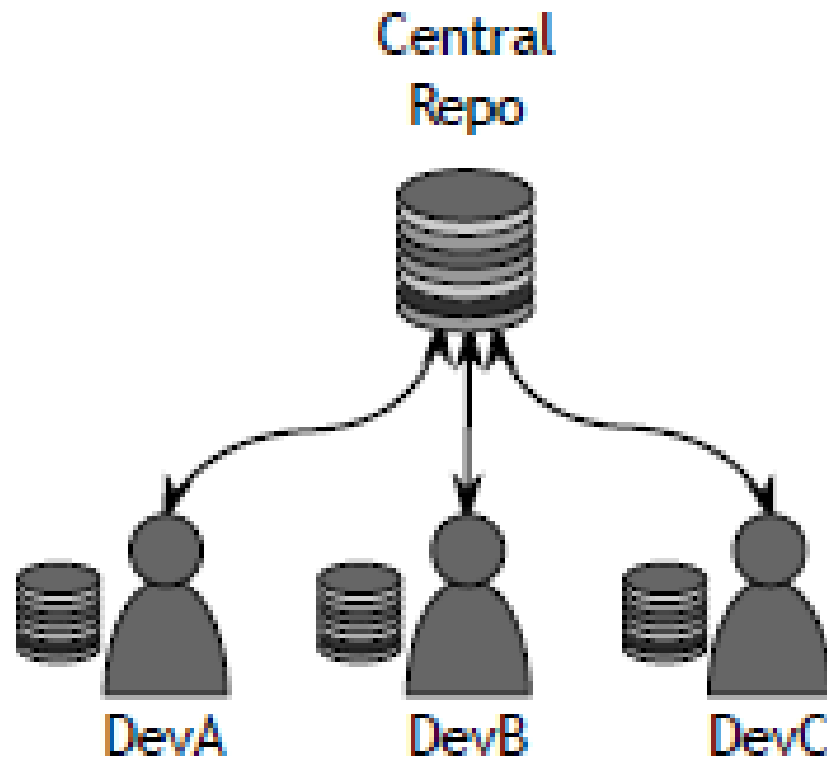
PyDev Package Explorer

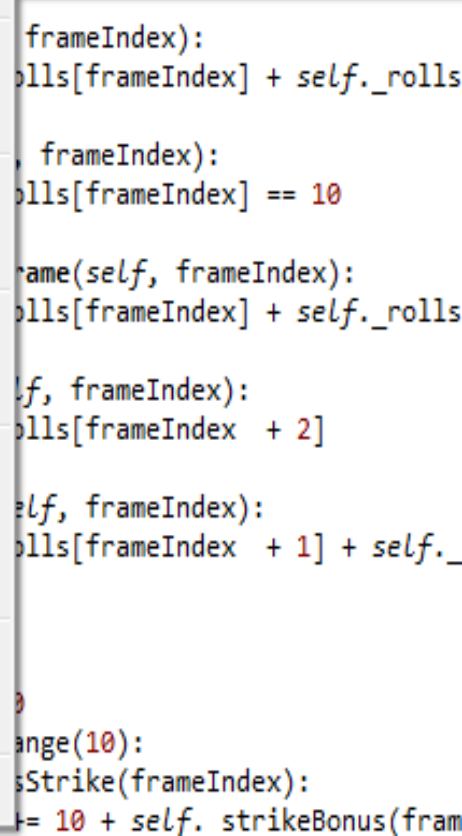
- Bowling [Repository]**
 - classGame.py
 - testGame.py
 - python (c:\Pytho
- RepositoryBowling [master] - C:\RepositorioBowling\.git**
 - Branches**
 - Local**
 - master 00fbaf1 Mi segundo commit
 - miPrimeraRama 72adc03 Merge branch 'master' into miPrimeraRam
 - miSegundaRama 2118866 C...
 - Remote Track**
 - Tags**
 - References**
 - Remotes**
 - Working Director**
 - .git
 - Bowling

Context Menu:


- Checkout
- Push Branch...
- Create Branch...
- Rename Branch... F2
- Configure Branch...
- Delete Branch Delete
- Merge
- Rebase on
- Synchronize with Workspace
- Reset...
- Show In Alt+ Shift+ W ▶
- Paste Repository Path or URI Ctrl+ V




Vámonos remoto






Push


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



Push

Push Ref Specifications

Select refs to push.



