

**UNIVERSIDAD DE GRANADA**  
**E.T.S. DE INGENIERÍA INFORMÁTICA**



**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

## **Ingeniería de Sistemas de Información**

<http://decsai.ugr.es>

### **Instalación y configuración inicial de Google App Engine en Eclipse**

Curso 2015-2016

Tercer Curso del Grado Ingeniero en Informática

# Índice general

1.	Objetivos . . . . .	2
2.	Instalar Java SDK 7 y Eclipse JEE . . . . .	3
3.	Instalar el plugin de Google App Engine en Eclipse . . . . .	4
4.	Instalar el entorno Google App Engine SDK . . . . .	6
5.	Crear nuestra primera aplicación: Hola Mundo . . . . .	8
6.	Lanzando nuestra aplicación <i>HolaMundo</i> en un servidor local . . . . .	11

## 1. Objetivos

En este breve guión se describen los pasos necesarios para instalar y configurar un entorno de desarrollo funcional para las prácticas de la asignatura.

Para elaborar dichas prácticas se va a utilizar la suite Google App Engine<sup>1</sup>. Esta suite nos facilitará la creación de los diferentes componentes de las prácticas incrementales, así como su despliegue final.

En este guión se desarrollará el proceso para tener un entorno funcional para el lenguaje Java. La documentación oficial de la plataforma se puede obtener en <https://cloud.google.com/appengine/docs/java/>. Además, se plantea el uso de Eclipse como entorno de desarrollo, emparejado con el plugin liberado por Google para poder realizar desarrollos más ágiles<sup>2</sup>. Además, se considerará Windows como Sistema Operativo de desarrollo.

El alumno puede usar otros lenguajes de programación que soporte Google App Engine u otros Sistemas Operativos bajo su responsabilidad.

---

<sup>1</sup><https://cloud.google.com/appengine/docs>

<sup>2</sup><https://developers.google.com/eclipse/docs/download>

## 2. Instalar Java SDK 7 y Eclipse JEE

Una de las primeras cosas que debemos tener en cuenta es que Google App Engine recomienda encarecidamente trabajar usando la versión 7 de Java. Si no la tenemos instalada en nuestro sistema, es conveniente hacerlo<sup>3</sup>. Para ello, navegaremos a la dirección <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>.

Es importante descargarse el SDK dependiendo de tipo de Windows que se disponga (actualmente 64 bits es lo más habitual). En la Figura 1 se destaca en rojo las 2 versiones de Java para Windows que deben usarse.

Una vez instalado Java 7 SDK, el siguiente paso es instalar Eclipse en nuestro equipo. Eclipse ofrece múltiples versiones adaptadas a diferentes objetivos de desarrollo. En nuestro caso, buscamos la versión de Eclipse para el desarrollo de Java estándar. Podemos escoger descargar el instalador (Figura 2) o directamente “Eclipse IDE for Java EE Developers” (Figura 3). Si escogemos el instalador automático, deberemos usar la opción “Eclipse IDE for Java EE Developers” como se resalta en la Figura 4.

Java SE Development Kit 7u79		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	130.4 MB	jdk-7u79-linux-i586.rpm
Linux x86	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u79-linux-x64.rpm
Linux x64	146.4 MB	jdk-7u79-linux-x64.tar.gz
Mac OS X x64	196.89 MB	jdk-7u79-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	16.44 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows x86	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

Figura 1: Últimas versiones de Java 7 para Windows destacadas en rojo

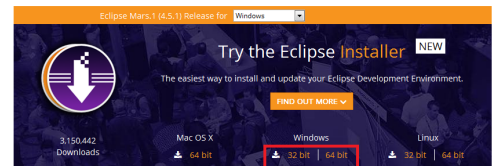


Figura 2: Instalador de Eclipse automático

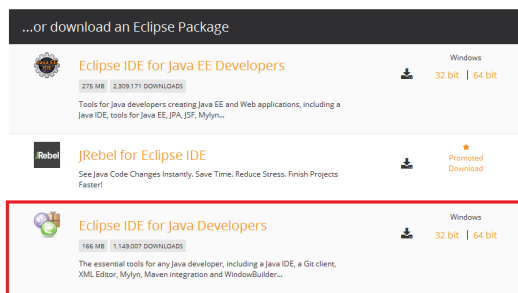


Figura 3: Eclipse para desarrollo en Java EE



Figura 4: Opción a escoger en el instalador automático de Eclipse

<sup>3</sup>En otro caso, deberemos asegurarnos que nuestros proyectos compilan Java con `target = 7`

