



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# Concurso Profesorial 2025 de la Facultad de Minas

Cátedra 0.2 → C1\_3010435 - Fundamentos de programación

Facultad de Minas – Departamento de Ciencias de la Computación y de la Decisión

Aspirante: Julián Andrés Castillo Grisales

Ingeniero de Sistemas, Magister en Ingeniería, Especialista en Analítica y Big Data

# Clase Fundamentos de Programación

Contenido de la clase: Condicionales --> Optimización de condicionales (Ladder & DRY)

Facultad de Minas – Departamento de Ciencias de la Computación y de la Decisión

Docente: Julián Andrés Castillo G.



# Contenido del curso

## Fundamentos de Programación

1. Introducción ✓
2. Conceptos Básicos ✓
  - i. Constantes y variables ✓
  - ii. Operadores ✓
  - iii. Expresiones ✓
  - iv. Arreglos ✓
3. Control de flujo (Condicionales) ✓
  - I. Sentencia IF ✓
  - II. Sentencia IF anidada ✓
4. Iteraciones → SOON
  - i. Ciclo FOR → SOON
  - ii. Ciclo FOR anidado → SOON
  - iii. Ciclo WHILE anidado → SOON
5. Funciones → SOON
6. Archivos y buenas prácticas de programación → SOON
7. Aplicaciones → SOON
  - I. Operaciones con matrices → SOON
  - II. Análisis de datos → SOON
  - III. Gráficos y procesamiento de imágenes → SOON
  - IV. Algoritmos con cadenas de texto y con fechas. Regular expressions y web Scraping → SOON
  - V. Aprendizaje de máquina → SOON
  - VI. Programación orientada a objetos → SOON

Profundicemos

### Descripción del Curso

El objetivo de este curso es introducir al estudiante en las técnicas básicas de programación. Busca desarrollar en el estudiante la capacidad el análisis de problemas y su solución empleando algoritmos. Para lograr este objetivo el alumno debe adquirir una serie de conocimientos, capacidades y aptitudes que le permitan enfrentar diversas situaciones relacionadas con su carrera.

### Objetivo General

Desarrollar en el estudiante la capacidad de análisis necesario para abordar problemas cuya solución pueda ser alcanzada empleando algoritmos.

### Objetivos Específicos

1. Comprender los fundamentos de la programación.
2. Entender los mecanismos de la programación estructurada.
3. Entender el flujo de ejecución de un programa de computador.
4. Analizar problemas y desarrollar su solución mediante algoritmos.

# Optimización de Condicionales

## DRY – Decision Chain – Ladder If-Else

- El principio DRY (Don't Repeat Yourself), introducido por Dave Thomas y Andy Hunt en *The Pragmatic Programmer* (1999), establece que cada pieza de conocimiento o lógica en un sistema debe tener una única representación clara y autorizada. Su propósito es evitar la duplicación de código, reglas de negocio o información, reduciendo inconsistencias, errores y costos de mantenimiento.
- Las decisiones en cascada o cadena (o decision chain, también conocida como ladder if-else simplificada) es una técnica de programación que organiza secuencias de condiciones de forma jerárquica y eficiente, usualmente de mayor a menor (o viceversa), para evitar comparaciones redundantes y mejorar la legibilidad del código.

# Caso Hipotético 1

## Mejora de espacios de trabajo

Supongamos que la Universidad Nacional desea clasificar a sus estudiantes en tres grupos [A,B,C] para comprar sillas ergonómicas de acuerdo a tres rangos de estaturas. Para esto la empresa Sillas&Sillas\* tiene tres sillas académicas que sirven para los siguientes rangos de estatura.

