Hello world!

- Hello world!
- ¿Qué es Java?
- Nuestro primer programa
- Antes de comenzar...
- ... vamos allá
- Donde encontrar mas recursos
- Funciones útiles para el aula
- Código utilizado en los ejemplos
- Tarea

Copyright 2026

Hello world!

En este primer módulo vamos a conocer que es Java y sus características principales. Además vamos a crear nuestro primer programa (lo cual derivará en otras necesidades que veremos entonces) y vamos a conocer donde obtener mas información sobre Java para complementar este curso.

En el siguiente esquema elaborado por el CATEDU podemos ubicar el lenguaje Java como un lenguaje ideal para abarcar el curriculo relativo a programación en alumnado a partir de 15-16 años.

Copyright 2026

¿Qué es Java?

Java es un lenguaje de programación multiplataforma, de propósito general y orientado a objetos.

En una frase de apenas 14 palabras hemos introducido 4 conceptos:

- **Lenguaje de programación**
- **Multiplataforma**
- **Propósito general**
- **Orientado a objetos**

Vamos a recurrir a la wikipedia para obtener una definición de estos conceptos y tener así una primera aproximación a los mismos

> Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras.
>
> https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
>
> En informática, multiplataforma es un atributo conferido a programas informáticos o métodos y conceptos de cómputo que son implementados e interoperan en múltiples plataformas informáticas.
>
> https://es.wikipedia.org/wiki/Multiplataforma
>
> Los lenguajes de propósito general, son lenguajes que pueden ser usados para varios propósitos, acceso a bases de datos, comunicación entre computadoras, comunicación entre dispositivos, captura de datos, cálculos matemáticos, diseño de imágenes o páginas.
>
> https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general
>
> La programación orientada a objetos (POO, u OOP según sus siglas en inglés) es un paradigma de programación que viene a innovar la forma de obtener

> resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.
>
> https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos

Con todo lo que hemos visto hasta el momento **tenemos que quedarnos con la idea de que Java es** un lenguaje de programación que va a permitirnos trabajar en distintas plataformas (Windows, Linux, Android,...), con distintas intenciones (hacer un programa que calcule factoriales, acceder a bases de datos, interactuar con el hardware de nuestros dispositivos,...) y además nos va a permitir trabajar con objetos en lugar de hacer uso de la tradicional programación estructurada (que también podremos usar en Java).

Copyright 2026

Nuestro primer programa

Vista una primera introducción llega el momento de realizar nuestro primer programa.

Una vez se ejecute nuestro primer programa únicamente va a saludarnos por pantalla pero nos va a servir para hablar de cuestiones que hasta el momento hemos obviado y a su vez nos va a servir para configurar nuestro entorno de trabajo.

Copyright 2026

Antes de comenzar...

Anteriormente hemos comentado que Java es multiplataforma, es decir, un mismo programa puede ser ejecutado en varias arquitecturas o sistemas operativos sin mayor problema. Para conseguir esto es necesario aclarar algunas cuestiones:

- Cuando trabajamos con código java los ficheros que contienen este código tienen la extensión .java Estos ficheros se llaman ficheros fuente y contienen texto plano, es decir, podemos abrirlos con cualquier editor de texto y ver su contenido.
- Cuando un fichero .java es compilado se obtiene un fichero .class del mismo nombre. Si nuestro fichero fuente se llama Saluda.java su versión compilada se llamará Saluda.class y si tratamos de abrirlo con un editor de texto veremos que su contenido nos resulta ilegible

Resumiendo, si yo tengo un fichero fuente ¿qué debo hacer? compilarlo, y para ello necesitarás el kit de desarrollo java (JDK) que incluye el programa *javac* que te permite compilar código. Y entonces, cuando ejecutamos un programa escrito en Java ¿qué estamos ejecutando? estamos ejecutando la versión compilada, es decir, el fichero .class ¿y qué programa interpreta estos ficheros? Lo interpreta un programa llamado *java* que está contenido dentro de la máquina virtual java (JRE) (y que no hay que confundir con las máquinas virtuales que creamos con Virtual Box o similares)

Pregunta Verdadero-Falso

Si quiero poder compilar un fichero .class debo instalar en mi equipo el kit de desarrollo java (JDK)

