

CONTENIDO

PREFACIO	XI
CAPÍTULO 1. INTRODUCCIÓN	1
Historia.....	2
Java 8	4
Java y JavaScript.....	4
Máquina virtual de Java	5
Descargar Java	7
Entorno de desarrollo	10
Creación de un “Hola mundo”	12
Comentarios.....	24
CAPÍTULO 2. VARIABLES Y TIPOS DE DATOS.....	27
Variables	27
Tipos de datos.....	28
Declarar variables en Java.....	31
Declarar constantes	32

CAPÍTULO 3. OPERADORES EN JAVA.....	33
Operadores matemáticos en Java.....	33
Operadores de asignación.....	34
Operadores de incremento y decremento.....	35
Operadores relacionales	37
Operadores lógicos	38
CAPÍTULO 4. CONDICIONALES	43
If	43
If anidados.....	46
Operador ternario	47
CAPÍTULO 5. BUCLES Y SELECCIÓN.....	49
For	50
While.....	52
Do while	54
Break y continue	55
Switch.....	59
CAPÍTULO 6. BASES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS	63
Introducción.....	63
Clases	64
Atributos	65
Métodos.....	67
Crear una clase en Java	68
Crear atributos en Java	70
Crear métodos en Java.....	71

CAPÍTULO 7. MODIFICADORES DE ACCESO	75
Ejemplo del uso de modificadores de acceso	77
CAPÍTULO 8. PROGRAMACIÓN ORIENTADA A OBJETOS Y OTROS CONCEPTOS..	81
Creación de un objeto.....	81
Constructor	84
Sobrecarga de métodos	89
This.....	92
Recolección de basura	93
Instrucción import.....	94
Herencia.....	97
Sobrescribir métodos y Super	103
Clases con atributo final.....	105
Notación de camello en la programación orientada a objetos	107
Cast	109
CAPÍTULO 9. ÁMBITOS	111
Ámbito de clase.....	111
Ámbito de método.....	112
Ámbito de bloque	115
CAPÍTULO 10. CLASES ABSTRACTAS E INTERFACES.....	117
Clases	117
Clase abstracta.....	118
Interfaz.....	121
Herencia versus interfaces.....	124

CAPÍTULO 11. POJO	127
CAPÍTULO 12. ARREGLOS.....	131
Crear un arreglo en Java	132
Los arreglos como objetos	133
Inicialización del arreglo.....	133
Índice o subíndice	134
Arreglos bidimensionales.....	134
Recorrer un arreglo bidimensional	136
For each	138
CAPÍTULO 13. EXCEPCIONES.....	141
Try y catch	143
Mecanismos para el manejo de excepción	144
Throw y throws	147
Finally.....	149
Volumen de excepciones	152
CAPÍTULO 14. CADENAS	153
Método concat()	156
Método compareTo().....	156
Método equals().....	158
Método endsWith().....	159
Método toLowerCase()	159
Método toUpperCase.....	160
Método valueOf().....	161
Método replace()	162

Método getChars()	162
Método length()	164
Método trim	164
CAPÍTULO 15. JAVA EN LA INDUSTRIA DEL SOFTWARE	167
Sobre Java	167
Java JRE y Java JDK	168
Java EE y Java SE	168
JavaFX	169
CAPÍTULO 16. ENTREVISTA DE TRABAJO	171
Descripción general del proceso	171
Mandar el CV	171
Redes sociales de empleo	172
Plataformas de código	175
Eventos tecnológicos	176
Bolsa de trabajo	177
Primer contacto con la empresa	178
Entrevista técnica	179
Programar en la entrevista	180
Algoritmos recurrentes en las entrevistas de trabajo	182
Algoritmos específicos	183
Preguntas a un desarrollador en Java	184
APÉNDICE. CERTIFICACIONES EN JAVA	197
ÍNDICE ANALÍTICO	201

PREFACIO

Es algo muy normal que hoy día al hablar de tecnología tengamos algunas palabras en la mente, como por ejemplo: cambio, velocidad, software, programas, aplicaciones.

Existen siempre el paradigma de la actualización y la velocidad de cambio cuando hablamos de tecnología, de hecho es “anormal” que en algunos casos el tiempo pase y no afecte o modifique de manera constante algo en cualquier industria tecnológica.

Java es un lenguaje de programación que pone a prueba muchas cosas que acabamos de mencionar y que podemos decir que en sus casi 20 años su evolución se ha dado de una manera paulatina incluso de forma “anti-natural” en comparación con toda la tecnología del siglo XXI.