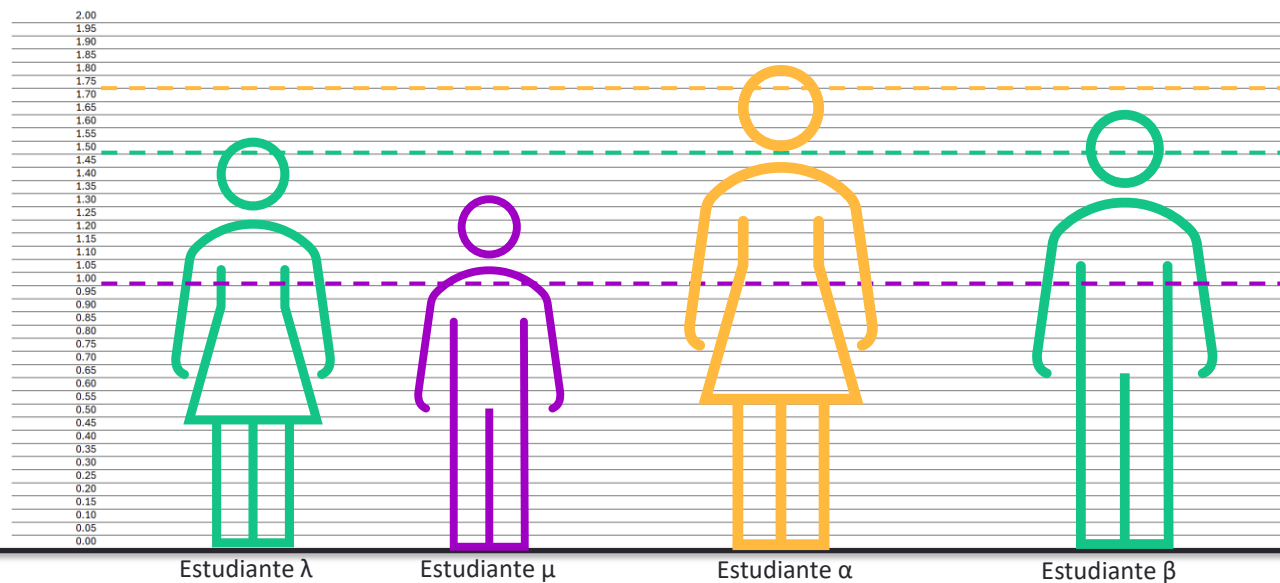
1. Silla A: Entre 1.00 m y 1.50 m
2. Silla B: Entre 1.50 m y 1.75 m
3. Silla C: De 1.75 m en adelante

Creemos un algoritmo usando condicionales que nos permita saber a que categoría pertenece un estudiante de la Universidad Nacional y así poderle asignar una silla acorde a su estatura.

# Caso Hipotético 1

## Mejora de espacios de trabajo

Vamos a tomar datos del público:



1. Silla A: Entre 1.00 m y 1.50 m
2. Silla B: Entre 1.50 m y 1.75 m
3. Silla C: De 1.75 m en adelante

- Pregunta 1: Cuantas personas entre 1.75 y 2.0 m.
- Pregunta 2: Cuantas personas entre 1.5 y 1.75 m.
- Pregunta 3: Cuantas personas entre 1.0 y 1.5 m.

## Caso Hipotético 1

Planteamiento de una solución algorítmica propia.

```
1 print('Favor ingresar la estatura del estudiante en metros: ')
2 estatura = float(input('-->'))
3 if estatura > 1.0 and estatura < 1.5:
4     categoria = "A"
5 elif estatura >= 1.5 and estatura < 1.75:
6     categoria = "B"
7 elif estatura >= 1.75:
8     categoria = "C"
9 else:
10     categoria = "NA"
11 print(f'La categoria del estudiante con estatura {estatura:,.2f}m es:\n\t--> {categoria}')
```



# Caso Hipotético 1.

## Planteamiento de una solución algorítmica usando IA

Prompt: Crea un algoritmo en Python para solucionar el presente problema: Supongamos que la Universidad Nacional desea clasificar a sus estudiantes en tres grupos [A,B,C] para comprar sillas ergonómicas de acuerdo a tres rangos de estaturas. Para esto la empresa Sillas&Sillas\* tiene tres sillas académicas que sirven para los siguientes rangos de estatura. Silla A: Entre 1.00 m y 1.50 m Silla B: Entre 1.50 m y 1.75 m Silla C: De 1.75 m en adelante Creemos un algoritmo usando condicionales que nos permita saber a que categoría pertenece un estudiante de la Universidad Nacional y poderle asignar una silla acorde a su estatura.

