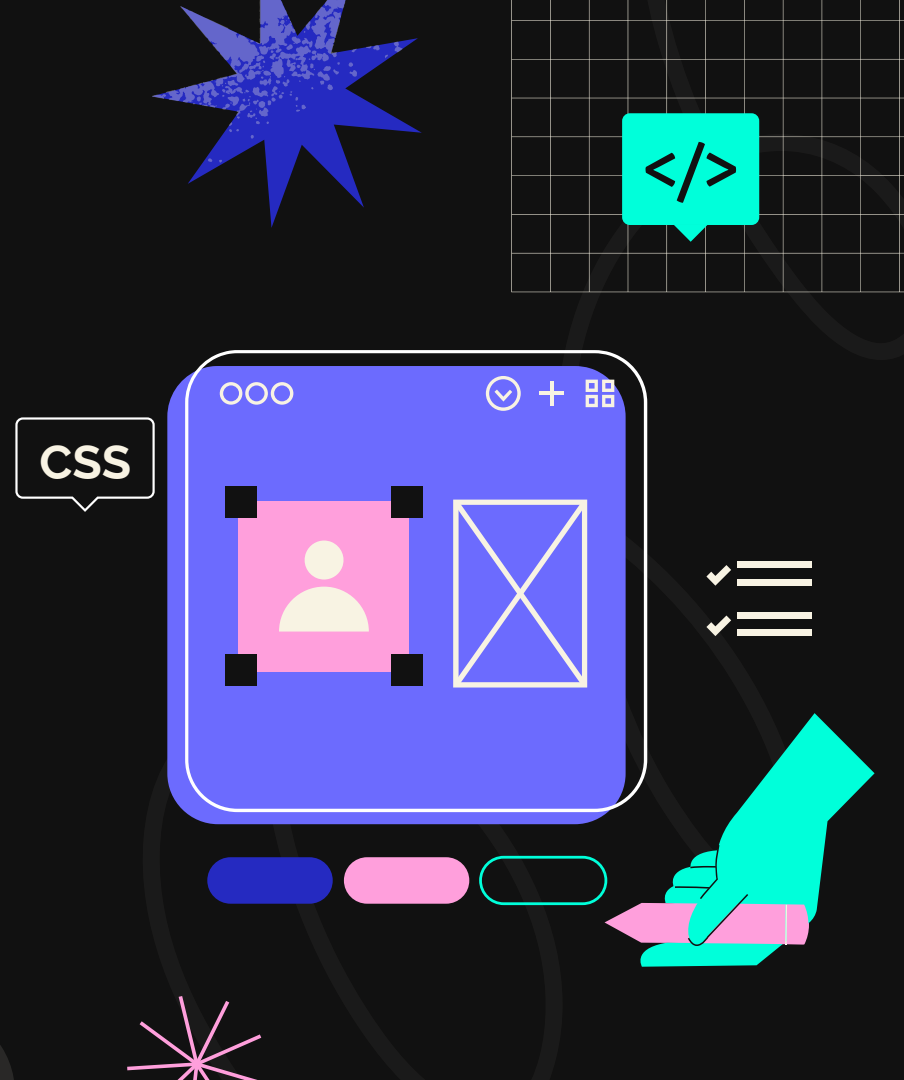


Algoritmos y Estructuras de Datos

UCAB



Quien soy?

Ingeniero en Informática de la UCAB

Presidente de Jayktec

Profesor

Poeta en esencia

Obsesionado con las ideas

Lector empdernido

www.yileone.com

yileone@gmail.com

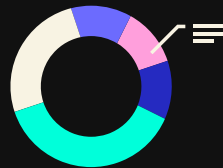
+584123172800



CSS

Whoa!

Y ustedes quienes son?



Plan de este semestre

01

Lenguajes de programación
estructurada

02

Punteros y gestión de
memoria.

03

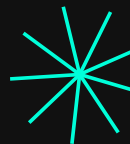
Estructuras de datos
dinámicas secuenciales:
Listas.

