

Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

CI3641 – Lenguajes de Programación 1

Trimestre: Septiembre - Diciembre 2023

Profesor: Ricardo Monascal

Estudiante: Junior Miguel Lara Torres, Carnet: 17-10303

**Parcial 3 (30 pts)**

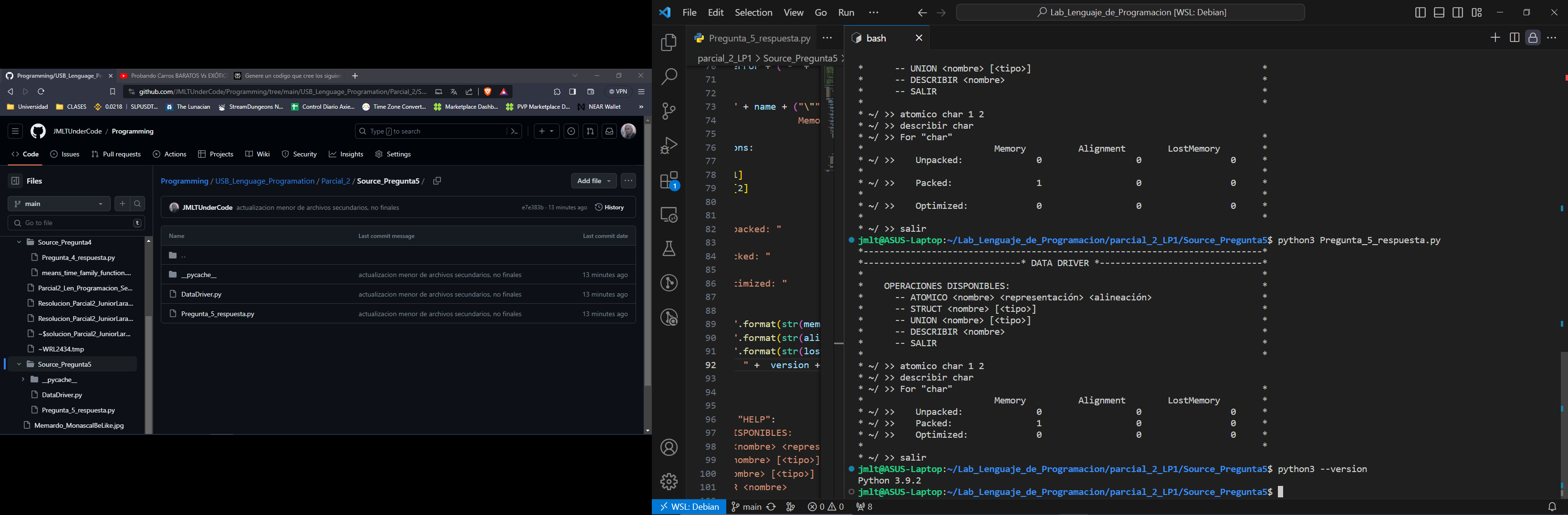
* “En algunas preguntas, se usarán las constantes X, Y y Z. Estas constantes debe obtenerlas de los últimos tres números de su carnet.”

Caso particular, 17-10303 entonces .

* “En aquellas preguntas donde se le pida implementar un programa, mantenga su código en un repositorio git remoto (preferiblemente Github) y coloque un enlace al mismo en lugar de su respuesta. Todo su código debe ser legible y estar debidamente documentado.”

Todos los códigos, este propio documento, documento oficial del enunciado de examen 2 y otras evaluaciones de la materia serán cargados en:

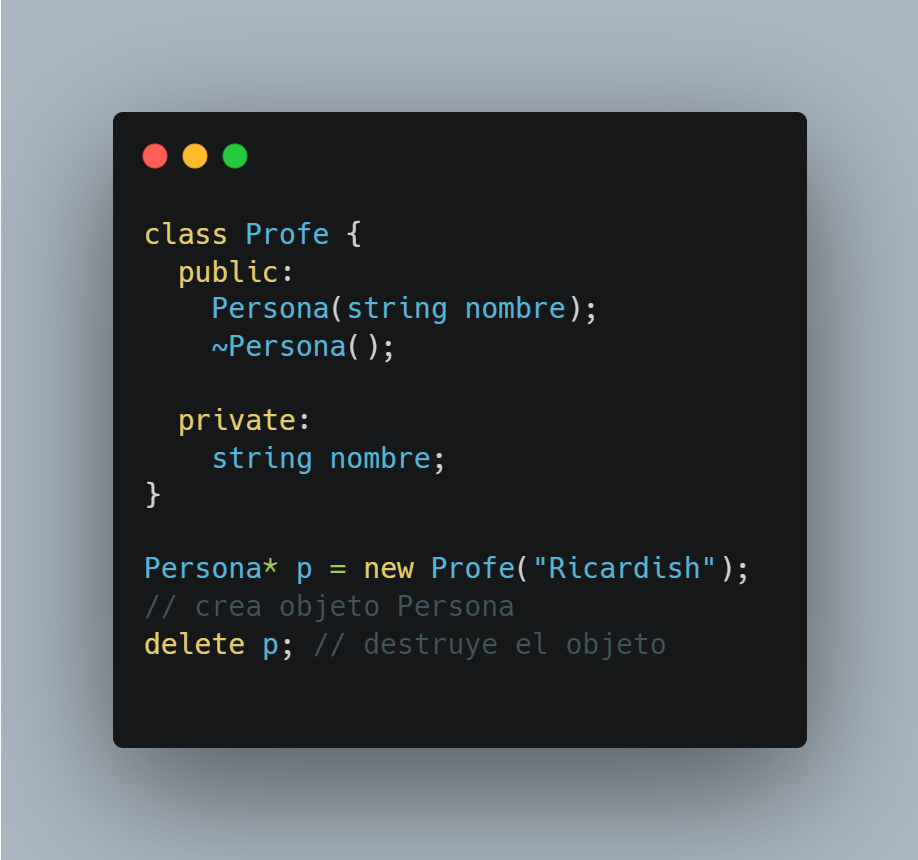
<https://github.com/JMLTUnderCode/Programming/tree/main/USB_Lenguage_Programation>

