

# Estrategia incremental

**Concepto General:** Es un modelo o enfoque el cual tiene como objetivo lograr un crecimiento progresivo de funcionalidad o productividad, se refiere a transferir sólo los datos o registros nuevos o que hayan sido modificados, en lugar de mover todos los datos va a tomar únicamente estos y esto reducirá los gastos en tiempo y procesos.

Entendí que este concepto es como un modelo el cual se usa tanto en procesos de la ingeniería o en la programación como en cosas más cotidianas, a la vez que este consta de que en cada entrega que se haga de este cada versión debe ser mejor y mostrar una evolución notable en comparación a la anterior, también entiendo que este modelo se usa mayormente en bases de datos en las cuales hayan muchos cambios constantes y que se requiera una buena utilización de recursos como tiempo y manejo de datos.

**Pregunta:** ¿Qué funciones en Python nos ayudarían a implementar esta estrategia?

**Fuentes:**

- Garzón Chris. 12 de Abril de 2024. Data Engineering: Incremental Data Loading Strategies [URL](#)
- Anna Pérez. 29 de Diciembre de 2025. Características y fases del modelo incremental [URL](#)

# Análisis Asintótico

**Concepto General:** Es cómo se evalúa el rendimiento de un algoritmo en función del tamaño que tenga de entrada y el tiempo de ejecución, es el mejor método para analizar algoritmos.

Entendí que esta es la manera de evaluar los algoritmos y esto es mediante el uso de temas matemáticos como lo sería: omega, theta, O y el uso de logaritmos. También que hay muchos tipos de algoritmos y que cada uno es mejor que otro dependiendo de muchos factores como lo podrían ser el tipo de entrada o el tipo de uso que se le dé a dichos algoritmos.



**Pregunta:** ¿Cómo sabemos que opción entre O, Omega o Theta debemos usar ?

**Fuente:**

- Geek For Geeks. 29 de Enero de 2026 [URL](#)

## Recursión

**Concepto General:** Es una técnica de programación en la que una función se invoca a sí misma para resolver un problema. Se descompone un problema complejo en subproblemas más simples y así cada llamada recursiva resuelve una instancia más pequeña del problema original.

Entendí que la recursión es dividir un problema en subproblemas y así este se volvera mas facil de resolver y a la vez se usarán las soluciones de estos subproblemas para resolver otros subproblemas y funciona con un caso base y un caso recursivo, en el caso base sería como una bandera la cual indique cuándo se detendrá la recursión y el caso recursivo va a ser la función que se llame a sí misma.

La recursión tiene ventajas de fácil entendimiento y que será usado en una gran variedad de problemas comunes lo cual les facilitará, en cuestión de rendimiento puede que llegue a tener sobrecarga por tener muchas llamadas lo cual no es tan eficiente y usara gran espacio de memoria al guardar cada llamado que se haga una función a si misma.

(Pila de llamadas: Es una estructura de datos LIFO (último en entrar será el primero en salir) funciona como una lista)

(Push o apilado: Cada vez que la función se llame a sí misma se crea un nuevo marco de pila que es como una posición en una lista la cual se pondrá en la parte superior y guardará parámetros y variables )

(Pop o Desapilado: Cuando se llega al caso base o a la bandera la función se retorna o se tomará en el llamado anterior y así libera su memoria )

**Pregunta:** No me quedó claro al 100% la estructura LIFO

**Fuente:**

Hash Mishra. 1 de Julio de 2024. Recursión, diseño y análisis de algoritmos [URL](#)