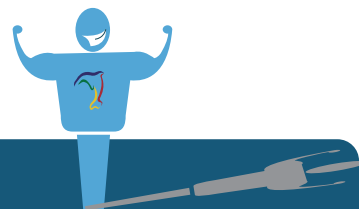


¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

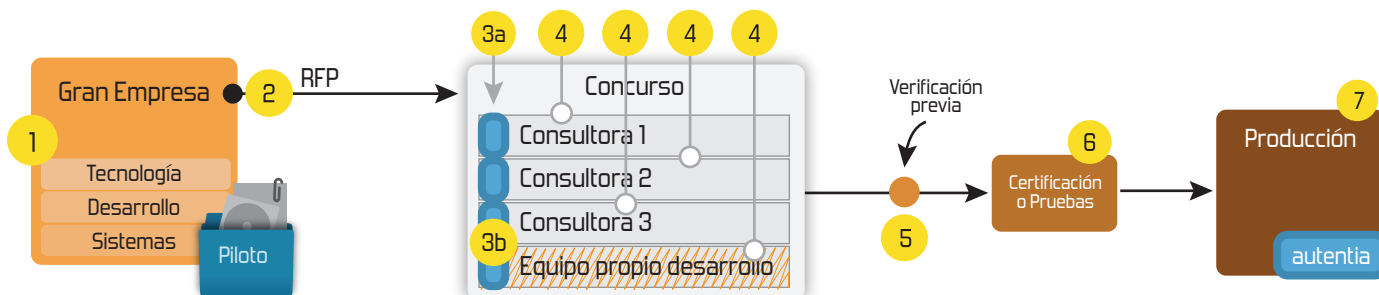
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)


 » Estás en: [Inicio](#) [Tutoriales](#) [Templates en Eclipse](#)

Alejandro Ortíz Martín

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero superior en Informática, especialidad en Ingeniería del Software.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)


Tu foto impresa en una original pieza con carcasa personalizable

 por sólo
9,95€

Fecha de publicación del tutorial: 2015-02-06

 Tutorial visitado 5 veces [Descargar en PDF](#)

Templates en Eclipse

0. Índice de contenidos.

- 1. Introducción
- 2. Entorno
- 3. ¿Que es un Template?
- 4. Creando nuestro primer Template
- 5. Algunos ejemplos
- 6. Conclusiones

1. Introducción

En este tutorial vamos a ver los Templates del entorno de programación Eclipse, y algunos ejemplos de sus posibles aplicaciones.

Eclipse cuenta con diversas herramientas de generación automática de código, accesibles desde el menú 'Source'. Aunque el código generado es eficaz, en muchas ocasiones nos gustaría poder personalizar ese código a nuestro gusto, o simplemente tener atajos para patrones que usamos muy a menudo. Los Templates de Eclipse nos permiten, como indica el nombre, disponer de unos Templates o trocitos de texto para generarlos rápidamente. En cierto modo parecido a tener copiado en todo momento lo que más nos interesa, listo para ser pegado a voluntad.

2. Entorno

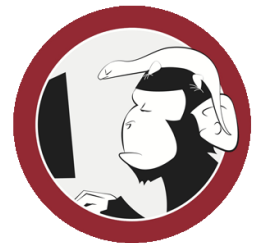
En este tutorial está escrito usando Eclipse Luna, aunque la funcionalidad de Templates de Eclipse no es exclusiva de esta versión, y de hecho tengo entendido que la interfaz de uso de la mismas no ha variado mucho tampoco.

3. ¿Que es un Template?

Comenzamos desde el principio, ¿Que es un Template?

Los Templates en Eclipse nos permiten expandir palabras clave en fragmentos de código. Si por ejemplo escribo do al escribir código en una clase java, y pulso Ctrl+Space, veo como entre las proposiciones de Eclipse se encuentra un Template predeterminado:

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» 2015: ¡Volvemos a la oficina!

