Módulo de Identificación de Pasos y Situaciones
MIPS

Rubén Agudo Santos Mikel Villamañe Gironés

2 de julio de 2014

Fe de erratas

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

Índice

1. Antecedentes
2. Ejemplo
3. Desarrollo
4. Gestión
5. Conclusiones
6. Líneas futuras

Antecedentes

ULISES

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

Antecedentes

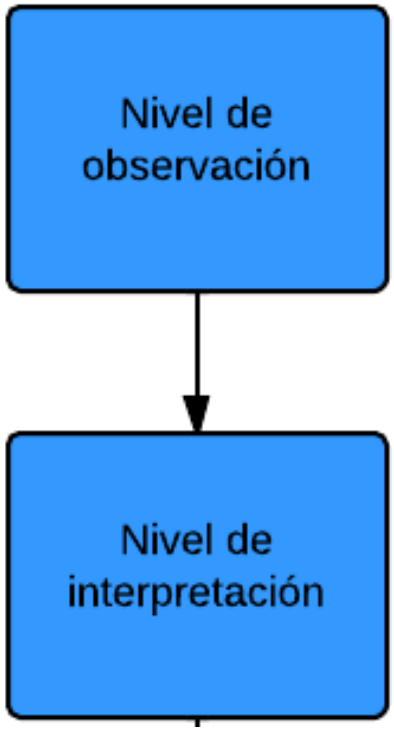
Funcionamiento actual: Nivel de observación

- Captura de datos
- Se crean
  - Propiedades
  - Observaciones

Antecedentes

Funcionamiento actual: Nivel de interpretación

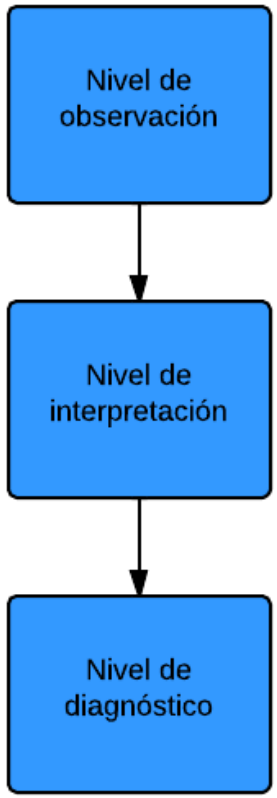
- Relaciones entre observaciones
  - Pasos
  - Situaciones



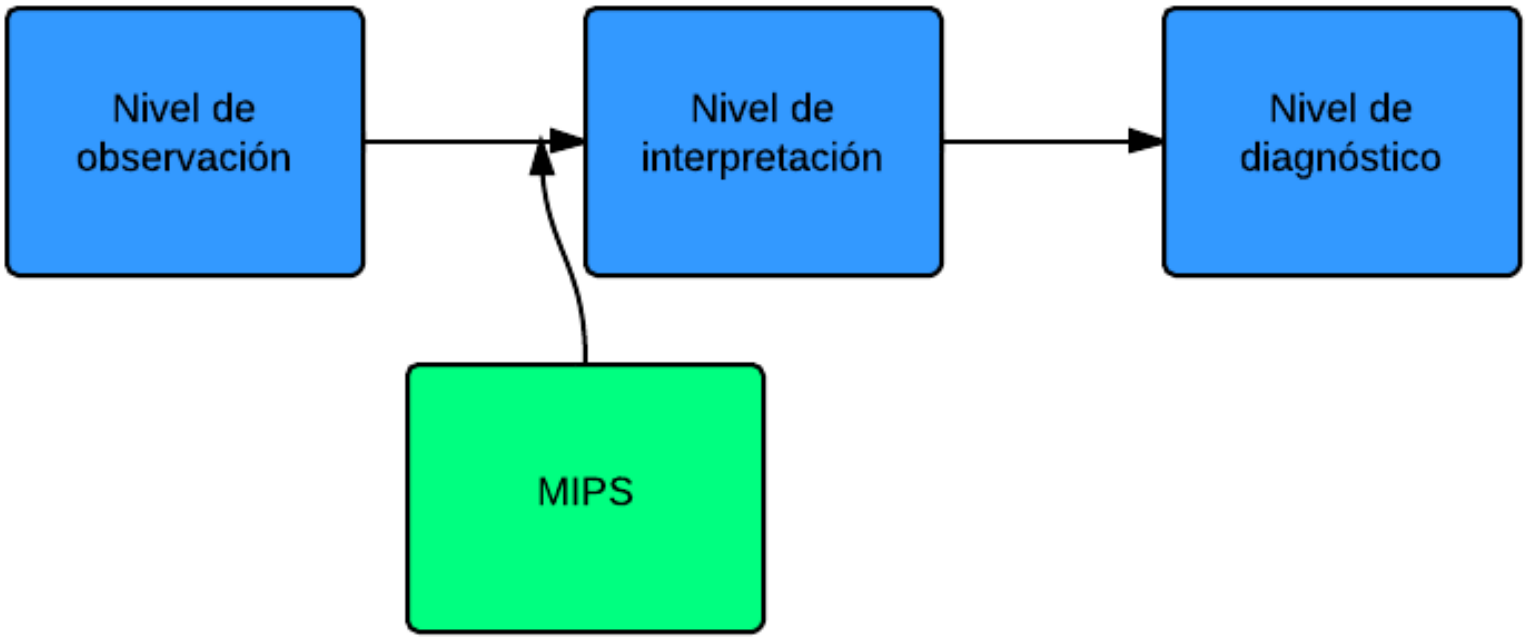
Antecedentes

Funcionamiento actual: Nivel de diágnostico

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



Después del proyecto



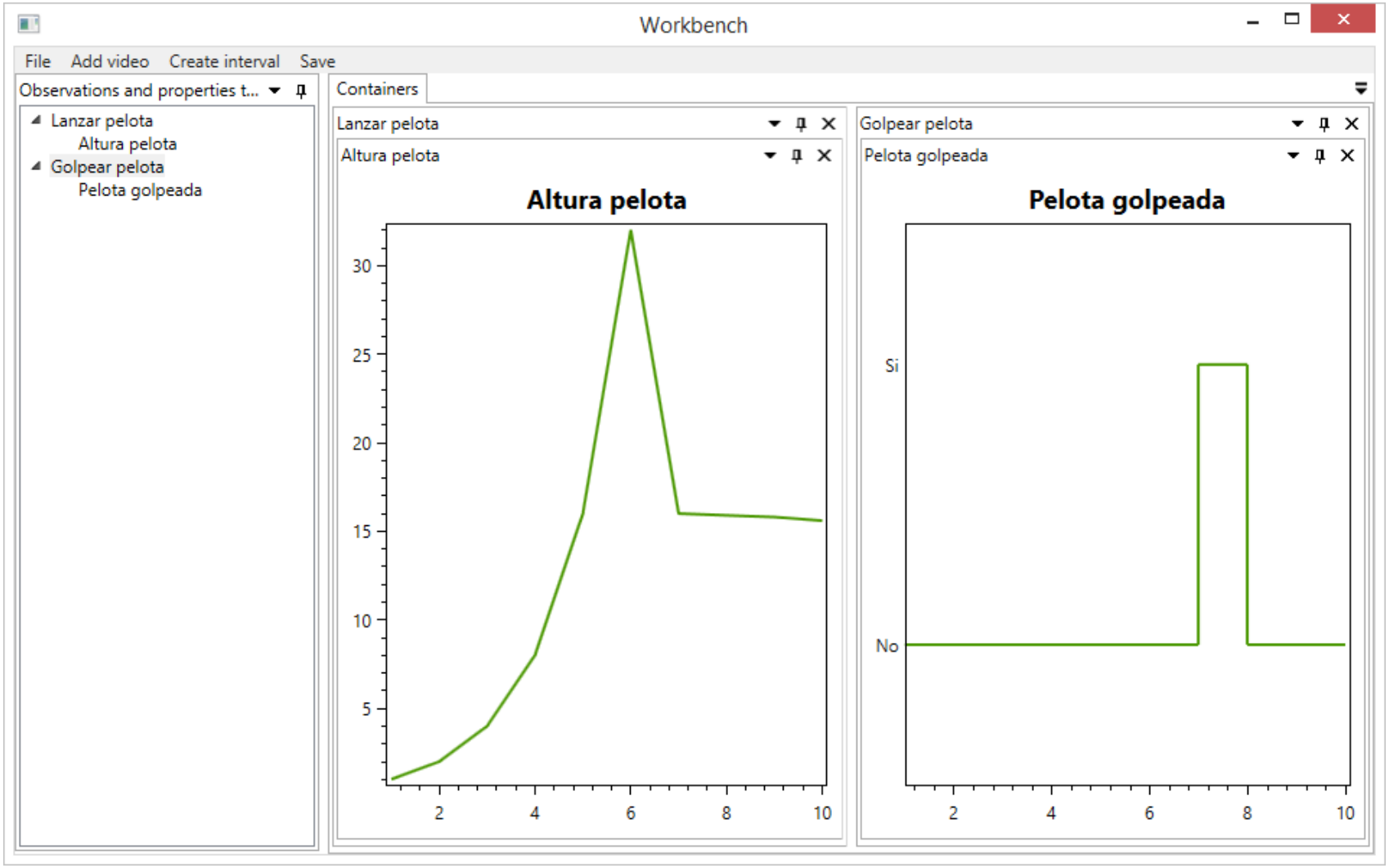
Ejemplo

Ejemplo de observaciones y propiedades

```
<?xml version="1.0" encoding="utf-8" ?>
<data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      EspObservation="ObservationModelExample.xml"
      instantLength="1">
    <observation name="Lanzar pelota">
        <property name="Altura pelota" type="1">
            <instant ins="1" value="1"/>
            <instant ins="2" value="2"/>
            <instant ins="3" value="4"/>
            <instant ins="4" value="8"/>
            <instant ins="5" value="16"/>
            <instant ins="6" value="32"/>
            <instant ins="7" value="16"/>
            <instant ins="8" value="15.9"/>
            <instant ins="9" value="15.8"/>
            <instant ins="10" value="15.6"/>
        </property>
    </observation>
    <observation name="Golpear pelota">
        <property name="Pelota golpeada" type="0">
            <instant ins="1" value="No"/>
            <instant ins="2" value="No"/>
            <instant ins="3" value="No"/>
            <instant ins="4" value="No"/>
            <instant ins="5" value="No"/>
            <instant ins="6" value="No"/>
            <instant ins="7" value="Si"/>
            <instant ins="8" value="No"/>
            <instant ins="9" value="No"/>
            <instant ins="10" value="No"/>
        </property>
    </observation>
</data>
```

Ejemplo

Ejemplo de visualización en MIPS



Diseño

SOLID y patrones

SOLID

- **S**ingle responsibility principle
- **O**pen-closed principle
- **L**iskov substitution principle
- **I**nterface segregation principle
- **D**ependency inversion principle

Patrones

- Model View View-Model
- Iterator
- Singleton

Desarrollo

Qué se ha hecho

Una aplicación que permite:

1. Cargar un XML con las observaciones y propiedades
2. Visualizar datos discretos y continuos
3. Cargar vídeos y visualizarlos
4. Seleccionar rangos
5. Exportar los rangos seleccionados en XML
6. Interfaz tipo IDE, como Eclipse o Visual Studio

Desarrollo

Cómo se ha hecho

Cargar un XML con las observaciones y propiedades

- Se ha utilizado LINQ to XML
- Permite realizar consultas similares a SQL a los XML