Add create/update specification

Source ref:

Destination ref:

*

*

+ Add Spec

Add delete ref specification

Remote ref to delete:

*

x Add Spec





Add predefined specification

Add Configured Push Specs


Add All Branches Spec

Add All Tags Spec

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
+ Update	refs/heads/master	HEAD	<input checked="" type="checkbox"/>	
+ Update	refs/heads/master	refs/heads/master	<input checked="" type="checkbox"/>	
+ Update	refs/heads/miPrimeraR...	refs/heads/miPrimeraR...	<input checked="" type="checkbox"/>	
+ Update	refs/heads/miSegunda...	refs/heads/miSegunda...	<input checked="" type="checkbox"/>	

Force Update All Specs

 Remove All Specs



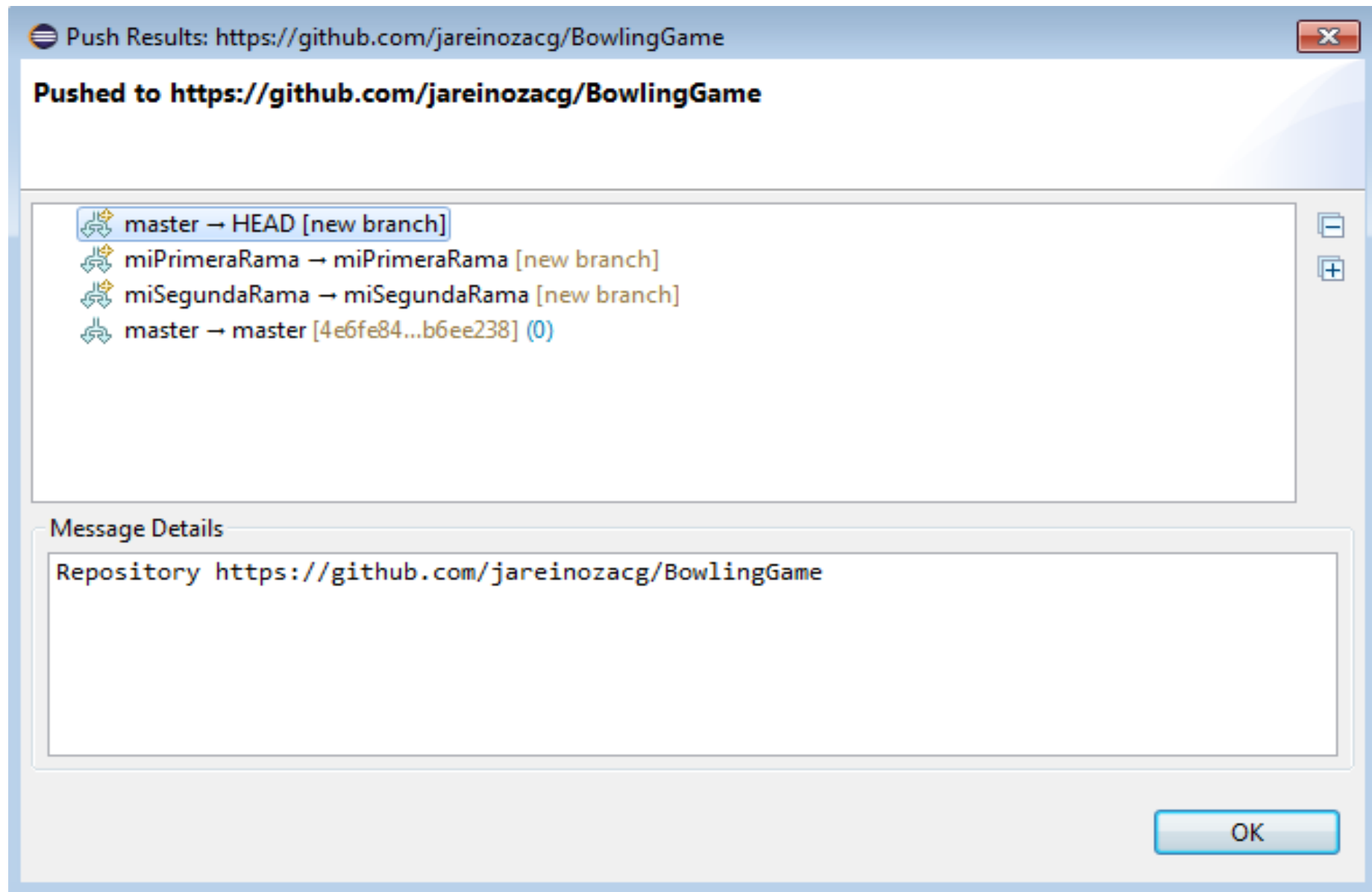
< Back

Next >

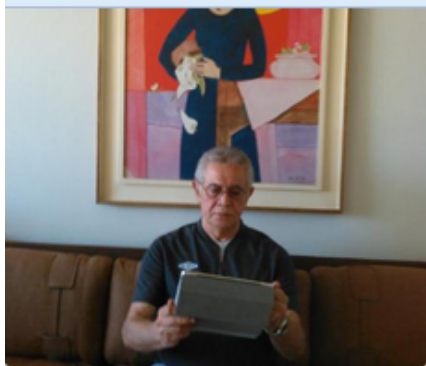
Finish

Cancel

Push



En el repositorio Github



Alfonso Reinoza

jareinozacg



Universidad Simón Bolívar



Caracas



jareinozacg@gmail.com



Joined on 1 May 2013

Popular repositories



BowlingGame

0 ★



CI3715-Enero---Abril-2015

0 ★



Marzullo

Marzullo's Algorithm

0 ★



SAGE

Prueba Fast Forward

0 ★




SAGE-Phoenix-Group


SAGE Phoenix Group

0 ★

En el repositorio Github



[Explore](#)
[Gist](#)
[Blog](#)
[Help](#)


jarei


jareinozacg / **BowlingGame**


Unwatch ▾
1

Description

Website

or [Cancel](#)