 » [Curso JBoss de Red Hat](#)

» Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris

» Portales, gestores de contenidos documentales y desarrollos a medida

 » [Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer](#)
[Histórico de noticias](#)

Últimos Tutoriales

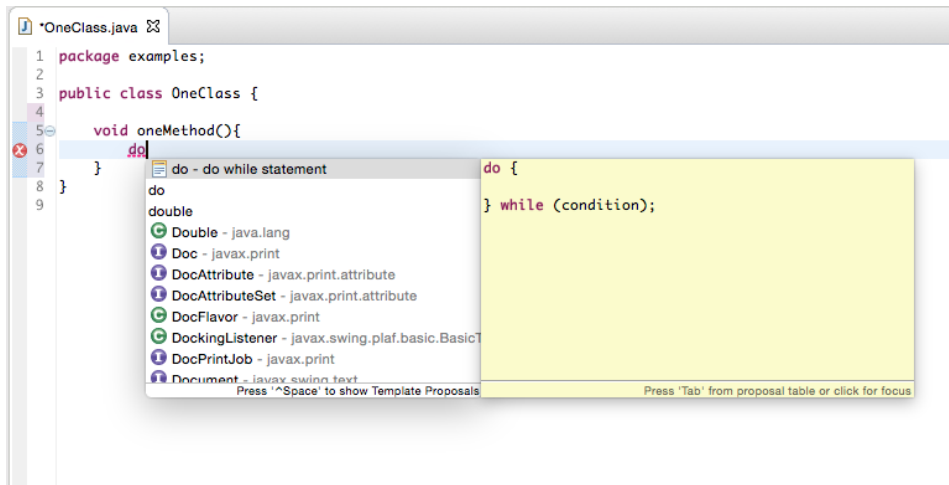
 » [Pruebas automáticas con FTP](#)

 » [Oracle - Importación de Datos con Data Pump Import \[impdp\]](#)

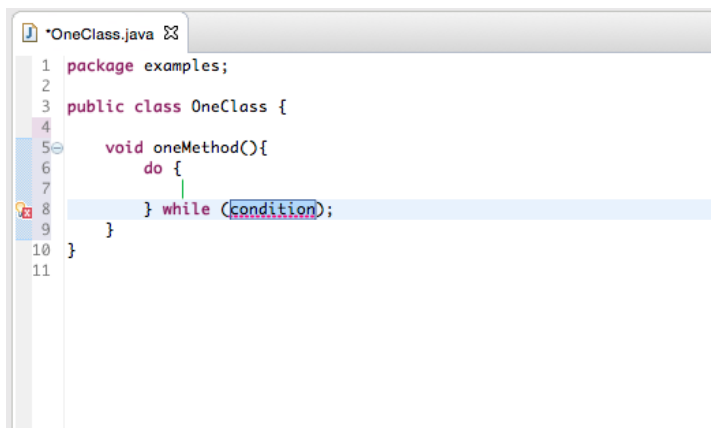
 » [Oracle - Exportación de Datos con Data Pump Export \[expdp\]](#)

 » [Flexibilidad laboral: mejoras para empleados y empresarios](#)

 » [Instalación de un entorno Hadoop con Ambari en AWS](#)



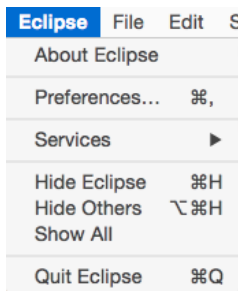
Si selecciono esa opción, se me escribirá el resto del bucle do-while.