Java es una tecnología muy peculiar, cualquier programador que conoce varios lenguajes de programación, sabe que Java tiene una forma “diferente” de abordar problemas, esto incluye también algunos retos para el desarrollador y la implementación de soluciones; además posee una gran motivación como la comunidad de desarrolladores, Java cuenta con una de las más grandes en el mundo del desarrollo, una gran cantidad de fuentes de trabajo para este lenguaje y además uno de los lenguajes mejor pagados en la industria del software.

Estas son solo algunas de las razones por las que un desarrollador debería darse la oportunidad de conocer Java y poder aprovechar tanto todas las ventajas que presenta este lenguaje, como además afrontar los retos del desarrollo que son maravillosos para cualquier programador.

JAVA DESDE CERO...

El libro aborda un tema muy importante para el desarrollador como el de la entrevista de trabajo, se hace énfasis en los conceptos y problemas a los que se enfrentará y que no siempre se está preparado, ya que eso no se enseña en el aula, se adquiere con experiencia y con algunas experiencias no del todo agradables.

Acerca del autor

El autor de este libro es un apasionado de la tecnología y la docencia. Comenzó en el mundo de la programación con el lenguaje BASIC a los 13 años de edad. Colaborador habitual de comunidades en español en temas como: desarrollo de videojuegos, programación orientada a objetos, desarrollo web y dispositivos móviles.

José Dimas Luján Castillo nació en 1986, tiene el grado de Maestría en Tecnologías de Información. En la docencia ha colaborado con más de 14 universidades a nivel presencial en Latinoamérica en los niveles de Licenciatura y Maestría. En la educación en línea es colaborador de las plataformas más importantes a nivel mundial con más de 90 cursos en línea en la actualidad; además de ser conferenciante habitual de eventos tecnológicos apoyando siempre la adopción de nuevas tecnologías.

Para que el lector pueda consultar y contactar con el autor, puede localizarlo en redes sociales con el alias `josedlujan`, twitter, Facebook, entre otras. En su web: www.josedlujan.com o por correo electrónico: `josedlujan@gmail.com`.

Agradecimientos

Este libro es el resultado de años de experiencia trabajando con una de las tecnologías a las que más cariño le tengo, Java.

En este camino apasionante de la tecnología he tenido muchos amigos y compañeros de historias y batallas, siempre conté con el cariño de mi familia y de mi novia, la comprensión de ellos en muchas de las actividades y tiempos siempre fue de gran apoyo.

Este libro se lo quiero dedicar con mucho cariño y de una manera muy especial a otros compañeros de viajes e historias de mi vida que son mis mascotas. A las cuales les agradezco compartir de forma parcial o completa su tiempo a mi lado en el que espero siempre haber cumplido con mi máximo esfuerzo por que tengan una vida digna, con cariño y calidad de atención. Así que este libro va con cariño para Mu, Chelsea, Rocko, Vaca, Dalvik, Pascal, Pato, Gorda, Botas, Gordo y muchos otros compañeros que tuve anterior a estos y que no por no colocarlos son menos importantes, todos los que he tenido me han marcado a lo largo de mi vida, siempre tendré un gran recuerdo de todos ellos y les estaré agradecido por todo.

1 INTRODUCCIÓN

En el año 2016 hablar de la industria del desarrollo de software incluye hablar de Java en mayor o menor medida dependiendo de las necesidades del proyecto, pero nunca está de más analizar si se consigue desarrollar la plataforma utilizando este lenguaje de programación o también combinándolo con algunas tecnologías para cubrir algunas debilidades y potencializar la plataforma que se quiere desarrollar.

Podría citar muchas encuestas en las que Java aparece normalmente en el top 3 de estas listas:

- Lenguajes con más demanda laboral.
- Lenguajes mejores pagados.
- Lenguajes con más plataformas desarrolladas.

Normalmente me gusta dejar la investigación de estos temas por cuenta del lector, estudiante o aprendiz, para que descubra más datos y logre sacar sus propias conclusiones para formar un criterio y opinión del mismo tema.

Los programadores normalmente tienen una opinión sobre Java, son amantes del desarrollo en este lenguaje o por el contrario quieren estar lo más lejos posible de él.

JAVA DESDE CERO...

Ya mencioné muchas razones por las cuales Java es un buen lenguaje para aprender, pero existen también aspectos que debemos de considerar en el momento de estudiarlo, por ejemplo un caso en el que yo no recomendaría aprender Java.

