|  | **CAMPUS LADS** |  |
| --- | --- | --- |
| **SOFTWARE REVIEW – CICLO I** | Rev 01/06/2023 |
| **Introducción a los Algoritmos y Programación** | Trainer Ing.  Carlos H. Rueda C. |

**INTRODUCCIÓN**

Investigue y conteste las preguntas en este mismo documento usando otro color de letra y usando sus propias palabras.

1. ¿Cuál es el ciclo de vida de un algoritmo?

Son instrucciones repetitivas como una serie de instrucciones para que se repitan determinadas o indeterminadas veces. Te permiten ahorrar tiempo en el momento en que debas realizar instrucciones un algoritmo, se trata de una estructura reiterativa.

El ciclo de vida de un algoritmo se compone de varias etapas. Estas etapas incluyen:

a) Definición del problema: Se establece el objetivo del algoritmo y los requisitos del problema que se desea resolver.

b) Diseño: Se desarrolla una solución al problema mediante la creación de un algoritmo.

c) Implementación: El algoritmo se traduce a un lenguaje de programación específico.

d) Prueba y depuración: El algoritmo se prueba con diferentes casos de prueba para asegurarse de que funciona correctamente y se corrigen los errores si los hay.

e) Mantenimiento: El algoritmo se puede mejorar o modificar con el tiempo para adaptarse a nuevas necesidades o corregir posibles problemas.

2. ¿Cuál es el origen de Python? ¿En que lenguajes se basó su diseño?

La historia de Python empieza con Guido Van Rossum empezando su desarrollo en 1989 y empezando a implementarlo en febrero de 1991, momento en el que se publicó la primera versión pública: la 0.9.0.

[Guido Van Rossum](https://www.tokioschool.com/noticias/guido-van-rossum-creador-de-python/) es el creador y responsable de que Python exista. Se trata de un informático de origen holandés que fue el encargado de diseñar Python y de pensar y definir todas las vías posibles de evolución de este popular lenguaje de programación

Python utiliza módulos de código que son intercambiables en lugar de una larga lista de instrucciones que era estándar para los lenguajes de programación funcional.

En las navidades de 1989 Van Rossum, mientras trabajaba en un centro de investigación holandés (CWI), decidió empezar un nuevo proyecto como pasatiempo personal. Pensó en darle continuidad a ABC, un lenguaje de programación que se desarrolló en el mismo centro en el que estaba trabajando.

Sin embargo, el proyecto no llegó mucho más lejos por las limitaciones del hardware de la época, así que Van Rossum decidió darle una segunda vida a su idea y partiendo de la base que tenía, empezó a trabajar en Python.

Esta primera versión ya incluía clases con herencias, manejo de excepciones, [funciones](https://www.tokioschool.com/noticias/funciones-python-curso-programacion/) y una de las [características principales de Python](https://www.tokioschool.com/noticias/caracteristicas-principales-de-python/): funcionamiento modular. Esto permitía que fuese un lenguaje mucho más limpio y accesible para la gente con pocos conocimientos de programación. Una característica que se mantiene hasta el día de hoy.

Hasta el año 2018, el desarrollo de este popular lenguaje de programación estaba dirigido personalmente por Van Rossum, pero decidió apartarse y, desde 2019, son cinco las personas que deciden como evoluciona y se desarrolla Python. Un consejo que se renueva de forma anual.

Para cuando salió la primera versión definitiva de Python, tal era la popularidad de este nuevo lenguaje de programación que se se creó comp.lang.python, un foro de discusión de Python que multiplicó, aún más, su cantidad de usuarios.

3. ¿Cuáles son las principales características de Python?

#### Programación Orientada a objetos (POO):

El código se organiza en unidades denominadas clases y objetos.

Lenguaje Interpretado:

Ejecuta las instrucciones directamente, sin una previa compilación del programa a instrucciones en lenguaje máquina.

#### Multiplataforma:

Python está disponible en los principales sistemas operativos como Linux, Windows, UNIX, Mac OS, etc lo que permite mayor facilidad a sus usuarios

#### Tipado dinámico

Python permite que las variables no tengan que especificar su tipo. Estas adoptan un tipo automáticamente en función del valor que se les asigne

#### Lenguaje open source

No se requieren licencias de pago para comenzar a trabajar con él.

#### Ampliamente respaldado:

Sus características y funcionalidades hacen que este lenguaje sea muy interesante. Por ello, Python ha generado una comunidad de usuarios muy grande a su alrededor que puede ser de utilidad cuando queremos encontrar información o pedir ayuda para desarrollar cualquier tipo de programa o algoritmo.

#### Es polivalente:

Es decir sirve para diferentes proyectos como lo son:

* Aprendizaje automático (Machine Learning)
* Inteligencia Artificial (IA)
* Big data y Análisis de datos
* Operaciones matemáticas
* Visualización de datos
* Programación de apps
* Desarrollo web
* Desarrollo de videojuegos
* Gestión financiera

4. ¿Qué son las variables?

Es una unidad de almacenamiento y recuperación de datos con valores que pueden cambiar, la cual se identifica con un nombre único en el código del programa. Puede entenderse como un nombre que representa un valor y se utiliza de forma periódica para almacenar diversos tipos de datos.

5. ¿Qué son con las constantes?

Es valor fijo que no cambia durante la ejecución del programa pueden ser un carácter o una cadena de caracteres. Las constantes son utilizadas en programas para evitar errores, asegurando que ciertas [variables](https://programacion.top/conceptos/variable-es/) y valores permanezcan inalterables.

6. ¿Cuáles son los tipos de datos primitivos en Python?

Python tiene **cuatro tipos primitivos:** enteros, flotantes, booleanos y cadenas o Strings.

* Los tipos de variables enteros en Python (int) se utilizan para representar datos numéricos, específicamente números enteros. Estos pueden ser tanto positivos como negativos.
* year=2021
* day=7
* temperatura=-10
* angulo=60

Los tipos de datos flotantes en Python (float) se utilizan para representar números de coma flotante o con decimales.

pi= 3.1416

Estatura=1.60

peso=50

edad= 28.0

* Los tipos primitivos boleanos en python (bool) son tipos de datos binarios, es decir que pueden tomar los valores: *Verdadero* (**True**) y *Falso* (**False**). Son útiles para expresiones con condicionales y de comparaciones.

esta\_caliente = False

es\_alto= True

* Los tipos de datos String en Python son un arreglo de caracteres que forman cadenas para formar un mensaje o oración generalmente. Se pueden crear usando comillas simples, dobles o triples.
* profesor = "Sergio Castaño Giraldo"
* web = 'Control Automático Educación'
* ursos = """
* 1. Python
* 2. Matlab
* 3. MicroPython
* 4. Arduino
* 5. PIC
* 6. Control de Procesos
* """

7. ¿Qué es el Upper Camel Case o Pascal Case?

La notación Pascal Case combina las palabras directamente, sin usar ningún símbolo, estableciendo que la primera letra de cada palabra esté en mayúscula sin excepciones, estando el resto de letras en minúsculas.

8. ¿Qué es Lower Camel Case (o también conocido como Camel Case)?

CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas.Muy utilizado en lenguajes orientados a objetos.

9. ¿Qué es el Snake Case?

Snake case se refiere al estilo de escritura en el que cada espacio se reemplaza con un carácter de subrayado y la primera letra de cada palabra se escribe en minúsculas.

10. ¿En qué consiste la Hungarian notation (notación húngara)?

Consiste en colocar al inicio del nombre de la variable una o más letras minúsculas que indique el tipo de dato, en el resto del nombre la descripción de la variable