



APRENDERAPROGRAMAR.COM

ORGANIZACIÓN DEL API DE
JAVA. PRINCIPALES CLASES
O LIBRERÍAS Y NOMBRES.
DIFERENCIAS ENTRE JAVA Y
JAVAX. (CU00646B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº46 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

ORGANIZACIÓN Y FORMA DE NOMBRAR LAS LIBRERÍAS DEL API DE JAVA

La biblioteca estándar de Java está compuesta por cientos de clases como *System*, *String*, *Scanner*, *ArrayList*, *HashMap*, etc. que nos permiten hacer casi cualquier cosa. Imagínate que quieres crear una lista de países donde ir añadiendo nombres de países y en un momento dado ordenarlos por orden alfabético.



O supón que quieres tener una lista de países relacionados con su continente (p. ej. México <-> América, España <-> Europa, Argentina <-> América, etc.). Para tener una primera idea de si el API de Java contendrá clases que nos puedan servir de ayuda, nos podemos hacer la pregunta: ¿habrán tenido muchos programadores necesidad de herramientas de este tipo? La respuesta es que obviamente sí. Ordenar, tener clasificado, hacer operaciones matemáticas, hacer búsquedas de texto, pedir datos al usuario y muchos más procesos son cuestiones que se repiten con frecuencia en programación, y por tanto se encontrarán resueltas en el API de Java. Por supuesto que podemos crear algoritmos propios para ordenar listas, pero lo más rápido y eficiente en general será usar las herramientas del API disponibles porque están desarrolladas por profesionales y han sido depuradas y optimizadas a lo largo de los años y versiones del lenguaje.

Saber usar la biblioteca y elegir las clases adecuadas es esencial para crear programas de forma rápida y eficiente. Es imposible conocer todas las clases y sus detalles (constructores, campos, métodos, etc.), ni siquiera sus nombres, pero gracias a internet lo tenemos todo “al alcance de la mano”.

Para programar en Java tendremos que recurrir continuamente a consultar la documentación del API de Java. Esta documentación está disponible en cds de libros y revistas especializadas o en internet tecleando en un buscador como yahoo, google o bing el texto “api java 8” (o “api java 6”, “api java 10” etc.) según la versión que estés utilizando. La documentación del API de Java en general es correcta y completa. Sin embargo, en casos excepcionales puede estar incompleta o contener erratas.

Cuando tengamos experiencia como programadores Java, posiblemente dispongamos de clases desarrolladas por nosotros mismos que utilicemos en distintos proyectos. En empresas grandes, es frecuente disponer de clases desarrolladas por compañeros de la empresa que usaremos de forma parecida a como se usa el API de Java: conociendo su interfaz pero no su implementación. Trabajar con una clase sin ver su código fuente requiere que exista una buena documentación que nos sirva de guía. Hablaremos de la documentación de las clases y proyectos en Java un poco más adelante. De momento, vamos a aprender a usar la documentación del API de Java.

En primer lugar, debemos tener una idea de cómo se organizan las clases del API. Esta organización es en forma de árbol jerárquico, como se ve en la figura “Esquema orientativo de la organización de librerías en el API de Java”. Esta figura trata de mostrar la organización del API de Java, pero no recoge todos los paquetes ni clases existentes que son muchos más y no cabrían ni en una ni en varias hojas.

Los nombres de las librerías responden a este esquema jerárquico y se basan en la notación de punto. Por ejemplo el nombre completo para la clase `ArrayList` sería `java.util.ArrayList`. Se permite el uso de `*` para nombrar a un conjunto de clases. Por ejemplo `java.util.*` hace referencia al conjunto de clases dentro del paquete `java.util`, donde tenemos `ArrayList`, `LinkedList` y otras clases.