En relación con mi experiencia de 9 años impartiendo clases, más de 100 cursos de forma online y más de 200 de forma presencial, comparto lo siguiente: “Aprender Java como primer lenguaje de programación puede ser un gran error”.

La mayoría de los estudiantes que han intentado aprender por su cuenta y como primer lenguaje Java sufren bastante, cuando en realidad Java cuenta con muchas desventajas si lo colocamos en posición de “primer lenguaje”, así que recomendaría siempre aprenderlo o entrar en el mundo de Java cuando se tengan las bases de la programación estructurada, un paradigma que hoy día es soportado por casi cualquier lenguaje como Pascal, C, Python, PHP, etc.

Si eliges Java como primer lenguaje, puedes estar perdiendo la oportunidad de ver con buenos ojos un lenguaje muy interesante, te recomendaría no quitarte esa oportunidad tú mismo.

Historia

Java fue creado por un equipo de trabajo aunque la cabeza del equipo y la mente creativa más relevante en las ideas y conceptos que son la esencia de Java es James Gosling.

Java se creó desde un equipo de la compañía Sun Microsystem al que se le conocía como “Equipo verde” o “Green Team”, el número de miembros siempre varía dependiendo de la fuente, se dice que eran 12 personas más James dando un total de 13, en otras fuentes se menciona un grupo de 15 personas contando a Gosling, lo que sí se sabe es que él era el mentor del equipo y que Java nació gracias a ese equipo dentro de Sun.

Java es uno de los lenguajes con más mitos desde su creación, por ejemplo el nombre es uno de ellos, puedes encontrar fuentes en donde se afirma que es un

acrónimo de los nombres de los desarrolladores más involucrados en la creación del lenguaje, aunque ya se ha dado por desechada esa teoría, ya que dejaría al 80% del equipo fuera del nombre y no tendría sentido. Por otro lado se dice que es un acrónimo que usa el sarcasmo para darle nombre al lenguaje, porque en ese época los acrónimos eran muy recurrentes para designar tecnología y en el caso de Java, se estaban burlando de todos ellos con: “Just Another Vague Acronym” la traducción en español quedaría como: “Solo otro acrónimo ambiguo”.

La última teoría y en la que durante ciertas entrevistas se han hecho bromas con el mismo James Gosling es la siguiente: cerca del centro de trabajo existía una cafetería en la que asistían de forma frecuente los miembros para despejar la mente, en esa cafetería se servía un café conocido por el nombre de “Java” del cual el equipo de desarrollo era gran consumidor, de hecho si buscas Java en cualquier navegador hoy día podrás ver el logo, en él se confirmaría esta teoría ya que el logo de Java es una simple taza de café.

La versión 1.0 de Java vio la luz en el año 1996, como lo mencioné en el prefacio, Java nunca ha tenido como característica la velocidad de actualización y desde el principio fue así. Un año más tarde sale la versión 1.1, es decir, en el año 1997, en comparación con los lenguajes de programación y con la tecnología que normalmente a la semana o a los días realizan siempre actualizaciones para resolver ciertos problemas, en Java todo se desarrollaba de una manera “anormal”, no quiere decir que esto sea malo o bueno, creo que no podemos juzgar la historia por un simple hecho, pero algo que ya hemos visto hasta este momento es que Java y velocidad no van de la mano.

Hoy día Java pertenece a Oracle, una de las compañías más grandes de software, ya que hace algunos años la compró a Sun Microsystems. Así que no tendremos que preocuparnos mucho por la implementación aunque recomendaría siempre leer las letras pequeñas de la licencia, nunca está de más analizar sus posibilidades y alcances, y cómo puede afectar a nuestro proyecto.

Java 8

Es la última versión lanzada en el mes de marzo de 2014. Dentro de las actualizaciones o mejoras que cuenta esta última versión, una de las más llamativas fue el uso de programación funcional, algo que no se consideraba en un principio, se volvió una realidad.

Otras mejoras bastantes populares dentro de la versión 8 es el manejo de la API de fecha y hora. Los métodos de extensión virtual son algo poco mencionado cuando se habla de mejoras pero que debemos de considerar, finalmente la integración de Java con JavaScript se mejoró, gracias a la implementación del motor de JavaScript Nashorn.

A lo largo del curso estaremos trabajando con Java 8, así que puedes consultar sin ningún problema la documentación oficial que se encuentra en:

<https://docs.oracle.com/javase/8/docs/>

Java y JavaScript

Cuando se está iniciando en el mundo del desarrollo con Java algunas de las preguntas más comunes son:

- ¿Qué relación tienen Java y JavaScript?
- ¿Cuáles son las diferencias entre estos lenguajes?
- ¿Cómo saber cuál tenemos que elegir?
- ¿Cuál es el mejor lenguaje de estos dos?

