



Última modificación oficial: 26 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

S03: Introducción a los Algoritmos	2
① ¿Por qué tu cerebro es el peor enemigo de un algoritmo perfecto?	2
Lógica del concepto:	2
Reglas / Condiciones (Las 3 Leyes de Oro):	2
La Trampa:	2
Mnemotecnia: "P-F-D" (Papas Fritas Deliciosas)	3
Resumen Express:	3
② ¿Qué tienen en común un juego de fresa y un sistema de reconocimiento facial?	3
Mecanismo / Proceso (Modelo EPS):	3
Lógica causal:	3
Sugerencia de Ilustración:	4
Resumen Express:	4
③ ¿Cómo pasar de una idea abstracta a un modelo que funcione?	4
Mecanismo / Proceso de Diseño:	4
Reglas de Representación Visual:	4
Tabla Comparativa: Representación	5
Resumen Express:	5



S03: Introducción a los Algoritmos

FICHA TÉCNICA:

- Curso:** Inteligencia Artificial Aplicada
- Unidad:** Fundamentos de Pensamiento Computacional e IA
- Objetivo:** Comprender la definición, características y componentes de un algoritmo para estructurar soluciones eficientes en el campo de la ingeniería.

[PROGRESO]:  19% Completado (*Semana 3 de 16*)

Si creías que una receta de cocina es el ejemplo perfecto de un algoritmo, estás a un paso de fallar tu próximo examen de lógica: las recetas suelen ser ambiguas, y un algoritmo **jamás** puede serlo.

Al terminar de leer este documento, dejarás de dar "instrucciones sueltas" y empezarás a diseñar procesos blindados que cualquier máquina o humano podrá ejecutar con éxito garantizado.

1 ¿Por qué tu cerebro es el peor enemigo de un algoritmo perfecto?

Definición y Características de Oro

Para que una secuencia de pasos sea considerada un señor Algoritmo en el mundo de la computación y la ingeniería, no basta con que sea una lista de tareas. Debe ser un conjunto de instrucciones **definidas** y **ordenadas**.

Lógica del concepto:

Un algoritmo es el puente entre un problema complejo y una solución automatizada. Es el resultado final del pensamiento computacional después de haber pasado por la descomposición y la abstracción.

Reglas / Condiciones (Las 3 Leyes de Oro):

- **Preciso:** Cada paso debe indicar exactamente qué hacer. No se permiten "un poquito de sal" o "mueve la pieza a donde creas".
- **Finito:** Debe tener un inicio y, obligatoriamente, un **final**. Un proceso que nunca termina es un error de bucle, no un algoritmo.
- **Definido:** Si metes los mismos datos 100 veces, debes obtener el **mismo** resultado las 100 veces.

La Trampa:

En la pc, te pondrán instrucciones como: "1. Piensa una idea, 2. Escribe algo interesante". **¡Eso NO es un algoritmo!** Es subjetivo. Lo "interesante" para ti no lo es para una computadora. Para ser algoritmo, debe ser objetivo y exacto.

PALABRAS CLAVE / ANOTACIONES

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

Lo que vimos en el anterior resumen locaso (DPAA)

Los algoritmos son instrucciones definidas y ordenadas que realizan una tarea específica

WASAAAA



Mnemotecnia: "P-F-D" (Papas Fritas Deliciosas)

- Preciso.
- Finito.
- Definido.

Resumen Express:

Característica	¿Qué significa?	Ejemplo en Ajedrez
Preciso	Sin dudas ni ambigüedad.	"Mover peón a E4".
Finito	Tiene un fin claro.	Lograr el Jaque Mate.
Definido	Siempre igual resultado.	Misma jugada, misma posición final.




Trata de ser lo más exacto posible en cuanto a esto para que no te resten puntos.

¿Qué tienen en común un jugo de fresa y un sistema de reconocimiento facial?

Componentes de un Algoritmo (Modelo EPS)

Todo proceso algorítmico, por más complejo que sea (como el algoritmo de recomendación de Netflix), se divide en tres bloques fundamentales de energía e información.

Mecanismo / Proceso (Modelo EPS):

1.  **Entrada (Input):** Son los datos iniciales o ingredientes que necesitamos para trabajar.
2.  **Proceso (Process):** La secuencia lógica de pasos que transforma los datos de entrada.
3.  **Salida (Output):** El resultado final, la solución al problema planteado.

