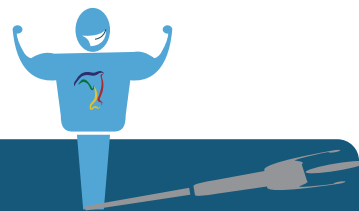


¿Qué ofrece Autentia Real Business Solutions S.L?

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

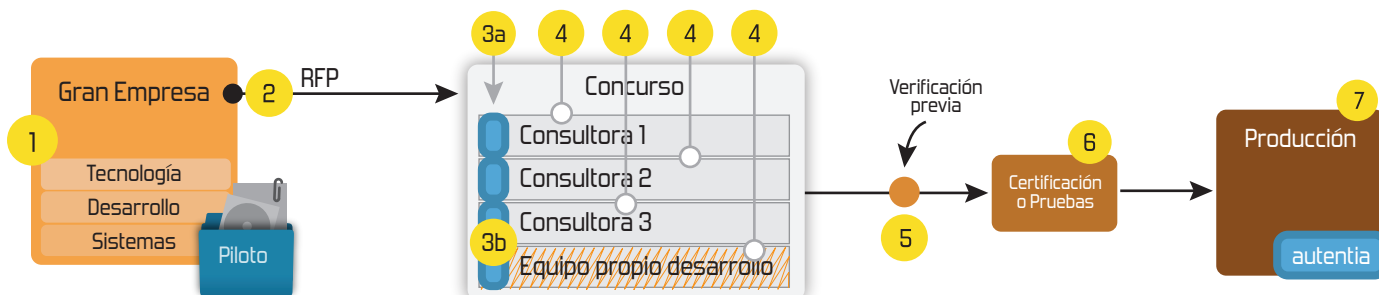
1. Desarrollo de componentes y proyectos a medida



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](#) Eclipse Indigo, la versión 3.7 de Eclipse



DESARROLLADO POR:
 Alejandro Pérez García

Alejandro es socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.

Ingeniero en Informática y Certified ScrumMaster

Si te gusta lo que ves, puedes contratarle para darte ayuda con soporte experto, impartir **cursos presenciales** en tu empresa o para que **realicemos tus proyectos como factoría** (Madrid). Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Catálogo de servicios Autentia

MÁSTER OFICIAL EN
DISEÑO GRÁFICO



Fecha de publicación del tutorial: 2011-06-27



Share |

[Regístrate para votar](#)

Eclipse Indigo, la versión 3.7 de Eclipse

Creación: 24-06-2011

Índice de contenidos

1. Introducción
2. Entorno
3. Instalar Eclipse Indigo
 - 3.1. Descarga de Eclipse Indigo
 - 3.2. Instalación de Eclipse Indigo
 - 3.3. Tuning de memoria de Eclipse Indigo
4. Ejecutar Eclipse Indigo
5. Instalando m2e (Maven Integration for Eclipse, o m2eclipse)
 - 5.1. Configurando m2e
 - 5.1.1. m2e integración con wtp
 - 5.1.2. Actualizar la versión de Maven que viene embebido con m2e
6. Conclusiones
7. Sobre el autor

1. Introducción

Ya llega el verano, así que como no podía ser de otra manera Eclipse nos presenta su nueva versión, la 3.7, llamada **Eclipse Indigo**.

Podemos encontrar que 62 proyectos forman parte de esta nueva versión. Entre estos proyectos podemos destacar:

Últimas Noticias

- [Alfresco Day 2011](#)
- [XVII Charla Autentia - Grails - Vídeos y Material](#)
- [iii 15 millones de descargas de tutoriales !!!](#)
- [XVII Charla Autentia - Grails](#)
- [Charla en WhyFLOSS en el IE: la ppt](#)

[Histórico de NOTICIAS](#)

Últimos Tutoriales

- [EpubLib, una librería Java para leer Epub](#)
- [Trabajando con GIT, introducción al uso de los branch y git-completion.bash](#)
- [Introducción a Spring Security 3.1](#)
- [Implementando SSO con CAS: ejemplo práctico](#)
- [Como desarrollar un plugin para Eclipse](#)

- Mejor integración con Maven, gracias a M2Eclipse de la mano de Sonatype. M2Eclipse nació siendo bastante pobre en funcionalidades, pero a lo largo de los años ha conseguido desbancar a Eclipse IAM.
- EGit para integrarnos con el, cada vez más famoso, sistema de control de versiones GIT. Aunque no os lo recomiendo, sobre todo si usáis GitHub, ya que parece que hay un bug en EGit que puede corromper los repositorios: <http://help.github.com/fix-egit-corruption/> En su lugar os recomiendo usar la línea de comandos; aquí tenéis algunos trucos para que os resulte cómodo y sencillo: [Trabajando con GIT, introducción al uso de los branch y git-completion.bash](#)
- Jubula, que proporciona test funcionales automático del GUI de nuestras aplicaciones, ya sean Java o HTML.
- WindowBuilder, es una popular herramienta de Google para construir GUI (Interfaces Gráficas de Usuario). Ahora es open source y parte de Indigo.

Todos estos plugins los podremos instalar de la manera tradicional indicando la URL (*Help --> Install New Software...*) o con el reciente incorporado Marketplace de Eclipse (*Help --> Eclipse Marketplace...*)

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 17' (2.8 GHz Intel i7, 8GB DDR3 SDRAM, 256GB Solid State Drive).
- NVIDIA GeForce GT 330M with 512MB
- Sistema Operativo: Mac OS X Snow Leopard 10.6.8
- Eclipse 3.7, codename Indigo

3. Instalar Eclipse Indigo

3.1. Descarga de Eclipse Indigo

Instalar Eclipse Indigo es tan fácil como sus versiones anteriores. Primero iremos a la página de Eclipse: <http://www.eclipse.org>

Últimos Tutoriales del Autor

 Trabajando con GIT, introducción al uso de los branch y git-completion.bash

 RVM y como actualizar Ruby a la versión 1.9.2 en Snow Leopard 10.6.7

 REST y como hacer con jQuery un PUT hacia Spring MVC

 Jackson y como deserializar objetos JSON usando un constructor

 Como editar XML o HTML con el plugin xmledit de Vim

Síguenos a través de:



Últimas ofertas de empleo

2011-06-20  Comercial - Ventas - SEVILLA.