En todas las respuestas dadas en este parcial se usó Python cuya versión es 3.9.2.

* **1er Pregunta:**

El lenguaje utilizado en esta pregunta es C++, dado que mi nombre es Junior Miguel Lara Torres.

* + (a)
    - i.

C++ permite crear y manipular objetos a través de clases y structs. Los objetos se crean mediante constructores, que son métodos especiales para inicializar un objeto. Los objetos tienen campos de datos (variables miembro) y métodos (funciones miembro) que operan sobre esos datos.

* + - ii.

Este lenguaje maneja la memoria de forma explícita a través de los operadores ***new*** y ***delete***. El programador debe reservar y liberar memoria manualmente. No hay recolección de basura automática. Sin embargo, existe una librería de nombre ***Boehm-Demers-Weiser (BDW)*** que funciona como recolector de basura, es una biblioteca de C que puede ser utilizada en proyectos de C++ para gestionar la memoria dinámica.

* + - iii.

En C++ se utiliza enlazado estático de métodos de forma predeterminada. Esto significa que los métodos se enlazan a la definición de la clase en tiempo de compilación. Pero se puede usar enlazado dinámico con punteros a funciones, la palabra clave que se usa para indicar que un método asocia dinámicamente es ***virtual***.

* + - iv.

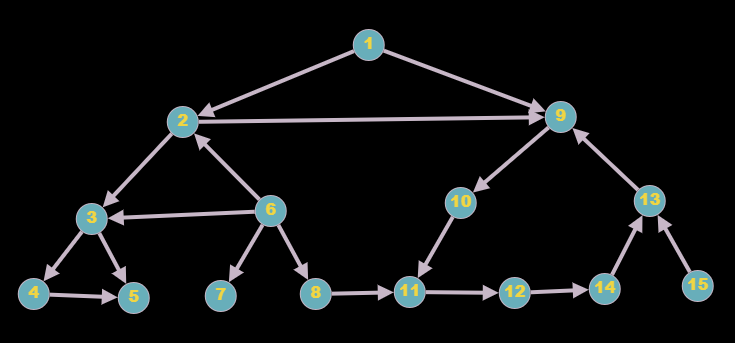
Es un lenguaje con tipado estático y fuerte. Soporta herencia simple y múltiple a través de clases base. Tiene polimorfismo paramétrico mediante ***template <typename T>***. Y maneja la varianza de tipos permitiendo especificar si una clase derivada es un subtipo covariante, contravariante o invariante de su clase base.

* + (b)
    - i.

El archivo principal de la respuesta a este apartado se encuentra en el siguiente link.

<https://github.com/JMLTUnderCode/Programming/blob/main/USB_Lenguage_Programation/Parcial_3/Source_Pregunta1/Pregunta_1_b_1_respuesta.cpp>

* + - ii.

El grafo usado como ejemplo es el siguiente.

El archivo principal de la respuesta a este apartado se encuentra en el siguiente link.

<https://github.com/JMLTUnderCode/Programming/blob/main/USB_Lenguage_Programation/Parcial_3/Source_Pregunta1/Pregunta_1_b_2_respuesta.cpp>

* **2er Pregunta:**
* **3er Pregunta:**
* **4er Pregunta:**

Se escoge Python3 como lenguaje de programación.

En el siguiente link encontrara los siguientes archivos esenciales:

* Pregunta\_4\_respuesta.py
* virtualtable.py
* test\_ virtualtable.py

[https://github.com/JMLTUnderCode/Programming/tree/main/USB\_Lenguage\_Programation/Parcial\_3/Source\_Pregunta4](https://github.com/JMLTUnderCode/Programming/tree/main/USB_Lenguage_Programation/Parcial_3/Source_Pregunta2)

La ejecución del archivo fuente principal se realiza por consola con:

[py o python3] Pregunta\_2\_respuesta.py

En el caso de la cobertura se requiere que haga la instalación **pytest** y **coverage** en su sistema operativo basado en Linux. Es decir, realice los siguientes pasos:

pip install pytest

pip install coverage

Luego debe realizar la siguiente corrida en consola:

coverage run -m pytest test\_AritmeticExprCalculator.py

Mostrándole en pantalla la ejecución correcta de las pruebas unitarias. Luego para ver la información detallada de la cobertura debe escribir en consola:

coverage report -m

Cuyos resultados corresponden a un 96% de cobertura.

* **5er Pregunta:**
* **6er Pregunta:**
* **Pregunta Extra:**

Sin tiempo. ☹