4 commits



6 branches



0 releases


1 contributor

Your recently pushed branches:


miPrimeraRama (1 minute ago)


Compare & pull request




branch: master ▾

BowlingGame / +

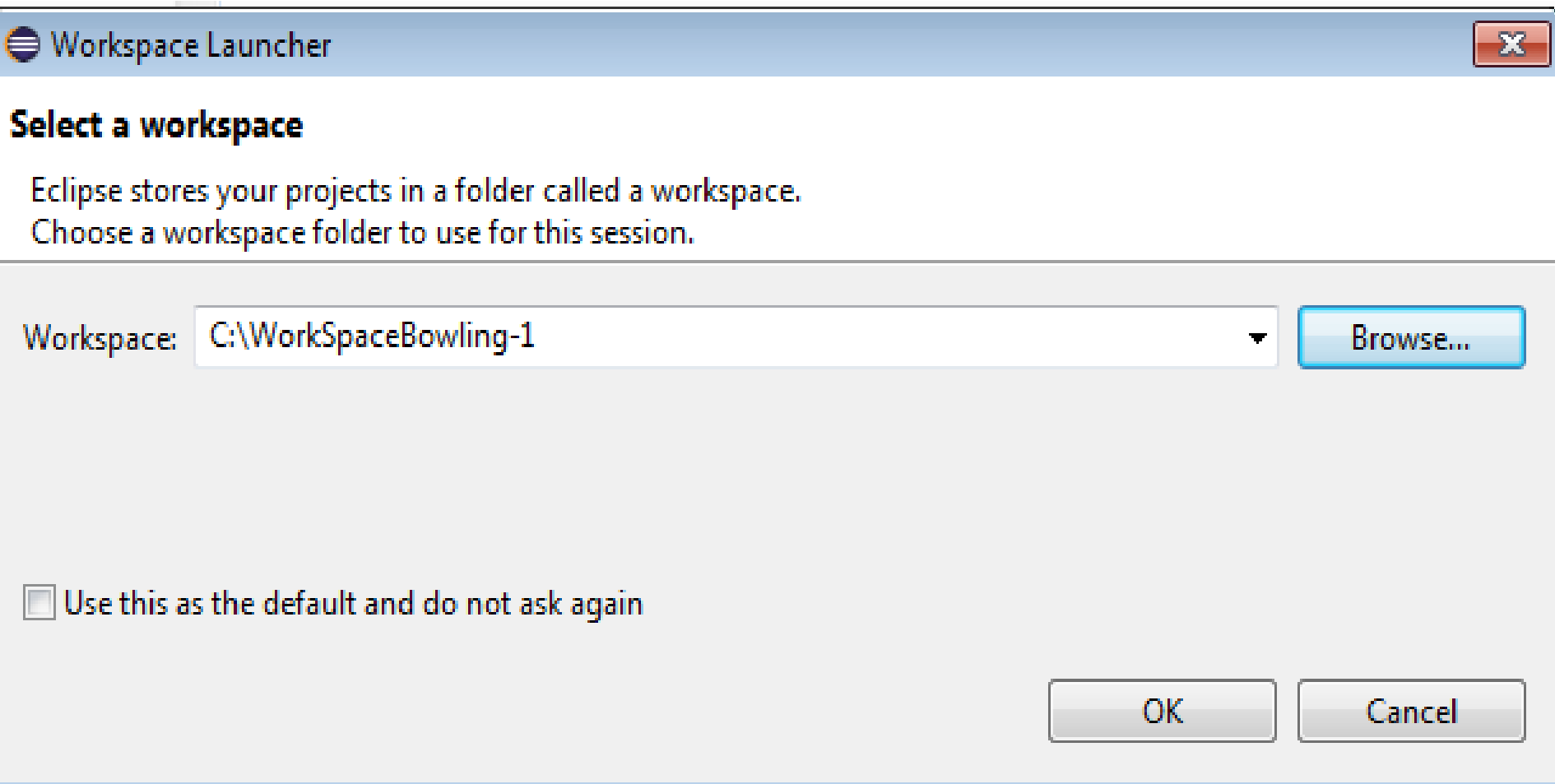


Merge branch 'miSegundaRama'
...