04

Estructuras de datos
dinámicas Jerárquicas:
Árboles

05

Estructuras de datos
dinámicas Jerárquicas:
Grafos



Contenidos de este curso



UNIDAD 1: Lenguajes de programación estructurada para la gestión dinámica de memoria.

1. Tipos de lenguajes de programación.
2. Aplicaciones.
3. Compiladores.
4. Interpretes.
5. Estructura de un programa, tipos de datos, comentarios, operadores, estructuras selectivas, estructuras iterativas.
6. Funciones.
7. Uso de librerías.

UNIDAD 2: Punteros y gestión de memoria.

1. Gestión de memoria dinámica: direcciones de memoria y apuntadores.
2. Definición de apuntadores y su utilidad.
3. Operadores: Dirección y contenido.
4. Operaciones básicas con apuntadores.
5. Apuntadores y arreglos.
6. Gestión de memoria dinámica.
7. El operador new.
8. Arreglos de datos dinámicos y arreglos dinámicos.



Contenidos de este curso

UNIDAD 3: Estructuras de datos dinámicas secuenciales: Listas.	<ol style="list-style-type: none">1. Representaciones estáticas Vs. Representaciones dinámicas.2. TDA Lista: simples, dobles, circulares y multi-enlazadas.3. TDA: Pila.4. TDA: Cola.
UNIDAD 4: Estructuras de datos dinámicas Jerárquicas: Árboles.	<ol style="list-style-type: none">1. Árboles General.2. Árboles Binario.3. Algoritmos de recorrido de árboles: preorden, inorden, postorden.4. Árboles Binarios de Búsqueda.5. Árboles AVL.6. Árboles 2-3.7. Árboles B-TREE.8. Árboles Rojo-negro.9. Algoritmos de ordenamiento basados en árboles (Montículos): Heapsort, mergesort.10. Caso de estudio: árbol de Merkle o árbol Hash.
UNIDAD 5: Estructuras dinámicas jerárquicas: Grafos.	<ol style="list-style-type: none">1. Grafos y sus aplicaciones.2. Representación de grafos: Matriz de adyacencia y matriz de incidencia, listas enlazadas.3. Recorridos de Grafos: Algoritmos DFS, Algoritmo BFS, Algoritmo de Dijkstra, Algoritmo Roy Warshall.4. Algoritmos de recubrimiento mínimo: algoritmo de Kruskal y algoritmo de Prim.5. Coloración de grafos: número cromático, algoritmo de Greedy y algoritmo de Brelaz.

Contenidos de este curso

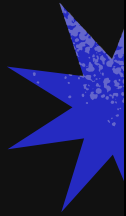


UNIDAD 1: Lenguajes de programación estructurada para la gestión dinámica de memoria.

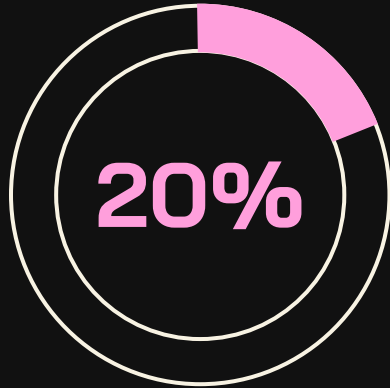
1. Tipos de lenguajes de programación.
2. Aplicaciones.
3. Compiladores.
4. Interpretes.
5. Estructura de un programa, tipos de datos, comentarios, operadores, estructuras selectivas, estructuras iterativas.
6. Funciones.
7. Uso de librerías.

UNIDAD 2: Punteros y gestión de memoria.

1. Gestión de memoria dinámica: direcciones de memoria y apuntadores.
2. Definición de apuntadores y su utilidad.
3. Operadores: Dirección y contenido.
4. Operaciones básicas con apuntadores.
5. Apuntadores y arreglos.
6. Gestión de memoria dinámica.
7. El operador new.
8. Arreglos de datos dinámicos y arreglos dinámicos.

