INSTITUTO KHIPU

SEMESTRE I
SESIÓN 02

CARRERA PROFESIONAL

DESARROLLO DE SISTEMAS DE INFORMACION

FUNDAMENTOS DE PROGRAMACIÓN

TEMA:

METODOLOGÍA E-P-S
ANÁLISIS DE PROBLEMAS

1. PREGUNTA DE INICIO

Si metes naranjas podridas en una máquina de jugo de última generación, ¿qué sale? ¿Sale jugo rico? No. Sale jugo podrido.

En sistemas, esto se llama **GIGO (Garbage In, Garbage Out): Basura entra, basura sale.** No importa qué tan bueno sea tu código (Proceso); si no identificaste bien los datos necesarios (Entrada), tu programa no sirve.

Hoy aprenderemos a diseñar la "máquina de jugo" correcta definiendo qué entra, qué pasa adentro y qué sale.

2. LA METODOLOGÍA E-P-S: LA RECETA DEL ALGORITMO

Todo software, desde una calculadora hasta Instagram, funciona bajo este esquema universal.

2.1. E - Entrada (Input)

- **Definición:** Son los "Ingredientes". Los datos que el usuario debe darte o que el sistema debe leer.
- **El Reto:** Identificar qué necesitas. Si quieres calcular el área de un triángulo, tus entradas OBLIGATORIAS son *Base* y *Altura*. Sin eso, no hay magia.

2.2. P - Proceso (Process)

- **Definición:** Es la "Cocina". Las fórmulas, cálculos y lógica que transforman la entrada.
- **El Reto:** El orden importa. No puedes dividir entre cero.
    - ***Ejemplo:*** Area = (Base * Altura) / 2.

2.3. S - Salida (Output)

- **Definición:** Es el "Plato Servido". La información útil que entregas al usuario.
- **El Reto:** El formato. No basta con mostrar un número ("50"); debes decir "El área es 50 metros cuadrados".

3. IDENTIFICACIÓN DE VARIABLES: LAS CAJAS

Una variable es un espacio en la memoria RAM donde guardas un dato. Imagínalas como **Cajas Etiquetadas**.

- **El Nombre (Identificador):** Debe ser claro. x es un mal nombre. precioUnitario es un buen nombre.
- **El Tipo de Dato:**
  - **Entero (int):** Para contar personas (no hay media persona).
  - **Real (float/double):** Para dinero o peso (10.50 soles).
  - **Cadena (string):** Para nombres ("Juan Pérez").
  - **Booleano:** Para decisiones (Verdadero/Falso).

El Chef

Entrada: Harina, Huevos, Leche (Variables).
Proceso: Mezclar, Batir, Hornear a 180° (Algoritmo).
Salida: Un Pastel (Resultado).

Si en la **Entrada** te olvidas del azúcar, el **Proceso** funciona perfecto (el horno calienta bien), pero la **Salida** es un pastel horrible.

Moraleja: El error no fue del horno (compilador), fue del chef (tú) al planificar los ingredientes.

EL COSTO DE NO PENSAR

Según el **NIST (National Institute of Standards and Technology)**, corregir un error de lógica (mal análisis E-P-S) durante la fase de diseño "en papel" cuesta **1 dólar.** Corregir ese mismo error cuando el sistema ya está programado y en producción cuesta **100 dólares**. La industria paga por desarrolladores que piensan antes de escribir, no por los que escriben rápido y "rompen" cosas.

NIST
National Institute of Standards and Technology

4. PREGUNTAS CON RESPUESTAS GUIADAS

1. ¿Una constante es una variable?
Técnicamente ocupan memoria igual, pero conceptualmente son opuestas.

- **Variable:** Su valor cambia (ej. *TipoDeCambio* del dólar).

- **Constante:** Su valor NUNCA cambia durante la ejecución (ej. *PI = 3.1416 o IGV = 0.18*). Usar constantes hace tu código más seguro.

2. ¿Cómo sé qué tipo de dato usar para un DNI?

Pregunta trampa. Parece un número ("44556677"), pero NO se hacen operaciones matemáticas con él (no sumas DNI + DNI).

- ***Respuesta:*** Úsalo como Cadena (String). Así evitas que el sistema borre el cero a la izquierda si el DNI empieza con 0 ("0123..." -> "123...").

3. ¿El "Proceso" es visible para el usuario?

No. Es una "Caja Negra". El usuario mete datos y ve resultados. Como programador, tu deber es que esa caja negra sea eficiente (rápida) y maneje errores (que no se cuelgue si meten letras en vez de números).

4. ¿Qué es una "Prueba de Escritorio"?

Es ejecutar tu lógica E-P-S en papel, paso a paso, con valores de prueba, antes de tocar la computadora. Es la herramienta más barata y efectiva para encontrar errores lógicos.

5. APLICACIÓN PRÁCTICA DEL CONTENIDO

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

Contexto: Debes crear un programa simple para que el cambista calcule cuántos dólares entregar al turista.

Análisis E-P-S:

1. Entrada (¿Qué necesito?):

- Monto en Soles a cambiar (*montoSoles* - Tipo: Real).
- Tipo de Cambio del día (*tipoCambio* - Tipo: Real).

2. Proceso (¿Cómo lo calculo?):

- *montoDolares* = *montoSoles* / *tipoCambio*
- *Regla de Negocio*: Si el *tipoCambio* es cero o negativo, mostrar error (validación).

3. Salida (¿Qué muestro?):

- "Usted recibe: " + *montoDolares* + " USD".

Identificación de Variables:

- *montoSoles*: Variable de entrada (cambia con cada cliente).
- *tipoCambio*: Variable de entrada (cambia cada día).
- *montoDolares*: Variable de salida (resultado calculado).

Resultado: Si el estudiante intenta programar esto sin definir que el tipo de cambio es divisor, su programa fallará cuando alguien ponga "0". El análisis E-P-S previo evita el desastre.

6. CONCLUSIONES

1. Abstracción: Programar es el arte de ignorar los detalles innecesarios y enfocarse en los datos clave (Entrada) para resolver una necesidad (Salida).

2. Orden: E-P-S es secuencial. No puedes mostrar la salida si no has procesado, y no puedes procesar si no tienes entrada. Este pensamiento estructurado es la base de la algoritmia.

3. Datos: Las variables son la sangre de tu programa. Definirlas mal (usar Entero para dinero) causará pérdidas financieras (pérdida de centavos).

7. MAPA MENTAL