### 3. Instalar el plugin de Google App Engine en Eclipse

En esta sección procedemos a instalar el plugin de Google App Engine en Eclipse. Nos permite tener de forma centralizada las tareas comunes de Google App Engine, como hacer *deploy* de nuestra aplicación o lanzarla en el servidor local. Además, también habilita Eclipse para configurar el SDK de Google App Engine, que veremos en la Sección siguiente.

El desarrollo de los plugins de Google App Engine se encuentra en la dirección <https://developers.google.com/eclipse/docs/download>. Debemos elegir la versión adecuada para nuestra versión de Eclipse. Tal y como se destaca en la Figura 5, deberemos usar y copiar la URL para la versión 4.4 de Eclipse (aunque tengamos instalada la versión 4.5 de Eclipse –Mars–).

#### Downloading and Installing the Plugin

To use the plugin you must be running **Java version 7** and a recent version of **Eclipse**. You can install the Google Plugin for Eclipse using the software update feature of Eclipse. Be sure to use the plugin that corresponds to your version of Eclipse. Follow the installation instructions provided at the links below or, if you are familiar with installing Eclipse plugins, you can simply paste the appropriate plugin link directly into Eclipse.

Eclipse version	Installation instructions	Direct plugin link
Eclipse 4.4 (Luna)	<a href="#">Plugin for Eclipse 4.4 (Luna)</a>	<a href="https://dl.google.com/eclipse/plugin/4.4">https://dl.google.com/eclipse/plugin/4.4</a>
Eclipse 4.3 (Kepler)	<a href="#">Plugin for Eclipse 4.3 (Kepler)</a>	<a href="https://dl.google.com/eclipse/plugin/4.3">https://dl.google.com/eclipse/plugin/4.3</a>
Eclipse 3.8/4.2 (Juno)	<a href="#">Plugin for Eclipse 3.8/4.2 (Juno)</a>	<a href="https://dl.google.com/eclipse/plugin/4.2">https://dl.google.com/eclipse/plugin/4.2</a>

Figura 5: URL para la instalación del plugin para Eclipse

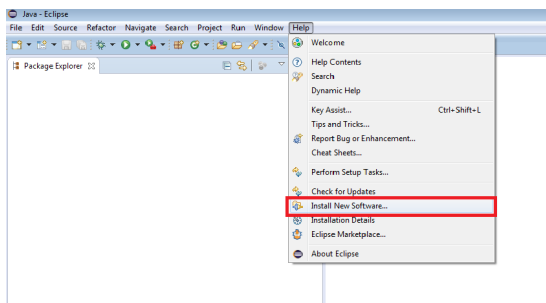


Figura 6: Opción para instalar nuevo software en Eclipse (plugins entre otros)

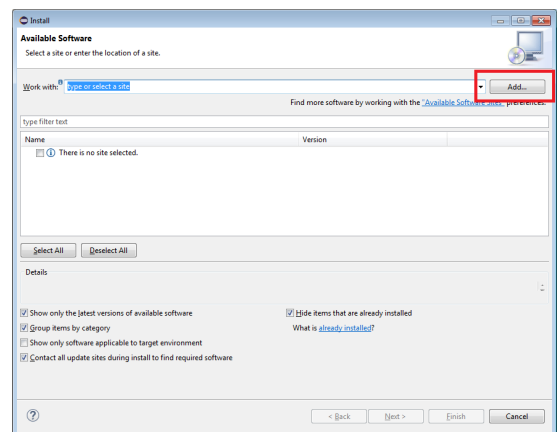


Figura 7: Añadir un nuevo repositorio de software a Eclipse

Para instalar el plugin, debemos hacer clic en la opción de “install new software” de Eclipse, tal y como se muestra en la Figura 6. En el cuadro de diálogo que nos aparece, hacemos clic en el botón “Add” para añadir el repositorio de software de Google App Engine, tal y como muestra la Figura 7.