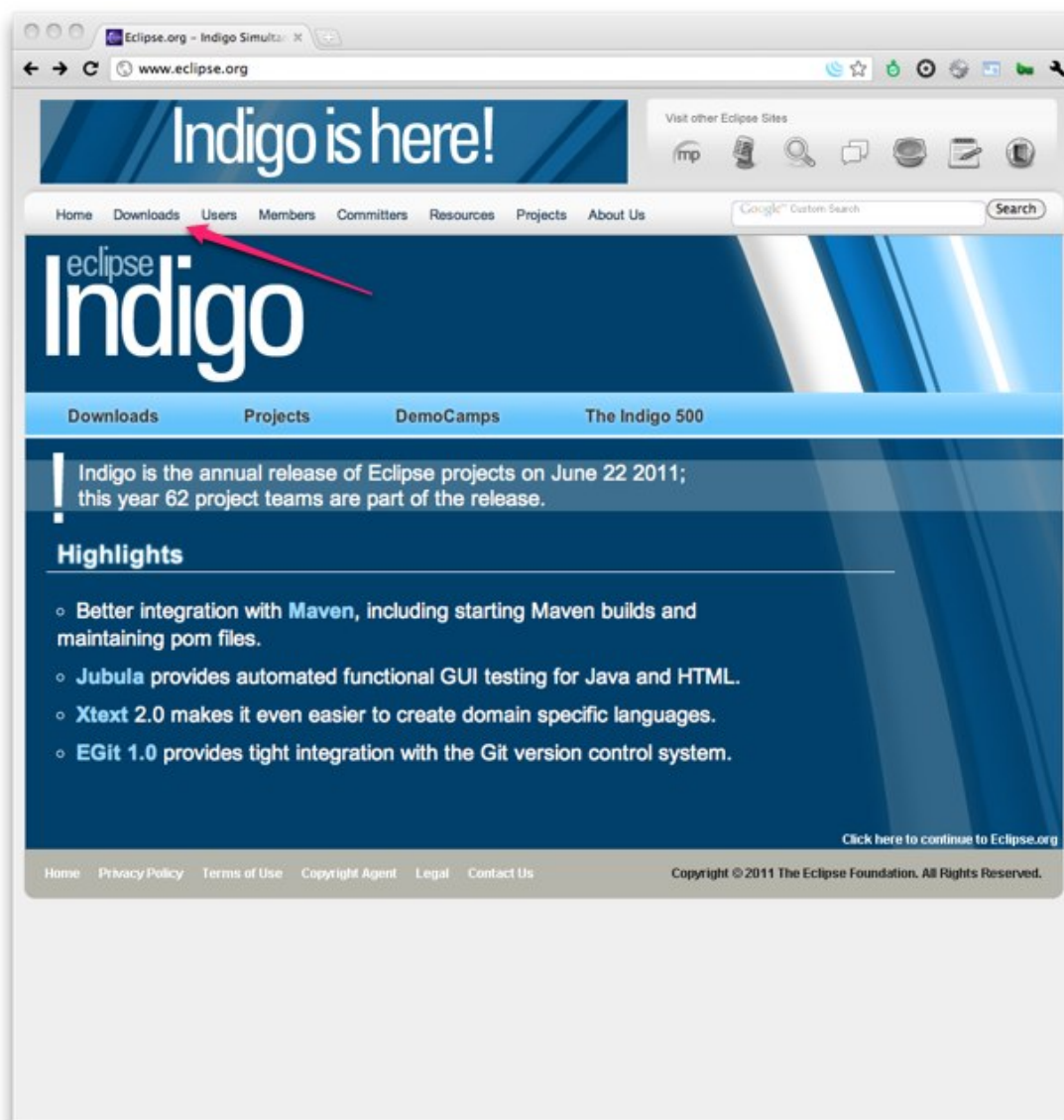
2011-05-24  Contabilidad - Especialista Contable - BARCELONA.

2011-05-14  Comercial - Ventas - TARRAGONA.

2011-04-13  Comercial - Ventas - VALENCIA.

2011-04-04  Comercial - Compras - CANTABRIA.

 Alejandro Pérez
alejandropgarc



Trabajando con #GIT, introducción al uso de los branch y git-completion.bash:
<http://bit.ly/iEduyH>
yesterday · reply · retweet · favorite

Introducción a #Spring #Security 3.1:
<http://bit.ly/iHDmPi>
yesterday · reply · retweet · favorite