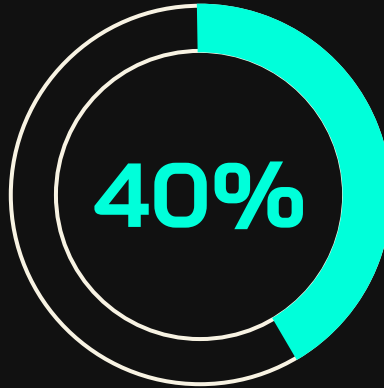


Evaluación



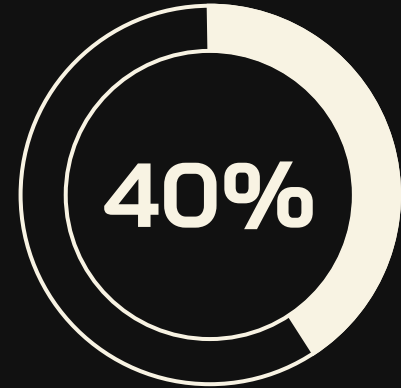
Exámenes Cortos

2 exámenes de teoría
2 talleres prácticos
5 % cada uno



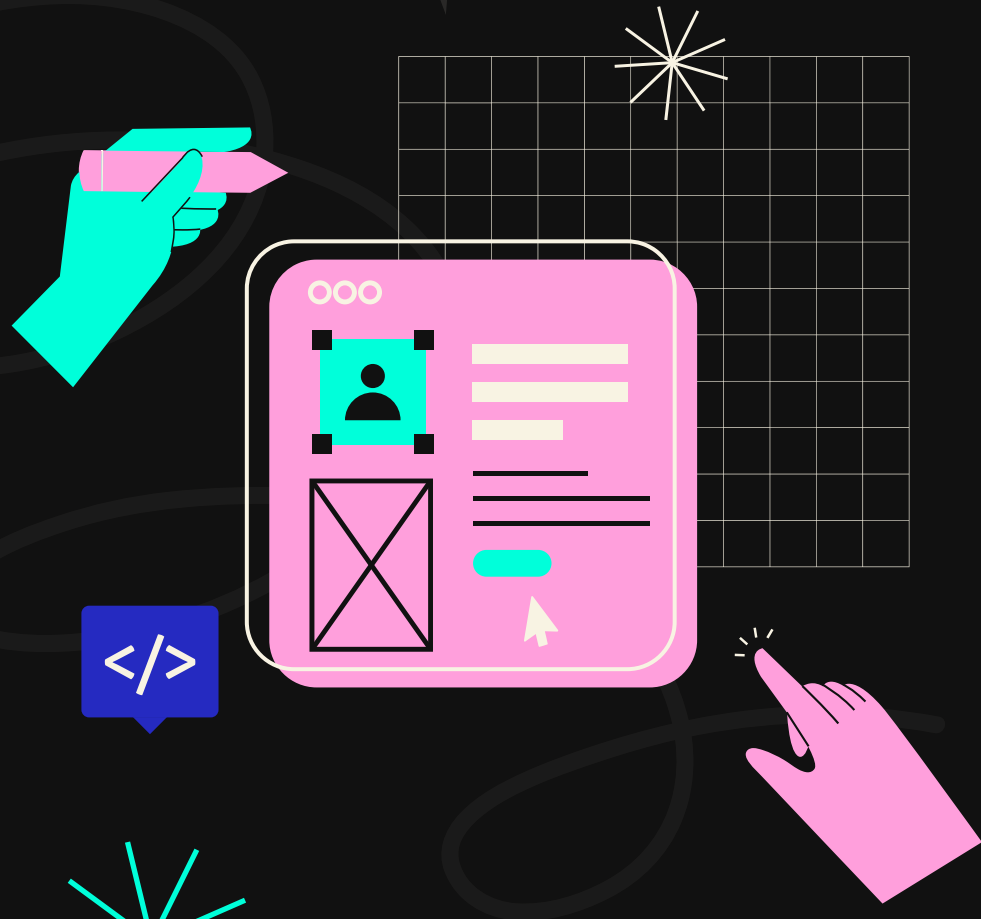
Parciales

2 parciales teóricos



Proyecto

Dos Fases
15% y 25 %



01

Nuestra Carrera



“Si lo puedes imaginar lo
puedes crear”

—Además lo puedes programar

Importancia de pasos

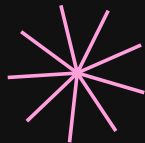


Establecer metas claras: Define lo que quieres lograr de manera específica y medible.