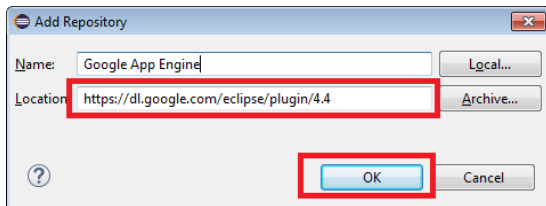


Figura 8: Añadir la URL de descarga

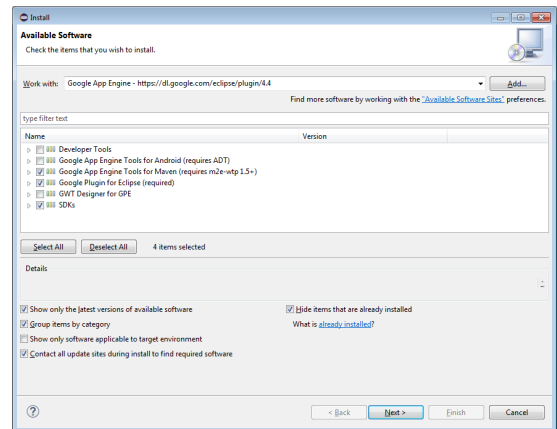


Figura 9: Software mínimo que se debe seleccionar

A continuación, pegamos la URL copiada desde la web mostrada en la Figura 5, tal y como se muestra en la Figura 8. Aceptamos las licencias de instalación de software y dejamos que el proceso termine. Es posible que aparezca alguna advertencia de software no firmado. La aceptamos para que continúe el proceso. Una vez finalizado, nos solicitará reiniciar Eclipse: aceptamos y ya estamos listos para configurar el SDK de Google App Engine.

## 4. Instalar el entorno Google App Engine SDK

Antes de poder crear nuestra primera aplicación, debemos descargar el Google App Engine SDK y configurar Eclipse para poder trabajar con él. Para ello, seguimos las instrucciones detalladas en <https://cloud.google.com/appengine/docs/java/webtoolsplatform>.

El primer paso es descargar el Google App Engine SDK de <https://cloud.google.com/appengine/downloads>. Este SDK es el que usará Eclipse (y todos los proyectos de Google App Engine en general) para poder crear el entorno de desarrollo y poder hacer el despliegue de nuestra aplicación en los servidores de Google una vez esté finalizada. Debemos seleccionar el SDK para Java [https://cloud.google.com/appengine/downloads#Google\\_App\\_Engine\\_SDK\\_for\\_Java](https://cloud.google.com/appengine/downloads#Google_App_Engine_SDK_for_Java), y descargar el fichero comprimido .zip con la última versión. En la Figura 10 se muestra el enlace de la versión disponible a fecha de escritura de este guión (versión 1.9.32).

Google App Engine SDK for Java

Version	Package	Size	SHA1 Checksum
1.9.32 - 2016-02-02	<a href="#">appengine-java-sdk-1.9.32.zip</a>	172.9 MB	e308024183cee85c427241bbf579bffd54286656

- ▶ Installing on Linux
- ▶ Installing on Mac OS X
- ▶ Installing on Windows

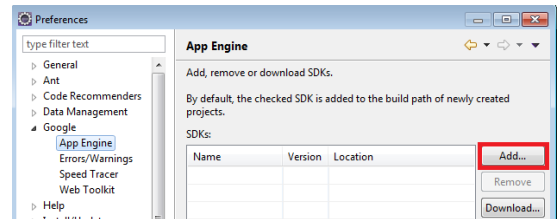


Figura 10: Descargar el Google App Engine SDK para java

Figura 11: Configuración de los Runtimes en Eclipse

Una vez descargado el fichero .zip, debemos descomprimirlo en algún directorio de nuestro computador. Es importante que se escoja una localización donde no se vaya a mover en el futuro. En nuestro caso lo descomprimiremos en *C:* para ilustrar el resto de ejemplos, aunque el alumno puede elegir cualquier otra ruta.

