
Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI3641 – Lenguajes de Programación I
Septiembre–Diciembre 2025

Examen 1
(25 puntos)

A continuación encontrará 5 preguntas (y una sorpresa al final), cada una de las cuales tiene un valor de 5 puntos. Sea lo más detallado y preciso posible en sus razonamientos y procedimientos.

En algunas preguntas, se usarán las constantes X , Y y Z . Estas constantes debe obtenerlas de los últimos tres números de su carné. Por ejemplo, si su carné es 09-40325, entonces $X = 3$, $Y = 2$ y $Z = 5$.

En aquellas preguntas donde se le pida decir qué imprime un programa, incluya los pasos relevantes de la ejecución del mismo con los cuales usted pudo alcanzar su conclusión.

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.

La entrega se realizará por correo electrónico a rmonascal@gmail.com hasta las 11:59pm (VET) del Lunes 16 de Octubre de 2022.

1. Pregunta 1

Escoja algún lenguaje de programación de alto nivel y de propósito general cuyo nombre empiece con la misma letra que su nombre (por ejemplo, si su nombre es “Pedro”, podría escoger “Perl”, “PLI”, “Python”, etc.).

(a) Dé una breve descripción del lenguaje escogido.

- I. Diga qué tipo de alcances y asociaciones posee, argumentando las ventajas y desventajas de la decisión tomada por los diseñadores del lenguaje, en el contexto de sus usuarios objetivos.
- II. Diga qué tipo de módulos ofrece (de tenerlos) y las diferentes formas de importar y exportar nombres.
- III. Diga si el lenguaje ofrece la posibilidad de crear *aliases*, sobrecarga y polimorfismo. En caso afirmativo, dé algunos ejemplos.
- IV. Diga qué herramientas ofrece a potenciales desarrolladores, como: compiladores, intérpretes, debuggers, profilers, frameworks, etc.

(b) Implemente los siguientes programas en el lenguaje escogido:

- I. Dada una cadena de caracteres w y un entero no-negativo k , calcular la rotación de k posiciones de la cadena w . Utilice la siguiente fórmula como referencia:

$$\text{rotar}(w, k) = \begin{cases} w & \text{si } k = 0 \vee |w| = 0 \\ \text{rotar}(w + [a], k - 1) & \text{si } k > 0 \wedge w = ax \wedge a \text{ es un caracter} \end{cases}$$

donde el operador $++$ corresponde a la concatenación de cadenas de caracteres.

Ejemplo:

- $\text{rotar}(\text{"hola"}, 0) = \text{"hola"}$
- $\text{rotar}(\text{"hola"}, 1) = \text{.olah"}$
- $\text{rotar}(\text{"hola"}, 2) = \text{"laho"}$
- $\text{rotar}(\text{"hola"}, 3) = \text{.ahol"}$
- $\text{rotar}(\text{"hola"}, 4) = \text{"hola"}$
- $\text{rotar}(\text{"hola"}, 5) = \text{.olah"}$

- II. Dada una matriz cuadrada A (cuya dimensión es $N \times N$), calcular el producto $A \times A^T$ (donde A^T es la transpuesta de A).

La multiplicación de dos matrices cuadradas A y B (de tamaño $N \times N$) viene dada por:

$$\forall i, j \in [1..N] : (A \times B)_{i,j} = \sum_{k \in [1..N]} A_{i,k} \times B_{k,j}$$

y la transpuesta de A se define como:

$$\forall i, j \in [1..N] : A_{i,j}^T = A_{j,i}$$

2. Pregunta 2

Considere el siguiente programa escrito en pseudo-código:

```
int a = Y + Z + 1, b = X + Y + 1, c = Z + Y + 1;
```

```
sub R (int b) {  
    a := b + c - 1  
}
```

```
sub Q (int a, sub r) {  
    b := a + 1  
    r(c)  
}
```

```
sub P(int a, sub s, sub t) {  
    sub R(int a) {  
        b := c + a + 1  
    }  
    sub Q (int b, sub r) {  
        c := a + b  
        r(c + a)  
        t(c + b)  
    }  
    int c := a + b  
    if (a < 2 * (Y + Z + 1)) {  
        P(a + 2 * (Y + Z + 1), s, R)  
    } else {  
        int a := c + 1  
        s(c * a, R)  
        Q(c * b, t)  
    }  
    print(a, b, c)  
}
```

```
P(a, Q, R);  
print(a, b, c)
```

Note que deberá reemplazar los valores para X , Y y Z como fue explicado en la introducción.

