# ENTORNOS DE DESARROLLO - 1º DAM

# UD1. ELEMENTOS DEL DESARROLLO DE SOFTWARE

## 

## E03 - PREGUNTAS SOBRE EL APARTADO 4

|  |  |
| --- | --- |
| **Resultado de Aprendizaje 1**: Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento. | |
| **Criterio de evaluación** | **Ejercicio/s** |
| RA01.b Se han clasificado los lenguajes de programación, identificando las principales características de Java. | TODOS |

1. ¿Cómo se llama el creador de Java?
2. ¿A partir de qué lenguaje está creado Java?
3. ¿En qué fecha fue lanzado Java?
4. Enumera las razones que motivaron la creación de Java.
5. Enumera las fechas y el número de versión de Java desde la versión 1.0, liberada en el año 1995.
6. Enumera las características de Java.
7. ¿Por qué Java es simple?
8. ¿Qué significa que Java es independiente de la arquitectura? ¿Y portable?
9. ¿Por qué Java es compilado e interpretado?
10. ¿Qué es y para qué sirve la máquina virtual de Java (JVM)?

**Respuestas:**

1. James Gosling

2. A partir de Pascal y C++.

3. En otoño de 1995.

4.

1. Creciente necesidad de interfaces mucho más cómodas e intuitivas que los sistemas de ventanas que proliferaban hasta el momento.
2. Fiabilidad del código y facilidad de desarrollo. Gosling observó que muchas de las características que ofrecían C o C++ aumentaban de forma alarmante el gran coste de pruebas y depuración. Por ello en los sus ratos libres creó un lenguaje de programación donde intentaba solucionarlos fallos que encontraba en C++.
3. Enorme diversidad de controladores electrónicos. Los dispositivos electrónicos se controlan mediante la utilización de microprocesadores de bajo precio y reducidas prestaciones, que varían cada poco tiempo y que utilizan diversos conjuntos de instrucciones. Java permite escribir un código común para todos los dispositivos.

5.

1. 1995: Java1.0.
2. 1997: Java 1.1, JavaBeans y JDBC.
3. 1998: Java 1.2, Collections, Swing y Reflection.
4. 2000: Java 1.3, Performance y JNDI.
5. 2002: Java 1.4, Assertions, Security y Regular Expression.
6. 2004: Java 1.5, Generics, Anotations y Threading.
7. 2006: Java 1.6, Compiler API y Web Services.
8. 2011: Java 1.7, JVM, T/C, Generics.
9. 2014: Java 1.8, Data/Time, Lambda y Streams.

6.

* Lenguaje Simple
* Orientado a objetos
* Distribuido
* Compilado e interpretado
* Robusto
* Seguro
* Portable
* Multihilo
* Dinámico

7. Java posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir programas interesantes desde el principio. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, y dado que la mayoría de la gente los conoce, aunque sea de forma elemental, resulta muy fácil aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.

8. Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución tan variopintos, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura, diseñado para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete de Java.

La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas. Estas dos últimas características se conocen como la Máquina Virtual Java (JVM).

9. Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (runtime).

10. Es un software necesario para poder ejecutar cualquier aplicación escrita en java y es el encargado de traducir el bytecode en lenguaje máquina, ya adaptado a la máquina concreta sobre la que se está ejecutando. Por tanto, será necesario instalarla JVM apropiada al S.O. del dispositivo.