

[Comience de forma gratuita](#)[Contáctese con nosotros](#)

[¿Qué es la informática en la nube?](#) / [Cloud Computing Concepts Hub](#) / [Herramientas para desarrolladores](#)

¿Qué es el entorno de ejecución de Java?

Obtenga más información sobre AWS para desarrolladores



[Explore las ofertas de machine learning gratuitas](#)

Construya, despliegue y ejecute aplicaciones de machine learning en la nube de forma gratuita



[Consulte los servicios de machine learning](#)

Innove más rápido con el conjunto más completo de servicios de IA y ML



[Examinar los cursos de machine learning](#)

Empezar con la formación para desarrolladores de machine learning con contenido creado por expertos de AWS



[Leer los blogs de machine learning](#)

Lea sobre las últimas novedades de productos de machine learning de AWS y las prácticas recomendadas

¿Qué es el entorno de ejecución de Java?

El entorno en tiempo de ejecución de Java (JRE) es un software que los programas de Java necesitan para ejecutarse correctamente. [Java](#) es un lenguaje de computación en el que se basan numerosas aplicaciones web y móviles actuales. El JRE es una tecnología subyacente que comunica el programa de Java con el sistema operativo. Actúa como traductor y facilitador, y brinda todos los recursos de modo que, una vez que escribe un software de Java, se ejecuta en cualquier sistema operativo sin necesidad de más modificaciones.

¿Por qué es importante el JRE?

Un programa de software necesita un entorno de tiempo de ejecución que proporcione acceso a la memoria y a otros recursos del sistema, como los archivos del programa y las dependencias. Antes, la mayor parte del software utilizaba el sistema operativo directamente como su entorno de tiempo de ejecución. Sin embargo, esto significaba que los desarrolladores tenían que escribir código diferente para cada sistema operativo en el que querían que se ejecutasen sus aplicaciones. La tecnología del entorno de tiempo de ejecución (JRE) de Java se creó como solución a este problema.

El entorno de ejecución de Java (JRE) es en realidad uno de los tres componentes de la plataforma de Java que se requieren para que cualquier programa de Java se ejecute correctamente. El kit de desarrollo de Java (JDK) y la máquina virtual de Java (JVM) son los otros dos componentes.

Java Development Kit

El kit de desarrollo de Java (JDK) es una colección de herramientas de software que se pueden utilizar para desarrollar aplicaciones de Java. Puede configurar el JDK en el entorno de desarrollo si lo descarga e instala. Seleccione la versión de software del JDK que coincida con la versión de Java que desee utilizar. Por ejemplo, la edición estándar de Java, o Java SE, requiere el kit de desarrollo de Java (JDK) de Java SE.

Máquina virtual de Java

La JVM es un software que ejecuta el programa de Java línea a línea. Los desarrolladores configuran los ajustes de la JVM para administrar recursos de programas cuando se ejecuta la aplicación de Java. Por ejemplo, puede configurar la configuración de memoria de la JVM y comprobar cuánta memoria interna consumen sus aplicaciones de Java durante el tiempo de ejecución.

Rol del JRE en el lenguaje de programación Java

El entorno de ejecución de Java (JRE) combina el código de Java creado. Para ello, utiliza el kit de desarrollo de Java (JDK) con código adicional incorporado que se denomina bibliotecas. A continuación, crea una instancia de JVM, o copia local, que finalmente ejecuta los programas de Java. Las JVM están disponibles para varios sistemas operativos, y el JRE genera una única copia de su código de Java que se ejecuta en todos los tipos de JVM. De esta manera, el JRE facilita la independencia de plataformas para aplicaciones de Java. Se pueden escribir una vez y ejecutar en cualquier lugar.

Diferencias entre el JRE, la JVM y el JDK

El kit de desarrollo de Java (JDK) es una capa de software situada sobre el entorno de tiempo de ejecución de Java (JRE) que contiene un compilador, un depurador y otras herramientas que se encuentran habitualmente en cualquier entorno de desarrollo de software. El código que se escribe en el JDK tiene una sintaxis similar a la lengua inglesa. El JDK lo compila y transfiere el código de bytes al JRE. Por el contrario, el JRE contiene bibliotecas de clases, que respaldan los archivos, y la JVM. Utiliza estos componentes de software para ejecutar el código intermedio en cualquier dispositivo.

¿Cómo funciona el JRE?

