












Última modificación oficial: 21 mar 2026

Para una mejor experiencia usamos la suite de Google.

Técnicas usadas: Bulletpoints + Estilo Cornell + Tablas y gráficos visuales

Puedes borrar el contenido de la sección de palabras clave o anotaciones para que puedas escribir las tuyas. ¡Éxitos!

CONTENIDO

S02: Pensamiento Computacional y Resolución de Problemas	2
① ¿POR QUÉ PENSAR COMO MÁQUINA TE HACE MEJOR INGENIERO?	2
 Lógica del concepto:	2
 Reglas / Condiciones:	2
 Resumen Express:	3
② ¿CÓMO DESCUARTIZAR UN PROBLEMA SIN MORIR EN EL INTENTO?	3
 Lógica del concepto:	3
 Mecanismo / Proceso:	3
 Resumen Express:	4
③ ¿CÓMO IGNORAR EL RUIDO Y CERRAR LA SOLUCIÓN?	4
 Lógica del concepto:	4
 Reglas / Condiciones: (Para que un Algoritmo sea válido)	4
 Mecanismo / Proceso: (Ejemplo - Diseñar un Cajero Automático)	5



S02: Pensamiento Computacional y Resolución de Problemas

FICHA TÉCNICA:

- Curso:** Inteligencia Artificial Aplicada
- Unidad:** Fundamentos y Estructura Jerárquica de la IA
- Objetivo:** Aplicar los 4 pilares del pensamiento computacional para estructurar, simplificar y resolver problemas de ingeniería complejos.

PALABRAS CLAVE / ANOTACIONES

Qué tal! Espero te sirva este material. Si tienes dudas u observaciones, date una pasada por la página web y contáctame!

[PROGRESO]:  13% Completado (*Semana 2 de 16*)

¿Pensabas que el "Pensamiento Computacional" era saber escribir código en una pantalla negra tipo Matrix? Falso. Es la forma en la que tu cerebro resuelve problemas diarios (como preparar un lomo saltado o cruzar la calle), pero hackeada a nivel god para aplicarla en la ingeniería.

Terminarás esta lección con el power de agarrar el problema más gigante, abstracto y caótico de algún proyecto y destrozarlo en piececitas tan fáciles de resolver que hasta un niño (o una máquina) podría hacerlo.

1 ¿POR QUÉ PENSAR COMO MÁQUINA TE HACE MEJOR INGENIERO?

El Pensamiento Computacional (PC)

Lógica del concepto:

El Pensamiento Computacional no es exclusivo de las máquinas. Es un proceso mental humano. Consiste en formular problemas de una manera tan clara y estructurada, que sus soluciones puedan ser representadas como secuencias de instrucciones que un procesador de información (una IA, una computadora o incluso otra persona) pueda ejecutar ciegamente.

Es chequear el problema y crinear una solución fácil y con pasos para que tu sobrino de 6 años lo pueda seguir

Reglas / Condiciones:

- No requiere una computadora para existir (puedes aplicarlo con papel y lápiz).
- Es la habilidad previa y obligatoria antes de escribir una sola línea de código o armar un prompt complejo.
- Es universal: Se aplica para diseñar un puente, crear un software o planear la logística de un evento.

De hecho es algo que podrías aplicar a cualquier cosa, en cualquier lugar y en cualquier momento

La Trampa: Cuidado en las evaluaciones! El error clásico es marcar en el examen que "Pensamiento Computacional = Programación". **Falso.** La



programación es dictar código a la máquina (el final del camino). El Pensamiento Computacional es *razonar* en la solución antes de dictarla.

Resumen Express:

El Pensamiento Computacional es el traductor universal: convierte la complejidad del mundo real en instrucciones simples que la tecnología puede procesar.

2 ¿CÓMO DESCUARTIZAR UN PROBLEMA SIN MORIR EN EL INTENTO?


Los 4 Pilares del Pensamiento Computacional

Lógica del concepto:

Todo problema inmenso se rinde ante este *framework* de 4 pasos secuenciales. No te saltes ninguno, van en orden lógico.

Mecanismo / Proceso:

1. **Descomposición:** Romper el caos en pedazos manejables.
2. **Reconocimiento de Patrones:** Buscar "lo repetido" o lo que ya conoces.
3. **Abstracción:** Eliminar la basura visual y quedarse con el jugo.
4. **Algoritmo:** Crear la receta mágica paso a paso.