Te podría responder todas ellas de una manera muy sencilla, Java y JavaScript son de las coincidencias más raras en el mundo de la informática, sus primeras 4 letras son iguales, pero son lo único que tienen de similar, en realidad ni se pueden comparar o no deberían, ya que tienen objetivos completamente diferentes.

Para ampliar mejor la respuesta permite que te dé un ejemplo: imagínate que le preguntamos a un piloto de avión cuál es la diferencia entre un avión y un patinete, algo así sucede con Java y JavaScript, lo único parecido entre los dos es que son lenguajes de programación.

Java es un lenguaje de programación que trabaja con el paradigma de programación orientada a objetos, también conocido como POO. JavaScript en sus inicios no tenía muchos conceptos de POO y además hoy día aún existe el debate si se puede considerar un lenguaje de programación orientado a objetos.

Por otro lado JavaScript es un lenguaje que funciona exclusivamente en el navegador, esto quiere decir que necesitamos Chrome, Firefox, Internet Explorer u otro navegador para ejecutarlo, mientras que con Java se utiliza una máquina virtual.

Como hemos visto en los párrafos anteriores los nombres de Java y de JavaScript son una gran coincidencia que puede llegar a confundir a cualquier desarrollador, pero leyendo un poco más sobre el tema se sale de dudas.

Máquina virtual de Java

Si se habla de Java se tiene que mencionar la máquina virtual de Java o en inglés conocida como la “JVM” “Java Virtual Machine”. La JVM es el principio que hizo popular a Java durante sus comienzos ya que fue de gran ayuda y soporte para el mundo de desarrollo.

El problema que vino a resolver la JVM es el siguiente: cuando un desarrollador hace un programa hay dos cosas que debe considerar a grandes rasgos:

1. Lenguaje de programación.
2. Sistema operativo para el programa.

Esto quiere decir que uno selecciona el lenguaje de programación con cuidado ya que no se cuenta con las mismas herramientas en todos los lenguajes de programación. En el punto 2 mencionamos que se debe de considerar el sistema operativo ya que cuando un programador programa en una computadora con el

JAVA DESDE CERO...

sistema operativo Windows se utilizan las librerías del sistema operativo, así que cuando quieres pasar ese mismo programa a un sistema operativo Apple o Linux se deben de reescribir líneas de código o peor aún, hacerlo de nuevo.

Este problema era una constante y uno de los principios de Java es ese, permitirle al desarrollador programar el código una vez y que este pueda ejecutarse en cualquier lado mientras se tenga una máquina virtual de Java.

Te lo explico de otro modo, con un ejemplo de idiomas. Una persona de España que habla español escribe una novela, si esa novela la manda a Alemania, a Italia o a Francia, simplemente esa novela no va a poder ser leída, tendríamos que hacer una traducción de esa novela; la máquina virtual es eso. La JVM es un intérprete del lenguaje de Java. Así que solamente necesitamos tener la JVM para Windows, para sistemas operativos de Apple, para Linux, para teléfonos, etc.

El programador se preocupa por programar y por compilar la aplicación para así generar un código. Este código es el que se recibe por la JVM de cada uno de los sistemas operativos y luego el JVM se encarga de interpretar y explicar todo a los sistemas operativos.

Este es uno de los motivos por los que Java en sus inicios era un lenguaje lento debido a la JVM, ya que tenía que hacer trabajo extra en comparación con todos los demás lenguajes, de hecho Java tiene una gran peculiaridad, es un lenguaje de programación que efectúa una compilación pero además realiza la interpretación por la JVM. La mayoría de los lenguajes de programación compilan o interpretan, en este caso Java hace las dos.

Hoy día la velocidad ya no se nota debido a que contamos con ordenadores diez veces más potentes que los de décadas anteriores, así que la desventaja más grande que se tenía como la velocidad se ha reducido bastante gracias a los avances de la tecnología, del procesamiento de cómputo y de la memoria.

Descargar Java

En principio todas las computadoras hoy día tienen Java, pero debemos de tener claro lo siguiente: una cosa es tener lo que necesitamos para ejecutar aplicaciones escritas en Java y otra es programar en Java. Dicho de otra forma, para programar en Java necesitas más cosas ya que no solo seremos consumidores de software escrito en Java, ahora vamos a generar código escrito en este lenguaje.

