

**ANEXO 5:**  
**MANUAL DE INSTALACIÓN Y USO**  
ANALIZADOR ESTÁTICO DE CÓDIGO PARA POLÍTICAS DE  
CONTROL DE ACCESO

ARIEL ARTURO LÓPEZ LESMES  
ING. JAIME ANDRÉS PAVLICH MARISCAL, Ph.D.

PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS  
BOGOTÁ, D.C.  
2012

## Tabla de contenido

<b>1 INTRODUCCIÓN</b>	<b>3</b>
<b>2 REQUISITOS</b>	<b>3</b>
2.1 JAVA	3
2.2 ECLIPSE	3
<b>3 INSTALACIÓN</b>	<b>3</b>
3.1 ACCELEO	3
3.2 EMFTEXT	3
3.3 PLUG-IN SARE	3
3.4 SUBCLIPSE	4
3.5 JBOSS Y SEAM FRAMEWORK	4
3.6 GENERADOR DE CÓDIGO SARE GENERATOR.M2T	4
3.7 GENERADOR DE INVERSO Y ANALIZADOR ESTÁTICO DE CÓDIGO SARE GENERATOR.T2M	4
<b>4 USO</b>	<b>4</b>
4.1 GENERADOR DE CÓDIGO SARE GENERATOR.M2TC	4
4.2 GENERADOR DE INVERSO Y ANALIZADOR ESTÁTICO DE CÓDIGO SARE GENERATOR.T2M	7
<b>REFERENCIAS</b>	<b>8</b>

## 1 Introducción

Este documento pretende definir los requerimientos mínimos para poder ejecutar la aplicación desarrollada de una forma óptima. Se ilustran la forma de descarga e instalación de las herramientas así como de todos los complementos necesarios no solo para ejecutar el analizador estático de código sino también todas las herramientas del proyecto SARE.

Finalmente se ilustra cómo se deben ejecutar cada una de las herramientas para obtener los resultados esperados.

## 2 Requisitos

### 2.1 Java

Descarga e instalar la última versión del java SE JDK disponible. Esto puede realizarse desde: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

### 2.2 Eclipse

Descargar e instalar el IDE Eclipse Indigo en su versión Eclipse Modeling Tools. Esto puede realizarse desde:

<http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/indigosr2>

## 3 Instalación

La versión de Eclipse descargada en el paso anterior incluye el proyecto EMF, de no haber descargado dicha versión, se debe descargar e instalar el proyecto EMF en su última versión disponible.

Ejecute Eclipse y en la pestaña *help /install new software* busque cada una de las siguientes herramientas:

### 3.1 Acceleo

Descargar e instalar desde el sitio de descarga:

<http://download.eclipse.org/modeling/m2t/acceleo/updates/releases/>

### 3.2 EmFtext

Descargar e instalar desde el sitio de descarga:

[http://www.emftext.org/update\\_trunk](http://www.emftext.org/update_trunk)

### 3.3 Plug-In SARE

Descargar e instalar desde el sitio de descarga:

<http://sare.googlecode.com/svn/trunk/sare.AccessControlModel/sare.AccessControlModel.AcmUpdateSite/>

### 3.4 Subclipse

Descargar e instalar desde el sitio de descarga:

[http://subclipse.tigris.org/update\\_1.8.x](http://subclipse.tigris.org/update_1.8.x)

### 3.5 JBoss y Seam Framework

Instalar Jboss 5.1 GA como se indica en la guía de instalación de la profesora Maria Consuelo Franky. Esta guía se presenta como un documento anexo [1].

Instalar JBoss-Seam como se indica en la guía de instalación de la profesora Maria Consuelo Franky. Esta guía se presenta como un documento anexo [1].

### 3.6 Generador de código SARE Generator.m2t

La versión más actualizada del generador se debe descargar de repositorio SVN:

<https://sare.googlecode.com/svn/trunk/sare.CodeGenerator/sare.m2t.generator/>

Descargar como proyecto en el workspace. Verificar que las referencias de a librerías en el build path correspondan a su máquina.