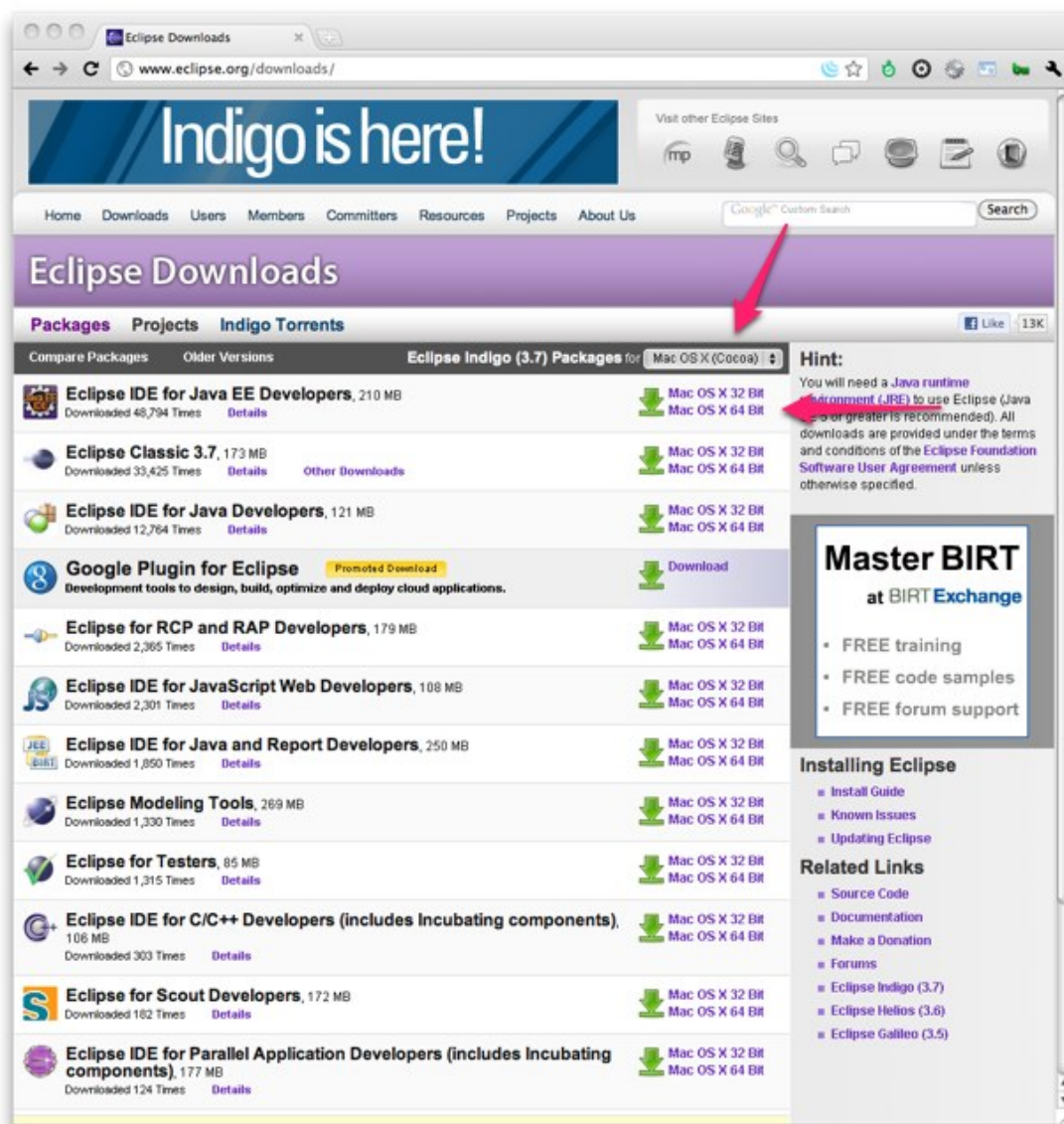
Implementando #SSO con #CAS: ejemplo práctico:
<http://bit.ly/IDRW4E>
yesterday · reply · retweet · favorite

Trabajando con GIT, introducción al uso de

 Join the conversation

Como podéis ver se trata de la página con el anuncio de la nueva versión. Es muy posible que cuando entréis vosotros ya hayan quitado el anuncio, pero no os preocupéis, ya que todo el proceso es igual.

Ahora iremos a la zona de descargas, podemos encontrar el enlace en la barra de menú de la parte superior (como se muestra en la imagen anterior).



Podemos ver que tenemos varias compilaciones distintas. Debemos elegir la que mejor se ajuste a nuestras necesidades. En mi caso es la de Eclipse IDE for Java EE Developers para trabajar con aplicaciones web. En esta misma página podemos seleccionar la plataforma para la que queremos el Eclipse y por último sólo queda pinchar sobre el enlace de descarga (en mi caso *Mac OS X 64 Bit*).

Algunas compilaciones son nuevas como *Google Plugin for Eclipse* que nos permite el desarrollo para GAE, o el *Eclipse for Testers* especialmente pensado para el desarrollo de pruebas.

De todas formas hay que tener en cuenta que al final lo que varía entre una compilación y otra es simplemente el número de plugins que lleva preinstalados, pero una vez instalado siempre podremos añadir o quitar los plugins que queramos.

3.2. Instalación de Eclipse Indigo

En la descarga nos habremos bajado un archivo *tar.gz*. Tendremos que descomprimirlo (en Mac es tan fácil como hacer doble click sobre el) y no aparecerá una carpeta llamada eclipse.

Esta carpeta la podemos copiar donde queramos, yo por ejemplo la voy a poner en */Applications* (en Mac es donde se suelen instalar las aplicaciones por defecto, aunque la podríamos haber puesto en cualquier otro directorio).

3.3. Tunning de memoria de Eclipse Indigo

Eclipse Indigo ya viene con una configuración de memoria bastante adecuada, pero teniendo en cuenta las capacidades de mi máquina a mi me gusta aumentar los márgenes un poco para no tener problemas cuando estoy trabajando con proyectos grandes.