El entorno de tiempo de ejecución de Java (JRE) se ejecuta sobre el sistema operativo, lo que proporciona recursos adicionales específicos de Java. El Java Development Kit (JDK) y el JRE interactúan para crear un entorno de tiempo de ejecución sostenible que ejecute archivos de programas de Java en cualquier máquina. El entorno de tiempo de ejecución de Java (JRE) utiliza tres componentes básicos para funcionar.

ClassLoader

Las bibliotecas de clase Java contienen colecciones de código previamente escrito que se puede llamar cuando sea necesario. Dichas colecciones simplifican el trabajo de los desarrolladores de Java al brindar métodos integrados para tareas comunes y no triviales como recibir entradas de los usuarios, mostrar resultados a los usuarios y más. Todos los programas de Java hacen referencia a diferentes bibliotecas de clases. La herramienta Java ClassLoader carga dinámicamente todos los archivos de clase necesarios en la máquina virtual Java (JVM) bajo demanda.

Verificador de bytecode

El kit de desarrollo de Java (JDK) cuenta con un compilador que convierte el código similar al inglés que se escribe en una versión en lenguaje de máquina llamada código intermedio de Java. El verificador de bytecode en el JRE verifica el formato y la precisión del código de Java antes de cargarlo en la JVM. Por ejemplo, si el código viola la integridad del sistema o los derechos de acceso, el entorno de tiempo de ejecución de Java (JRE) no cargará el archivo de clase.

Intérprete

Una vez que el bytecode se carga correctamente, el intérprete de Java crea la instancia de la JVM que ejecuta el programa de Java en la máquina subyacente.

¿Cuáles son los componentes del JRE?

Además de los componentes principales, el entorno de ejecución de Java (JRE) contiene muchos otros componentes de software que ayudan a ejecutar programas de Java de manera más eficiente. A continuación se muestran algunos ejemplos:

Herramientas de desarrollo

El JRE contiene herramientas de desarrollo como kits de herramientas de interfaz de usuario que puede utilizar para mejorar la calidad de sus aplicaciones. A continuación se muestran algunos ejemplos:

Java 2D

Java 2D es una interfaz de programación de aplicaciones (API) que puede utilizar para diseñar gráficos en dos dimensiones y crear completas interfaces de usuario, videojuegos, animaciones y efectos especiales en lenguaje Java.

Swing

Swing es una interfaz de usuario gráfica (GUI) ligera que brinda personalizaciones flexibles y sencillas.

Abstract Window Toolkit

Abstract Window Toolkit (AWT) es una interfaz de usuario gráfica que puede utilizar para crear objetos de interfaz de usuario como botones, ventanas y barras de desplazamiento.

Soluciones de implementación

El JRE incluye tecnologías que simplifican el proceso de publicar cambios en el software para usuarios de aplicaciones. Estas tecnologías también brindan compatibilidad avanzada para actualizaciones en la aplicación. Las tecnologías de implementación como Java Web Start y el complemento de Java se incluyen como parte de la instalación del JRE. Dichas tecnologías simplifican la activación de aplicaciones y también ofrecen soporte avanzado para próximas actualizaciones del JRE. A continuación se muestran algunos ejemplos:

Java Web Start

Con Java Web Start puede lanzar aplicaciones repletas de características solamente con un clic desde su navegador web.

Complemento de Java

El complemento de Java establece una conexión entre navegadores populares y la plataforma de Java, de modo que pueda ejecutar sus applets para sitios web en un navegador de escritorio.

Bibliotecas de lenguaje y utilidades

Una colección de archivos de clase de Java se conoce como paquete en Java. El JRE incluye varios paquetes en Java para dar soporte al control de versiones, la administración y la supervisión. A continuación se muestran algunos ejemplos de paquetes:

Marco de colecciones

El marco de colecciones es una arquitectura unificada que incluye interfaces para mejorar el almacenamiento y el procesamiento de datos de aplicaciones.

API de preferencias

La API de preferencias permite a varios usuarios de la misma máquina definir su propio grupo de preferencias de aplicación.

Registro

Los paquetes de registro producen informes de registro para la resolución de incidentes como errores de seguridad, fallos de rendimiento y errores de configuración.

Java Archive

Java Archive (JAR) es un formato de archivo independiente de plataformas que le permite agrupar varios archivos para reducir el tamaño de los archivos de su aplicación y mejorar de manera notable la velocidad de descarga.

Librerías de integración

El JRE incluye un buen número de bibliotecas de integración que ayuda a los desarrolladores a crear conexiones de datos integrales entre sus servicios y aplicaciones. A continuación se muestran algunas bibliotecas de ejemplo:

Java IDL

Java Interface Definition Language (IDL), basado en la arquitectura Common Object Request Broker Architecture (CORBA), admite objetos de datos distribuidos, es decir, objetos que interactúan en diferentes plataformas dentro de una red. Por ejemplo, Java IDL permite que objetos escritos en Java interactúen con otros escritos en otro lenguaje, como C, C++ o COBOL.

Java Database Connectivity

Los desarrolladores utilizan la API de Java Database Connectivity (JDBC) para escribir aplicaciones que pueden acceder a bases de datos, hojas de cálculo y archivos remotos.

Java Naming and Directory Interface

Java Naming and Directory Interface (JNDI) es un servicio de directorio que permite a los clientes crear aplicaciones portátiles que obtienen información de bases de datos externas mediante reglas de asignación de nombres.

¿Qué es AWS SDK para Java?

El [kit de desarrollo de software \(SDK\) de AWS para Java](#) simplifica el uso de los servicios de AWS, ya que ofrece un conjunto de bibliotecas que resultan coherentes y conocidas para los desarrolladores que utilizan Java. Admite abstracciones de alto nivel para un desarrollo simplificado. Las bibliotecas de Java de código abierto centradas en AWS están disponibles junto con ejemplos de código y una guía de referencia de la API de Java.

Los ejemplos de código de [AWS SDK para Java 2.0](#) en GitHub contienen ejemplos de código de Java y casos de uso del mundo real para los servicios de AWS destinados a acelerar el desarrollo de las aplicaciones. Además, la [Guía de referencia de la API de AWS SDK para Java](#) describe las operaciones de la API para la versión más reciente de AWS SDK para Java. La guía de referencia también ofrece ejemplos de solicitudes, respuestas y errores de los protocolos de servicios web compatibles.

Comience a utilizar AWS SDK para Java con la [Guía para desarrolladores de AWS SDK para Java](#), o visite el [Centro para desarrolladores de Java de AWS](#).

Siguientes pasos en AWS

[Iniciar sesión en la consola](#)

Información sobre AWS

[¿Qué es AWS?](#)
[¿Qué es la informática en la nube?](#)
[Inclusión, diversidad e igualdad en AWS](#)
[¿Qué es DevOps?](#)
[¿Qué es un contenedor?](#)
[¿Qué es un lago de datos?](#)
[Seguridad en la nube de AWS](#)
[Novedades](#)
[Blogs](#)
[Notas de prensa](#)

Recursos para AWS

[Introducción](#)
[Formación técnica y certificación](#)
[Cartera de soluciones de AWS](#)
[Centro de arquitectura](#)
[Preguntas frecuentes sobre cuestiones técnicas y productos](#)
[Informes de analistas](#)
[Red de socios de AWS](#)

Desarrolladores en AWS

[Centro de desarrolladores](#)
[SDK y herramientas](#)
[.NET en AWS](#)
[Python en AWS](#)
[Java en AWS](#)
[PHP en AWS](#)
[JavaScript en AWS](#)

Ayuda

[Contáctenos](#)
[Oportunidades de trabajo en AWS](#)
[Abra un tique de soporte técnico](#)
[Centro de conocimientos](#)
[AWS re:Post](#)
[Información general de AWS Support](#)
[Legal](#)

[Crear una cuenta de AWS](#)



Amazon es un empleador que ofrece igualdad de oportunidades: *minorías, mujeres, discapacitados, veteranos, identidad de género, orientación sexual y edad.*

Idioma

[عربي](#) |
[Bahasa Indonesia](#) |
[Deutsch](#) |
[English](#) |
[Español](#) |
[Français](#) |
[Italiano](#) |
[Português](#) |
[Tiếng Việt](#) |
[Türkçe](#) |
[Русский](#) |
[ไทย](#) |
[日本語](#) |
[한국어](#) |
[中文 \(简体\)](#) |
[中文 \(繁體\)](#)

[Privacidad](#)

|

[Términos del sitio](#)

|

[Preferencias de cookies](#)

|

© 2022, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.