### 3.7 Generador de inverso y analizador estático de código SARE Generator.t2m

La versión mas actualizada del generador inverso se debe descargar de repositorio SVN:

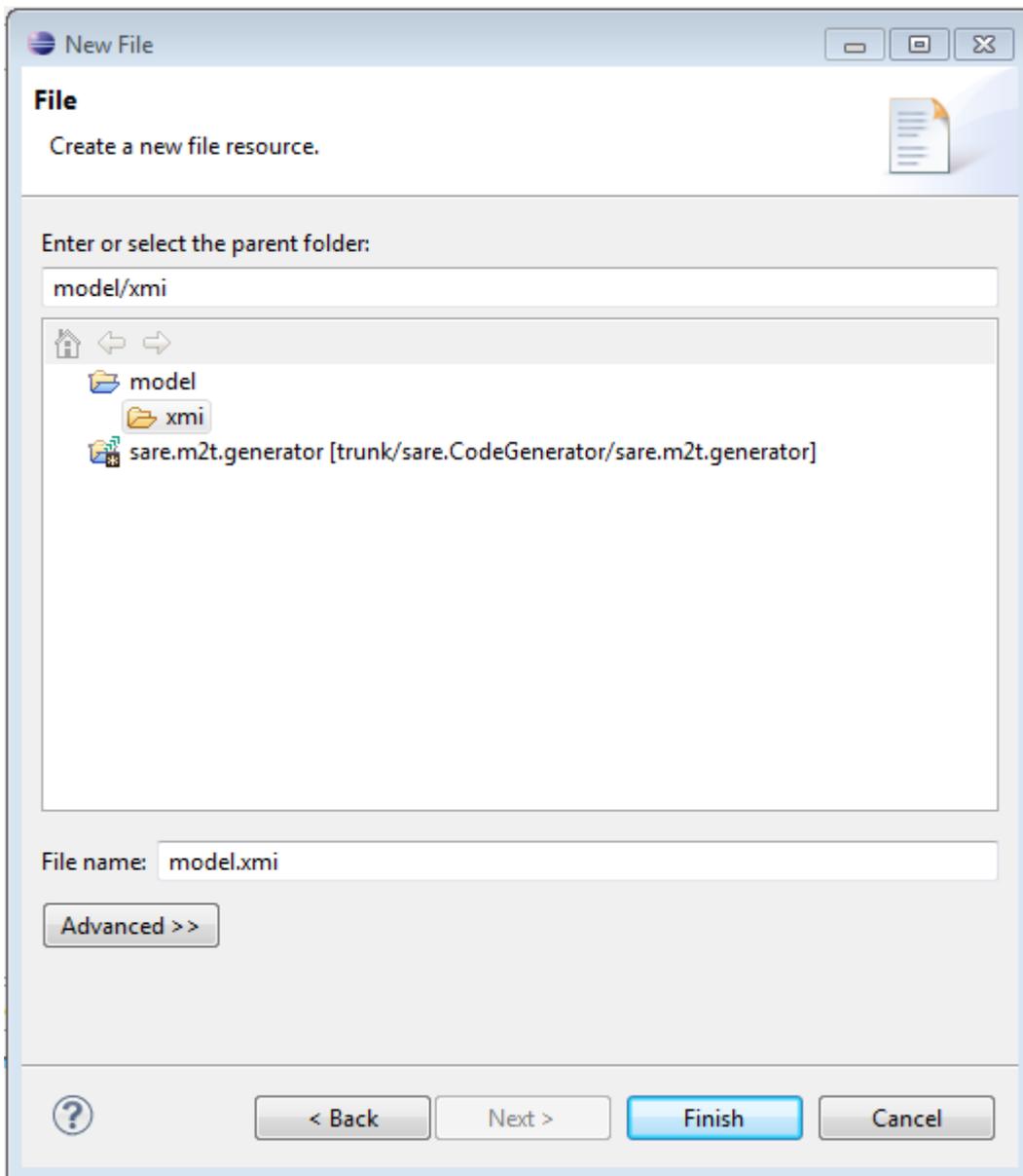
<https://sare.googlecode.com/svn/trunk/sare.ReverseEngineeringGenerator/sare.t2m.generator/>

Descargar como proyecto en el workspace. Verificar que las referencias de a librerías en el build path correspondan a su máquina.

