



Procesadores de Lenguajes 2

Transformaciones de modelo a texto con Xtend

Curso 2013-2014

Iván Ruiz Rube

Departamento de Ingeniería Informática

Escuela Superior de Ingeniería

Universidad de Cádiz



Contenidos

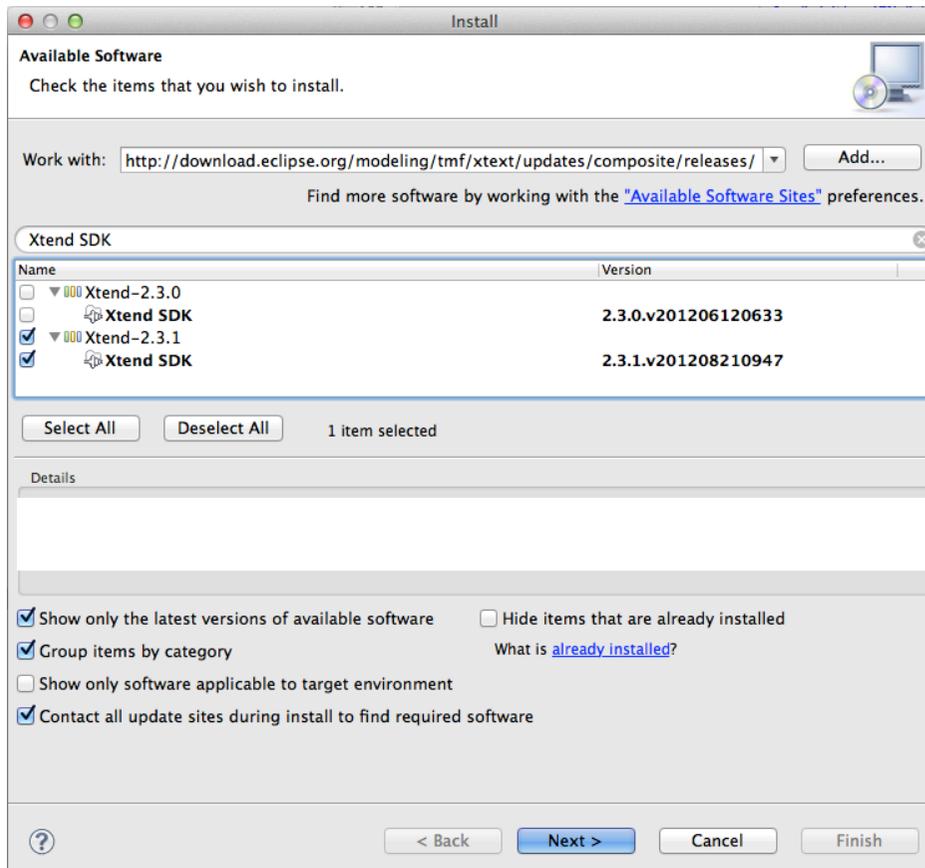
- Instalación
- Introducción
- Desarrollo de una transformación

TRANSFORMACIONES DE MODELO A TEXTO CON
XTEND



INSTALACIÓN

Instalación Xtend



Incluido en la distribución de **Xtext**.

En otro caso, instalar por el procedimiento general

Help → Install New Software

TRANSFORMACIONES DE MODELO A TEXTO CON
XTEND



INTRODUCCIÓN

Xtend

- Xtend es un pequeño lenguaje que permite simplificar la programación en Java
- Construido sobre la base de Xtext
- Este lenguaje “compila” a un código Java legible y bien formado
- El código generado es más eficiente que el desarrollado manualmente
- Diseñado para trabajar con APIs ya existentes

Características

- 100% compatible Java
- Eliminación de ruido sintáctico: visibilidad por defecto, paréntesis vacíos, etc.
- Expresiones lambda, para evitar clases anónimas
- Inferencia automática de tipos
- Instrucciones *switch* más potentes
- Expresiones para plantillas

