

# TEMAS PRE-REQUISITOS

## Programación

1. • Programación básica
2. • POO (OOP)
3. • Estructuras de datos.

1. Resolver un problema → convertir algo de palabras a código.

a. Definir Funciones.

Datos Entrada →  $\boxed{F}$  → Datos de salida.

Ejemplos:

1. PAPA  
# creditos vistos →  $\boxed{F}$  → PAPA resultante.  
# creditos algoritmos

2. Lista de notas  
y creditos vistos →  $\boxed{F}$  → PAPA result.  
# creditos algoritmos

Def

Datos Entrada  
Nombre (arg1, arg2, arg3) {

return

Salida.

}

b. Operaciones de control.

- Ciclos  $\begin{cases} \text{For} \\ \text{while} \end{cases}$
- Condicionales → if
- Operaciones  $\begin{cases} \text{switch} \\ \rightarrow +, -, *, /, \wedge, \& \end{cases}$

Dominio de un lenguaje de programación

## 1.2 Tipos de Datos

### Datos Primitivos.

int  
float/double  
char.  
bool.

⇒

### colecciones básicas

arrays.

int[]  
float[]  
char[] → string  
bool[]

matriz

int [][]  
float [][]

Objeto → Agrupación de primitivas

↳ Nivel de abstracción.

Representar de manera mejor un obj del mundo real.

Interfaz

```
Class Estudiante {  
    string Nombre;  
    string correo;  
    Materias[] cursadas;  
    float Calcular Promedio();  
}  
Materia Obtener Mejor Materia();
```

```
Class Materias {  
    string Nombre  
    int credits  
    float nota  
}
```

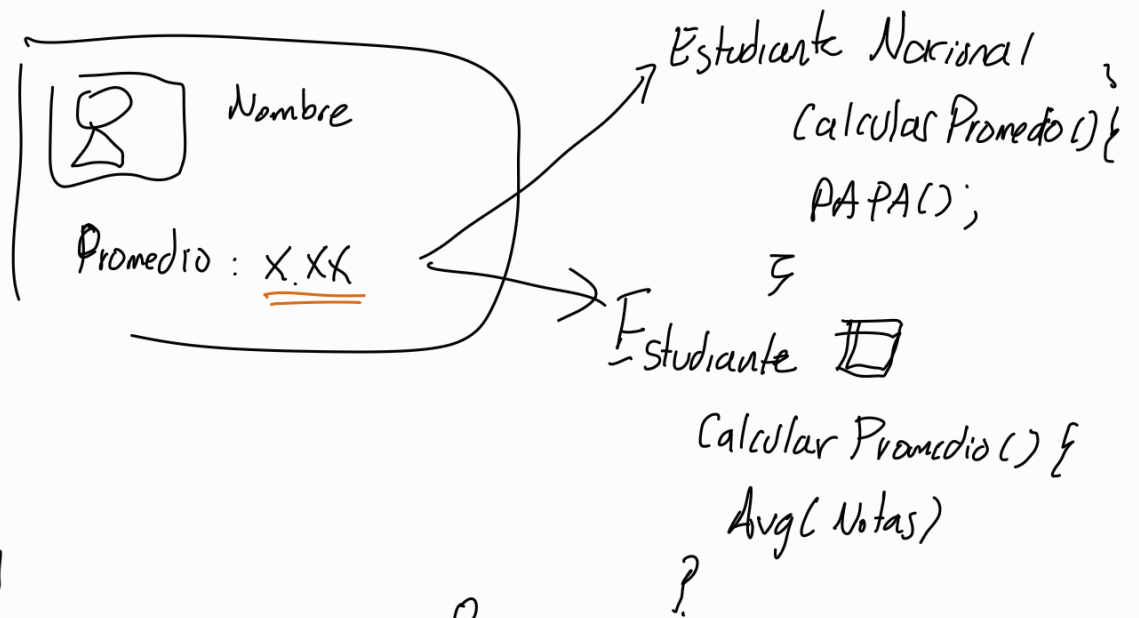
Abstracta - interfaz.

