# Wollok: en el aula y más allá

Javier Fernandes<sup>1,2</sup> Nicolás Passerini<sup>1,2,4</sup>
Pablo Tesone<sup>3,1,2,4</sup> Débora Fortini<sup>1,4</sup>
Nahuel Palumbo<sup>4</sup> Juan Contardo<sup>4</sup> Carlos Raffellini<sup>4</sup>

<sup>1</sup>Universidad Nacional de Quilmes
 <sup>2</sup>Universidad Nacional de San Martin
 <sup>3</sup>Universidad Nacional del Oeste
 <sup>4</sup>Universidad Tecnológica Nacional - F.R. Buenos Aires.

Workshop de Ingeniería en Sistemas y Tecnologías de la Información 19/09/2015

## Agenda

- Introduccion
- 2 Desarrollo de Wollok
- S Features Avanzados
- Wollok Game
- 6 Experiencia en el Aula
- 6 Próximos Pasos

# Introducción ¿Por qué es difícil aprender OOP?

- Enfoque en un lenguaje particular
- Demasiados conceptos

```
package examples;
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

- Entornos de desarrollo limitados o inadecuados
- Aprender a programar exige aprender a abstraer

# Introducción ¿Por qué es difícil aprender OOP?

- Enfoque en un lenguaje particular
- Demasiados conceptos

```
package examples;
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

- Entornos de desarrollo limitados o inadecuados
- Aprender a programar exige aprender a abstraer

Desarrollo de Wollok Features Avanzados Experiencia en el Aula Próximos Pasos

### Introducción Antecedentes

#### **∆**Ozono

http://ozono.ugbar-project.org/

- Basado en Smalltalk
- Recorrido incremental

Metamodelo simplificado: sin clases ni herencia Foco en objeto - mensaje - referencia - polimorfismo

 Herramientas de visualización de código Diagramas de objetos / clases

### **♦**Gobstones

- Elimina la necesidad de entrada-salida
- Separación entre elementos con efecto y elementos puros

### Introducción Antecedentes

#### **₩**Ozono

http://ozono.uqbar-project.org/

- Basado en Smalltalk
- Recorrido incremental

Metamodelo simplificado: sin clases ni herencia Foco en objeto - mensaje - referencia - polimorfismo

Herramientas de visualización de código
 Diagramas de objetos / clases

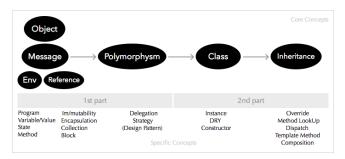
### **<b> ♥**Gobstones

http://www.gobstones.org/

- Cuidadosa selección de los elementos sintácticos
- Elimina la necesidad de entrada-salida
- Separación entre elementos con efecto y elementos puros

# Introducción ¿Qué es Wollok?

- Lenguaje + muchas herramientas
- Optimizados para la enseñanza
- Cercanos a las herramientas profesionales mainstream



# Introducción

¿Qué es Wollok? - Un entorno optimizado para la enseñanza

- Sintaxis cuidada
  - selección de keywords
  - return obligatorio
  - énfasis en objetos y mensajes (aunque no todo es objeto mensaje)
- Combina object-based con class-based programming
- APIs minimalistas (ej: colecciones)
- Import system

## Introducción

Cuidado: No perderle pisada la evolución de las herramientas industriales

- Ambiente de objetos basado en archivos
- Framework de testing integrado
- Sintaxis concisa y "moderna"
   Ej: lambdas, literales para colecciones, excepciones, constructores
- Sistema de tipos pluggeable (en proceso)
- Mixins (planificado)

## Desarrollo de Wollok

- OpenSource: LGPLv3
- Stack: ■Eclipse XText + Xtend Lang
- SCM:
  - Código: OGitHub (uqbar-project/wollok)
  - **Build**: Maven + Tycho
  - Continuous Integration: Travis
  - Continuous Deployment
  - **Coverage**: coveralls + jacoco
- Testing & TDD