Ejemplo: Movies.xtend

```
@Data class Movie {  
    String title  
    int year  
    double rating  
    long numberOfVotes  
    Set<String> categories  
}
```

Ejemplo (gen): Movie.java (I)

```
import java.util.Set;
import org.eclipse.xtend.lib.Data;
import org.eclipse.xtext.xbase.lib.util.ToStringHelper;

@Data
@SuppressWarnings("all")
public class Movie {
    private final String _title;

    public String getTitle() {
        return this._title;
    }

    ...
}
```

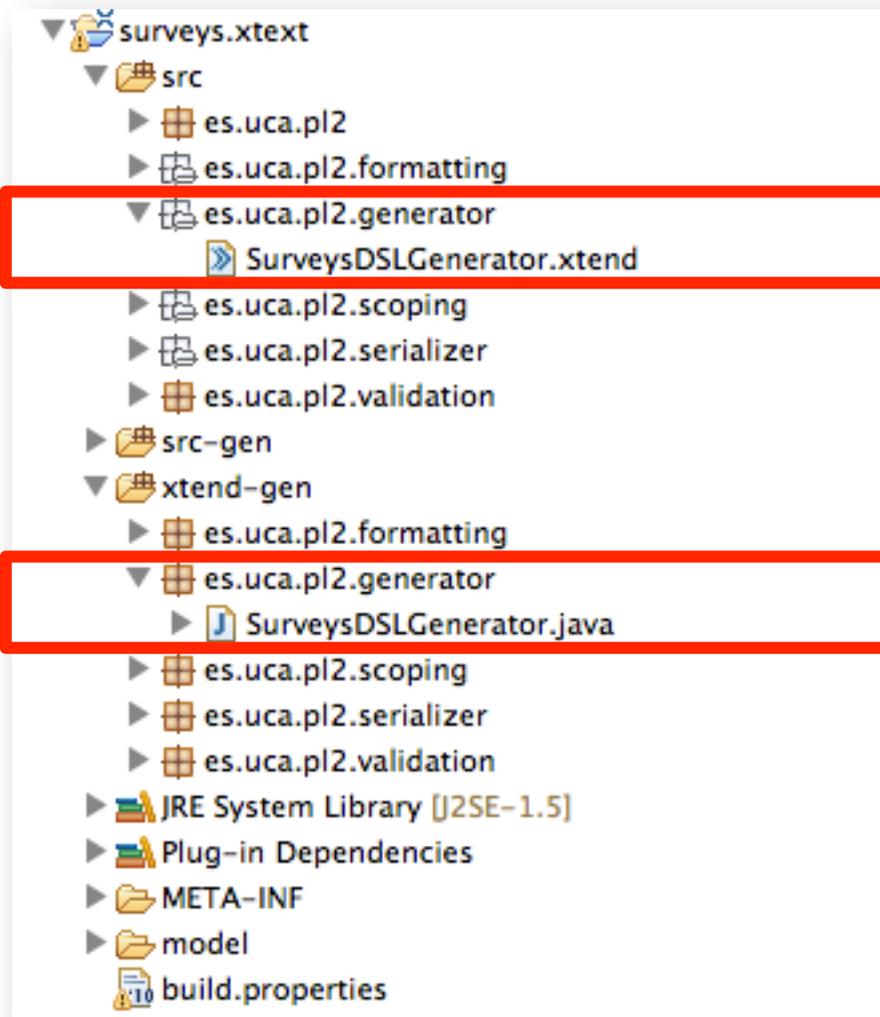
Ejemplo (gen): Movie.java (II)

```
public Movie(final String title, final int year, final double rating, final
long numberOfVotes, final Set<String> categories) {
    super();
    this._title = title;
    ..
}

@Override
public int hashCode() {
    ...
}

@Override
public boolean equals(final Object obj) {
    ...
}
```