Para ello editaremos el fichero */Applications/eclipse/Eclipse.app/Contents/MacOS/eclipse.ini*, y cambiaremos los valores **-XX:MaxPermSize** y el **-Xmx**. El fichero quedará algo como:

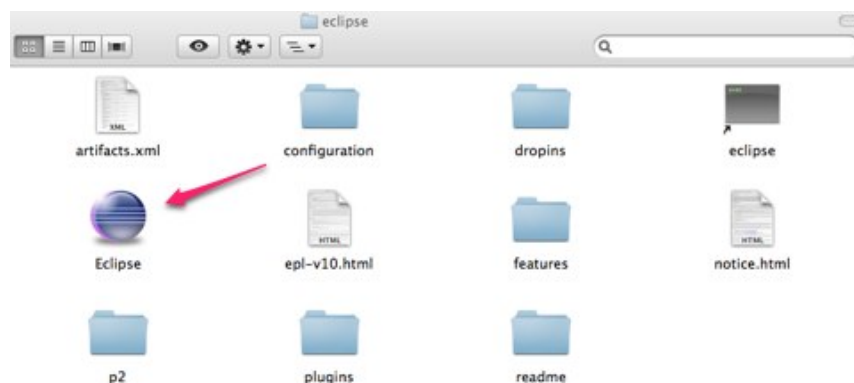

```

1 -startup
2 ../../../../plugins/org.eclipse.equinox.launcher_1.2.0.v20110502.jar
3 --launcher.library
4 ../../../../plugins/org.eclipse.equinox.launcher.cocoa.macosx.x86_64_1.1.100.v20110502
5 -product
6 org.eclipse.epp.package.jee.product
7 --launcher.defaultAction
8 openFile
9 -showsplash
10 org.eclipse.platform
11 --launcher.XXMaxPermSize
12 512m
13 --launcher.defaultAction
14 openFile
15 -vmargs
16 -Dosgi.requiredJavaVersion=1.5
17 -XstartOnFirstThread
18 -Dorg.eclipse.swt.internal.carbon.smallFonts
19 -XX:MaxPermSize=512m
20 -Xms40m
21 -Xmx2048m
22 -Xdock:icon=../Resources/Eclipse.icns
23 -XstartOnFirstThread
24 -Dorg.eclipse.swt.internal.carbon.smallFonts

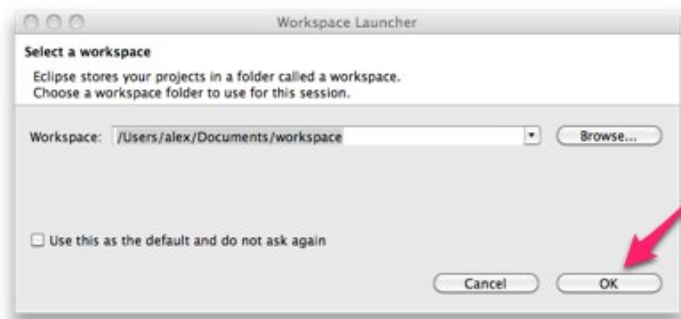
```

4. Ejecutar Eclipse Indigo

Para ejecutarlo basta con entrar en la carpeta `/Applications/eclipse`, y hacer doble click sobre el icono de *Eclipse*.



Como siempre, nos pedirá el workspace donde queremos trabajar.



Damos al botón ok, y nos aparecerá la ventana de bienvenida.



Ya estamos listos para trabajar.

5. Instalando m2e (Maven Integration for Eclipse, o m2eclipse)

m2e es un plugin que nos permite integrar Maven dentro de Eclipse, de forma que podamos trabajar fácilmente con proyectos Java gestionados por Maven.

La instalación la podemos hacer por el Marketplace, o de la forma tradicional proporcionando nosotros la URL. Vamos a elegir este segundo método para tener un poco más de control sobre lo que vamos a instalar.

En la [página de documentación de m2e](#) nos indica dos URL de instalación, una para la releases oficiales, y otra para las builds estables de desarrollo. Vamos a escoger la primera ya que la segunda parece no funcionar.

[Page](#) [Discussion](#) [View source](#) [History](#) [Edit](#)

Maven Integration

Maven Integration for Eclipse [provides tight integration for Maven \[into the IDE and providing the following features:\]\(#\)](#)

- Launching Maven builds from within Eclipse
- Dependency management for Eclipse build path based on Maven's pom.xml
- Resolving Maven dependencies from the Eclipse workspace without installing to local Maven repository
- Automatic downloading of the required dependencies from the remote Maven repositories
- Wizards for creating new Maven projects, pom.xml and to enable Maven support on plain Java project
- Quick search for dependencies in Maven remote repositories
- Quick fixes in the Java editor for looking up required dependencies/jars by the class or package name

Project is originally known as m2eclipse, which [been developed at the Codehaus \[and currently is being moved to Eclipse.\]\(#\)](#)

You can install last m2eclipse release by using the following update site from within Eclipse:

<http://download.eclipse.org/technology/m2e/releases>

Stable m2eclipse development builds can be installed from separate update:

<http://download.eclipse.org/technology/m2e/updates/N/>

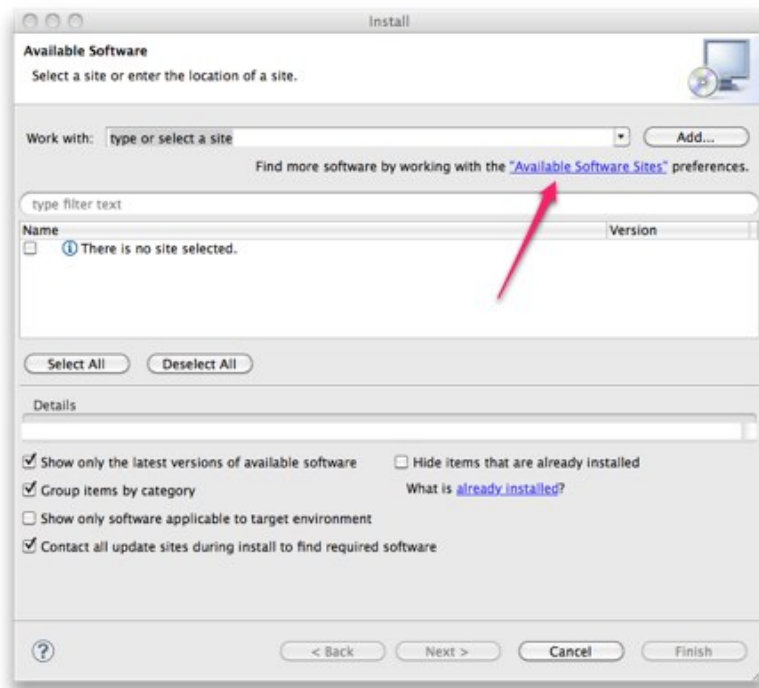
Additional information about project will be posted here and to [Eclipse project web site \[.\]\(#\)](#)

M2E Extension Development

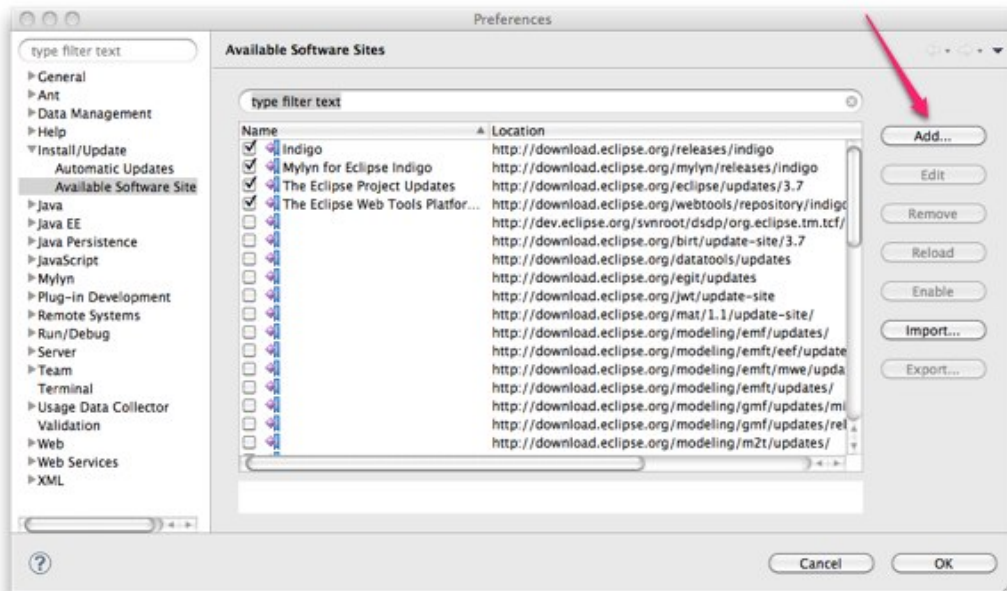
Category: M2E

Así que la URL que vamos a usar es: **<http://download.eclipse.org/technology/m2e/releases>**

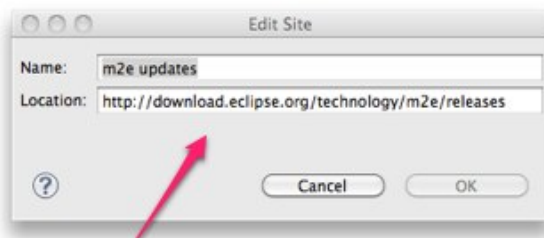
Para hacer la instalación seleccionamos en el menú de *Eclipse Help --> Install New Software...*



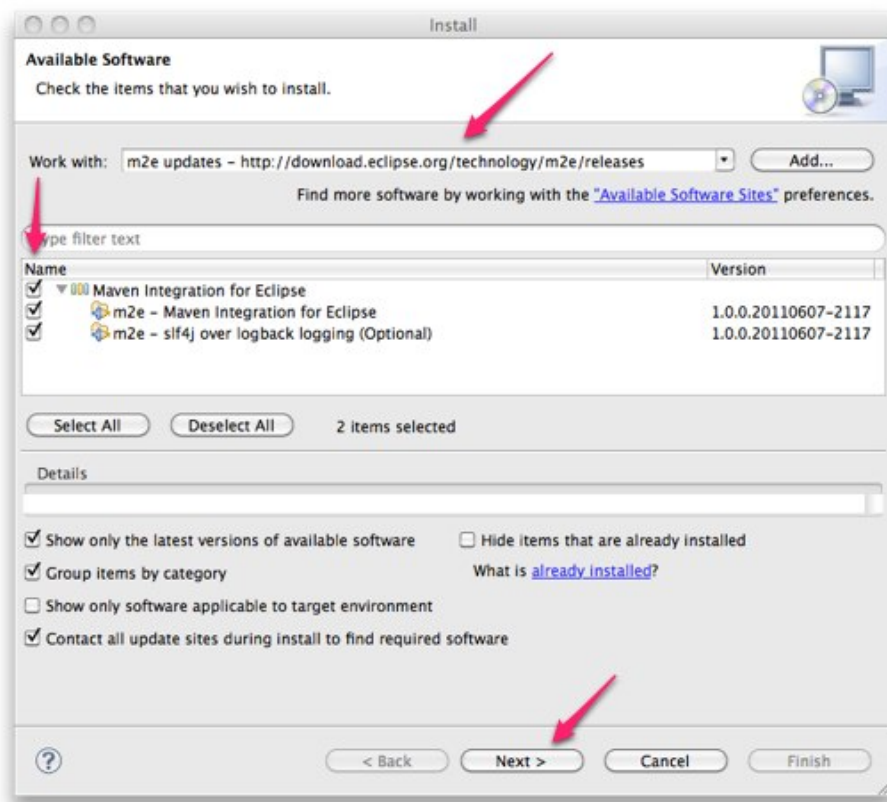
Pulsamos sobre "Available Software Sites"