Planificación: Desarrolla un plan detallado de cómo vas a alcanzar tus metas.

Persistencia: Mantén la determinación de seguir adelante, incluso cuando enfrentan obstáculos.

Apoyo: Busca el apoyo de amigos, familiares o mentores que puedan ayudarte en tu camino hacia el éxito.



Estilo



Bello es mejor que feo.
Explícito es mejor que implícito.
Simple es mejor que complejo.
Complejo es mejor que complicado.
Plano es mejor que anidado.
Espaciado es mejor que denso.
La legibilidad es importante.

Los casos especiales no son lo suficientemente especiales como para romper las reglas.

Sin embargo la practicidad le gana a la pureza.

Los errores nunca deberían pasar silenciosamente.

A menos que se silencien explícitamente.

Frente a la ambigüedad, evitar la tentación de adivinar.

Debería haber una, y preferiblemente sólo una, manera obvia de hacerlo.

A pesar de que eso no sea obvio al principio a menos que seas Holandés.

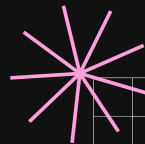
Ahora es mejor que nunca.

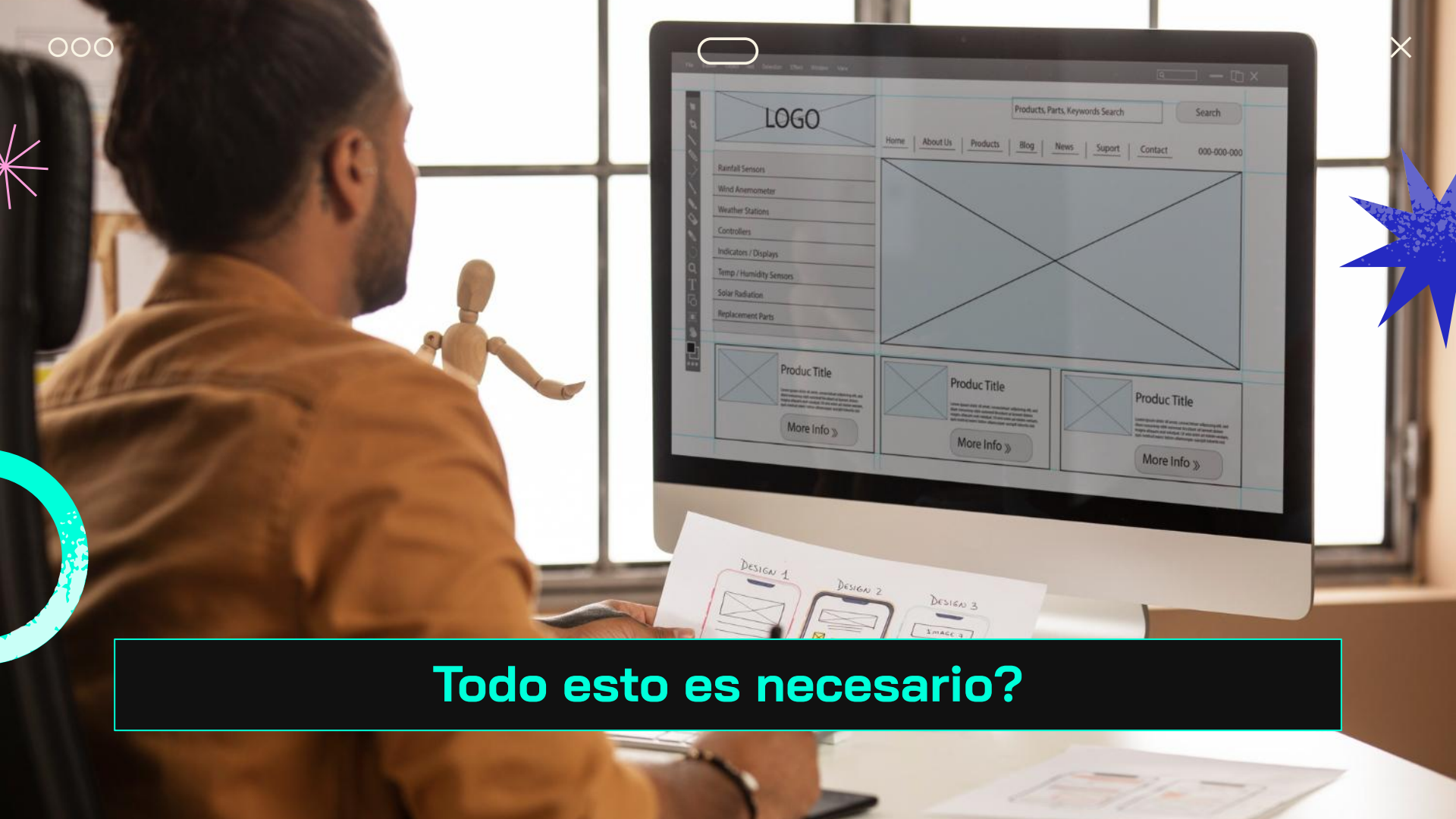
A pesar de que nunca es muchas veces mejor que *ahora* mismo.