 **Mnemotecnia: D-P-A-A** (Da Pasos A-Avanzar)
Descomposición, Patrones, Abstracción, Algoritmos.

 **Tabla Comparativa:** (La Analogía de la "Pizzería Locasa")

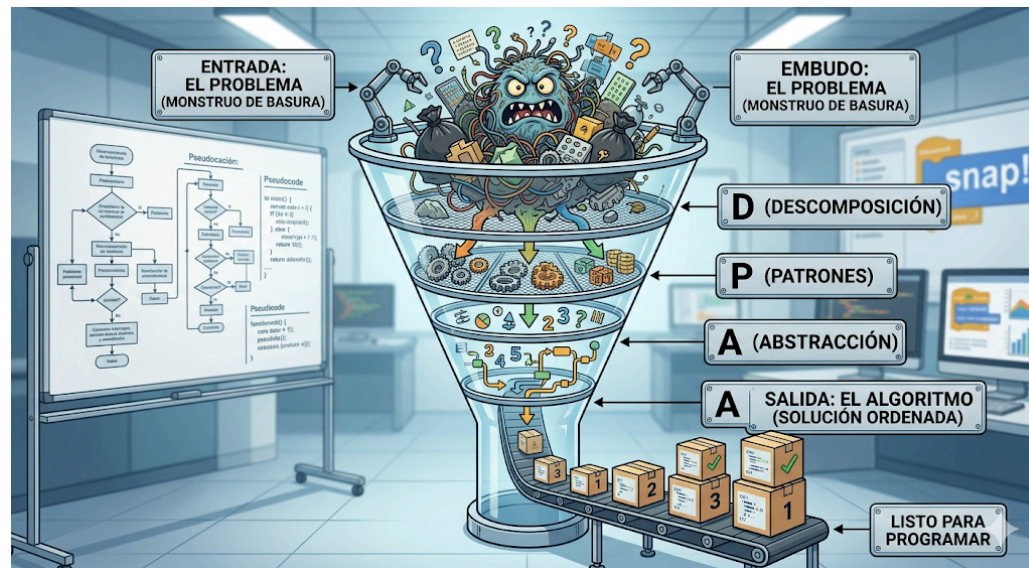
Pilar	¿Qué hace tu brain?	Ejemplo: Pizzería
1. Descomposición	Divide el problema complejo en subproblemas pequeños, precisos y aislados.	No ves "una pizza", ves: la masa, la salsa, el queso y la caja. Lo separas por estaciones.
2. Patrones	Identifica similitudes o comportamientos iterativos dentro del problema (o en tu pasado).	Notas que, sin importar el sabor, la base SIEMPRE se repite: Masa → Salsa → Queso
3. Abstracción	Ignora el "ruido" irrelevante y se concentra SOLO en los datos críticos para que funcione.	No importa el color de la caja ni el nombre del repartidor; solo te

Son los pasos para resolver cualquier problema, trata de hacerlo siempre de la forma más sencilla posible y luego le vas agregando o modificando pasos



		importa que el horno esté a 200°C.
4. Algoritmo	Crea una serie de instrucciones automatizables.	La receta pegada en la pared de la cocina, enumerada estrictamente del paso 1 al 10.

Sugerencia de Ilustración:



>>>> Yo los sábados a las 7am

Resumen Express:

Para vencer a un monstruo gigante, **Divídelo**, encuéntrale sus **Puntos débiles** repetidos, **Aísla** lo que importa y crea un **Ataque** paso a paso. (D-P-A-A).

También puedes aplicar pasos para la resolución de problemas:

- Detectar el problema
- Observar el problema
- Proponer posibles soluciones
- Ejecutar las soluciones
- Repetir

3 ¿CÓMO IGNORAR EL RUIDO Y CERRAR LA SOLUCIÓN?

Abstracción profunda y Algoritmos

Lógica del concepto:

En ingeniería, nos ahogamos en datos. La Abstracción es el arte de decir "esto no me importa". Una vez que tienes solo los datos vitales, construyes el Algoritmo (el producto final del Pensamiento Computacional).

Reglas / Condiciones: (Para que un Algoritmo sea válido)

- **Ordenado:** La secuencia debe tener lógica estricta (no puedes freír el huevo antes de romperlo).



- **Claro:** Sin ambigüedades. Las instrucciones deben ser a prueba de tontos.
- **Preciso:** Acciones y movimientos concretos y medibles.
- **Finito:** Tiene que tener un inicio y, sobre todo, un FIN (lograr el objetivo o detenerse).

⚠ **La Trampa:** Esta es pregunta fija: Si en un caso práctico te piden optimizar la ruta de unos camiones de reparto usando el pilar de **Abstracción**, y tú incluyes en tu análisis "el color del logo del camión" o "la estética del puente"... Fuiste! Abstraer significa eliminar variables inútiles. Solo importan pesos, distancias y resistencia.

⚙ **Mecanismo / Proceso: (Ejemplo - Diseñar un Cajero Automático)**

1. **D:** Separo el cajero en: Lector de tarjeta, teclado, dispensador, pantalla.
2. **P:** Noto que todos los usuarios pasan por la misma validación de PIN de 4 dígitos.
3. **A:** Ignoro el color de la ropa del cliente; me enfoco solo en su saldo y número de cuenta.
4. **A:** Creo los pasos: 1) Pedir PIN, 2) Validar PIN, 3) Mostrar saldo, 4) Entregar dinero.

💡 **Resumen Express:**

Abstraer es quitarle toda la grasa al problema; el Algoritmo es puro músculo trabajando de forma finita, clara y precisa.

Genial! Si llegaste hasta aquí es más que suficiente para entender la semana. Pero si quieres profundizar alguna parte o concepto puedes visitar los recursos adicionales en www.ingenierolocaso.com en la sección de CONTENIDO.

Utiliza ese material como guía o complemento, de todas formas tienes que darle una repasada, ESE EXÁMEN NO SE VA A PASAR POR ARTE DE MAGIA. TU PUEDES!

A partir de acá viene en el examen

Síguenos en nuestras redes para enterarte de todas las novedades!