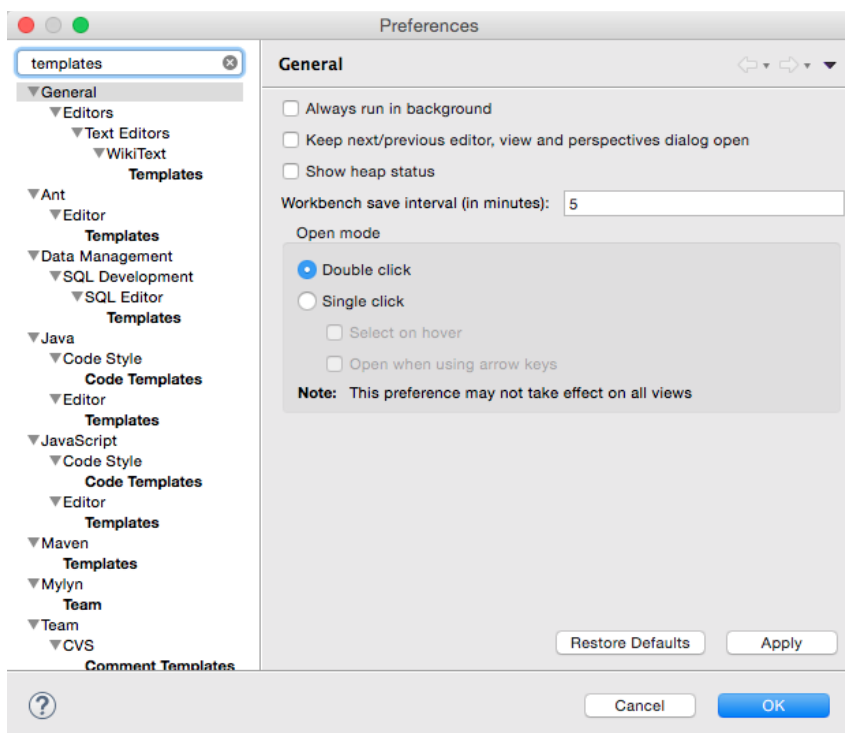


Este caso que acabamos de ver es un ejemplo de Template. Estos Templates son fáciles de hacer, y hoy vamos a ver algunos ejemplillos.

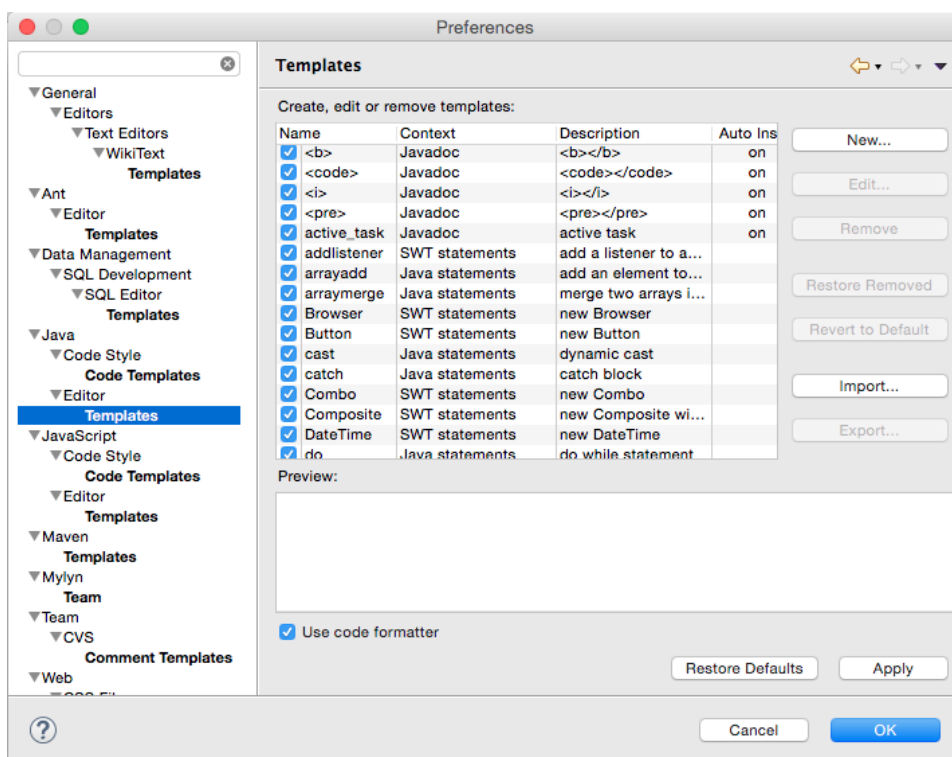
4. Creando nuestro primer Template

En primer lugar accedemos a las conocidas preferencias de Eclipse, y ahí buscamos templates.