VerdaderoFalsoLos ficheros que se compilan son los ficheros .java y se compilarían con el programa javac contenido dentro del JDK

Si quiero ejecutar un programa Java del cual me han facilitado su fichero .class en la máquina en la que quiero ejecutarlo deberé tener instalada la máquina virtual java (JRE)

VerdaderoFalso

Si quiero ejecutar un programa Java del cual me han facilitado su fichero .class en la máquina en la que quiero ejecutarlo deberé tener instalada la máquina virtual java (JRE)

VerdaderoFalsoLo que se ejecuta son ficheros .class y se hace con el programa java que está

dentro del JRE

Como trabajar en consola con comandos resulta tedioso lo que nosotros vamos a hacer para facilitarnos la vida va a ser trabajar con un entorno de desarrollo integrado (IDE) que nos va a aportar soporte para el lenguaje java evitándonos tener que hacer uso de la consola y facilitándonos el aprendizaje al reseñarnos los errores sintácticos que cometamos.

Para ello he elegido trabajar con el IDE Netbeans que es libre, gratuito y tiene una gran comunidad de gente detrás desarrollándolo y utilizándolo. Entre otras opciones tendríamos el trabajar con Eclipse o IntelliJ IDEA, además de utilizar un editor de texto plano como bloc de notas, notepad++ o Sublime Text y utilizar aparte el terminal.

Netbeans puede ser ejecutado en equipos con sistemas operativos Windows, Linux o Mac OS en sus diferentes arquitecturas de 32 o 64 bits. Para ahorrarnos el descargar por un lado JDK y por otro lado netbeans vamos a descargar e instalar una versión en la que ya viene todo. Para ello nos dirigiremos a http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html y descargaremos e instalaremos la versión que se corresponda a nuestro sistema operativo. A continuación he realizado 2 videotutoriales en los que muestro como proceder en Windows 7 y Ubuntu 16.04 LTS.

https://www.youtube.com/embed/DldsDzidscg?list=PL1ubUMkNBAR_98ka0Q393l3fpzSjo3FGq

https://www.youtube.com/embed/gm2ICqWz4P8?list=PL1ubUMkNBAR_98ka0Q393l3fpzSjo3FGq

Si queremos prescindir del uso de un entorno de desarrollo integrado podemos hacerlo, aunque yo no lo haría con mi alumnado excepto que trabajase con ellos en FP de la familia de Informática y comunicaciones. En el siguiente enlace tenemos un pequeño manual en el cual nos explican como hacer la compilación y ejecución a través del terminal (previamente habrá que haber instalado el JDK). No obstante a lo largo del curso haré uso de NetBeans en mis explicaciones y vídeos con el fin de facilitar la tarea a docentes y alumnado.

... vamos allá

Como hemos indicado con anterioridad en este curso vamos a utilizar NetBeans como entorno de desarrollo así que ha llegado el momento de familiarizarnos con él y crear nuestro primer programa. En el siguiente videotutorial se muestran aquellas funcionalidades de NetBeans que utilizaremos en el curso y crearemos nuestro primer programa. En este mismo capítulo, con posterioridad al vídeo, desgranaremos este programa.

https://www.youtube.com/embed/plwtjTiiXwo

Voy a extraer el código del programa realizado en el vídeo para comentarlo con mayor detalle. Voy a quitar los comentarios con el fin de facilitar la lectura del código y centrarnos en lo relevante en este momento.

```
1  package modulo1holamundo;
2  public class Modulo1HolaMundo {
3      public static void main(String[] args) {
4          System.out.println("Hola mundo");
5      }
6  }
```

La primera línea hace referencia al **paquete** al que pertenece nuestro fichero. Hablaremos de la organización del código en capítulos posteriores.

En la segunda línea **defino la clase** Modulo1HolaMundo con la palabra reservada class, es importante fijarse que tras la definición de la clase abrimos una llave { que cerramos en la línea 6 } Todo lo contenido entre esas llaves pertenece a la clase. No comento nada de la palabra reservada public ya que la veremos mas adelante pero si es importante reseñar que **el nombre de la clase debe coincidir con el nombre del fichero**. En un mismo fichero puede definirse mas de 1 clase pero solo 1 de ellas será pública.

