**Java** es un lenguaje de programación de propósito general, concurrente y orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su objetivo es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "*write once, run anywhere*"), lo que quiere decir que el código puede escribirse una sola vez y ser ejecutado en cualquier tipo de dispositivos (PC, móvil, etc.).

# JAVA SE – versiones y distribuciones.

**J2SE o simplemente Java SE**: Java 2 Standard Edition o Java Standard Edition. Orientado al desarrollo de aplicaciones cliente / servidor. No incluye soporte a tecnologías para internet. Es la base para las otras distribuciones Java y es la plataforma que utilizaremos nosotros en este curso por ser la más utilizada.

**J2EE:** Java 2 Enterprise Edition. Orientado a empresas y a la integración entre sistemas. Incluye soporte a tecnologías para internet. Su base es J2SE.

**J2ME**: Java 2 Micro Edition. Orientado a pequeños dispositivos móviles (teléfonos, tabletas, etc.).

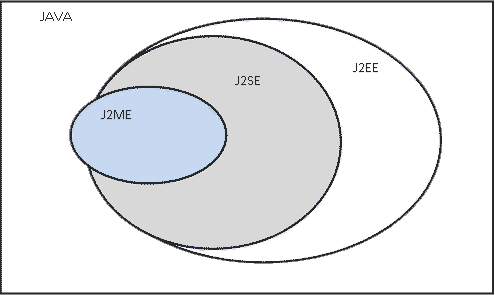
# API Java

La **API Java** es una [interfaz de programación de aplicaciones](https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones) (**API**, por sus siglas del [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s): *Application Programming Interface*) provista por los creadores del [lenguaje de programación Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java), que da a los programadores los medios para desarrollar aplicaciones Java.

Como el lenguaje Java es un [lenguaje orientado a objetos](https://es.wikipedia.org/wiki/Lenguaje_orientado_a_objetos), la API de Java provee de un conjunto de [clases](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)) utilitarias para efectuar toda clase de tareas necesarias dentro de un programa.

La API Java está organizada en paquetes lógicos, donde cada paquete contiene un conjunto de clases relacionadas semánticamente.

La **interfaz de programación de aplicaciones**, conocida también por la sigla **API**, en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s), *Application Programming Interface*,[1](https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones#cite_note-1)​ es un conjunto de [subrutinas](https://es.wikipedia.org/wiki/Subrutina), funciones y procedimientos (o [métodos](https://es.wikipedia.org/wiki/M%C3%A9todo_(inform%C3%A1tica)), en la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos)) que ofrece cierta [biblioteca](https://es.wikipedia.org/wiki/Biblioteca_(programaci%C3%B3n)) para ser utilizado por otro [*software*](https://es.wikipedia.org/wiki/Software) como una [capa de abstracción](https://es.wikipedia.org/wiki/Capa_de_abstracci%C3%B3n).



En esta imagen vemos, de forma orientativa, como J2EE “expande” a J2SE, mientras que J2ME “recorta” a J2SE al tiempo que tiene una fracción de contenido diferenciada exclusiva de J2ME. En realidad hablar de expansiones y recortes no es correcto, porque cada distribución es en sí misma distinta puesto que están concebidas con distintas finalidades. Por tanto no puede decirse que sean expansiones o recortes, pero de forma coloquial muchas veces se interpreta así.

# SCRIPT

Es una “secuencia de comandos” o “guion” que ayuda a ordenar o interpretar código.

# SDK o JDK

el kit de desarrollo de software (SDK) de JAVA (también llamado JDK)