### Propuesta por Chat GPT (26 lc)

```
Perfecto 🌟
Te propongo el siguiente algoritmo en Python para clasificar al estudiante según su estatura y asignarle la silla correcta.

python
# Programa para clasificar estudiantes y asignar silla ergonómica
# Universidad Nacional - Empresa Sillas&Sillas

# Entrada de datos
estatura = float(input("Ingrese la estatura del estudiante en metros (Ej: 1.68): "))

# Clasificación según los rangos de estatura
if 1.00 <= estatura < 1.50:
    categoria = "A"
    silla = "Silla A (1.00 m - 1.50 m)"
elif 1.50 <= estatura < 1.75:
    categoria = "B"
    silla = "Silla B (1.50 m - 1.75 m)"
elif estatura >= 1.75:
    categoria = "C"
    silla = "Silla C (1.75 m en adelante)"
else:
    categoria = None
    silla = "Estatura fuera de rango válido."

# Salida de datos
if categoria:
    print(f"Categoría: {categoria}")
    print(f"Asignación: {silla}")
else:
    print("La estatura ingresada no es válida. Debe ser mayor o igual a 1.00 m.")
```

### Propuesta por Chat Gemini (30 lc)

```
Python
def asignar_silla(estatura_metros):
    """
    Clasifica a un estudiante en una categoría de silla (A, B o C) según su estatura.

    Args:
        estatura_metros: La estatura del estudiante en metros.

    Returns:
        Una cadena de texto indicando la categoría de la silla asignada.
        "Estatura no válida" si la estatura es menor que 1.00 m.
    """
    if estatura_metros < 1.00:
        return "Estatura no válida"
    elif 1.00 <= estatura_metros < 1.50:
        return "Silla A"
    elif 1.50 <= estatura_metros < 1.75:
        return "Silla B"
    else: # 1.75 m o más
        return "Silla C"

# Ejemplos de uso del algoritmo
estudiante1 = 1.45
estudiante2 = 1.68
estudiante3 = 1.80
estudiante4 = 0.95

print(f"Al estudiante con una estatura de {estudiante1} m se le asigna la: {asignar_silla(estudiante1)}")
print(f"Al estudiante con una estatura de {estudiante2} m se le asigna la: {asignar_silla(estudiante2)}")
print(f"Al estudiante con una estatura de {estudiante3} m se le asigna la: {asignar_silla(estudiante3)}")
print(f"Al estudiante con una estatura de {estudiante4} m se le asigna la: {asignar_silla(estudiante4)}")
```

### Propuesta por Claude (34 lc)

```
v2: Más reciente

# Script para clasificar estudiantes y asignar sillas ergonómicas
# Universidad Nacional - Sillas&Sillas

print("==== CLASIFICADOR DE SILLAS ERGONÓMICAS ====")
print("Grupo A - Silla A: 1.00m a 1.50m")
print("Grupo B - Silla B: 1.50m a 1.75m")
print("Grupo C - Silla C: 1.75m en adelante")
print("-" * 45)

# Solicitar la estatura del estudiante
estatura = float(input("Ingrese la estatura del estudiante en metros: "))

# Clasificación usando condicionales
if estatura < 1.0:
    print(f"Estatura {estatura}m está fuera del rango mínimo")
    print("No hay silla disponible para esta estatura")

elif estatura >= 1.0 and estatura < 1.5:
    print(f"Estatura: {estatura}m")
    print("Grupo: A")
    print("Silla asignada: Silla A")
    print("Rango: Entre 1.00m y 1.50m")

elif estatura >= 1.5 and estatura < 1.75:
    print(f"Estatura: {estatura}m")
    print("Grupo: B")
    print("Silla asignada: Silla B")
    print("Rango: Entre 1.50m y 1.75m")

else: # estatura >= 1.75
    print(f"Estatura: {estatura}m")
    print("Grupo: C")
    print("Silla asignada: Silla C")
    print("Rango: De 1.75m en adelante")
```

- Pregunta 1: ¿Qué diferencia podemos evidenciar con nuestra propuesta de solución?
- Pregunta 2: ¿Es posible mejorar nuestro algoritmo?