Si la implementación es difícil de explicar, es una mala idea.

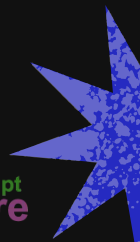
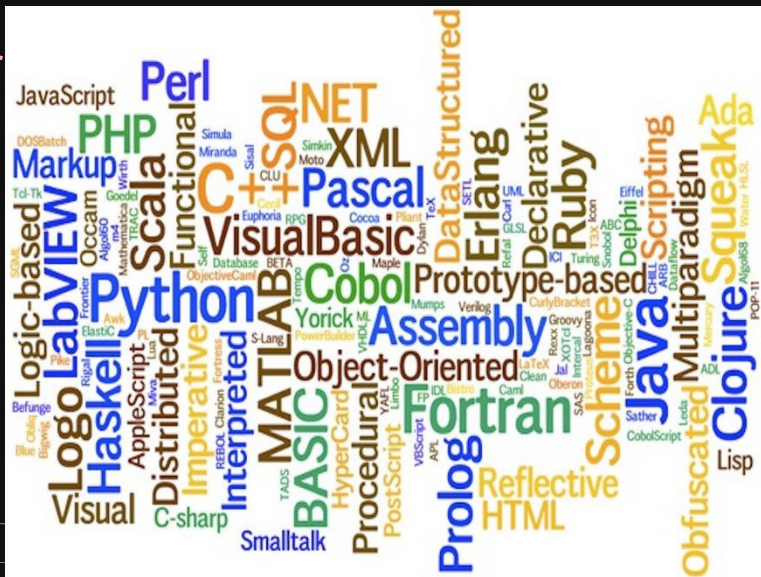
Si la implementación es fácil de explicar, puede que sea una buena idea.

Los espacios de nombres son una gran idea, ¡tengamos más de esos!





Todo esto es necesario?



Lenguaje c++



Bases de Datos: MySQL, una de las bases de datos más utilizadas está escrita en C++.

Navegadores WEB: Utilizan C++ porque necesitan rapidez a la hora de mostrar los resultados en pantalla.

Sistemas operativos: La columna principal tanto de Windows, como Linux o Mac OS, están escritas en C++. Su potencia y rapidez lo hace un lenguaje de programación ideal para programar un sistema operativo.

Compiladores: los compiladores de muchos lenguajes de programación están escritos en C++.

Videojuegos: C++ es utilizado aún en el mundo de los videojuegos, bien para programar motores gráficos o para alguna parte concreta del videojuego.

También tiene otras aplicaciones como en máquinas médicas, relojes inteligentes, etc. por su capacidad de estar cerca del lenguaje máquina que otros lenguajes de alto nivel.