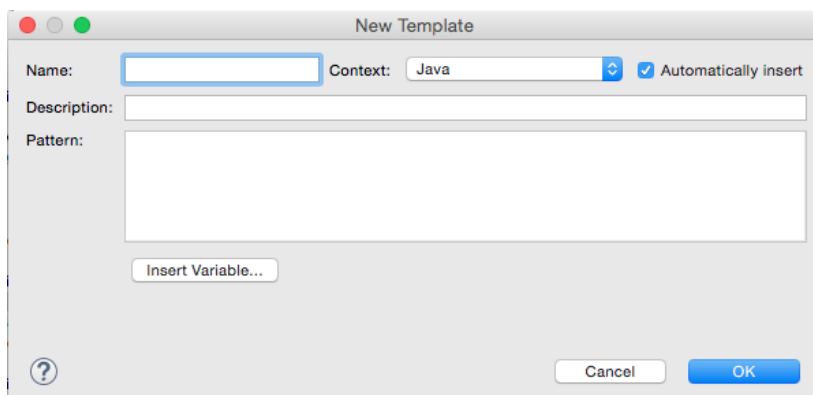




Como podemos ver, Eclipse tiene templates para cada uno de los editores que contiene: SQL, CSS, HTML... En nuestro caso crearemos un par de Templates para el editor de java, pero el funcionamiento es el mismo en todos los editores. Al seleccionar Java > Editor > Templates, vemos como ya hay un monton creados por defecto.



Se nos permite crearlos, editarlo, eliminarlos... e incluso importarlos desde un fichero xml, como veremos más adelante. Vamos a crear un nuevo Template para ver como funcionan.



La interfaz de crear un nuevo Template es la misma que la usada para editar uno creado. Tenemos una serie de campos a

rellenar:

- Name: Identificador del Template. Será el nombre que se usará para acceder al mismo.
- Contexto: Como ya hemos visto antes, Eclipse distingue los Templates para los distintos editores que contiene. El contexto nos permite seleccionar en que momentos estará disponible el Template en el editor. Por ejemplo en java contamos con los contextos javadoc o sentencias java.
- Automatically insert: Esta opción permite expandir automáticamente el Template con Ctrl+Space si no coincide ningún otro.
- Description: Descripción del Template que nos permitirá distinguirlo con más facilidad.
- Pattern: Código del Template en si.

Por último, vemos un botón de 'Insert Variable'. Los Templates no son simplemente trozos de texto que se insertan, cuentan con un poco de lógica que, aunque limitada, nos permitirá hacerlos más útiles y cómodos. Esto se lleva a cabo a través de las variables.

Al pulsar el mencionado botón, vemos una lista de las variables que están disponibles con una breve descripción de cada una. Estas variables son distintas dependiendo del editor donde estes creando el Template. Por ejemplo en java tenemos 'enclosing_method', que proporciona el nombre del método en el cual se expanda el Template, o 'primary_type_name', que proporciona el nombre de la clase principal del archivo donde se expanda.

Además de las variables con funcionamientos especiales como los ejemplos mencionados, se pueden definir variables nuevas escribiendo '\${nombre_variable}'. Estas variables tienen la gracia de que al repetir el mismo nombre a lo largo del template, al expandirlo se modifican todas al modificar una. Esto lo veremos en acción en los ejemplos.

5. Algunos ejemplos

5.1. hashCode y equals

Este ejemplo me gusta porque no solo es útil tener este template, sino que muestra dos ejemplos que nos vienen muy bien.

Además nos crea el hashCode y el equals a la vez, para que no se nos olvide poner ambos y que el Sonar siga calladito.