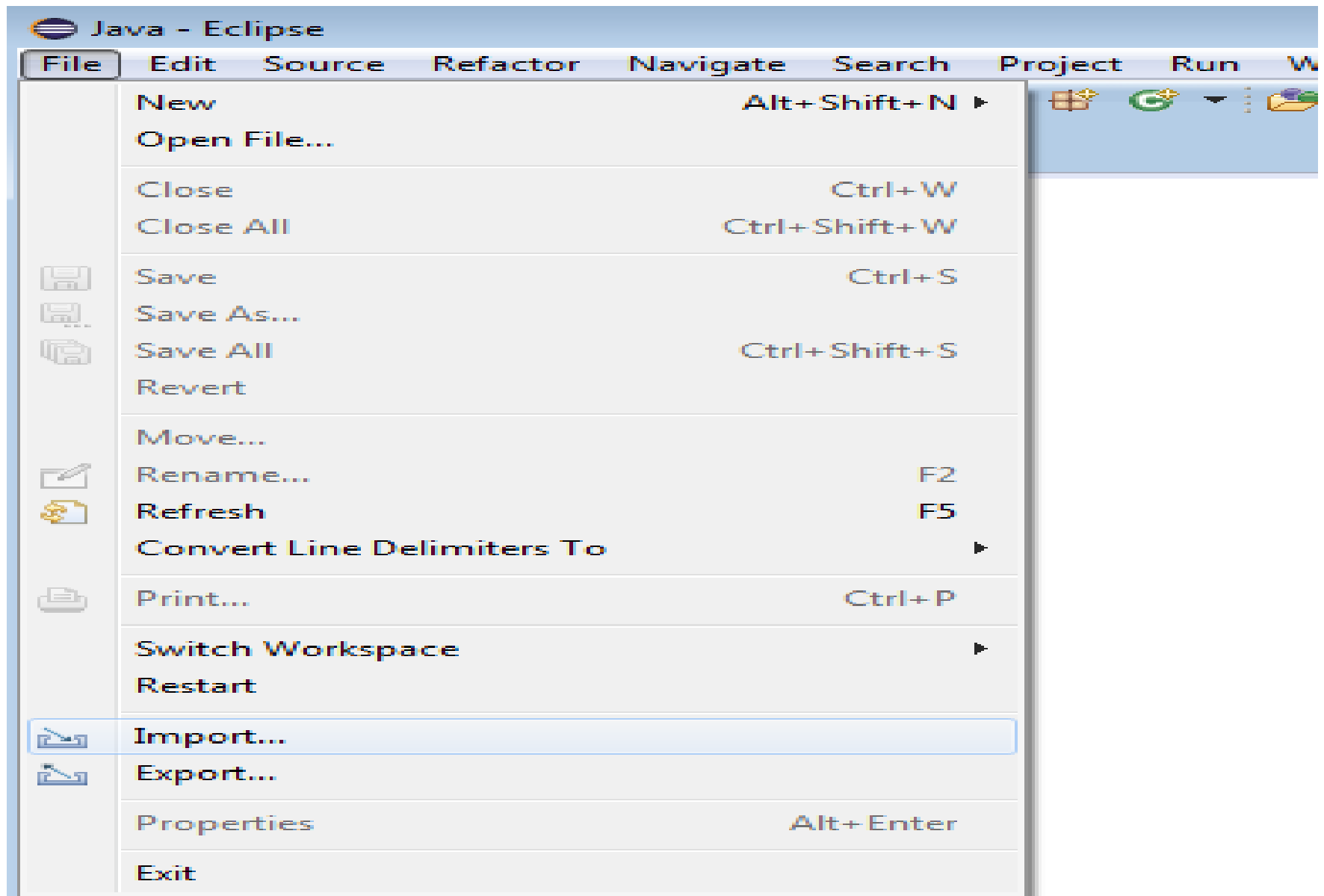

jareinozacg authored 10 minutes ago
latest commit 4e6fe84214


Bowling
Merge branch 'miSegundaRama'
10 minutes ago