↳ Yo no necesito definir todo. ¿cómo funciona?

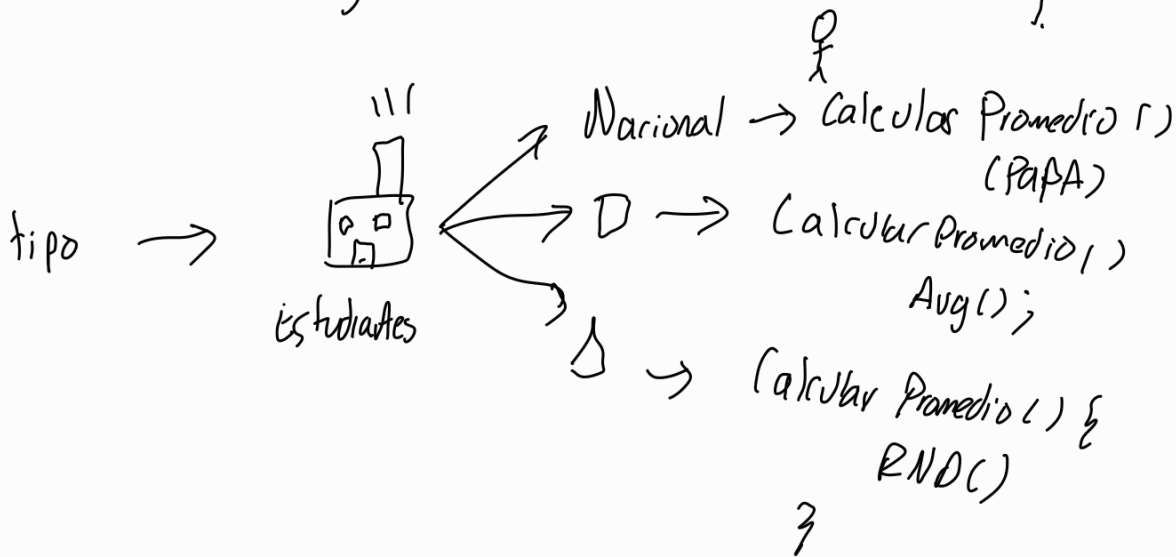
↳ Delegar la responsabilidad de definir el cómo implementar a otra parte.

↳ comportamientos.

Ejemplo.



Factory



### 3. Estructuras de Datos.

- Organizar de manera adecuada y optima los datos de acuerdo a una necesidad (contexto).

Datos → Variables → Espacio de Memoria.

int a:      Ram  
             [a]

int[n] A      Ram  
             [A<sub>0</sub> | A<sub>1</sub> | A<sub>2</sub> | ... |   |   | A<sub>n</sub>]

## Arreglo de Notas.

Notas  $\rightarrow$  La nota más alta está al comienzo.

3, 3.5, 2.8, 3.2, 4.5, 4.0

Float 

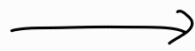
4.5	3	2.8	3.2	3.5	4.0	
-----	---	-----	-----	-----	-----	--

adicionar Nota / Float nota) {

add  $\rightarrow$  adicionar elemento al final

```
if (notas.empty()) notas.add(nota)
elseif (notas[0] >= nota) notas.add(nota)
elseif (notas[0] < nota) {
    notas.add(notas[0])
    notas[0] = nota;
}
```

## Metodos.



Metodos para adicionar, encontrar y sacar datos.

add()
   
erase()
   
find()
   
pop-back()
   
pop-front()

No enfocarnos en esto:
   
Asumimos que usarla en este caso es lo mejor en tiempo y memoria.

## Tarea:

	Estructura Dato	Definicion o Para que sirve.	Cuando usar	Operaciones
Mapas	• Arreglos Dinamicos	Arreglar elementos FIFO		
	• Queues			
	• Stacks			
	• Deque			
	• Listas Enlazadas			
	• Diccinario $\rightarrow$ Hash table			
	• Diccinario $\rightarrow$ Arbol			
	• Heaps - Priority Queue			

Sets {  
• Hash table  
• Arbol balanceado.  
• Multisets

!

!