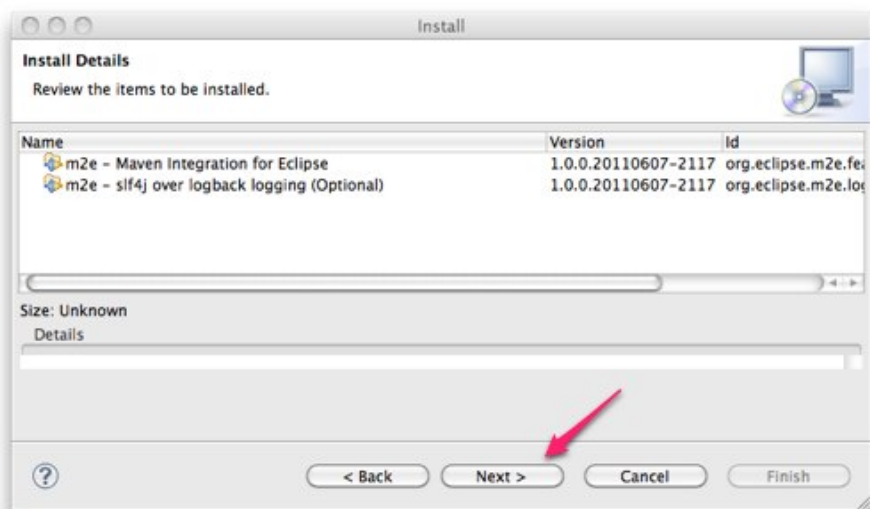
Ahora sobre el botón *Add...*, y añadimos la URL que hemos visto antes.



Damos al botón *Ok*, volviendo a la ventana anterior con el listado de todas las URL, debería aparecer al que acabamos de añadir, así que volvemos a dar sobre el botón *Ok*. Volviendo a la ventana de instalación, donde seleccionamos el sitio que acabamos de añadir en al pestaña de *Work with*:



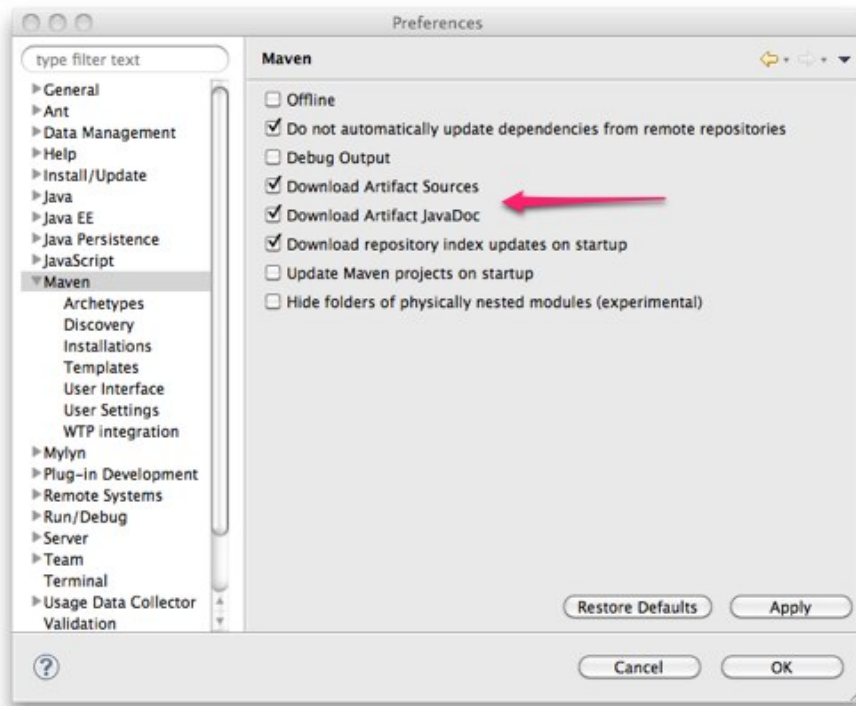
Vemos como nos ha aparecido el plugin de *Maven Integration for Eclipse*, lo marcamos tal como se ve en la imagen y pulsamos el botón *Next >*.



Volvemos a pulsar sobre *Next >*. Aceptamos la licencia, y otra ahora sobre el botón *Finish*. Comenzará la descarga e instalación del plugin. Ahora sólo queda esperar, y reiniciar el Eclipse cuando se nos indique.