Java JDK

Java JDK significa en inglés “Java Development Kit”. Es decir, “Kit de desarrollo de Java”, eso es lo que vamos a tener que descargar para poder programar utilizando el lenguaje de programación en Java.

Para descargar Java lo que necesitamos es ir a un buscador y escribir:

“Java jdk download”


O también puedes escribir de forma directa en tu navegador:

www.oracle.com/technetwork/es/java/javase/downloads/

Ya que contamos con el enlace lo abrimos y vemos una web como la siguiente:


Resumen Descargas Documentación Comunidad Tecnología Formación

Java SE Downloads



DOWNLOAD ▾

Java Platform (JDK) 8u101 / 8u102



DOWNLOAD ▾

NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 8u101 / 8u102
Java SE 8u101 includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u102 is a patch-set update, including all of 8u101 plus additional features (described in the release notes).
[Learn more](#) ▶

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
 - JDK ReadMe
 - JRE ReadMe

JDK

DOWNLOAD ▾

Server JRE

DOWNLOAD ▾

JRE

DOWNLOAD ▾

Como puedes observar vienen dos opciones:

1. Java.
2. NetBeans.

En realidad las dos opciones son Java, la única diferencia es que la primera es únicamente Java y la segunda es Java + Netbeans. Entonces nos viene una pregunta:

¿Qué es Netbeans?

Netbeans es un IDE, es decir, una herramienta usada en la exposición durante la mayoría del desarrollo de software. En otras palabras, donde programamos y

siempre es de gran utilidad ya que contiene otras herramientas y otras maneras de agilizar y dar velocidad al desarrollo de plataformas o de software.

En nuestro caso vamos a seleccionar la primera, la que solamente dice “Java” es con el objetivo de que no utilizaremos como IDE a Netbeans, más adelante comentaré sobre el nuevo IDE que manejaremos.

Dicho esto ahora vamos a entrar a una web similar a esta:

Overview
Downloads
Documentation
Community
Technologies
Training

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u101 Checksum
JDK 8u102 Checksum

Java SE Development Kit 8u101

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement
 Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

JAVA DESDE CERO...

Como se puede apreciar en la imagen anterior, tenemos un apartado para aceptar la licencia, así que antes de descargar cualquier versión de Java lo primero que tenemos que hacer es aceptar la licencia, ya que así podrás descargar sin ningún problema la versión de Java que necesites.

Solo ten en cuenta que la versión X86 hace referencia a un procesador de 32 bits y una máquina de x64 hace referencia a un procesador de 64 bits, por lo tanto considera qué tipo de procesador tiene tu computadora antes de seleccionar una opción.

A continuación selecciona una, haz clic en el que más te conviene y listo, esperar a que se descargue el instalador.

Una vez descargado, haz doble clic al instalador en MAC o Windows y listo, en el caso de Linux depende de cómo seleccionaste manejar tus paquetes, eso sería todo.

Entorno de desarrollo

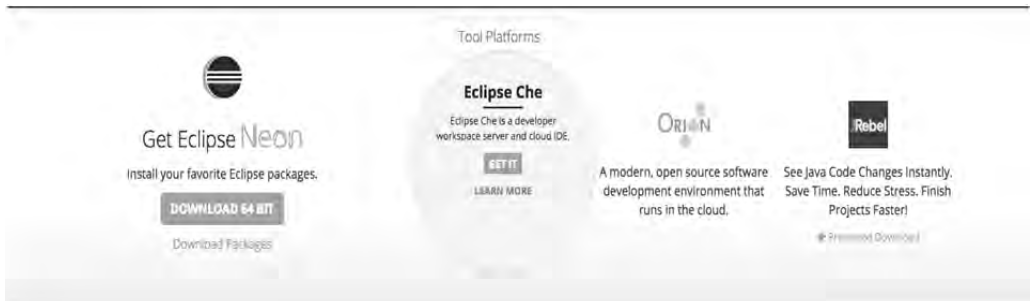
Es la herramienta que utilizamos para programar, esta normalmente viene con algunas funcionalidades y opciones que nos van ayudar a poder programar, por ejemplo: nosotros podemos escribir una tesis o un libro utilizando el bloc de notas, pero nadie lo hace. ¿Por qué? En realidad la mayoría usamos una herramienta que es un procesador de texto como Word, esto se debe a que Word nos facilita ciertas opciones u operaciones que nos van a permitir avanzar más rápido y mucho más sencillo. Lo mismo pasa con los IDE, son entornos de desarrollo que se adaptan a ciertas funcionalidades que nos van a ser de utilidad, en este caso vamos a usar el IDE conocido como ECLIPSE.

