# 

# **TRABAJO GRUPAL: Tipos de Datos y Sentencias**

**Nombres:** Constanza Hurtado Ávila

Pablo Morales

Emanuel Olguín

Alejandro Venegas

DESARROLLO CUESTIONARIO

Para el avance de nuestro proyecto debemos incorporar algunos tipos de datos y así dar una solución óptima a nuestro problema. Continuando en nuestro desarrollo anterior, debemos incluir lo siguiente:

1. **Discutan ¿Qué es un dato booleano? ¿Qué utilidad puede tener para el desarrollo de un programa?**

Una variable booleana es una variable que sólo puede tomar dos posibles valores: True (verdadero) o False (falso).

Los booleanos son especialmente útiles con estructuras de control condicionales como declaraciones if, se evalúa la expresión y, en caso de ser ésta verdadera, se ejecuta el código a continuación.

1. **Investigar qué significa que python sea un lenguaje de tipado dinámico.**

Un módulo en Python es un archivo con extensión .py y consta de un código en lenguaje Python.

Se puede hacer referencia a cualquier archivo de Python como módulo. Algunos de estos módulos se encuentran a través de la biblioteca estándar de Python (vienen incluidos en su instalación). Otros se pueden instalar con el administrador de paquetes pip de Python. También, se encuentra la posibilidad de crearlos.

Entonces para crear un módulo, necesitamos guardar el código en una extensión .py. Luego, se utiliza import para llamar al módulo, seguido de su nombre.

1. **Investigar y documentar sobre la creación de Módulos en Python.**

Un lenguaje de programación es dinámicamente tipado si una variable puede tomar valores de distinto tipo.

La mayoría de lenguajes de tipado dinámico son lenguajes interpretados, como Python.

1. **Investigar y documentar sobre la creación de Paquetes en Python.**

Un paquete es una carpeta que contiene varios módulos. Podemos diseñar un paquete matemático creando una carpeta con la siguiente estructura.

matemático/

|-- \_init\_.py

|-- aritmetica.py

|-- geometria.py

Debe contener siempre un archivo \_init\_.py (por el momento vacío) para que Python entienda que se trata de un paquete y no de una simple carpeta. Así, podemos acceder a alguno de los módulos del paquete de la siguiente manera.

import matematica.aritmetica

print(matematica.aritmetica.sumar(7, 5))

O bien de la siguiente.

from matematica import aritmetica

print(aritmetica.sumar(7, 5))

También, esta otra:

from matematica.aritmetica import sumar

print(sumar(7, 5))

1. **Investigar e implementar el uso del archivo \_\_init\_\_.py**

El método \_init\_ es un método especial de una clase en Python. El objetivo fundamental del método \_init\_ es inicializar los atributos del objeto que creamos. Básicamente el método \_init\_ remplaza al método inicializar que habíamos hecho en el concepto anterior.

Las ventajas de implementar el método \_init\_ en lugar del método inicializar son:

* El método \_init\_ es el primer método que se ejecuta cuando se crea un objeto.
* El método \_init\_ se llama automáticamente. Es decir, es imposible de olvidarse de llamarlo ya que se llamará automáticamente.
* Quien utiliza POO en Python (Programación Orientada a Objetos) conoce el objetivo de este método.
* Otras características del método \_init\_ son:
* Se ejecuta inmediatamente luego de crear un objeto.
* El método \_init\_ no puede retornar dato.
* el método \_init\_ puede recibir parámetros que se utilizan normalmente para inicializar atributos.
* El método \_init\_ es un método opcional, de todos modos, es muy común declararlo.