



CICLO: [DAM]
**MÓDULO DE [ENTORNOS DE
DESARROLLO]**

[Tarea N° 01]

Alumno:
[Juan Carlos Filter Martín]
[REDACTED]

Contenido

1. Documentos que se adjuntan a este informe.....	3
2. (RA1_e) Se han clasificado los lenguajes de programación.....	3
Describir con tus propias palabras el lenguaje de programación Java indicando, entre otras cosas, qué paradigma, generación y principales características.....	3
3. (RA1_d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.....	4
Describir la creación del código intermedio de Java.....	4
4. (RA1_g) Se han identificado las características y escenarios de uso de las metodologías ágiles de desarrollo de software.....	5
Comentar, al menos, 3 factores que se deben dar en un proyecto para que elijamos una metodología ágil para desarrollarlo.....	5

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.

2. (RA1_e) Se han clasificado los lenguajes de programación.

Describir con tus propias palabras el lenguaje de programación Java indicando, entre otras cosas, qué paradigma, generación y principales características.

Java es un lenguaje de programación orientado a objeto desarrollado por Sun Microsystems a principios de los años 90. Toma mucha de su sintaxis de C y C++, pero tiene un modelo de objeto más simple y elimina herramientas de bajo nivel, como la manipulación directa de punteros o memoria.

Las principales características que podemos encontrar en Java son:

- **Lenguaje simple:** Java posee una curva de aprendizaje muy rápida.
- **Se basa en el paradigma de la programación orientada a objetos:** Java fue diseñado como un lenguaje orientado a objeto desde el principio.
- **Es Distribuido:** Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- **Interpretado y compilado a la vez:** Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina. También diremos que es interpretado ya que el código objeto de Java se interpreta en tiempo real a través de la máquina virtual.
- **Robusto:** Java fue diseñado para crear software altamente fiable.
- **Seguro:** Dada la naturaleza distribuida de Java, donde las applets se bajan desde cualquier punto de la Red, la seguridad se impuso como una necesidad de vital importancia.
- **Indiferente a la arquitectura:** Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red.
- **Portable:** Los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, son iguales en todas las plataformas. Esto se conocen como la Máquina Virtual Java (JVM).

- **Alto rendimiento.**
- **Multiebra:** Java soporta la sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje.
- **Dinámico:** Las clases sólo se enlazan a medida que son necesitadas.
- **Applets:** Java puede ser usado para crear dos tipos de programas: Aplicaciones independientes y applets.
Los Applets son pequeños programas que aparecen embebidos en las páginas web, pero con la capacidad de ejecutar acciones muy complejas, como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo para luego emprender acciones...

3. (RA1_ d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.

Describir la creación del código intermedio de Java.

El código intermedio de Java se refiere generalmente al bytecode de Java (Conjunto de instrucciones de bajo nivel para representar programas escritos en lenguaje Java).

La creación del código intermedio de Java sigue varios pasos:

1. **Escritura del código fuente:** Comienza con la escritura del código fuente en lenguaje Java. Lenguaje de alto nivel para facilitar la programación orientada a objetos.
2. **Compilación:** Una vez que el código fuente se ha escrito, se utiliza un compilador de Java para traducir el código fuente en código bytecode de Java. Se realiza varias tareas como la verificación de tipos, la optimización del código y la generación del bytecode.
3. **Generación de archivos .class:** El resultado de la compilación es un conjunto de archivos con extensión ".class". Cada archivo .class contiene el bytecode correspondiente a una clase Java específica dentro del programa.
4. **Almacenamiento y distribución:** Los archivos .class se almacenan en el sistema de archivos y se pueden distribuir en forma de archivos JAR o WAR.
5. **Ejecución en una máquina virtual Java (JVM):** Para ejecutar el código, se utiliza una Máquina Virtual Java (JVM) lo que permite que las aplicaciones escritas en Java sean portables y se ejecuten en múltiples plataformas.

4. (RA1_ g) Se han identificado las características y escenarios de uso de las metodologías ágiles de desarrollo de software.

Comentar, al menos, 3 factores que se deben dar en un proyecto para que elijamos una metodología ágil para desarrollarlo.

Respuesta ágil a los cambios del proyecto:

Si los requisitos iniciales son erróneos o propensos a cambiar a medida que se avanza en el proyecto, la metodología ágil se convierte en la mejor opción ya que permite la flexibilidad para ajustar los objetivos y las prioridades a medida que se obtiene una comprensión más clara de las necesidades del cliente.

Necesidad de entregas periódicas del producto:

Si se requiere el proyecto de manera temprana y constante, las metodologías ágiles puede ser un acierto. Esta metodología permite hacer entregas en intervalos cortos conforme se realiza el proyecto con la finalidad de que los clientes comiencen a utilizar el software sin necesidad a esperar hasta el final del proyecto.

Equipo colaborativo:

Una metodología ágil puede ser la adecuada si el equipo de proyecto está compuesto por profesionales que pueden tomar decisiones de manera conjuntas. En esta metodología los miembros del equipo trabajan juntos, comparten la responsabilidad y tienen la libertad de tomar decisiones operativas. Esto puede llevar a una mayor eficiencia y calidad en el desarrollo del proyecto.