Tecnología

JAVASCRIPT | Módulo:JAVASCRIPT & POO

Profesor Gabriel Muñoz | g.munoz@sistemas-utnfra.com.ar **Ayudante Ibrahin Rodriguez** | i.rodriguez@sistemas-utnfra.com.ar

El programa de formación DevJump te permitirá alcanzar los **skills técnicos** necesarios y las **competencias laborales** para iniciarte como desarrollador trainee/junior en la industria IT con experiencia en proyectos reales.

JAVASCRIPT y la POO |

En cada versión de ES Javascript evoluciona en temas de POO

Los objetos suelen comunicarse entre sí para crear códigos más robustos, pero a la vez más organizados y simples.

podemos saltar prototipos y tener una forma más consistente de usar los conceptos bases de estos paradigmas.

JAVASCRIPT y la POO |

¿QUÉ ES POO?

Para definir POO podemos definir cada sigla con lo que sabes hasta ahora:

Programación: Escribir código con el objetivo de darle órdenes al ordenador
Orientado a: dirigido a
Objetos: modelo informático de un objeto real/ficticio
Obtenemos que POO es: Escribir código dirigido a modelar objetos

Es un paradigma de programación (un método para crear sistemas)

JAVASCRIPT y la POO |

¿QUÉ ES POO?

Consiste en abstraer los elementos importantes que conforman objetos (cosas) del mundo real en código.

Esos Objetos pueden comunicarse entre sí para crear códigos más robustos, pero a la vez más organizados y simples.

Ejemplo: Un Objeto puede ser una Consola de juegos y este objeto se comunica con otros objetos como el Joystick .

JAVASCRIPT y la POO |

ANALOGÍA

Imaginen que tenemos una TV en la sala y otra en la cocina.

En la POO a ambos TV se denominan OBJETOS

Cada TV Tiene funcionalidades,como por ejemplo el botón para encender o el botón para subir el volumen; a cada funcionalidad la llamamos MÉTODOS (Funcionalidad que puede ejecutar el objeto).

Las TV tienen diferentes colores y características, (no son funcionalidades) son PROPIEDADES, son variables ligadas al objeto. cada tv puede tener su propio color , forma etc.

JAVASCRIPT y la POO |

ANALOGÍA

Para que estas dos TV puedan estar en nuestro hogar alguien las tuvo que fabricar a esto le llamamos CLASES

las Clases podemos verlas como una plantilla para fabricar Objetos.

Las CLASES se encargan de encapsular las propiedades y funciones que posteriormente nuestros objetos podran utilizar, en la jerga de la POO le llamamos Instanciar un objetod de la clase en nuestro ejemplo tenemos 2 ejemplos de la clase LG (creamos 2 objetos: la sala y la cocina)

PILARES DE LA POO |

- ABSTRACCIÓN
- ENCAPSULACIÓN
- HERENCIA
- POLIMORFISMO

PILARES DE LA POO |

❖ ABSTRACCIÓN

Abstraer, es eliminar los detalles innecesarios para solo nos enfocamos en los aspectos que son necesarios para el contexto o sistema que estamos desarrollando.

Ejemplo:

Supongamos que eres un estudiante y además empleado.Para el sistema de la escuela tu solo les interesas como estudiante, no le importan tus propiedades como empleado,así que el sistema de la escuela solamente abstrae las propiedades que le interesan de ti como podrían ser: nombre, grado y calificaciones. Pero al sistema de tu trabajo, les interesan tus propiedades como empleado, las cuales podrían ser: numero_empleado, puesto y sueldo

PILARES DE LA POO |

- ENCAPSULACIÓN

Se centra en ocultar los detalles que no son relevantes para el exterior

Es la agrupación de datos (propiedades) y los métodos que actúan sobre esas propiedades de manera que el acceso a esos datos está restringido desde fuera del paquete (clase).

En POO, esto significa que un objeto almacena su estado de forma privada,y solo los métodos del objeto tienen acceso para cambiarlo.

PILARES DE LA POO |

- ENCAPSULACIÓN

Ejemplo:

Si un usuario de motocicleta quiere encender la moto; entonces hay un interfaz para hacer eso; y el usuario no tiene acceso directo para encenderla desde el propio cableado interno (sabiendo que eso puede resultar mal),incluso, podríamos concluir que al usuario de la moticicleta no le importa cómo funciona el sistema de encendido,solamente le interesa utilizarlo para comenzar a moverse.

Entonces, los detalles del funcionamiento permanecen ocultos al usuario,pero en cambio podriamos darle acceso a una interfaz de fácil uso (si así se requiere).

PILARES DE LA POO |

- HERENCIA

Es la forma en la que una clase (hija) hereda de otra clase (padre) sus métodos y atributos.

La herencia permite la reutilización de código de una clase en otra y es ideal implementarla cuando los objetos tienen similitudes entre sí.

Ejemplo, Un libro tiene título y autor, y por otro lado un Comic también tiene título y autor,

En POO podríamos utilizar la herencia para hacer que Comic herede de Libro y evitar repetir código innecesario.

PILARES DE LA POO |

- POLIMORFISMO

Se refiere a la capacidad de realizar una misma acción en diferentes formas.

Polimorfismo es una palabra griega que significa "con muchas formas"

Ejemplo si tenemos una clase A con método M y la clase B que hereda de A, la clase B puede utilizar el mismo método M de la clase A,pero con una realización (algo) diferente.

MODIFICADORES DE ACCESO |

Los lenguajes de programación que aceptan POO suelen contar con palabras reservadas conocidas como modificadores de acceso. Estas palabras reservadas sirven para especificar si queremos que el acceso de una propiedad o método sea público o sea privado y precisamente los dos modificadores de acceso más populares son: public y private

- Público (public)
  se puede acceder desde cualquier parte del código
- Private (private)
  sólo es accesible desde dentro de la clase que lo define.

MODIFICADORES DE ACCESO |

Podemos específicar que no deseamos que puedan modificar la data de un objeto y es normal,en un ejemplo si no o hay un sistema de usuarios y permisos el dejar abierto al público que se puedan modificar propiedades podría ocasionar que cualquier usuario cambiará propiedades de un objeto (ejemplo el precio de un libro).

En realidad todas las propiedades son datos sensibles, no quisiéramos que cualquier pudiera modificar el título ni el autor, podría ocasionar un caos y el resto igual.

TERMINOLOGÍA POO |

OBJETOS:
Son la pieza central de la POO.
Los objetos se componen de datos (propiedades) y métodos (funciones) que operan sobre esos datos

PROPIEDAD:
Es un dato que representa una característica de un objeto. Una propiedad es una variable.
Un objeto puede tener muchas propiedades

TERMINOLOGÍA POO |

MÉTODO:

Es una función que puede ejecutar nuestro objeto. Un objeto puede tener muchos métodos

CLASE

Funcionan como las plantillas y son utilizadas para instanciar (crear) objetos.
Una clase encapsula (contiene) todas las propiedades y métodos que después almacenarán los objetosninstanciados. Una clase Representa a un tipo de objeto; ejemplo: Libro, Automóvil, Perro.

TERMINOLOGÍA POO |

CONSTRUCTOR:
Es un método que se llama en el momento de la creación de instancias (objetos).
Los constructores son útiles para (valga la redundancia) "construir" o inicializar las propiedades de los objetos.

GETTER:
Un getter es un método que permite acceder a una propiedad privada de un objeto en lugar de usar la propiedad directamente

SETTER:
Un setter es un método que permite acceder y modificar una propiedad privada de un objeto en lugar de modificar la propiedad directamente