

# Módulo de Identificación de Pasos y Situaciones

## MIPS

Rubén Agudo Santos    Mikel Villamañe Gironés

30 de junio de 2014

# Fe de erratas

## Errores en la memoria

- ▶ La palabra “continuo” varias veces.
- ▶ La palabra “que”. Página 43, párrafo 1, línea 1.

# Antecedentes

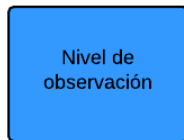
## ULISES

- ▶ Enseñar a alumnos habilidades
- ▶ Unir un sistema interactivo a uno educativo
- ▶ En definitiva, evaluar

# Funcionamiento actual

## Nivel de observación

- ▶ Captura de datos
- ▶ Se crean
  - ▶ Propiedades
  - ▶ Observaciones



# Funcionamiento actual

## Nivel de interpretación

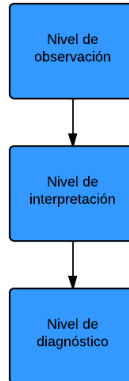
- ▶ Relaciones entre observaciones
  - ▶ Pasos
  - ▶ Situaciones



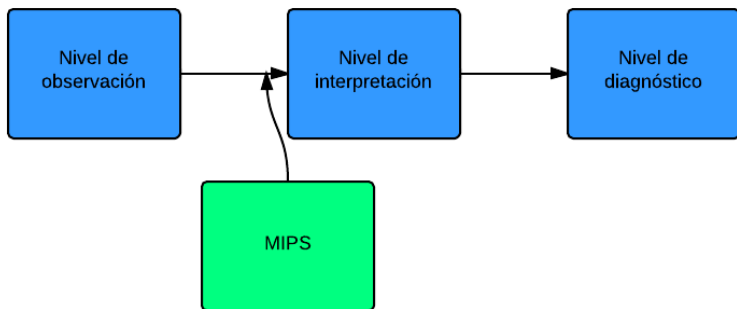
# Funcionamiento actual

## Nivel de diagnóstico

- ▶ Usando distintos métodos de diagnóstico
  - ▶ *Clustering*
  - ▶ Clasificación supervisada
  - ▶ ...



## Después del proyecto



# ¿Qué es MIPS?

- ▶ Herramienta de autor para el experto
- ▶ Permite
  - ▶ Selección de rangos
  - ▶ Ayuda la identificación de pasos y situaciones
  - ▶ Exportar esos rangos



## ¿Por qué es necesario?

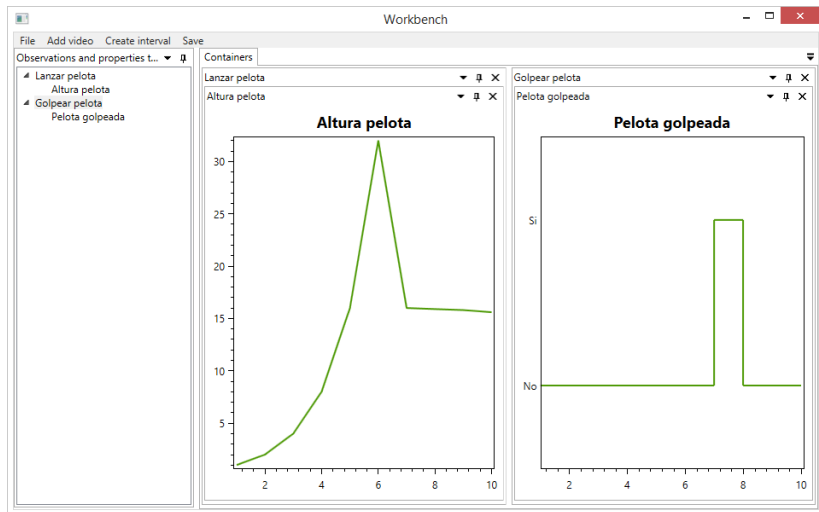
- ▶ Identificación manual de pasos y situaciones
- ▶ Prueba y error
- ▶ Relaciones erróneas
- ▶ Resultados subóptimos

