

## 1. <sup>1</sup>Introducción a Java

### Test de autoevaluación

1) ¿Qué es el código de ensamblaje y el programa traductor ensamblador?

*En mis palabras, el código de ensamblaje es el lenguaje más básico, y está constituido por números binarios, es el lenguaje que entiende directamente la computadora. Y, el programa traductor es un programa que traduce el lenguaje humano (en formato de código, es decir, no literalmente) a código de ensamblaje*

2) ¿Qué es la abstracción en informática?

*La abstracción se podría definir como cada vez los lenguajes se van haciendo más al entendimiento del habla humana que al de la máquina. Omitiendo información para que sea más entendible.*

3) Explique los diferentes niveles de abstracción en lenguajes de programación y de algunos ejemplos de lenguajes por nivel de abstracción

*Existen tres niveles de abstracción, el de **bajo nivel** que sería programación básicamente a nivel máquina, ya que permiten controlar directamente el software y se entiende con las características del hardware. Un ejemplo es el lenguaje **ensamblador o assembler**. El **nivel medio** se entiende como un lenguaje que trabaja directamente sobre **procedimientos**, algunos de estos como C o Basic pueden ser interpretados como de alto nivel, pero en realidad no lo son ya que como se menciona antes, éstos trabajan sobre procedimientos en el software como los de bajo nivel y también con algunas características de alto nivel. El **nivel alto** se caracteriza por un, valga la redundancia, alto nivel de **abstracción**, haciendo que sea entendible con leerlo (ya que se basa en el lenguaje humano), ya que trabaja sobre objetos (más adaptado al pensamiento humano en sí) y estos emergen a procedimientos.*

4) Defina interprete, compilador y explique la diferencia entre estos dos conceptos

*Un intérprete es un programa que realiza la traducción a medida que sea necesaria, es decir, línea por línea, tiene la ventaja de que una vez*

---

<sup>1</sup> Alumna: Mikaela Rosas

*interpretado puede ser ejecutado (entendido) en cualquier software, aunque al interpretar línea por línea; almacena mucha memoria. Por el contrario, el compilador traduce de una sólo vez un programa, entonces consume menos memoria y es más rápido, pero sólo puede ser ejecutado en el software donde fue compilado.*

5) ¿Qué es un paradigma de programación? Nombre algunos de ellos  
*Son estilos o maneras de programar adoptadas por un gran grupo de programadores, como un consenso, los paradigmas buscan resolver problemas “comunes” dentro de la comunidad. Algunos ejemplos son el paradigma imperativo, el declarativo y el con orientación a objetos.*

6) Explique que es el paradigma orientado a objetos  
*Se basa en una entidad llamada objetos que al mismo tiempo puede relacionarse con otras de este tipo, es muy usado ya que presenta soluciones de forma clara, ordenada, es flexible y posibilita a la reutilización del código.*

7) ¿Cómo es que Java es tan popular en tecnologías webs, si cuando se creó el lenguaje no existía Internet tal y como se la conoce en la actualidad?

*Es así gracias a que fue pensado para funcionar en los aparatos electrónicos domésticos, aunque no evolucionó de ese modo (como se esperaba), junto a ese avance y una “predicción” de cómo avanzaría internet conocemos a Java tal y como es hoy.*

8) Liste las características más relevantes del lenguaje y desarrolle brevemente cada una de ellas

- *Paradigma orientado a objetos: como se mencionó antes es flexible, claro, ordenado y trabaja con una entidad llamada “objetos” que se puede relacionar con otras del mismo tipo.*
- *Facilidad de uso: al estar basada en otros lenguajes como C y C++, es de fácil entendimiento para programadores de estos otros dos lenguajes*
- *Independencia de la plataforma: gracias a la máquina virtual que se encarga de interpretar y compilar todo en tiempo real, en esa y otras máquinas.*

- *Soporte en red: facilita el uso de redes, pudiendo ser aplicadas con tan solo una URL y un socket.*
- *Seguridad: tiene un sistema de seguridad por capas que además restringe algunos usos como el portapapeles.*

9) ¿Cómo es que Java es multiplataforma y al mismo tiempo sus programas se ejecutan de una manera eficiente respecto al tiempo de ejecución?

*Gracias a su máquina virtual*

10) ¿Qué es y cómo funciona la máquina virtual de Java?

*Funciona ya que la máquina virtual trabaja con un compilador del código escrito por el humano “traduciéndolo” a un código binario especial llamado bytecode, que es generado por el mismo compilador, y después este código es ejecutado por la máquina virtual y ésta traduce el código bytecode a uno nativo de cualquier software (Windows, Mac, Linux).*

## **2. Elementos básicos del lenguaje**

### **Test de autoevaluación**

1) ¿Qué es un linker o cargador, y para qué sirve?