## 4 Uso

### 4.1 Generador de código SARE Generator.m2tC

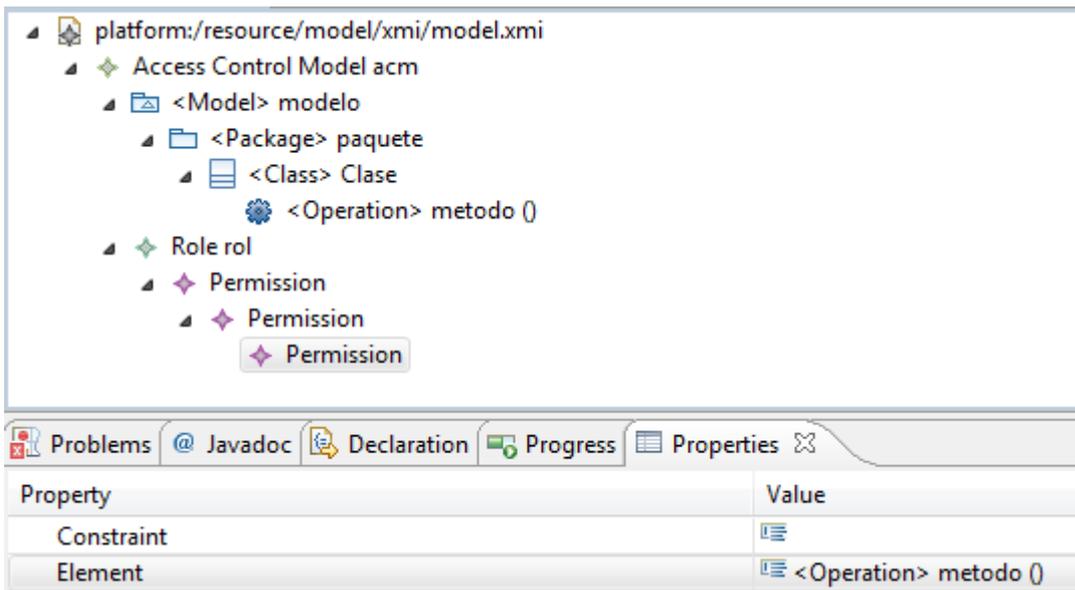
Crear un diagrama de entidades donde reflejen entidades y beans de sesión de la aplicación a desarrollar, creado los roles y los permisos a los elementos del diagrama de entidades.



Abrir el archivo como un archivo de texto y copiar lo siguiente:

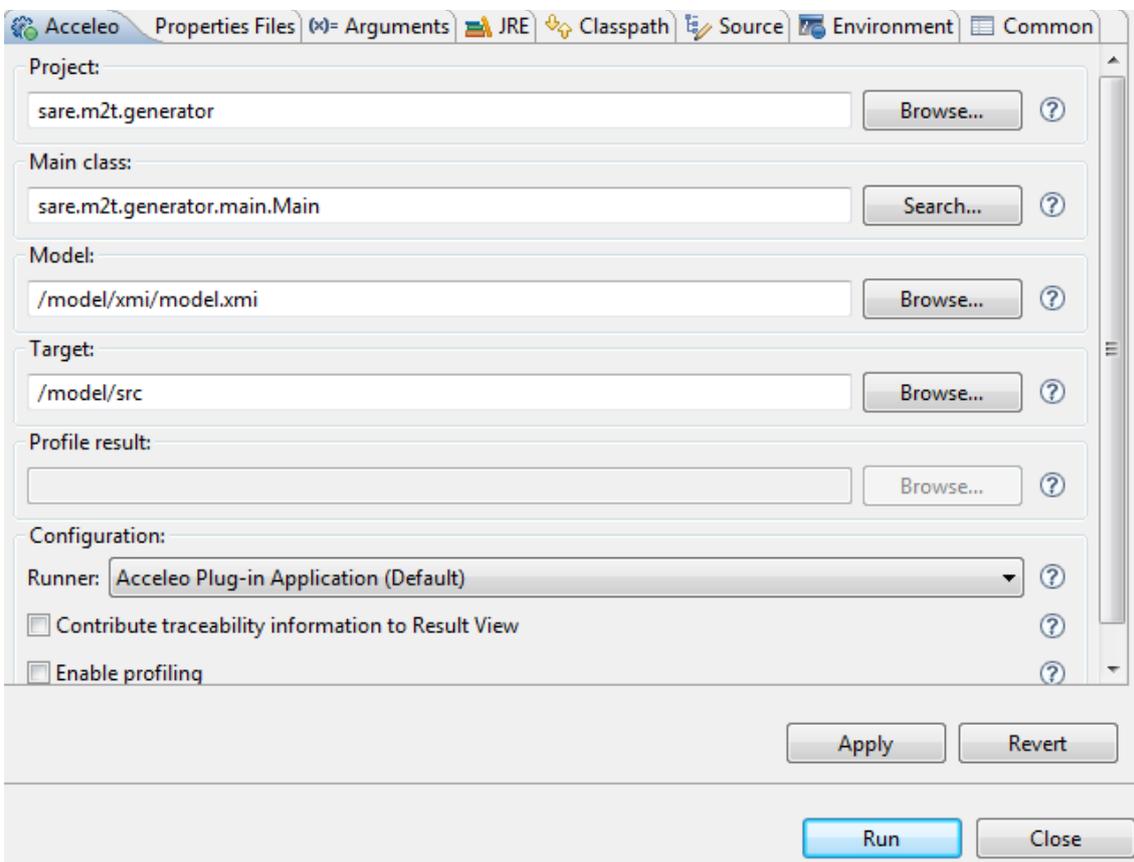
```
<?xml version="1.0" encoding="ASCII"?>
<acm:AccessControlModel xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance "
xmlns:acm="http://sare/language/acm "
xmlns:uml="http://www.eclipse.org/uml2/3.0.0/UML ">
</acm:AccessControlModel>
```