Generador de código



- El paquete *generator* incluye todo lo necesario para la transformación M2T.
- En el fichero *.xtend* tendremos que codificar las reglas de transformación
- La carpeta *xtend-gen* contendrá el código Java resultante

Desarrollo de la transformación

- Función principal que genera un nuevo archivo HTML.

```
class SurveysDSLGenerator implements IGenerator {  
  
    override void doGenerate(Resource resource, IFileSystemAccess fsa) {  
    for (myEncuesta : resource.allContents.tolterable.filter(typeof(Encuesta))) {  
        fsa.generateFile(myEncuesta.nombre + ".html", myEncuesta.toHTML)  
    }  
}
```

Desarrollo de la transformación (II)

- Función auxiliar que genera el contenido

```
def toHTML(Encuesta myQuiz) ""
  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
  <head><title>Mis Encuestas</title></head>
  <body>
  <table border="1">
    <tr>
      <td><b>Pregunta</b></td>
      <td><b>Descripcion</b></td>
      <td><b>Valoracion</b></td>
    </tr>
    «FOR myQuestion : myQuiz.preguntas»
      «myQuestion.toHTML»
    «ENDFOR»
  </table>
  </body>
  </html>
```

...

Desarrollo de la transformación (III)

- Otra función auxiliar...

```
def toHTML(Pregunta myQuestion) ""  
  
    <tr>  
        <td>  
            «myQuestion.titulo»  
        </td>  
        <td>  
            «myQuestion.descripcion»  
        </td>  
        <td>  
            «myQuestion.puntuacion»  
        </td>  
    </tr>  
  
    ""  
}
```