Cuando tengamos descomprimido el fichero con el Google App Engine SDK, debemos incluirlo en Eclipse. En el momento en el que incluyamos el SDK, podremos crear proyectos usando el SDK y el entorno de ejecución que incorpora. Para incorporar el SDK de Google App Engine hacemos clic en “Window → Preferences” y seleccionamos la categoría “Google → App Engine”. En este cuadro de diálogo, seleccionamos la opción “Add”, tal y como ilustra la Figura 11. Navegamos hasta la ruta donde descomprimimos el .zip con las SDK, lo seleccionamos y hacemos clic en “OK”, como se muestra en la Figura 12. Si

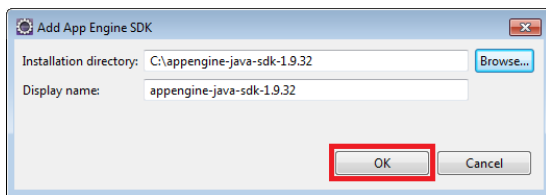


Figura 12: Localizar la ruta donde hemos descomprimido el Google App Engine SDK

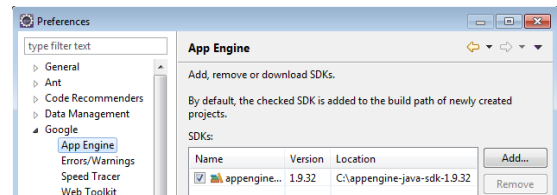


Figura 13: Configuración final del Google App Engine

hemos seguido estos pasos correctamente, deberemos tener configurado nuestro Runtime de forma similar a la Figura 13.



## 5. Crear nuestra primera aplicación: Hola Mundo

En el siguiente paso, vamos a compilar un pequeño ejemplo proporcionado por Google para crear una mini-aplicación que mostrará el mensaje “Hola Mundo” por pantalla. El ejemplo está tomado de <https://cloud.google.com/appengine/docs/java/> y adaptado a nuestro entorno de desarrollo.

El primer paso es crear una nueva aplicación “Dynamic Web Project” en Eclipse. Para ello, hacemos clic en “Archivo → Nuevo”, tal y como se muestra en la Figura 14.

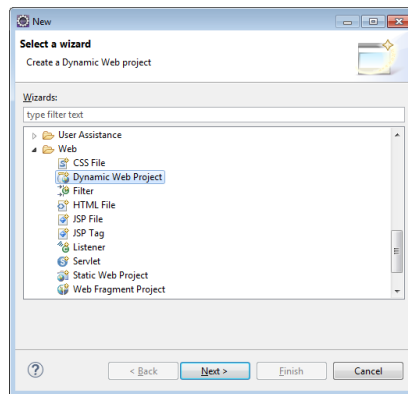


Figura 14: Crear un nuevo proyecto en Eclipse

En ese momento nos aparecerá un cuadro de diálogo donde se nos preguntará por los parámetros de nuestro proyecto. Debemos introducir un nombre (por ejemplo “HolaMundo”) y elegir un Runtime para nuestro proyecto. Para poder escoger un Runtime, haz clic en el botón “New Runtime” tal y como muestra la imagen de la Figura 15 y en el siguiente paso en “Next” (nos aseguramos de que está seleccionado “Google App Engine”) como se ilustra en la Figura 16. Veremos que deberá aparecer el Runtime que configuramos en la sección anterior (en el caso de esta guía, el Google App Engine versión 1.9.32). Hacemos clic en Finalizar (Figura 17. Nuestro asistente para crear un nuevo proyecto debería tener un aspecto similar al de la Figura 18. Hacemos clic en “Finish” y dejamos que Eclipse cree el proyecto.