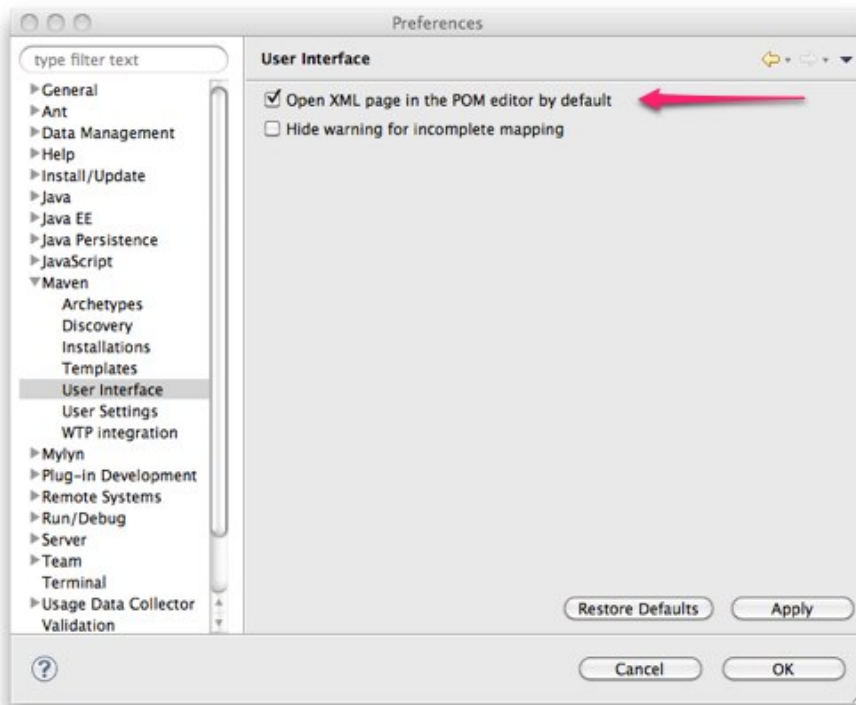
5.1. Configurando m2e

Para comprobar la instalación, podemos irnos a las opciones del Eclipse y deberíamos tener una categoría más llamada *Maven*.



Aquí a mi me gusta marcar las opciones de “Download Artifact Sources” y “Download Artifact Javadoc” para tener toda la información disponible mientras desarrollo (los fuentes vienen especialmente bien para disipar algunas dudas de como funcionan ciertas librerías).

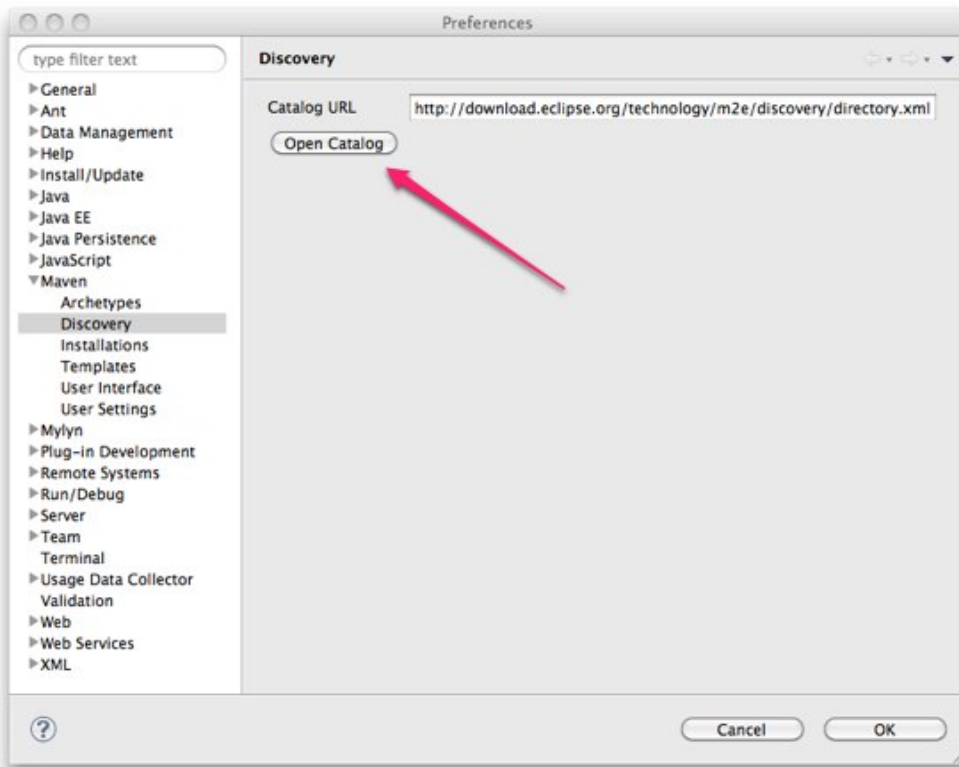
En la sección *Maven* --> *User Interface* me gusta marcar la opción “Open XML page in the POM editor by default”, ya que al final voy más rápido si toco el XML directamente que si tengo que ir por la interfaz visual.