Posteriormente se debe abrir el modelo con el “sample reflective model editor”. Y se inicia la construcción del modelo siguiendo la estructura que se muestra en la siguiente imagen:



De esta forma se crea un modelo de control de acceso que contiene un modelo uml2 y unos roles. El modelo UML tiene un paquete que contiene una clase con un método. Existe un rol que tiene permiso sobre el paquete y a su vez un permiso sobre la clase y a su vez un permiso sobre el método. Esa es la estructura que se debe seguir en el modelado.

Una vez finalizado el modelo, se debe configurar la ejecución como se muestra en la siguiente imagen. Para eso, se da clic derecho sobre el generador y se selecciona run as, luego aparece la siguiente ventana:



En caso de presentarse errores asegure de que la ruta del plug-in de sare.AccessControlModel corresponde a una ruta de su computador.

## 4.2 Generador de inverso y analizador estático de código SARE Generator.t2m

Antes de ejecutar:

- Se debe contar con una carpeta que contenga clases java.
- Se debe contar con el modelo original de políticas de control de acceso.
- Se debe contar con los .jar de las librerías referenciadas en el código. Como las de javax.persistence y las anotaciones de sean.

Como ejemplo de los parámetros de entrada se muestra:

```
/Users/arl0le07/Documents/workspace/JaMoPPCTest/src  
/Users/arl0le07/Documents/workspace/JaMoPPCTest/xmi/model.xmi  
/Users/arl0le07/Documents/workspace/jars/javax.persistence_2.0.3.v201010191057.jar  
/Users/arl0le07/Documents/workspace/jars/jboss-seam.jar
```

- El primer parámetro se reserva para la carpeta que contiene las clases java.
- El segundo parámetro se reserva para el modelo destino.
- Del tercer parámetro en adelante se registran las librerías referenciadas en el código.

Para el analizador estático de código se debe ingresar la ruta del modelo original de políticas de control de acceso. Para esto, en *sare.t2m.generator/src/jamoppc/Jamoppc.java*

```
public class Jamoppc {  
  
    protected static final ResourceSet rs = new ResourceSetImpl();  
    protected static final ResourceSet rsFinal = new ResourceSetImpl();  
    protected static final String OriginalModelFile= "/Users/arl0le07/Do
```

Se debe modificar el atributo OriginalModelFile con la ruta del modelo.

Una vez configurado lo anterior se podrá ejecutar la aplicación.

## Referencias

- [1] Franky, M. Instalación y configuración de las herramientas para Java EE 5.
- [2] Franky, M. Guía de los generadores del framework Seam.