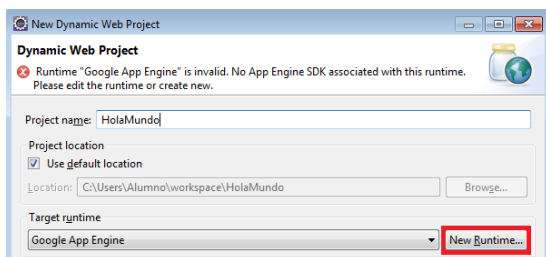


Figura 15: Elegir el Runtime para nuestro proyecto

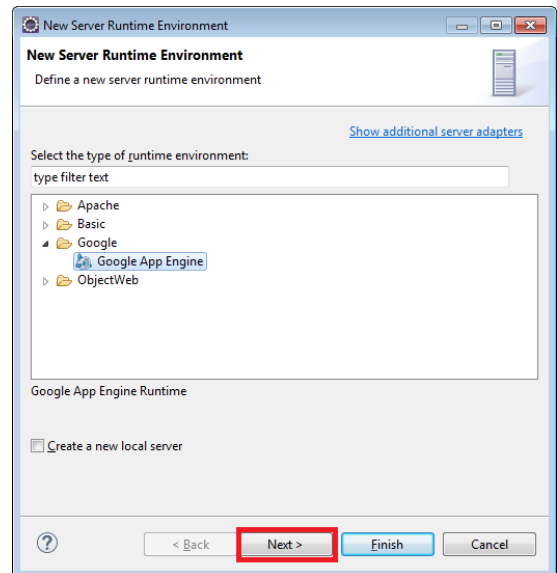


Figura 16: Elegir el Runtime para nuestro proyecto (2)

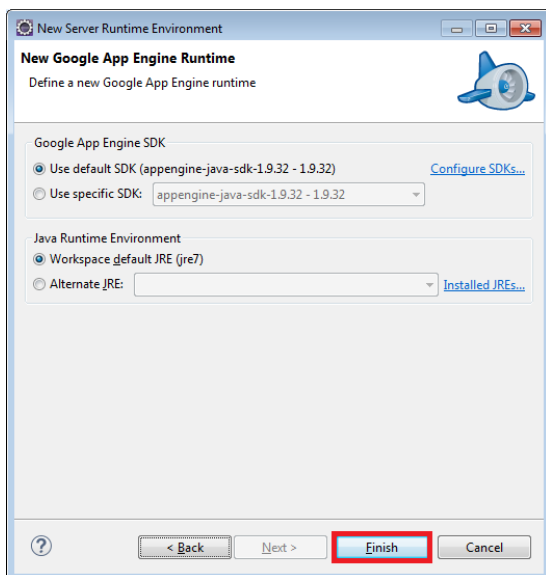


Figura 17: Elegir el Runtime para nuestro proyecto (y 3)

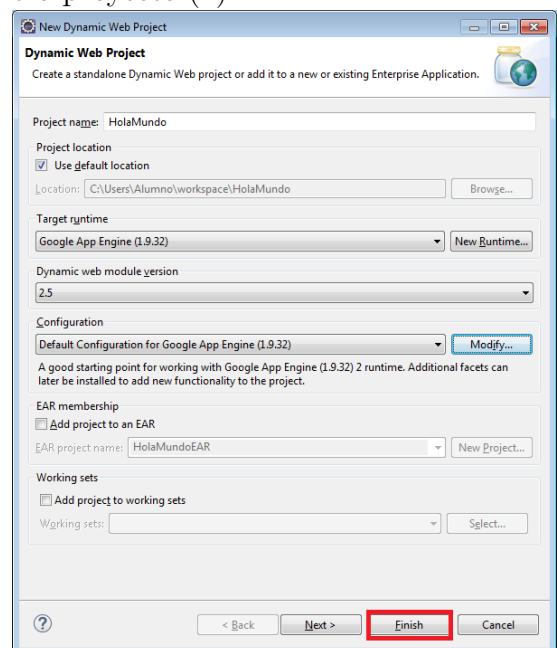


Figura 18: Elegir el Runtime para nuestro proyecto (Final)

Una vez Eclipse termine de crear el proyecto, deberemos tener una estructura de paquetes similar a la que aparece en la Figura 19. Ya estamos listos para lanzar nuestro proyecto en la Sección siguiente.