Diga qué imprime el programa en cuestión, si el lenguaje tiene:

- (a) Alcance estático y asociación profunda
- (b) Alcance dinámico y asociación profunda
- (c) Alcance estático y asociación superficial
- (d) Alcance dinámico y asociación superficial

Recuerde mostrar los pasos de su ejecución (por lo menos al nivel de cada nuevo marco de pila creado).

3. Pregunta 3

Se desea que modele e implemente, en el lenguaje de su elección, un programa que simule un manejador de memoria que implemente el *buddy system*. Este programa debe cumplir con las siguientes características:

- (a) Al ser invocado, recibirá como argumento la cantidad de bloques de memoria que manejará.
- (b) El programa pedirá repetidamente una acción al usuario, que puede ser:
 - I. **RESERVAR** *<cantidad><nombre>*
 - II. **LIBERAR** *<nombre>*
 - III. **MOSTRAR**
 - IV. **SALIR**

Cada acción debe ser validada con mensajes de error claros e informativos. Además, debe incluir pruebas unitarias y alcanzar una cobertura de al menos 80 %.

4. Pregunta 4

Implemente un módulo que defina el tipo de vectores tridimensionales y operadores aritméticos sobre estos, cumpliendo con las siguientes características:

- Soporte operaciones: suma (+), resta (-), producto cruz (*), producto punto (%), y norma (&).
- Permita expresiones naturales como:

```
b + c
a * b + c
(b + b) * (c - a)
a % (c * b)
```

- Soporte operaciones con escalares por la derecha:

```
b + 3
a * 3.0 + &b
(b + b) * (c % a)
```

donde $(x, y, z) \oplus n = (x \oplus n, y \oplus n, z \oplus n)$.

Incluya pruebas unitarias y cobertura mayor al 80 %.

5. Pregunta 5

Modele e implemente un programa que simule programas, intérpretes y traductores, como en los diagramas de T. El programa debe permitir definir:

- **PROGRAMA** <nombre><lenguaje>
- **INTERPRETE** <lenguaje_base><lenguaje>
- **TRADUCTOR** <lenguaje_base><lenguaje_origen><lenguaje_destino>

Debe manejar el lenguaje especial **LOCAL**, correspondiente al lenguaje ejecutable por la máquina local.

Debe permitir acciones del usuario:

- **DEFINIR** <tipo>[<argumentos>]
- **EJECUTABLE** <nombre>
- **SALIR**

Incluya ejemplos, validaciones de error y pruebas unitarias con cobertura mayor al 80 %.

Reto Extra: Golf

Considere los números de Narayana:

$$N_{n,k} = \frac{1}{n} \binom{n}{k} \binom{n}{k-1}$$

y la función Fibonacci:

$$\text{fib}(n) = \begin{cases} n & \text{si } 0 \leq n < 2 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{si } n \geq 2 \end{cases}$$

Así como el logaritmo en base 2:

$$\log_2(n) = m \Leftrightarrow 2^m = n$$

Definimos:

$$\text{wadefoc}(n) = \text{fib}(\lfloor \log_2(N_{n+1,n-1}) \rfloor + 1)$$

Desarrolle un programa que reciba un argumento $n \geq 2$ y calcule $\text{wadefoc}(n)$, con tiempo de ejecución menor a 1 segundo hasta $n = 100$.

Reglas del reto: minimice la cantidad de caracteres de su programa. Los ganadores recibirán puntos extra (5, 3 y 1 respectivamente).