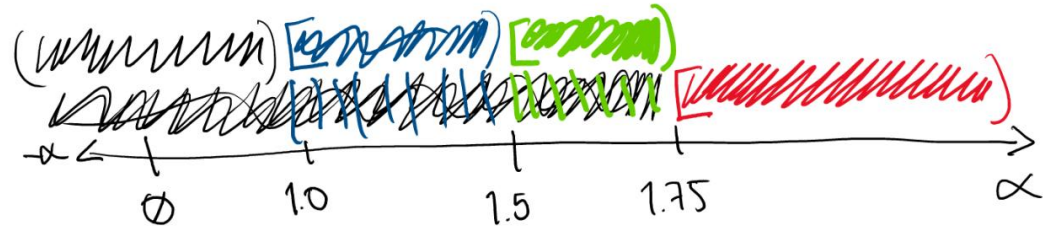
■ Nota: Copilot, DeepSeek, Grok y Meta AI brindan respuestas similares.

Prompt: Crea un algoritmo en Python para solucionar el presente problema: Supongamos que la Universidad Nacional desea clasificar a sus estudiantes en tres grupos [A,B,C] para comprar sillas ergonómicas de acuerdo a tres rangos de estaturas. Para esto la empresa Sillas&Sillas\* tiene tres sillas académicas que sirven para los siguientes rangos de estatura. Silla A: Entre 1.00 m y 1.50 m Silla B: Entre 1.50 m y 1.75 m Silla C: De 1.75 m en adelante Creemos un algoritmo usando condicionales que nos permita saber a que categoría pertenece un estudiante de la Universidad Nacional y poderle asignar una silla acorde a su estatura.

## Caso Hipotético 1.

### Planteamiento de una solución algorítmica usando cadenas de decisión

Evaluemos el concepto del algoritmo desde el punto de vista de la cantidad de preguntas necesarias para solucionar el problema y los conceptos de evaluación del algoritmo usando el concepto de cadenas de decisión o decisión escalonada.



```

if e >= 1.75:
    cat = "C"
elif e >= 1.5:
    cat = "B"
elif e >= 1.0:
    cat = "A"
else:
    cat = "NA"
  
```

## Caso Hipotético 2.

### Descuentos en Cafetería Central – El Ágora



Supongamos que la Cafetería Central desea aplicar un criterio de descuentos para incentivar la compra de productos. Para ello se definen niveles asociados a un valor de compra en donde se asigna un descuento porcentual de acuerdo a los siguientes rangos.

1. Compras inferiores a 25000. Sin descuento.
2. Compras superiores a 25000 e inferiores a 50000. Descuento del 5%.
3. Compras superiores a 50000 e inferiores a 100000. Descuento del 10%.
4. Compras superiores a 100000 e inferiores a 200000. Descuento del 15%.
5. Compras superiores a 200000. Descuento del 20%

Creemos un algoritmo usando condicionales anidados usando el criterio de If-Else escalonado para solucionar el presente problema. Usar f-strings aplicando formato para visualizar los resultados. Mostrar el valor inicial y el valor con descuento más un mensaje final.

Reto: Crear una factura tipo impresora térmica de recibos para un papel de 80mm con un estimado de 60 caracteres por línea.



## Caso Hipotético 2.

### Descuentos en Cafetería Central – El Ágora

Realicemos la solución en nuestro entorno de desarrollo favorito.



```
1 print('Bienvenido a la Cafetería Central'.center(60))
2 print('Ingresa el valor comprado: ')
3 compra = int(input('--> '))
4 print('Comprado-->', compra)
5 print('*'*60 + '\n')
6
7 if compra > 200_000:
8     descuento = 0.8
9 elif compra > 100_000:
10     descuento = 0.85
11 elif compra > 50_000:
12     descuento = 0.90
13 elif compra > 25_000:
14     descuento = 0.95
15 elif compra > 0:
16     descuento = 1
17 else:
18     descuento = 0
19
20 subtotal = compra
21 valor_descuento = compra - (compra * descuento)
22 total = compra * descuento
23 print(f'Subtotal--> ${subtotal:,.0f}')
24 print(f'Descuento--> ${valor_descuento:,.0f}')
25 print(f'Total a Pagar--> ${total:,.0f}')
```



*Gracias*

*Espacio para preguntas*

Universidad Nacional de Colombia

---