En la tercera línea creo un **método** llamado main que pertenece a la clase Modulo1HolaMundo. En java para definir un método debemos definir el tipo que devuelve, el nombre y los argumentos que requiere. Posteriormente abrimos una llave { que cerramos en la línea 5 }, todo lo que se encuentra dentro de ambas llaves pertenece al método main En este caso el tipo que devolvemos es void (nada), el nombre de la función es main y tenemos un argumento llamado args de tipo String[]. Los modificadores public y static los dejamos para capítulos posteriores. El **método main es un método especial**, es el que se ejecutará en nuestro programa cuando iniciemos este. En caso de que nuestro proyecto tenga mas de un método main para configurar cual queremos que se

ejecute al iniciar el proyecto deberemos dirigirnos a las propiedades del proyecto (properties), seleccionar ejecución (Run) y allí establecer la clase principal (Main Class) que queremos que se ejecute.

En la cuarta línea hago una llamada al sistema y le digo que escriba por pantalla en una línea la frase Hola mundo. Es muy común al principio olvidar el ; del final de la línea.

Copyright 2026

Donde encontrar mas recursos

Hasta el momento hemos añadido enlaces a distintos recursos de utilidad para aquellas cuestiones que hemos ido tratando.

En este apartado el objetivo es realizar una pequeña recopilación de todos aquellos recursos que puedan resultarnos de utilidad y ampliación a curso. **NO es necesario descargar ni instalar nada de este listado** si has seguido mis instrucciones hasta la fecha.

- Página oficial de Java: https://www.java.com
- Academia Oracle: https://www.java.com/en/about/oracleacademy.jsp en inglés, incluye recursos de formación para estudiantes desde secundaria hasta el mundo de la empresa
- JRE en su versión 8: http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
- JDK en su versión 8: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- API Java 8: https://docs.oracle.com/javase/8/docs/api/
- Tutoriales Java: http://docs.oracle.com/javase/tutorial/
- Aprendiendo el lenguaje: http://docs.oracle.com/javase/tutorial/java/index.html
- Creación de una nterface gráfica: http://docs.oracle.com/javase/tutorial/uiswing/index.html

Copyright 2026

Funciones útiles para el aula

Cuando hemos creado nuestro primer programa hemos utilizado una función que muestra texto en una línea por pantalla. Voy a recoger en este apartado algunas funciones de la biblioteca de Java (API) que pueden resultarnos de utilidad para los programas que realicemos en el aula:

|Comando|Acción| |:--:|:--:| |System.out.println("Texto");|Escribe un texto y salta a la siguiente línea| |System.out.print("Texto");|Escribe un texto y permanece en la línea| |BufferedReader br = new BufferedReader(new InputStreamReader(System.in));int lado = Integer.parseInt(br.readLine());|Nos permite almacenar en una variable lo que introduzcan en consola| |Math.PI|Equivale a pi| |Math.E|Equivale a e| |Math.pow(base, exponente);|Eleva la base al exponente| |Math.sqrt(número);|Devuelve la raíz cuadrada positiva del número|

En los siguientes capítulos haremos haciendo uso de estas útiles funciones que procuraré ir introduciendo poco a poco pero me ha parecido interesante mostrar un pequeño adelanto.

Copyright 2026

Código utilizado en los ejemplos

Módulo 1 Proyecto Hola Mundo (zip - 17825 B).

Copyright 2026

Tarea

Tu tarea una vez acabado este primer módulo consiste en:

- Descargar e instalar Netbeans
- Crear un proyecto llamado Modulo1_Apellidos_Nombre es decir, si te llamas Pablo Ruiz Soria, tu proyecto se deberá llamar Modulo1_Ruiz_Soria_Pablo
- En el proyecto deberá haber un fichero llamado Saluda.java
- Saluda.java debe contener un método principal (main)
- Cuando se ejecute el método principal se deberán escribir por pantalla 2 líneas.
  - En la primera deberá poner tu nombre y apellidos
  - En la segunda línea deberá aparecer "He terminado el primer módulo"

Copyright 2026