Es la interfaz encargada de suministrar el bloque de memoria donde será ejecutado el código

2) ¿Cuáles son las tres zonas de memoria utilizadas por Java?

La zona de datos, la pila (stack) y el montículo (heap)

3) Explique para qué sirve la zona de memoria de datos

En esta zona se guardan las instrucciones del programa ejecutable en código máquina, las constantes, el código correspondiente a los métodos, y a las clases. Es una zona “estática” por lo cual se le debe informar el tamaño de los elementos, ya que no procesa la variabilidad de éstos al ejecutar el programa.

4) Explique para qué sirve la zona de memoria conocida como pila o stack

La pila sirve para guardar variables locales, variables de referencia, parámetros y valores de retorno, resultados parciales, y el control de la invocación y retorno de métodos. Tiene memoria limitada y se basa en (por ejemplo, una pila de

platos), donde lo último que se guarda será lo primero en ser eliminado y lo primero que se guarda será lo último en ser eliminado.

5) Explique para qué sirve la zona de memoria conocida como montículo o heap  
El montículo o heap es parte de lo que antes se encargaba el programador, se utiliza para guardar variables de instancia y objetos. El encargado de administrar este espacio de memoria es el Garbage Collector o recolector de basura. Es dinámica.

6) ¿Qué es un atributo y de qué tipos de datos puede ser definido?

Es un dato, puede almacenar diferentes cosas y tener diferentes características (como un número entero, decimal, cadena de texto, etc.) pueden ser primitivos o de referencia.

7) ¿Cuál es la diferencia entre tipos de datos primitivos y de referencia? Explique cada una de estas categorías

Los tipos de datos primitivos son sencillos, cortos y concretos. Tienen un tamaño predeterminado y se colocan en pila para que el programa sea más eficiente.

Los tipos de datos “de referencia”, justamente hacen referencia a objetos y almacenan ese espacio en la memoria (es decir, no está predefinida) y cuando se hace intercambios de datos entre dos de este tipo, no se copia la información sino que se “referencia” el lugar donde está almacenado.

8) El tipo de dato String, ¿Es un tipo de dato primitivo o de referencia? Explique las particularidades correspondientes al mismo

Si bien el tipo de dato String no es ninguno de estos, es más “apropiado” tratarlo como tipo de dato primitivo ya que su valor es constante, aunque también tiene características de uno de referencia, ya que String es básicamente un objeto (en lo que respecta al almacenado de memoria).

9) Qué es la conversión entre tipos de datos y en qué casos resultaría útil hacer este tipo de conversión. Detalle los problemas que pueden surgir al trabajar con conversiones

La conversión es pasar un tipo de dato a otro, por ejemplo int a string, existen formas específicas de hacerlo, son útiles a la hora de sumar, por ejemplo, un entero con un decimal, y podría causar problemas de precisión a la hora de convertir.

10) Liste y explique las convenciones de escritura adoptado por el común denominador de programadores

**Clases:** Las clases deben ser escritas en letras minúsculas, con la primera letra de cada palabra en mayúsculas, esta forma de escritura es llamada lomo de camello o CamelCase en inglés.

**Métodos:** Los métodos deben ser verbos escritos en minúsculas con la primera letra de cada palabra en mayúsculas con excepción de la primera palabra del método.

**Atributos:** Al igual que los métodos deben ser escritos en minúsculas con la primera letra de cada palabra en mayúsculas con excepción de la primera palabra del identificador del atributo.

**Constantes:** Las constantes deben escribirse en su totalidad con mayúsculas y para separar palabras se debe usar el guión bajo (underscore) "\_".

**Paquetes:** Los nombres de los paquetes se escriben completamente en minúsculas.

**Estructuras de control:** Cuando una sentencia forma parte de una estructura de control como por ejemplo de flujo como for o while, es necesario escribir los corchetes ({}), aunque solo se tenga una línea de código.

**Espacios:** Los espacios son necesarios para entender el código de mejor manera, para mantenerlo limpio, ordenado y distinguir el principio y el final de estructuras de control. Cada vez que se escribe algo dentro de una llave "{", se hace luego de una tabulación.

**Comentarios:** Los comentarios son útiles y recomendados para explicar ciertas partes del código o hacer aclaraciones sobre el mismo. Especialmente si es un proyecto en el que pueda trabajar más de un programador.