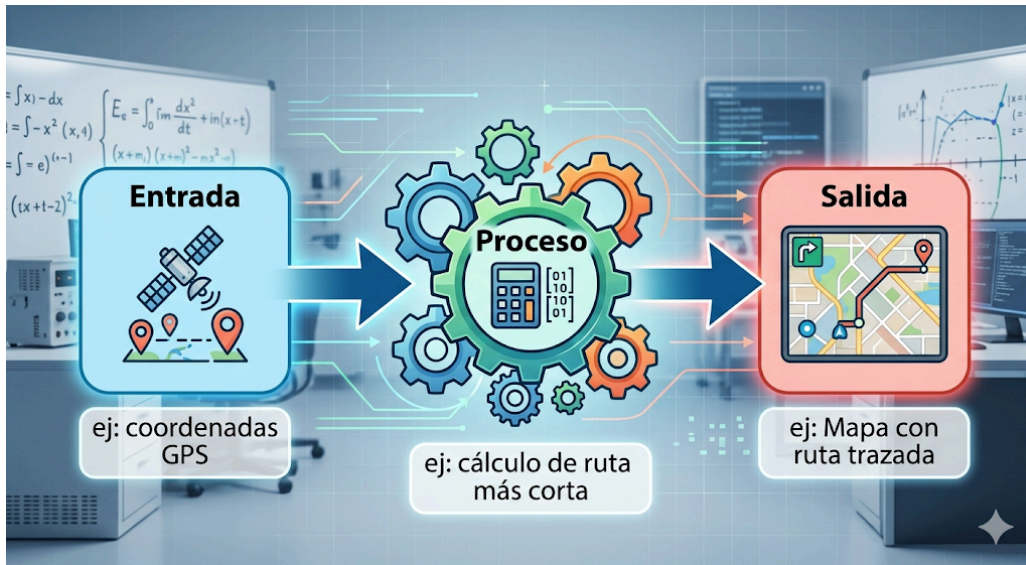
Lógica causal:

Si la **Entrada** es basura, el **Proceso** procesará basura y la **Salida** será basura (*Garbage In, Garbage Out*). La calidad de tu algoritmo depende de qué tan bien definas qué entra y cómo se transforma.

Luego de eso está la retroalimentación para que el algoritmo sea mejor o se ajuste a los objetivos definidos



🎨 Sugerencia de Ilustración:



En este caso el feedback vendría a ser las reseñas o comentarios de los usuarios para que los programadores hagan modificaciones al algoritmo para que sea más exacto y mitigar los errores.

💡 Resumen Express:

- **Entrada:** Datos crudos.
- **Proceso:** "La magia" o lógica matemática.
- **Salida:** Información útil o acción ejecutada.

3 ¿Cómo pasar de una idea abstracta a un modelo que funcione?

📌 Diseño y Representación de Algoritmos

Diseñar no es solo escribir. Es una metodología de ingeniería que evita que pierdas tiempo programando algo que no funciona desde la base.

⚙️ Mecanismo / Proceso de Diseño:

1. **Análisis:** Entender el problema a fondo (¿Qué me piden? ¿Qué datos tengo?).
2. **Diseño:** Crear la lógica (aquí es donde usas diagramas de flujo o pseudocódigo).
3. **Implementación:** Traducir esa lógica a un lenguaje que la máquina entienda (ej: Python o Snap!).

📏 Reglas de Representación Visual:

- **Diagramas de Flujo:** Usan figuras geométricas (Óvalos para inicio/fin, Rombo para decisiones).
- **Pseudocódigo:** Un lenguaje intermedio, más humano que el código pero más estructurado que el habla común.

Esto es una pequeña introducción al diseño de algoritmos, en la siguiente semana veremos cómo usar los diagramas de flujo y pseudocódigo en PSaint.



Tabla Comparativa: Representación

Método	Ventaja	Desventaja
Diagrama de Flujo	Muy visual, fácil de entender el camino.	Difícil de actualizar si el proceso es gigante.
Pseudocódigo	Más rápido de escribir y fácil de pasar a código real.	Menos intuitivo para personas no técnicas.

Resumen Express:

Diseñar un algoritmo requiere: **Analizar** (qué hay) → **Diseñar** (cómo se hace) → **Probar** (si funciona). No te saltes el análisis o tu código será un desastre de "espagueti".

CONSEJO DE AMIGO: Para tu práctica de la semana (la del cajero automático o la historia corta), recuerda aplicar siempre los **4 pilares** de la semana pasada antes de escribir tu primer paso del algoritmo. ¡El éxito está en la estructura!

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!

Síguenos en nuestras redes para enterarte de todas las novedades!