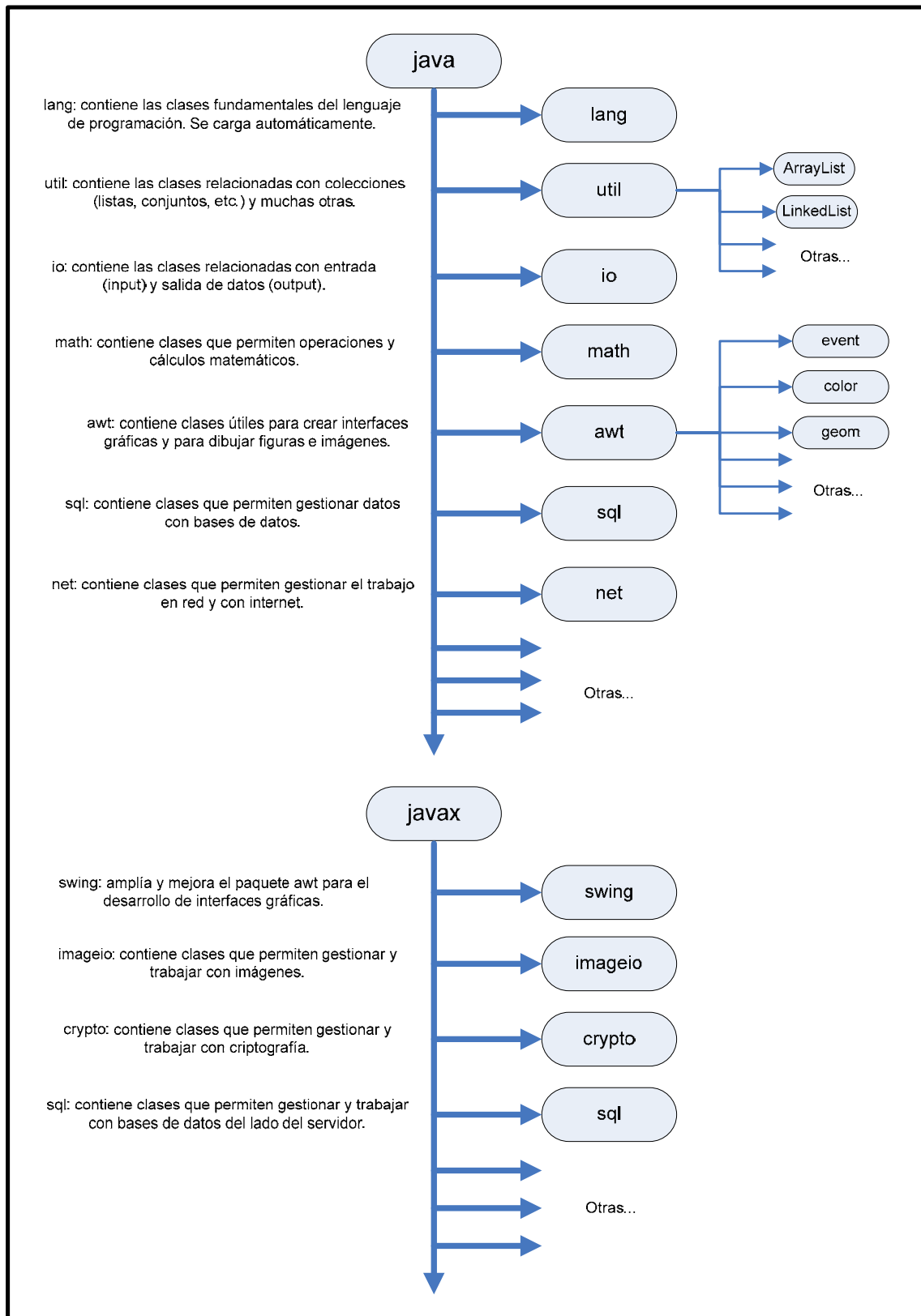
Para utilizar las librerías del API, existen dos situaciones:

- a) Hay **librerías o clases que se usan siempre** pues constituyen elementos fundamentales del lenguaje Java como la clase `String`. Esta clase, perteneciente al paquete `java.lang`, se puede utilizar directamente en cualquier programa Java ya que se carga automáticamente.
- b) Hay **librerías o clases que no siempre se usan**. Para usarlas dentro de nuestro código hemos de indicar que requerimos su carga mediante una sentencia `import` incluida en cabecera de clase. Por ejemplo `import java.util.ArrayList;` es una sentencia que incluida en cabecera de una clase nos permite usar la clase `ArrayList` del API de Java. Escribir `import java.util.*;` nos permitiría cargar todas las clases del paquete `java.util`. Algunos paquetes tienen decenas o cientos de clases. Por ello nosotros preferiremos en general especificar las clases antes que usar asteriscos ya que evita la carga en memoria de clases que no vamos a usar. Una clase importada se puede usar de la misma manera que si fuera una clase generada por nosotros: podemos crear objetos de esa clase y llamar a métodos para operar sobre esos objetos. Además cada clase tendrá uno o varios constructores.

Las librerías podemos decir que se organizan en ramas como si fueran las ramas de un árbol. Vamos a fijarnos en dos grandes ramas: la rama “java” y la rama “javax”. La rama java parte de los orígenes de Java, mientras que la rama javax es más moderna. Iremos conociendo poco a poco tanto la una como la otra.

Encontrar un listado de librerías o clases más usadas es una tarea casi imposible. **Cada programador, dependiendo de su actividad, utiliza ciertas librerías** que posiblemente no usen otros programadores. Los programadores más centrados en programación de escritorio usarán clases diferentes a las que usan programadores web o de gestión de bases de datos. Las clases y las librerías básicas deberás ir conociéndolas mediante cursos o textos de formación básica en Java. Las clases y librerías más avanzadas deberás utilizarlas y estudiarlas a medida que te vayan siendo necesarias para el desarrollo de aplicaciones, ya que su estudio completo es prácticamente imposible. Podemos citar clases de uso amplio. En el paquete `java.io`: clases `File`, `FileWriter`, `FileReader`, etc. En el paquete `java.lang`: clases `System`, `String`, `Thread`, etc. En el paquete `java.security`: clases que permiten implementar encriptación y seguridad. En el paquete `java.util`: clases `ArrayList`, `LinkedList`, `HashMap`, `HashSet`, `TreeSet`, `Date`, `Calendar`, `StringTokenizer`, `Random`, etc. En los paquetes `java.awt` y `javax.swing` una biblioteca gráfica: desarrollo de interfaces gráficas de usuario con ventanas, botones, etc.

Insistimos en una idea: no trates de memorizar la organización detallada del API de Java ni un listado de clases más usadas porque esto tiene poco sentido. Lo importante es que conozcas la forma de organización, cómo se estructuran y utilizan las clases y que aprendas a buscar información para encontrarla rápidamente cuando te sea necesaria.



Esquema orientativo de la organización de librerías en el API de Java

Próxima entrega: CU00647B

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188