### Desarrollo de Wollok

Continuous Integration & Deployment

- GitFlow
  - Feature Branches
  - Pull-Requests
  - $dev \rightarrow master \leftarrow hotfixes$
- Integration:
  - Travis
  - compile, test, coverage, deploy
- Deployment:
  - Productos (IDE): multiples pataformas
  - Update Sites
  - WDK
  - 2 Ambientes: Stable & Dev

### Desarrollo de Wollok Testing & TDD

- 87% Cobertura
- Runtime
  - Testean ejecución
  - Interprete
  - JUnit + iDSL
- Estáticos
  - **Chequeos**: XPect
  - Type System: JUnit + iDSL
  - Autocomplete: XPect
  - Formateo: JUnit + iDSL
- Pendientes
  - Quick-Fixes
  - Refactors

# Testing & TDD Runtime

### Testeo del Intérprete

```
{\tt class\ PostFixOperationTestCase\ extends\ AbstractWollokInterpreterTestCase\ \{}
```

```
@Test
def void testPlusPlus() {'''
    program p {
       var n = 1
       n++
       assert.that(n == 2)
    }'''.interpretPropagatingErrors
}
```

# Testing & TDD Chequeos Estáticos

### Testeo de Chequeos Estáticos

```
/* XPECT_SETUP org.uqbar.project.wollok.tests.xpect.WollokXPectTest END_SETUP */

class Golondrina {
    var energia = 100

    method energia() {
        // XPECT errors | --> "Cannot assign a variable to itself. It does not have any effect" at "energia"
    energia = energia }
```

## Features Avanzados

- Debugger
- Diagramas: Clases y Objetos (en desarrollo)
- Tests
- ContentAssist y QuickFixes
- Sublime Plugins
- I18N

# Features Avanzados Debugger

### Debugger

- UI integrada a Eclipse Debug
- Breakpoints: agregar, remover, deshabilitar, etc
- Step, into, out
- Inspeccionar variables
- Diagrama de Objetos

## Features Avanzados Tests

#### Tests

### Features Avanzados Soporte para Sublime

### Soporte para Sublime

- WDK
  - No IDE
  - $\bullet \sim 70 \mathrm{MB} \; (\mathrm{vs} \sim 140)$
  - Headless: wchecker, winterpreter, wtest
- Syntax highlight
- Templates
- Linter

# Sublime Support Syntax Highlight

```
3 polymorphism.wpgm - wollok-tests
                                                                                        UNREGISTERED
OPEN FILES
                             1.1_booleans.wpgm ×
                                                  3 polymorphism.wpgm •
                                                                        2 object literals.wpgm ×
× 1.1 booleans.wpgm
                            program polymorphism {

    3_polymorphism.wpgm

× 2 object literals.wpgm
                                  // golondrina pepona con energia
FOLDERS
                         4
                                  val pepona = object {
var energia = 100
 ▶ 🗀 .settings
                                       method comer(comida) {
 ▶ ( bin
                                            energia = energia + comida.energia()
 ▶ Ĉ¬ META-INF
                         8
 ▼ 🗁 src
                         9
                                       method energia() { return energia }
   ▶ 🗀 bugs
                       10
   ▶ 🗀 org
                       11
   ▶ ( wollok
                                  /* el alpiste tambien tiene energia */
     (9 1.1 booleans.wpgm
                                  val alpiste = object {
     1.2_console.wpgm
                       14
                                       method energia() {
     (A) 10 collection literals
                                            return 5
     1_variables_and_num
                       16
     2_object_literals.wpg
                       18
     3_polymorphism.wp
                       19
                                  pepona.comer(alpiste)
     4 objectliteral as par
                       20
     In 5 var ref vs method
                                  tester.assert(pepona.energia() == 105)
     6_custom_toString.w
                       22
     7 operator overloadi
                       23
     9_closures.wpgm
     ConsoleObject.wpgm
                                                                              Tab Size: 4
```