# Ejemplo de observaciones y propiedades

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     EspObservation="ObservationModelExample.xml"
4     instantLength="1">
5     <observation name="Lanzar pelota">
6         <property name="Altura pelota" type="1">
7             <instant ins="1" value="1"/>
8             <instant ins="2" value="2"/>
9             <instant ins="3" value="4"/>
10            <instant ins="4" value="8"/>
11            <instant ins="5" value="16"/>
12            <instant ins="6" value="32"/>
13            <instant ins="7" value="16"/>
14            <instant ins="8" value="15.9"/>
15            <instant ins="9" value="15.8"/>
16            <instant ins="10" value="15.6"/>
17        </property>
18    </observation>
```

```
19 <observation name="Golpear pelota">
20     <property name="Pelota golpeada" type="0">
21         <instant ins="1" value="No"/>
22         <instant ins="2" value="No"/>
23         <instant ins="3" value="No"/>
24         <instant ins="4" value="No"/>
25         <instant ins="5" value="No"/>
26         <instant ins="6" value="No"/>
27         <instant ins="7" value="Si"/>
28         <instant ins="8" value="No"/>
29         <instant ins="9" value="No"/>
30         <instant ins="10" value="No"/>
31     </property>
32 </observation>
33 </data>
```

# Ejemplo de visualización en MIPS



# Desarrollo

## Qué se ha hecho

Una aplicación que permite:

- ▶ Cargar un XML con las observaciones y propiedades
- ▶ Visualizar datos discretos y continuos
- ▶ Cargar vídeos y visualizarlos
- ▶ Seleccionar rangos
- ▶ Exportar los rangos seleccionados en XML

# Desarrollo

## Cómo se ha hecho

Una aplicación que permite:

- ▶ Cargar un XML con las observaciones y propiedades
- ▶ Visualizar datos discretos y continuos
- ▶ Cargar vídeos y visualizarlos
- ▶ Seleccionar rangos
- ▶ Exportar los rangos seleccionados en XML

# Herramientas utilizadas

- ▶ Visual Studio 2013 Ultimate
- ▶ OxyPlot
- ▶ AvalonDock
- ▶ GVim
- ▶ L<sup>A</sup>T<sub>E</sub>Xy TeXstudio

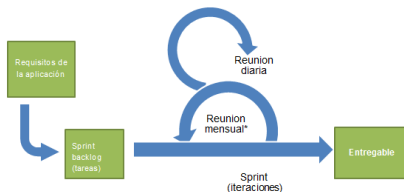
- ▶ Se han utilizado metodologías ágiles de desarrollo
  - ▶ Scrum
  - ▶ Kanban

# Gestión

## Scrum

### ¿Qué es?

- ▶ Método de desarrollo iterativo e incremental
- ▶ En cada ciclo de desarrollo (sprint) se genera un entregable.
- ▶ Lo importante es que el diferencial de valor incremente



<https://upload.wikimedia.org/wikipedia/commons/e/e5/Scrumm.PNG>

Autor: Maxie Ayala Licenciado bajo CC BY-SA 3.0



# Gestión

## Kanban

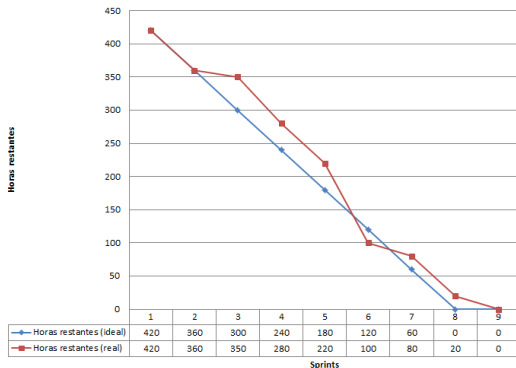
**¿Qué es?**

- ▶ blabla

# Conclusiones

## Sobre la gestión

- ▶ Ha habido retraso
- ▶ Para mitigarlo
  - ▶ Scrum
  - ▶ Kanban
  - ▶ Meter más horas



# Conclusiones

## Personales

- ▶ Síndrome del programador
- ▶ Difícil programar sin documentación
- ▶ Estar fuera de la zona de “confort”

# Líneas futuras

## Mejoras

Ordenadas de más importante a menos importante

1. Que el software sea más abstracto
2. Utilizar Desarrollo Dirigido por Pruebas (Test Driven Development)
3. Mejorar el procesamiento paralelo.
4. Eliminar Singleton por patrones Factory
5. Utilizar los *data bindings* de MVVM

**Refactorizar, refactorizar, refactorizar**