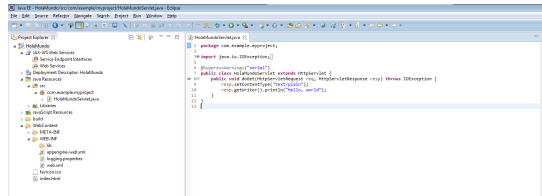


Figura 19: Proyecto HolaMundo terminado y listo para lanzar

## 6. Lanzando nuestra aplicación *HolaMundo* en un servidor local

El SDK de las Google App Engine trae un servidor local integrado, capaz de almacenar objetos y servir páginas web con diferentes tecnologías. El proyecto *HolaMundo* que acabamos de crear es un simple Servlet que nos mostrará el mensale “Hola Mundo” en nuestro navegador. Sin embargo, antes de poder ejecutar nuestro proyecto, debemos configurar nuestro servidor local para poder lanzar (cargar) nuestra aplicación en él. Gracias al plugin que hemos instalado de Eclipse es una tarea sencilla. Vamos a seguir los pasos que se indican en [https://cloud.google.com/appengine/docs/java/webtoolsplatform#running\\_the\\_project\\_on\\_the\\_server](https://cloud.google.com/appengine/docs/java/webtoolsplatform#running_the_project_on_the_server). Para ello, seleccionamos la pestaña “Servers” y hacemos clic en el enlace para invocar el asistente (Figura 20).

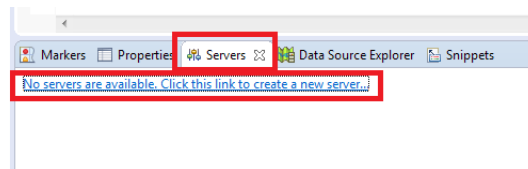


Figura 20: Crear un nuevo servidor para alojar nuestras aplicaciones

En el siguiente cuadro de diálogo (Figura 21) es importante que seleccionemos “Google App Engine” y la versión que tenemos instalada (en esta guía es 1.9.32). A continuación elegimos los parámetros para nuestro servidor (podemos cambiar el puerto por ejemplo), y hacer clic en “Finish” como se indica en la Figura 22.

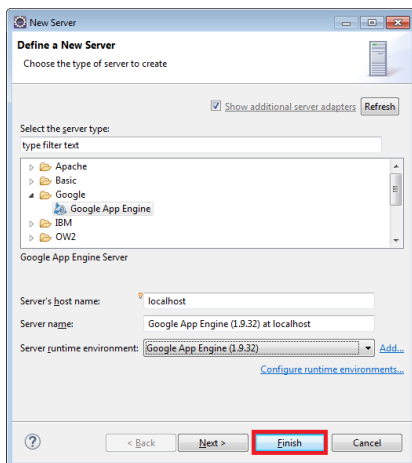


Figura 21: Seleccionar Google App Engine para crear un servidor

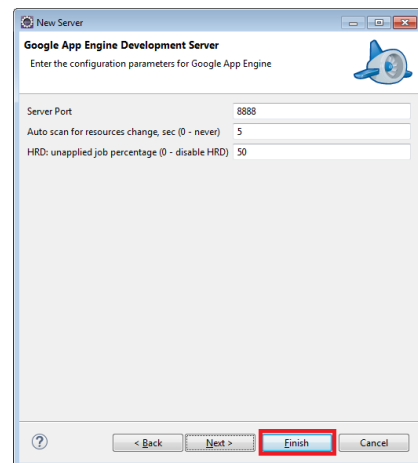


Figura 22: Parámetros del servidor

En este momento ya tenemos nuestro servidor listo. Está parado (“Stopped”) a la espera de alojar alguna aplicación que nosotros lancemos. Para lanzar nuestra aplicación, debemos simplemente hacer clic con el botón derecho en nuestro proyecto *HolaMundo* y elegir la opción “Run As → Run on Server” como se muestra en la Figura 23. A continuación debemos elegir un servidor para alojar nuestra aplicación (aquí también podemos crear un servidor nuevo si no hemos hecho los pasos de las Figuras 20 a 22). Por defecto nos aparecerá el servidor que acabamos de crear hace un momento. Lo seleccionamos (si no lo está) y hacemos clic en “Finish” tal y como nos muestra la Figura 24. También podemos marcar la casilla resaltada para que el proyecto se lance siempre en este servidor (y ahorrarnos este cuadro de diálogo cada vez que lo lancemos).

