

**CARRERA PROFESIONAL**

# DESARROLLO DE SISTEMAS DE INFORMACIÓN

## FUNDAMENTOS DE PROGRAMACIÓN



## TEMA: HERRAMIENTAS DE PROGRAMACIÓN



## 1. PREGUNTA DE INICIO

**¿Por qué un Arquitecto hace una maqueta de cartón antes de construir el edificio de concreto?** Porque el cartón es barato, rápido de cortar y fácil de corregir. El concreto es caro y duro. **PSeInt y los Diagramas** son tu cartón. **Java, C# y Python** son el concreto. Si cometes un error lógico en la maqueta (PSeInt), lo arreglas en 10 segundos. Si lo cometes en el edificio (Java), puedes tardar horas depurando ("Debugging") código complejo. Hoy aprenderás a ser Arquitecto de Software, no solo albañil de código.



## 2. PSEINT: TU PRIMER IDE (PSEUDOCÓDIGO)

### 2.1. ¿POR QUÉ PSEUDOCÓDIGO?

Es un lenguaje intermedio entre el humano (Español) y la máquina (Binario).

- **El Objetivo:** Centrarte en el **Algoritmo** (la receta), no en la Sintaxis (la gramática).
- **Entorno de Trabajo:**
  - **Configuración:** Usaremos el **Perfil Estricto**.
  - *¿Por qué?* Porque el perfil "Flexible" te permite ser desordenado (no declarar variables). El perfil "Estricto" te obliga a definir tipos de datos, preparándote para la disciplina de lenguajes reales como Java o C++.

### 2.2. LA VENTAJA OCULTA

PSelnt te permite ejecutar paso a paso (Step-by-Step). Verás cómo la flecha verde salta de línea en línea y cómo cambian los valores de las variables en tiempo real. Esto es vital para desarrollar tu "ojo de programador".

## 3. HERRAMIENTAS DE DIAGRAMACIÓN: LUCIDCHART / DRAW.IO

"Si no puedes dibujarlo, no lo entiendes".

### 3.1. EL FLUJOGRAMA (FLOWCHART)

Es la representación visual del algoritmo.

- **Ovalo:** Inicio/Fin.
- **Rombo:** Decisión (If / Else). Es el símbolo más importante. Aquí el camino se bifurca.
- **Rectángulo:** Proceso (Cálculo).
- **Paralelogramo:** Entrada/Salida de datos.

### 3.2 ¿POR QUÉ DRAW.IO?

Es gratuito, web y se integra con Google Drive.

- **Uso Profesional:** En la vida real, antes de programar una "Pasarela de Pagos", tendrás que dibujar el diagrama de estados para que el cliente lo apruebe. El código viene después.

ResearchGate



Un estudio académico sobre la enseñanza de la programación demostró que los estudiantes que aprendieron primero con Pseudocódigo superaron significativamente a los que empezaron directamente con Java, especialmente en la resolución de problemas complejos. El pseudocódigo actúa como un "puente cognitivo" que permite al cerebro concentrarse en la lógica difícil sin saturarse con la sintaxis del idioma inglés/técnico. Además, en la industria, se estima que el ratio de tiempo que un programador pasa leyendo código versus escribiéndolo es de 10 a 1. Aprender a leer lógica clara (como en PSeInt) es más rentable que aprender a escribir rápido.

## 4. PREGUNTAS CON RESPUESTAS GUIADAS

### 1. ¿PSeInt es un lenguaje de programación real?

No. Es un intérprete educativo. No puedes crear Facebook con PSeInt. Pero la lógica que aprendes en PSeInt (Si-Entonces, Mientras, Para) es idéntica en el código de Facebook (PHP/Hack). Solo cambian las palabras, no la lógica.

### 2. ¿Por qué tengo que dibujar el diagrama si ya sé qué hacer?

Porque tu memoria es frágil.

- El Diagrama es documentación. Si te enfermas y otro programador debe seguir tu trabajo, el diagrama le explica en 1 minuto lo que el código le explicaría en 1 hora.

### 3. ¿Qué hago con los archivos .psc de PSeInt?

Guárdalos en una carpeta organizada por temas (/Semana02/Condicionales/). Acostúmbrate al orden. En el futuro, esa carpeta será un Repositorio Git. El orden es la primera virtud del desarrollador Senior.


### 4. ¿Cuándo dejo de usar PSeInt?

Cuando tu cerebro pueda visualizar el "Rombo" (la decisión) y el bucle sin necesidad de dibujarlo. PSeInt son las rueditas de la bicicleta; cuando tengas equilibrio, las quitaremos y pasaremos a Python.

## 5. APLICACIÓN PRÁCTICA DEL CONTENIDO

**Caso: "La Casa de Cambio de la Av. El Sol"**

**Contexto:** Debes diseñar la lógica para una máquina de venta de boletos en la entrada de un museo.



**Contexto:** Debes diseñar la lógica para una máquina de venta de boletos en la entrada de un museo.

**Reglas de Negocio:**

1. Si el turista es Cusqueño -> Entrada Gratis.
2. Si no es Cusqueño pero es Niño (<12 años) -> Paga 50%.
3. Si no -> Paga 100%.

**Solución en Diagrama (Draw.io):** Dibujas dos Rombos anidados (decisiones en cascada).

1. Primer Rombo: ¿Es Cusqueño?
  - o SÍ -> Precio = 0
  - o NO -> Segundo Rombo: ¿Edad < 12?
    - SÍ -> Precio = Tarifa \* 0.5
    - NO -> Precio = Tarifa

**Solución en PSeInt:**

Fragmento de código

Si origen = "Cusco" Entonces

Escribir "Bienvenido, paisano. Pase gratis."

SiNo

Si edad < 12 Entonces

Escribir "Tarifa Niño: " + (tarifa \* 0.5)

SiNo

Escribir "Tarifa Plena: " + tarifa

FinSi

FinSi

Lección: PSeInt te permite validar esta lógica compleja anidada antes de pelearte con la sintaxis de if/else y las llaves {} de Java.

## 6. CONCLUSIONES

1. **Disciplina:** El entorno de trabajo ordenado y el uso del perfil estricto en PSeInt te preparan para la rigurosidad de la industria.
  2. **Abstracción:** Los diagramas te enseñan a pensar en flujos de datos, no en líneas de código.
  3. **Puente:** No veas a PSeInt como un retraso, sino como un impulso. Lo que aprendas bien aquí, lo aplicarás por el resto de tu carrera en cualquier lenguaje.
- 