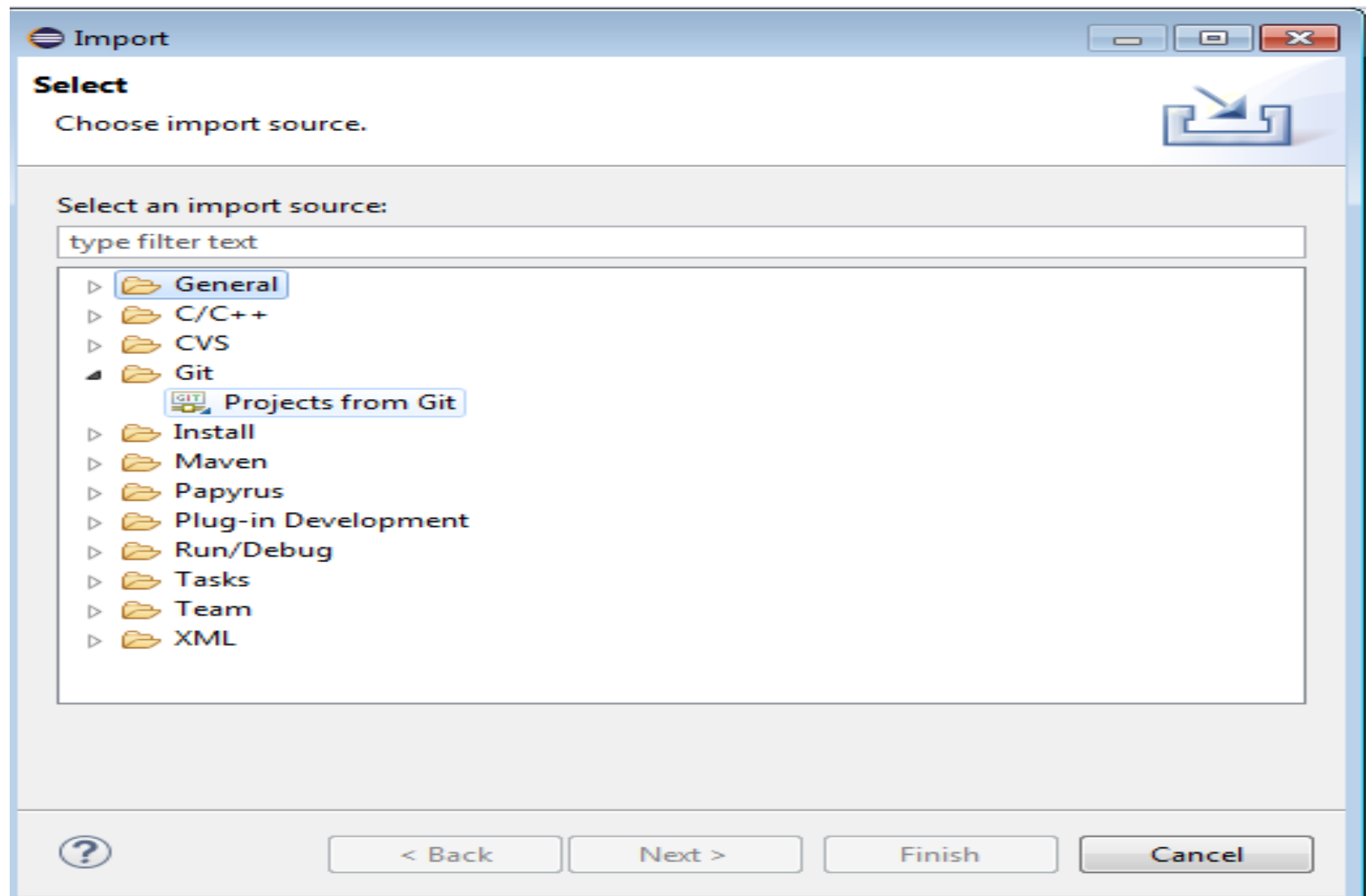
Vamos a clonar



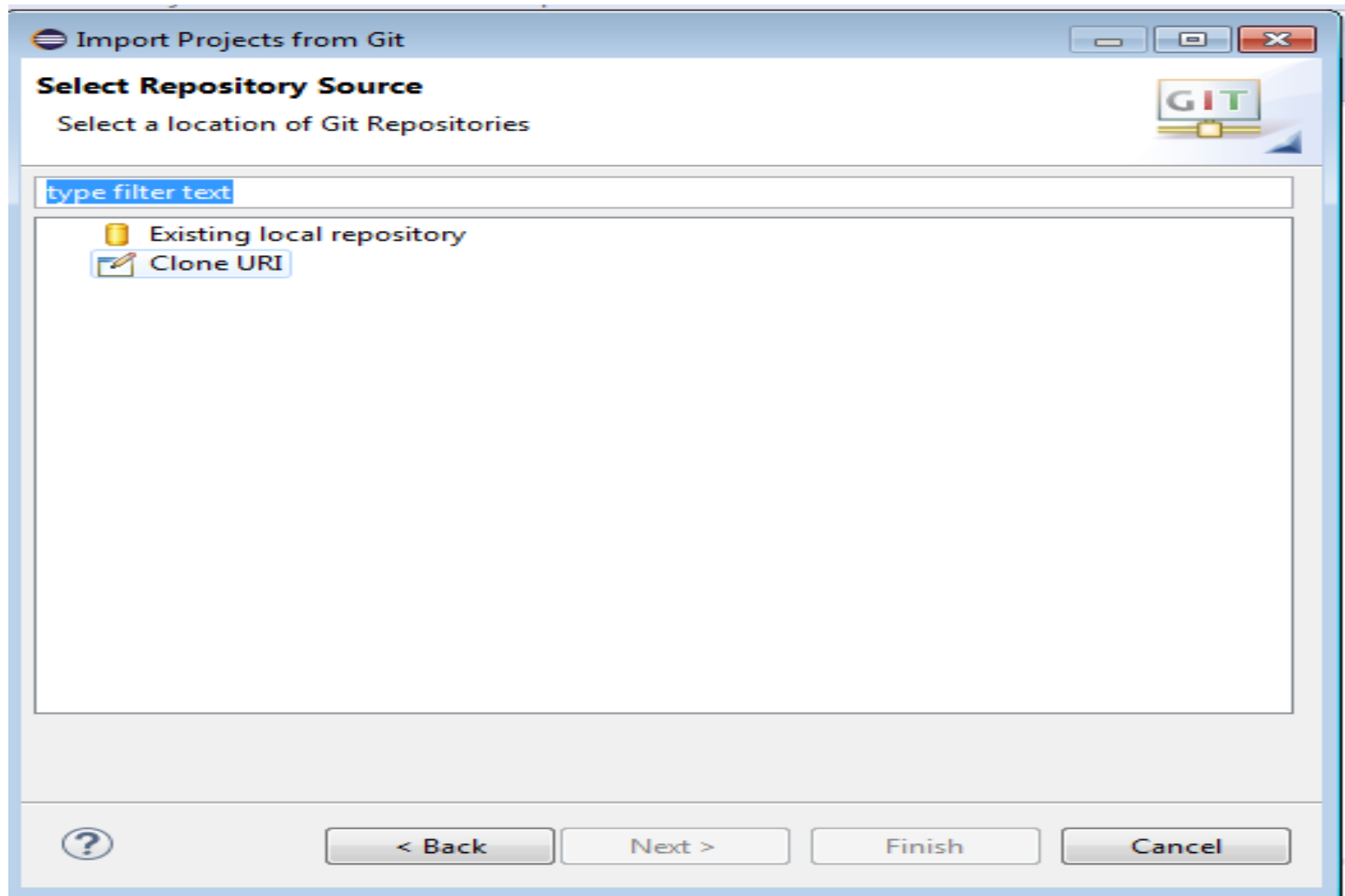
Importar




Traer de Git




Traer de Git remoto



Traer de Git remoto