Código Java generado por Xtend

```
EncuestasDSLGenerator.java ✖
package es.uca.pl2.generator;

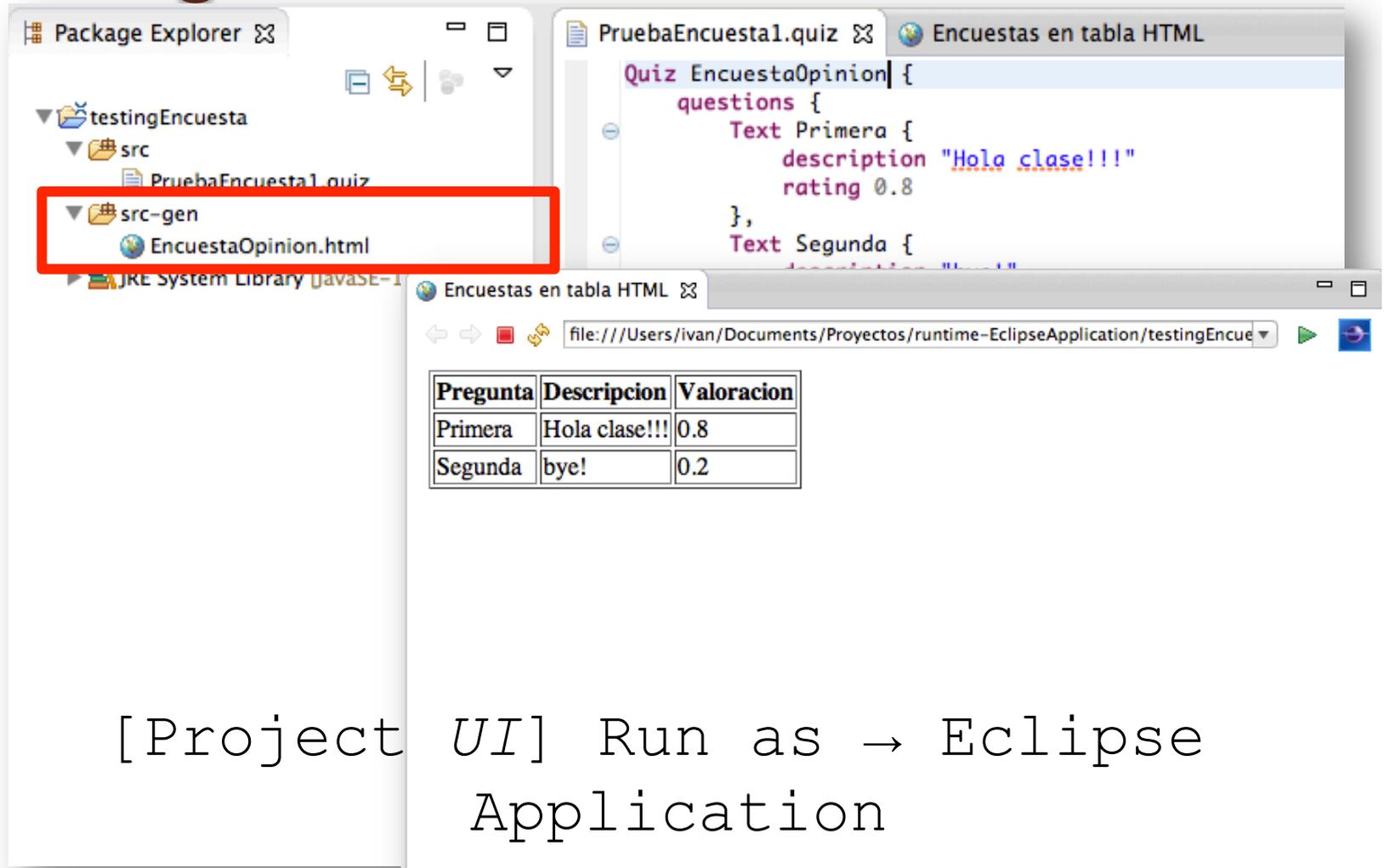
import com.google.common.collect.Iterables;

@SuppressWarnings("all")
public class EncuestasDSLGenerator implements IGenerator {
    @Inject
    private IQualifiedDataProvider _iQualifiedDataProvider;

    public void doGenerate(final Resource resource, final IFileSystemAccess myHTMLFile) {
        TreeIterator<EObject> _allContents = resource.getAllContents();
        Iterable<EObject> _iterable = IteratorExtensions.<EObject>toIterable(_allContents);
        Iterable<Quiz> _filter = Iterables.<Quiz>filter(_iterable, Quiz.class);
        for (final Quiz myQuiz : _filter) {
            QualifiedName _fullyQualifiedName = this._iQualifiedDataProvider.getFullyQualifiedName(myQuiz);
            String _string = _fullyQualifiedName.toString("/");
            String _plus = (_string + ".html");
            CharSequence _hTML = this.toHTML(myQuiz);
            myHTMLFile.generateFile(_plus, _hTML);
        }
    }

    public CharSequence toHTML(final Quiz myQuiz) {
        StringConcatenation _builder = new StringConcatenation();
        _builder.newLine();
        _builder.append("<html xmlns=\"http://www.w3.org/1999/xhtml\" xml:lang=\"es\" lang=\"es\">");
        _builder.newLine();
        _builder.append("<head>");
        _builder.newLine();
        _builder.append("\t");
        _builder.append("<title>Encuestas en tabla HTML</title>");
        _builder.newLine();
        _builder.append("</head>");
        _builder.newLine();
        _builder.append("<body>");
        _builder.newLine();
    }
}
```

Test del DSL con generación de código



```
Quiz EncuestaOpinion| {
  questions {
    Text Primera {
      description "Hola clase!!!"
      rating 0.8
    },
    Text Segunda {
      description "bye!"
      rating 0.2
    }
  }
}
```

Pregunta	Descripcion	Valoracion
Primera	Hola clase!!!	0.8
Segunda	bye!	0.2

[Project *UI*] Run as → Eclipse Application

Transformaciones de modelo a texto con Xtend

Iván Ruiz Rube

ivan.ruiz@uca.es