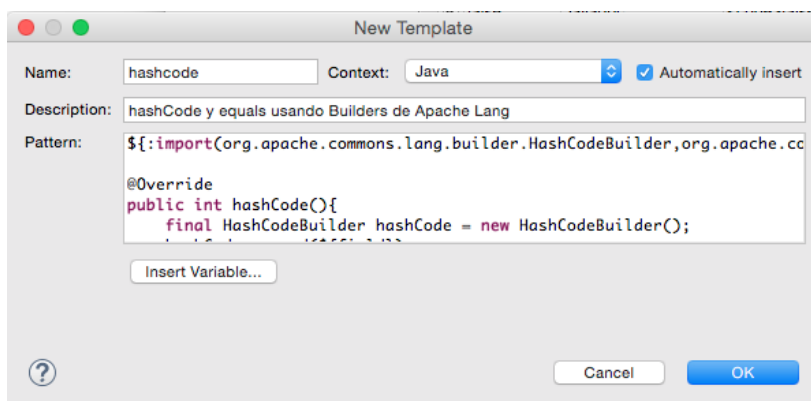
Aunque el hashCode y el equals que genera automáticamente Eclipse son correctos (es decir, funcionan) mucha gente prefiere otros métodos para generarlos. En mi caso, me gustan las herramientas que proporciona [Apache Lang](#):

HashCodeBuilder y EqualsBuilder.

Veamos el código del Template:

```
1  ${:import(org.apache.commons.lang3.builder.HashCodeBuilder,org.apache.commons.lang3.builder.EqualsBuilder)}
2
3  @Override
4  public int hashCode(){
5      final HashCodeBuilder hashCode = new HashCodeBuilder();
6      hashCode.append(${field});
7      return hashCode.toHashCode();
8  }
9
10 @Override
11 public boolean equals(Object obj){
12     boolean isEqual = false;
13     if (obj instanceof ${enclosing_type}) {
14         final ${enclosing_type} param = (${enclosing_type})obj;
15         final EqualsBuilder equals = new EqualsBuilder();
16         equals.append(this.${field}, param.${field});
17
18         isEqual = equals.isEquals();
19     }
20     return isEqual;
21 }
```

Este código lo insertamos en la creación del Template con el nombre 'hashCode', contexto java y una breve descripción. Dado que estas opciones no afectan al rendimiento del Template, podeis poner las que querais:



Analicemos el código del Template... Vemos como usamos 3 variables distintas. En primer lugar tenemos una variable 'import', que se asegura de añadir los imports correspondientes si no estuvieran ya incluidos.

Después tenemos 'field', una variable genérica de las que no tiene lógica, que al desplegar el Template nos permitirá editar a la vez los 3 sitios en que aparece.

Por último tenemos 'enclosing_type'. Esta variable se sustituye al desplegar por el nombre del fichero, que suele corresponder con el nombre de la clase.

Para hacer uso del nuevo Template bastará con escribir 'hashCode' (el nombre del Template) y pulsar Ctrl+Space. Al seleccionarlo en el desplegable se expandirá haciendonos el trabajo. Al rellenar el

5.2 Imports de testing

Otro Template que uso bastante es el de los imports estáticos que usamos para los tests. Antes de tenerlo tenía que abrir otro test y copiarlos, así que este Template me ahorra un par de clicks bastante a menudo.

Además Eclipse no resuelve automáticamente los imports estáticos, por lo que es especialmente útil.

```
1  ${:importStatic('org.hamcrest.Matchers.*','org.junit.Assert.*','org.mockito.Matchers.*')}
```

Este es especialmente útil, pues Eclipse no resuelve automáticamente los imports estáticos.

6. Conclusiones

Bueno esta claro que los Templates del Eclipse no son una panacea y tienen bastantes limitaciones, pero son los pequeños detalles como estos los que hacen que uno se sienta agusto con su entorno de programación. Espero que os haya sido útil este tutorial... y si se os ocurre algún uso para estos Templates dejadlo en los comentarios!

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

Share |

8+1 0

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

¿Ayuda?

sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by [karmacracy](#)

Copyright 2003-2015 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

W3C XHTML 1.0

W3C CSS

XML RSS

XML ATOM