 Import Projects from Git



Source Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:


Port:

Authentication

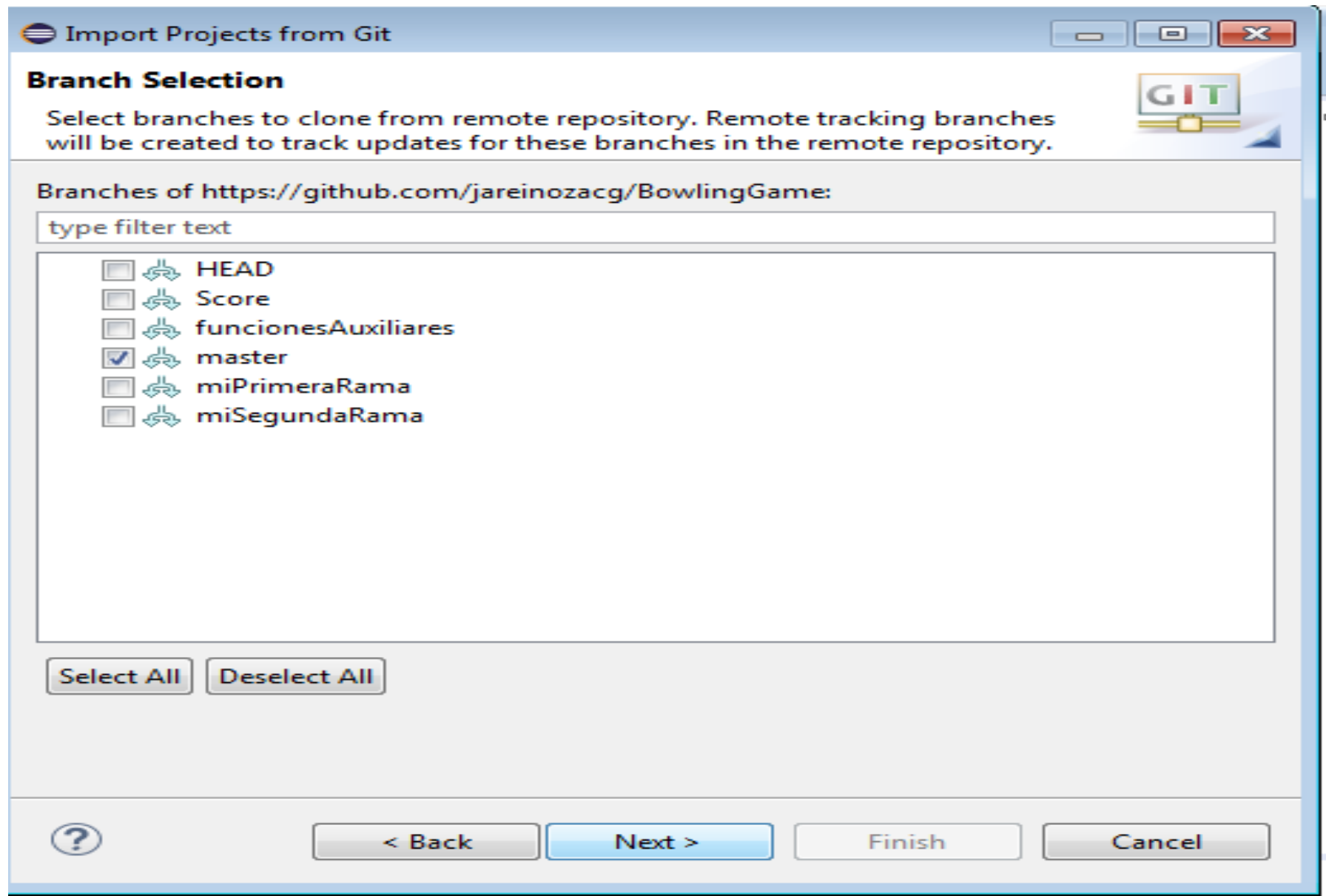
User:

Password:

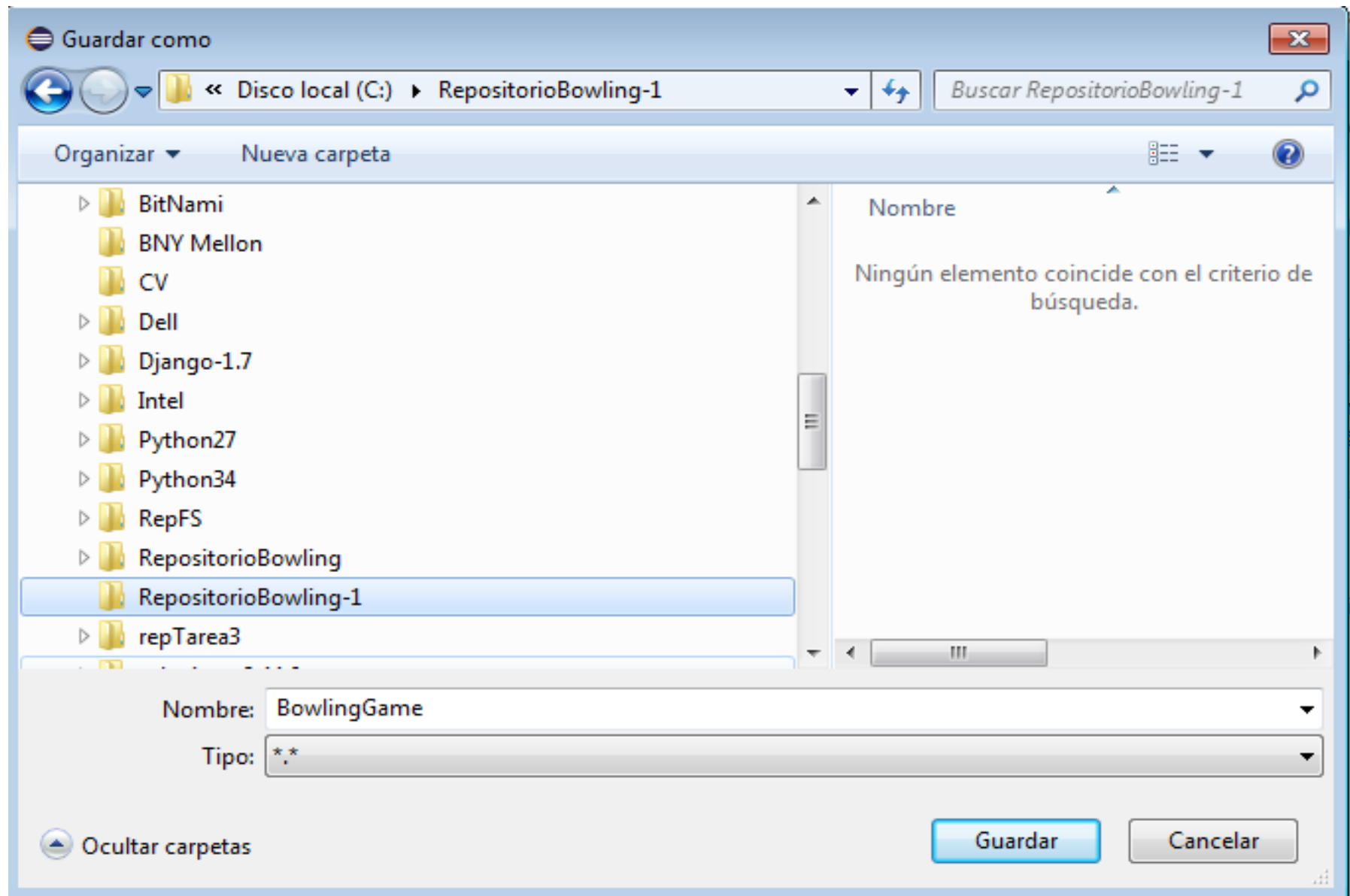
☐ Store in Secure Store



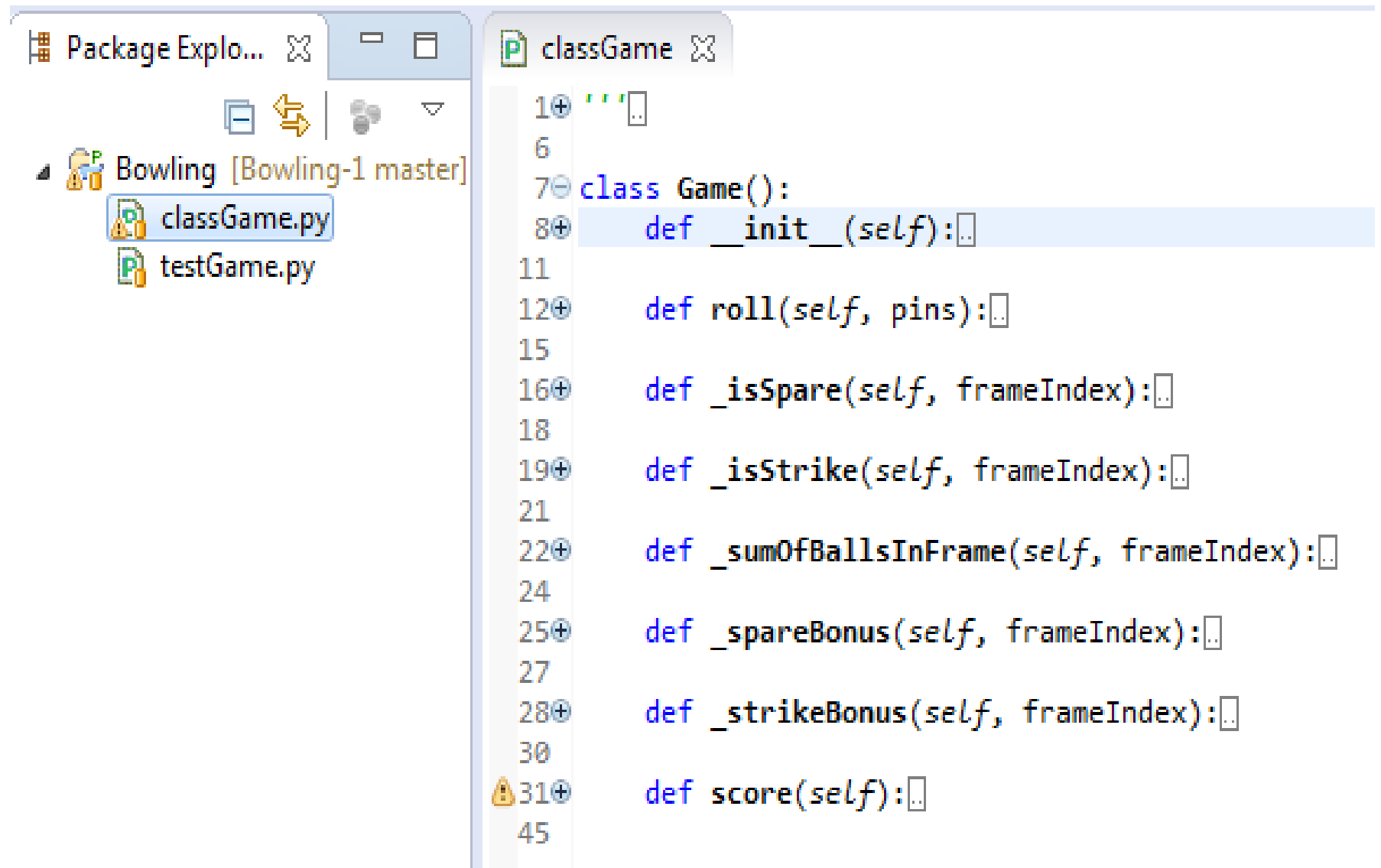
Traer de Git remoto



Indicamos donde lo guardamos



Clonado



The screenshot displays an IDE interface. On the left, the 'Package Explorer' shows a project named 'Bowling [Bowling-1 master]' containing two Python files: 'classGame.py' and 'testGame.py'. The 'classGame.py' file is selected and its contents are shown in the main editor area. The code defines a 'Game' class with several methods for simulating a bowling game.

```
classGame.py
1  """
6
7  class Game():
8      def __init__(self):
11
12      def roll(self, pins):
15
16      def _isSpare(self, frameIndex):
18
19      def _isStrike(self, frameIndex):
21
22      def _sumOfBallsInFrame(self, frameIndex):
24
25      def _spareBonus(self, frameIndex):
27
28      def _strikeBonus(self, frameIndex):
30
31      def score(self):
45
```

Agenda

- Verificación:
 - PyUnit
- Gestión de Configuraciones y Versiones de software:
 - Introducción a *Git* básico
- **Tarea 2**

**Gracias por su
atención**