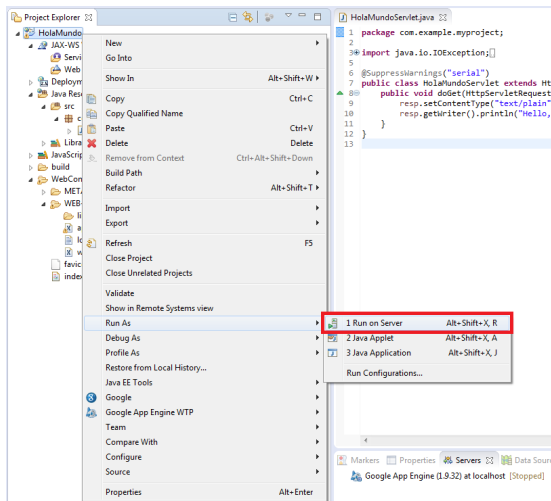


Figura 23: Lanzando nuestro proyecto *HolaMundo* en el servidor

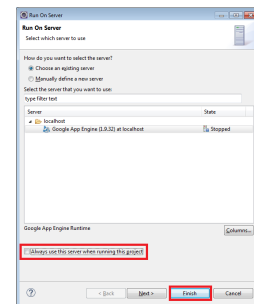
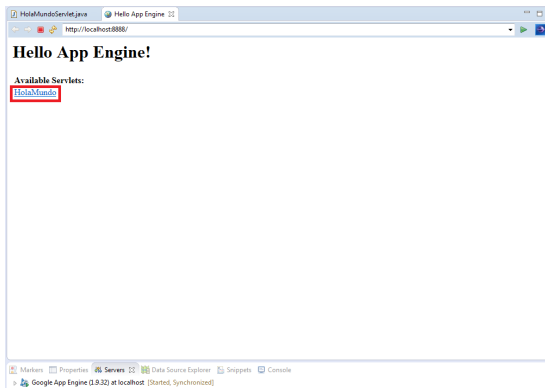
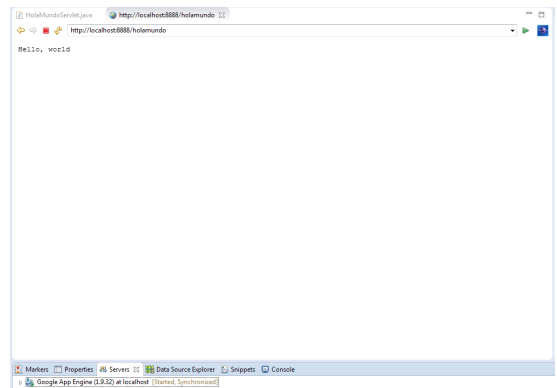


Figura 24: Elegimos el servidor que hemos creado antes para alojar nuestra aplicación

Figura 25: *HolaMundo* ejecutándoseFigura 26: Servlet de *HolaMundo*

En el momento que hemos hecho clic en “Finish”, el proyecto se compila y se aloja en el servidor local. A continuación se lanza, mostrándonos el contenido del *index.html* en un navegador local a Eclipse. Podemos hacer clic en el enlace HolaMundo para hacer funcionar nuestro como un Servlet, tal y como se destaca en la Figura 25. Si hacemos clic, se mostrará el mensaje que estaba en nuestro código Java para el Servlet: “Hello, world”. Date cuenta también de que el estado del servidor ha cambiado de “Stopped” a “Started, Synchronized”.

Podemos hacer cambios en el String del fichero HolaMundoServlet.java y ver que efectivamente nuestro proyecto funciona adecuadamente respecto a los cambios. Antes de que los cambios tengan efecto y poder relanzar nuestro proyecto, debemos parar el servidor. Para ello, haz clic en el botón de *stop* rojo de Eclipse después de seleccionar el servidor. Una vez hechos los cambios, solo deberás lanzar el proyecto otra vez al servidor para que anote los cambios y se vuelva a cargar



Figura 27: Deteniendo el servidor para poder hacer cambios en el proyecto