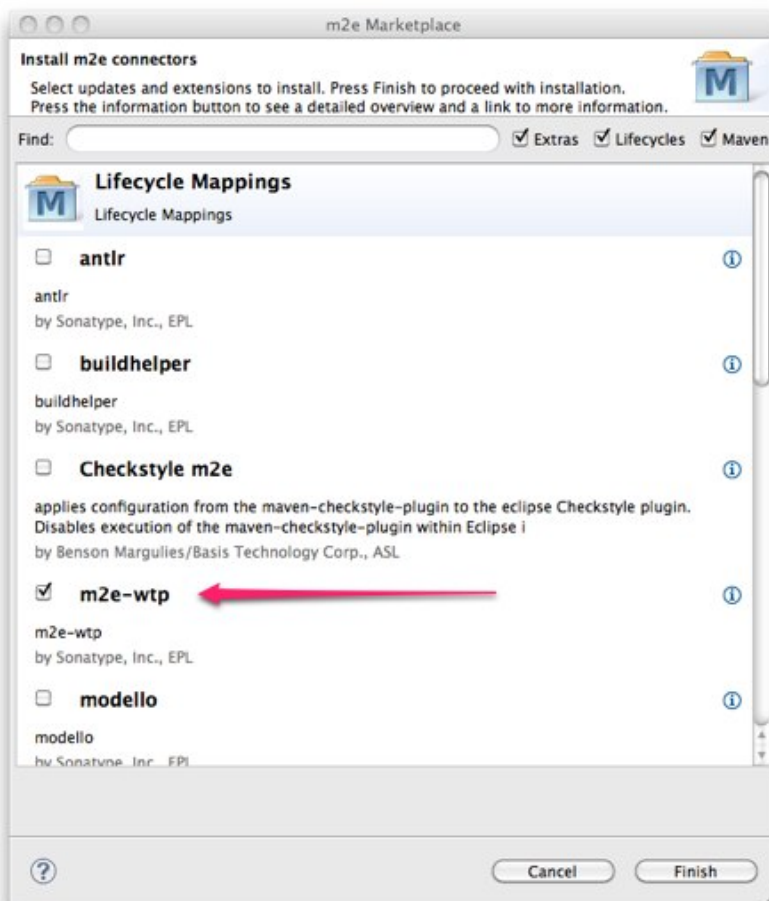
5.1.1. m2e integración con wtp

WTP es el plugin de Eclipse que permite desarrollar aplicaciones web. En este punto vamos a ver como instalar la integración de m2e con wtp de forma que al importar nuestros proyectos Maven, eclipse reconozca los proyectos de tipo “war” como proyectos web.

Para ello nos vamos a las opciones del Eclipse, a la sección *Maven* --> *Discovery*, y pulsamos sobre el botón *Open Catalog*.



Veremos como se nos abre una nueva ventana con varios conectores entre m2e y otros tantos plugins. En esta ventana debemos seleccionar **m2e-wtp**

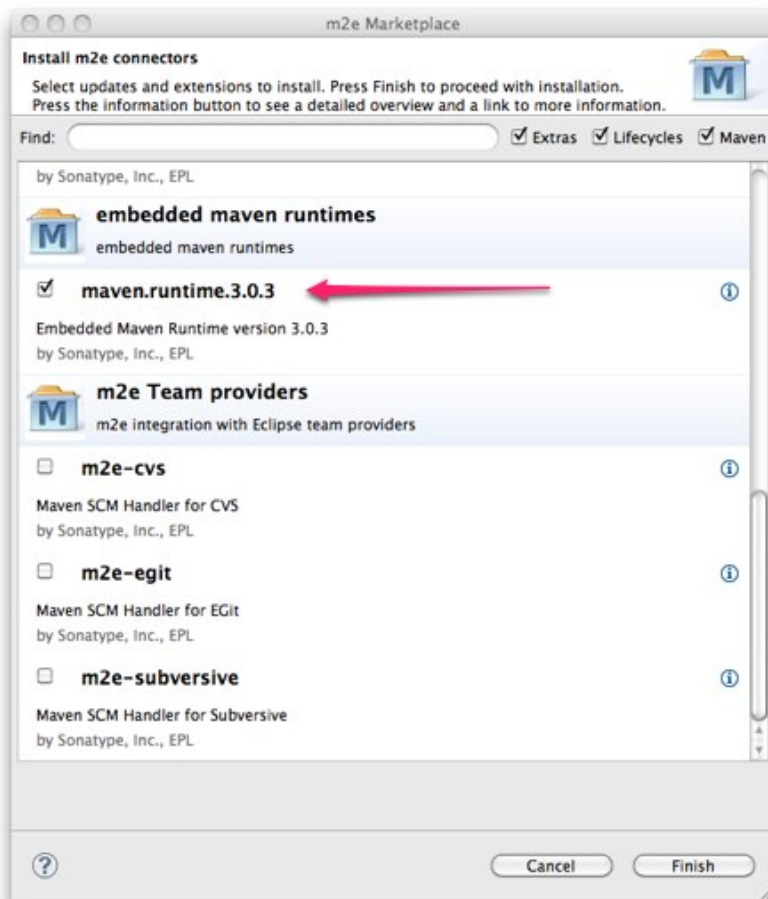


Pulsamos sobre el botón *Finish*. Comenzará el proceso de instalación y tan sólo tenemos que ir siguiendo las instrucciones (básicamente next, next, next, •)

5.1.2. Actualizar la versión de Maven que viene embebido con m2e

Por defecto m2e viene con la versión 3.0.2 de Maven, pero la podemos actualizar a la 3.0.3 que es la última que hay a día de hoy.

Para ello hacemos igual que antes y nos vamos a las opciones de Eclipse: *Maven --> Discovery --> Open Catalog*, y marcamos la opción *maven.runtime.30.3*.



Igual que antes pulsamos el botón *Finish* y seguimos el proceso de instalación.

Aunque yo casi os recomiendo que configuréis m2e para que use la versión de Maven que tengáis instalada en vuestro sistema (opciones del Eclipse, a la sección *Maven --> Installations*).

6. Conclusiones

Eclipse cada año gana en funcionalidad y en usuarios. Si bien en determinadas ocasiones hay algunos momentos de inestabilidad (recientemente se ha rumoreado mucho sobre como WTP ha ido perdiendo calidad con cada nueva versión), la verdad es que cada vez funciona mejor y se integra mejor con las herramientas ampliamente adoptadas por el sector como pueden ser Maven o GIT.

Os recomiendo que intentéis tener siempre vuestros entornos actualizados y "limpios" (tenéis que saber qué tenéis instalado y por qué, cómo está configurado y por qué). Revisar cada opción, explorar cada menú, aprender cada atajo de memoria. Si es la herramienta que usáis a diario más vale que la conozcáis bien para sacarle todo el provecho y que no os haga pasar malos ratos.

Y sobre todo aprender a trabajar también sin esa herramienta, porque nunca se sabe ;)

7. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software) y Certified ScrumMaster

Socio fundador de Autentia (Desarrollo de software, Consultoría, Formación)

<mailto:alejandropg@autentia.com>

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"

<http://www.autentia.com>

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

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

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

COMENTARIOS



franferri

2011-06-27 - 11:54:07

Como molan las flechas! :)



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

Copyright 2003-2011 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