Desarrollo

Cómo se ha hecho

Ejemplo LINQ To XML

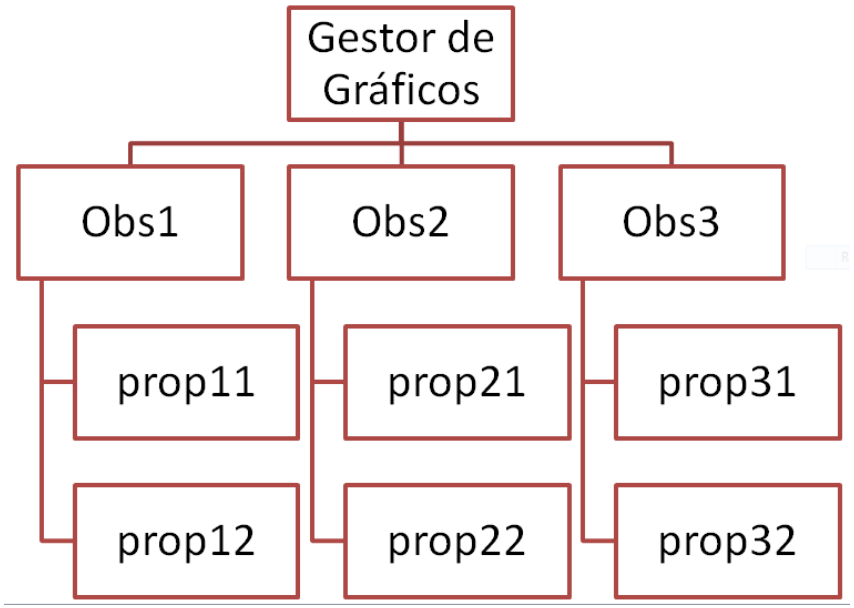
```
1          private IEnumerable<XElement> getData(
               string observacion)
2          {
3              return from prop in xml.
                   Descendants("property")
4                  where (string)prop.Parent.
                       Attribute("name") ==
5                  observacion
6                  select prop;
7          }
```

Desarrollo

Cómo se ha hecho

Seleccionar rangos

1. Cada propiedad notifica a su padre cuando cambia
2. Esa observación notifica al Gestor de gráficos
3. El gestor cicla a través de todas las observaciones
4. Cada observación actualiza el valor de sus propiedades



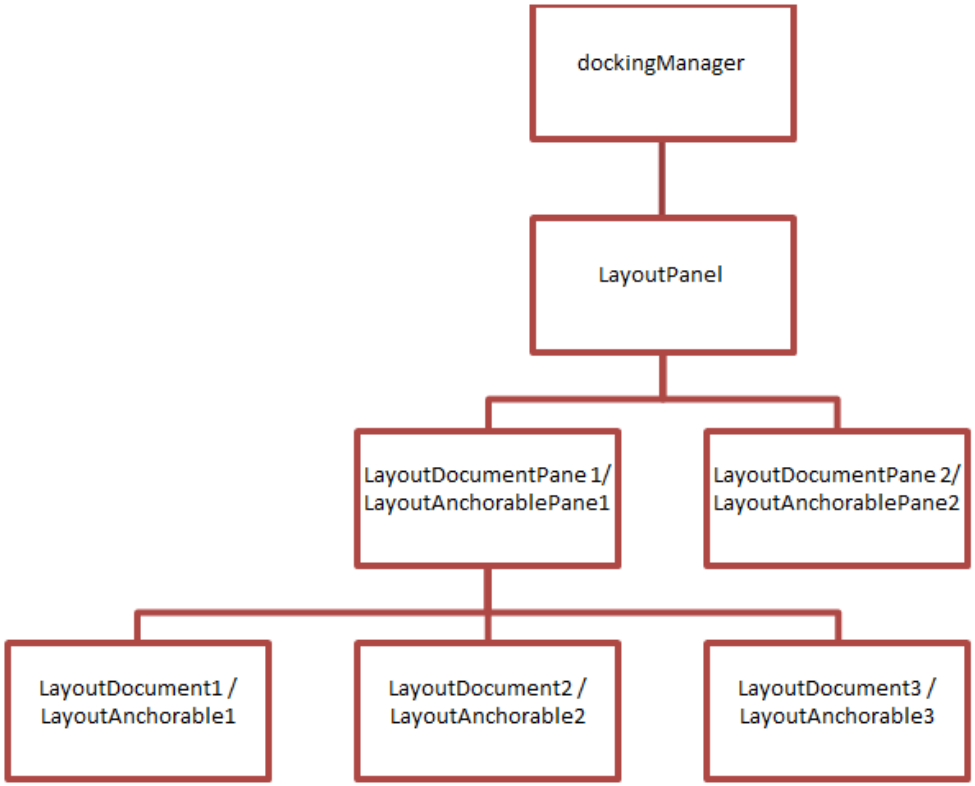
Exportar los rangos seleccionados

- Igual que cargar, pero al revés
- Se usa LINQ para obtener los datos de los View-Models
- Se crean los elementos XML y se van guardando
- Finalmente se exportan haciendo *.toString()*