Por todos estos usos y aplicaciones podemos concluir que la importancia del lenguaje C++ es muy grande y está presente en muchos sitios.

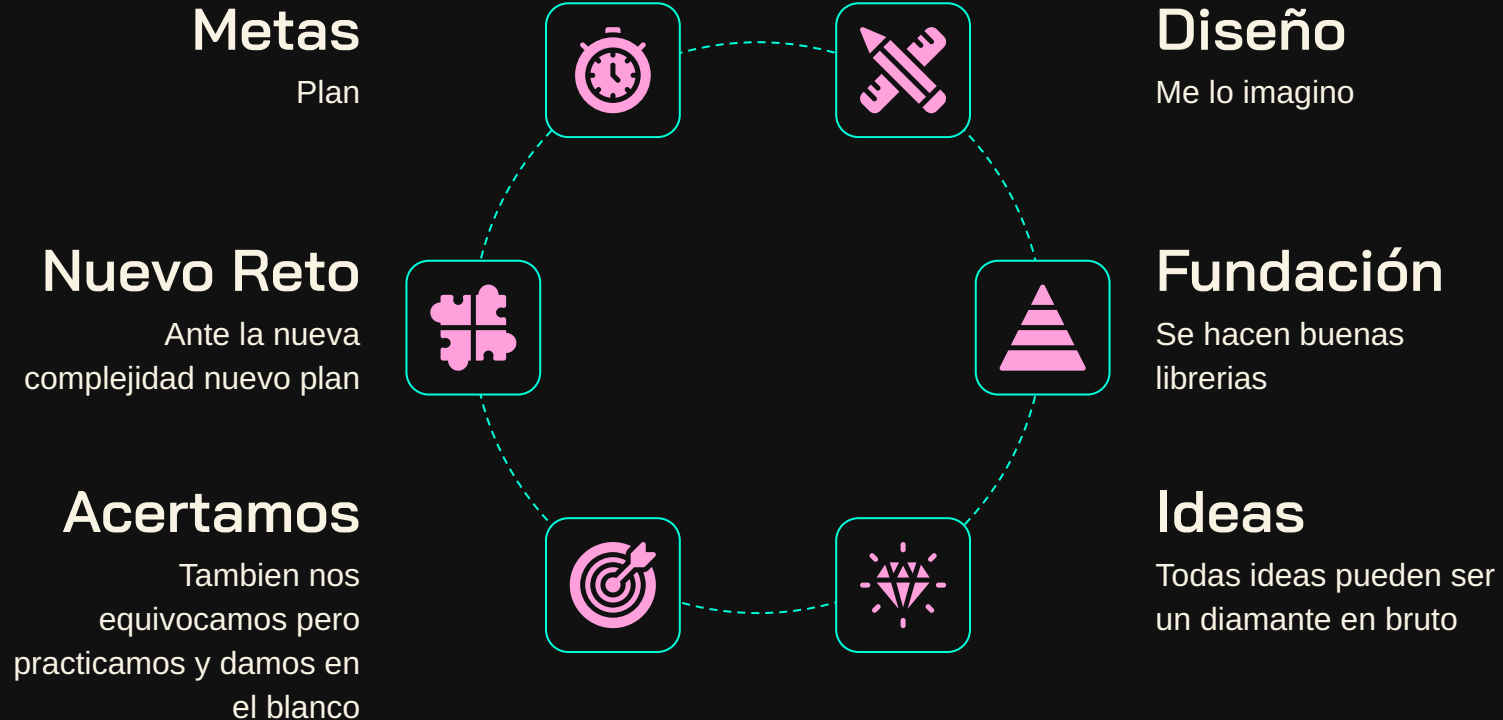


¿cómo estudio?

Libros recomendados
Nada de estar leyendo Wikipedia
Canales especializados de youtube
Internet (que sea serio)
PRACTICAR, PRACTICAR, PRACTICAR



Metodología para mejorar mi proyecto





Todo está dicho!

Empéccemos