# Sublime Support Linter

```
3 polymorphism.wpam - wollok-tests
OPEN FILES
                                                                                    2_object_literals.wpgm ×
                                  1.1 booleans.wpgm ×
                                                          3 polymorphism.wpgm ×
 × 1.1_booleans.wpgm
                                  progr
 × 3 polymorphism.wpgm
                                                                                                          100.00
 × 2 object literals,wpgm
                                         12 Couldn't resolve reference to WReferenciable 'algo'
                             3
FOLDERS
                             4
                                          pepe.comer(algo)→
 5
                                         12 Couldn't resolve reference to WReferenciable 'pepe'
  ▶ Ĉ¬ .settings
                             6
                                          pepe.comer(algo)→
  ▶ M bin
                             7
                                                                                                   a()
                                         23 Couldn't resolve reference to WReferenciable 'tester'
  ▶ ( ) META-INF
                             8
                                          tester.assert(pepona.energia() == 105)→
  ▼ 🗁 src
                             9
    ▶ 🗀 bugs
                           10
                           11
    ▶ Mora
                          •12
    ▶ \( \backsquare
\) wollok
                                       pepe.comer(algo)
      1.1 booleans.wpgm
                            14
                                       /* el alpiste tambien tiene energia */
      A 1.2 console.wpgm
                           15
                                        val alpiste = object {
      10_collection_literals
                                             method energia() {
      1 variables and num
                           16
                                                   return 5
      2 object literals.wpg
                           18
      3_polymorphism.wp
      A 4 objectliteral as par
      5_var_ref_vs_method
                                       pepona.comer(alpiste)
      6 custom_toString.w
      7 operator overloadi
                          •23
                                       tester.assert(pepona.energia() == 105)
      3 9_closures.wpgm
                           24
      ConsoleObject.wpgm
1-2 of 3 errors: Couldn't resolve reference to WReferenciable 'pepe'; Couldn't resolve reference to WReferenciable 'algo', 2 lines, 1 characters sele
```

## Wollok Game

- Herramienta complementaria al testeo unitario y consola interactiva.
- Mejorar la comprensión de conceptos.
- Visualización de comportamiento
- Motivación en el aprendizaje fomentando la participación.

# Wollok Game FarmVille

#### FarmVille - Demo



# Wollok Game

#### Sokoban - Demo



# Wollok Game

#### **Futuro**

- + Tipos de Juegos
  - Survival
  - Por turnos
- + Tipos de Interacciones
- Features Gráficos
  - Animaciones
  - Fondos infinitos
  - Distintos vistas (lateral, isométrica, etc)

Los alumnos se apropian intuitivamente de las herramientas

- Integración class-based / object-based
- El REPL resulta más intuitivo que los workspaces de Smalltalk
- Mayor control sobre los tests unitarios
- Editores

Un recorrido incremental apoyado en herramientas adecuadas, permite aprovechar la intuición del estudiante fomentando su autonomía, creatividad y motivación

Los alumnos se apropian intuitivamente de las herramientas

- Integración class-based / object-based
- El REPL resulta más intuitivo que los workspaces de Smalltalk
- Mayor control sobre los tests unitarios
- Editores

Un recorrido incremental apoyado en herramientas adecuadas,

permite aprovechar la intuición del estudiante fomentando su autonomía, creatividad y motivación

Los alumnos se apropian intuitivamente de las herramientas

- Integración class-based / object-based
- El REPL resulta más intuitivo que los workspaces de Smalltalk
- Mayor control sobre los tests unitarios
- Editores

Un **recorrido incremental** apoyado en **herramientas** adecuadas, permite aprovechar la **intuición** del estudiante

fomentando su autonomía, creatividad y motivación

Los alumnos se apropian intuitivamente de las herramientas

- Integración class-based / object-based
- El REPL resulta más intuitivo que los workspaces de Smalltalk
- Mayor control sobre los tests unitarios
- Editores

Un **recorrido incremental** apoyado en **herramientas** adecuadas, permite aprovechar la **intuición** del estudiante fomentando su **autonomía**, **creatividad y motivación** 

## Próximos pasos

#### Próximos Pasos

- Varias discusiones sobre la mejor sintaxis
- Herencia basada en mixins
- Implementar wollok-game en el aula
- Plataforma p/interacción Alumno ↔ Docente

Y muchas actividades para sumar más gente al proyecto.

# Muchas gracias

# ¡Muchas Gracias!