Desarrollo

Cómo se ha hecho

Interfaz tipo IDE



Desarrollo

Herramientas utilizadas

- Visual Studio 2013 Ultimate
- OxyPlot
- AvalonDock
- Git
- LaTeXy TeXstudio

Gestión

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

Gestión

Scrum: Cómo se ha usado

Ha habido 7 sprints, de un mes de duración

1. Se ha creado un listado de tareas (Scrum backlog)
2. Se han ordenado las tareas por prioridad
3. Por cada sprint
   - 3.1 Se seleccionan máximo 3 tareas por sprint (Sprint backlog)
   - 3.2 Se desarrollan las funcionalidades
   - 3.3 Se documentan
   - 3.4 Se despliega

Gestión

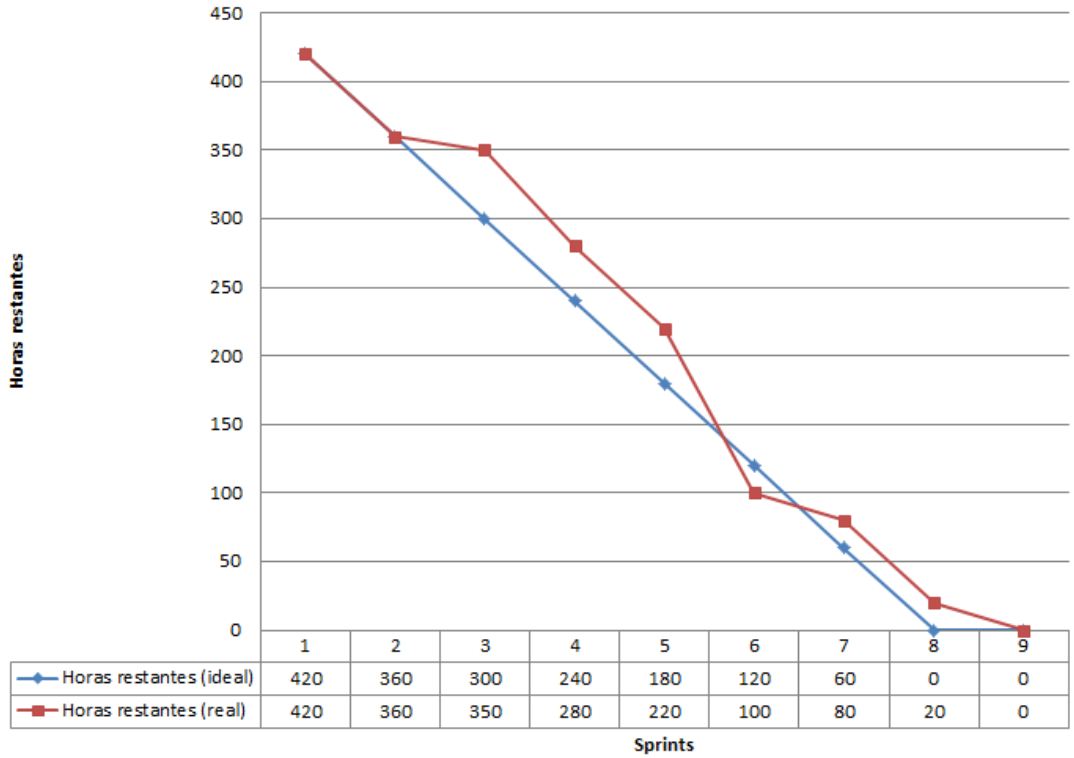
Kanban: Cómo se ha usado

- Columna “Por hacer”: Scrum backlog
- Columna “En progreso”: Sprint backlog
- Columna “Finalizado”: Tareas finalizadas organizadas por sprint

Conclusiones

Sobre la gestión

- Pese a las metodologías ágiles ha habido retraso



	1	2	3	4	5	6	7	8	9
Horas restantes (ideal)	420	360	300	240	180	120	60	0	0
Horas restantes (real)	420	360	350	280	220	100	80	20	0

Conclusiones

Riesgos que se han cumplido

- Tener que cambiar de biblioteca
- Nuevos requisitos
  - Seleccionar el momento de inicio de sincronización del vídeo
  - En un mismo XML poder añadir más de un intervalo
  - Visualizar un árbol lateral con las propiedades y observaciones

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