¿Qué es Programación Orientada a Objetos (POO)?

Es un **paradigma de programación** donde organizamos el código usando **clases** y **objetos**. Permite modelar entidades del mundo real de forma más natural.

- **Clase**: es el plano o modelo. Define atributos y comportamientos.
- **Objeto**: es una instancia concreta de una clase.

Elementos básicos en Dart

1. Clases

Se crean con la palabra clave `class`:

```
class Persona {
  String nombre;
  int edad;
}
```

Nota: Los atributos (variables de la clase) pueden tener cualquier tipo: `String`, `int`, `double`, etc.

2. Constructores

Un **constructor** sirve para inicializar los atributos cuando creamos un objeto.

Ejemplo:

```
class Persona {
  String nombre;
  int edad;

  // Constructor
  Persona(this.nombre, this.edad);
}
```

Al usar `this`, asignamos el valor recibido al atributo correspondiente.

3. Crear objetos

Se usa `new` (opcional) o simplemente llamando al constructor:

```
var persona1 = Persona('Juan', 30);
```

4. Métodos

Son **funciones** que están dentro de una clase. Representan comportamientos de los objetos.

```
class Persona {
  String nombre;
  int edad;
  Persona(this.nombre, this.edad);
  void saludar() {
    print('Hola, soy $nombre y tengo $edad años.');
  }
}
```

Para llamarlo:

```
persona1.saludar();
```

5. Uso de `return` en métodos

Un método puede devolver un valor:

```
class Rectangulo {
  double ancho;
  double alto;
  Rectangulo(this.ancho, this.alto);
  double area() {
    return ancho * alto;
  }
}
```

Llamar al método:

```
var rect = Rectangulo(5, 10);
print('Área: ${rect.area()}');
```

6. Modificar atributos

Podemos leer y modificar atributos de un objeto:

```
persona1.nombre = 'Carlos';
print(persona1.nombre);
```

7. Condicionales simples en métodos

Podemos usar condicionales (`if`) para controlar comportamientos:

```
void retirar(double monto) {
  if (monto <= saldo) {
    saldo -= monto;
  } else {
    print('Saldo insuficiente.');
  }
}
```

8. Resumen: Estructura típica de una clase

```
class NombreDeLaClase {
  // Atributos
  Tipo nombreAtributo;
  // Constructor
  NombreDeLaClase(this.nombreAtributo);
  // Métodos
  Tipo metodo() {
    // instrucciones
  }
}
```

Tipos de datos más usados

Tipo	Ejemplo
`String`	`'Hola'`
`int`	`25`
`double`	`3.14`
`bool`	`true`, `false`

Algunas operaciones útiles

- **Concatenar texto**: `"Hola " + nombre`
- **Interpolar texto**: `"Hola, $nombre"`
- **Multiplicar números**: `ancho * alto`

- **Sumar números**: `saldo + monto`
- **Reducir número sin que sea negativo**:

```
velocidad = (velocidad - decremento).clamp(0, double.infinity);
```

(`clamp` asegura que el número no baje de 0).

Consejos para resolver el TP

- **Primero** crear la clase, atributos y constructor.
- **Luego** crear un método sencillo.
- **Finalmente** crear objetos en `main()` para probar.
- Imprimir resultados con `print()`.
- Revisar errores de mayúsculas, tipos de datos y paréntesis.

Plantilla mínima de ejemplo para cualquier ejercicio

```
void main() {
  // Crear objeto
  var miObjeto = Clase(parametros);
  // Usar métodos
  miObjeto.metodo();
}
class Clase {
  // Atributos
  Tipo atributo;
  // Constructor
  Clase(this.atributo);
  // Método
  void metodo() {
    print('...');
  }
}
```