Para poder descargarlo accederemos a la web del proyecto:

<https://eclipse.org/downloads/>

Esta web siendo la misma muchos años, así que es muy probable que se mantenga con la misma ruta bastante tiempo. Por otro lado Eclipse es el IDE por excelencia en el desarrollo Java, con casi el 80% de los desarrolladores Java que lo utiliza, por eso es el que vamos a usar para hacer la mayoría de los ejercicios de este libro.

Es importante que sepas que Eclipse es un IDE en constante evolución y normalmente los nombres de las versiones van cambiando, así que por ejemplo lo que uno descarga durante este año el siguiente puede ser otra versión, pero no importa, lo significativo es que lo que descargues sea la última versión estable del IDE. En este caso la última lleva por nombre “Eclipse Neon” .

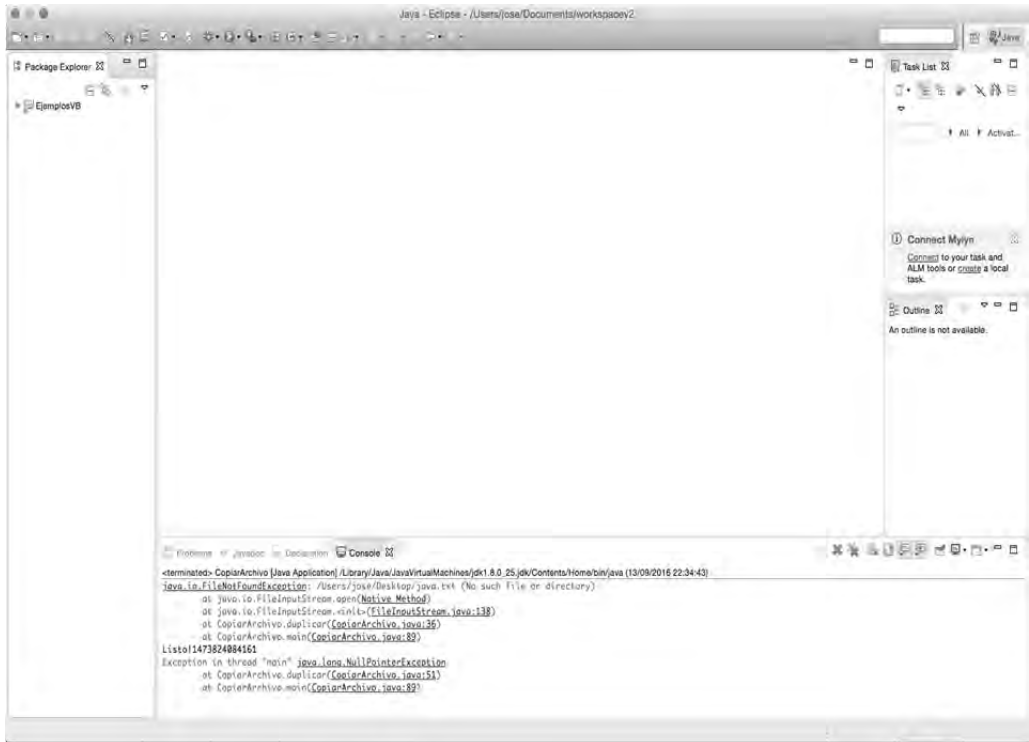


Sucede lo mismo que con el JAVA jdk, solamente se tiene que seleccionar el botón de descarga y listo, Eclipse hoy día es un proyecto que se puede utilizar sin tener que pagar por una licencia, es decir, que cualquier persona puede utilizarlo sin tener problemas.

Después de la descarga, hay que descomprimir el paquete y colocarlo en un directorio que no sea de fácil acceso, ya que se pueden perder los proyectos y el trabajo que ya ha realizado uno, así que te recomiendo que lo guardes bien.

Cuando abras Eclipse vas a ver una ventana como la siguiente:

JAVA DESDE CERO...



En mi caso tengo algunos proyectos o líneas de código, pero en cualquier versión nueva únicamente vas a ver los espacios en blanco que nos permiten escribir código y abrir proyectos.

Creación de un “Hola mundo”

Para establecer un clásico “Hola mundo” primero vamos a crear un proyecto.

File -> New -> Java Project

Después veremos una ventana como la siguiente:

