JS

PROFE VIDERMID

Programación en Java Script

(Parte 6)

Facilitador: Ing. Esp. Vidermid Sánchez

@vidermid

+584147106623

@ingenieriadigitalsc

+584147464801

Objetos

PROFE VIDERMID

Es una cosa tangible que podemos sentir y manipular

- Televisor
- Carro
- Computador
- Libro
- Nevera
- Semáforo

Objetos en desarrollo de Software

PROFE VIDERMID

En desarrollo de software trata sobre un modelo informático que representa a objetos reales o ficticios. en otras palabras, es un conjunto de datos con comportamientos asociados.

Objeto Real

Computador

Modelo Informático

Computador
1. Procesador 2. Memoria 3. Disco 4. Tarjeta Madre
encender() procesar() calcular() apagar()

Programación Orientada a Objetos

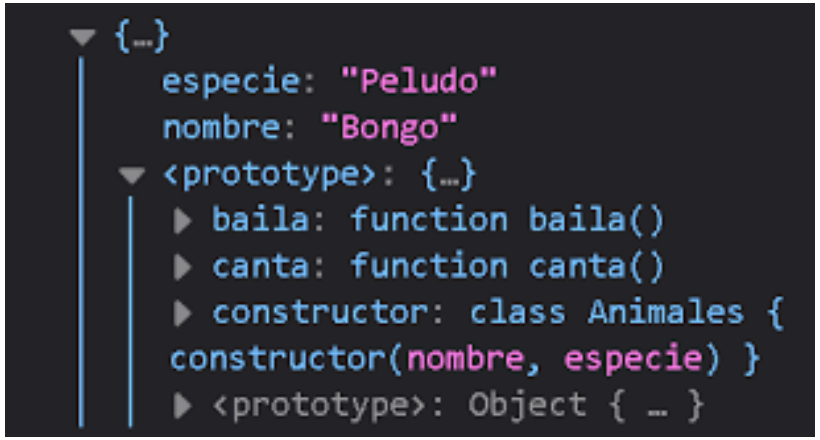
PROFE VIDERMID

POO quiere decir programar o crear líneas de código dirigidas a modelos informáticos de un objeto real o ficticio, donde debe existir un área para elaborarlo tomando en cuenta sus características y posibles acciones que realizará en diferentes escenarios.
Todo ello con la finalidad de reutilizar de manera óptima el código de programación y minimizar el tamaño de los script que conforman los aplicativos diseñados.

Modelo Informático

Computador
1. Procesador 2. Memoria 3. Disco 4. Tarjeta Madre
encender() procesar() calcular() apagar()

Líneas de Código en POO

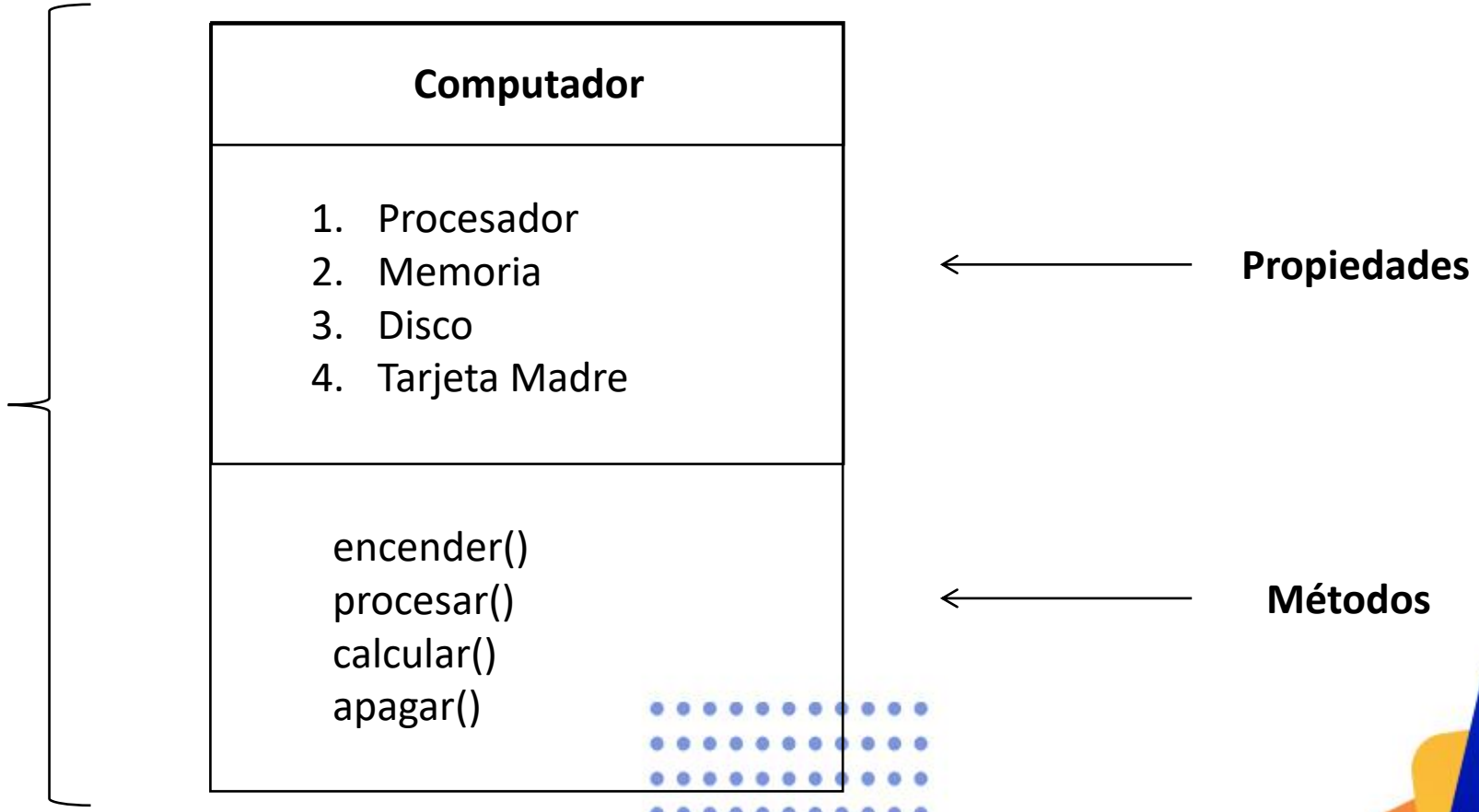


Clase

PROFE VIDERMID

Es la fabrica donde se estructuran los objetos, es decir, éstos se crean a partir de una Clase, en ella se asignan las características que conformarán el Objeto, además se establece la operatividad o comportamientos del mismo.

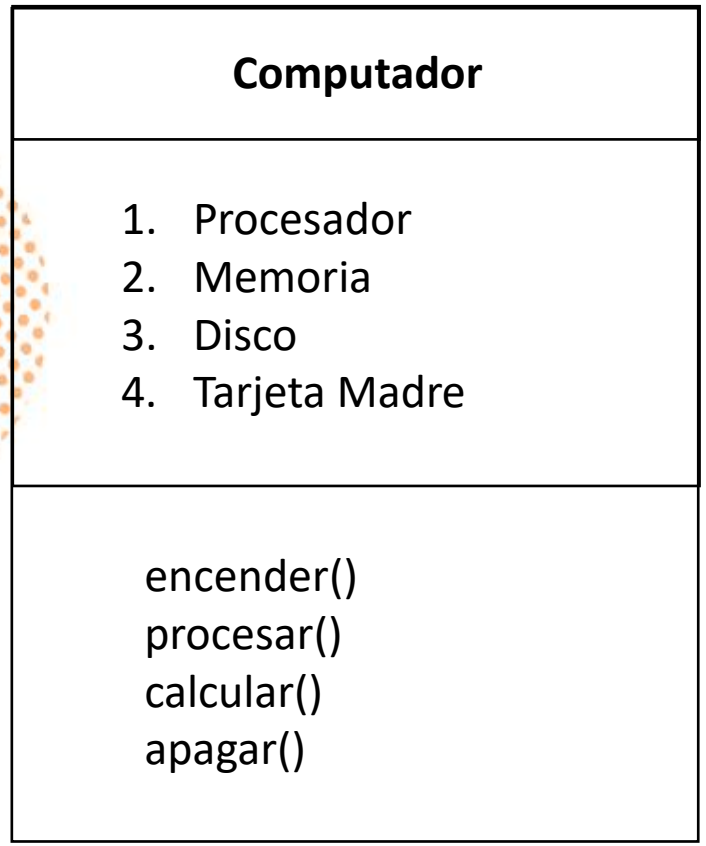
Clase



Clases y Objetos

PROFE VIDERMID

Clase



Objeto: conjunto de datos con un comportamiento asociado.

Propiedades: datos que componen al objeto.

Métodos: el comportamiento del objeto.

Clase: contenedor de propiedades y métodos para crear un tipo objeto.

Instancia: crear un Objeto desde una clase.

Ingeniería Digital

Los 4 pilares de la POO

PROFE VIDERMID

- Abstracción
- Encapsulamiento
- Herencia
- Polimorfismo

Caso práctico a diseñar

PROFE VIDERMID

Requerimientos del Sistema:
Desarrollar un sistema para registrar películas y consultar su información.

Características de las películas:
id, titulo, genero, duración, clasificación

Funciones:
Mostrar información.

Exigencia adicional: los usuarios no pueden modificar las características de las películas durante la primera versión del sistema.

Caso práctico a diseñar

PROFE VIDERMID

Análisis del Sistema:
El objeto principal del sistema serán las Películas.

Tareas:
- Modelar una Película con las propiedades mencionadas en los requerimientos.
- No se pueden modificar las propiedades.
- Los objetos Película deberán ser capaces de mostrar su información.

Procedimiento:
Definir una clase llamada Película con las propiedades: id, titulo, genero, duración, clasificación. Además el respectivo método necesario para mostrar la información.

Abstracción

PROFE VIDERMID

Es la base donde comienza la POO, trata sobre eliminar los detalles innecesarios para solo enfocarse en los aspectos que son relevantes para el contexto del sistema en desarrollo.

Clase

Película
id ----> entero titulo ----> cadena genero ----> cadena duración ----> entero clase ----> caracter
consultar()

Características eliminadas porque no son necesarias para el caso de estudio o sistema a desarrollar:

- protagonistas
- estudio
- costo
- tiempo de rodaje
- recaudación en taquilla

Ingeniería Digital

Abstracción e Instanciación

PROFE VIDERMID

Es la creación de un Objeto a partir de una clase, se asignan los valores a cada atributo. Esto trae como beneficio la reutilización de código, es decir si se construyen N objetos de un mismo tipo, todos podrán instanciar a la misma clase ya existente.

pelicula_1 = nueva **Película**

id= 001
titulo= “La Fuga”
genero= “Acción”
duración=”2:15”
clase=”B”

Clase

Película
id ----> entero titulo ----> cadena genero ----> cadena duración ----> entero clase ----> caracter
consultar()

pelicula_1

Encapsulamiento

PROFE VIDERMID

Se entiende como ocultar los detalles que no son relevantes para el exterior, agrupar propiedades y métodos de manera que el acceso está restringido desde fuera del paquete.

Clase

Película
id ----> entero titulo ----> cadena genero ----> cadena duración ----> entero clase ----> caracter
consultar()

Modificadores de Acceso

Public: Es el nivel mas permisivo. Funciona para indicar que el método o propiedad de la clase es publico. En este caso se puede acceder a este atributo, para visualizarlo o editarlo, esto lo podría hacer cualquier otro elemento del propio script o aplicativo

Private: Indica que el método o propiedad de la clase es privado y que solo puede tener acceso desde dentro de la misma clase

Encapsulamiento

PROFE VIDERMID

Para el caso de estudio de las Películas todas las propiedades serán privadas ya que no se permiten que modifiquen sus valores, pero el método de consultar será publico para que puedan consultar todas las películas almacenadas

Clase

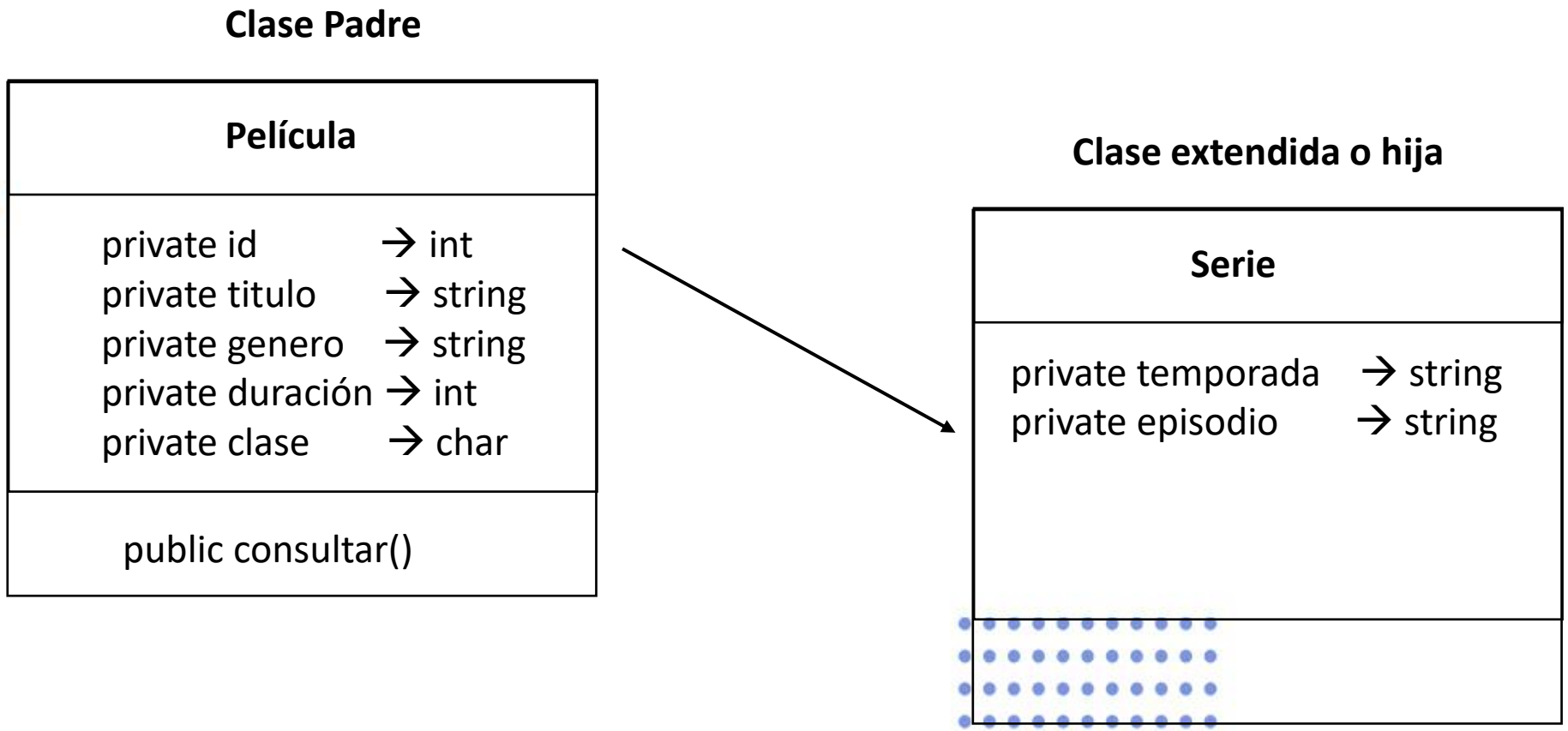
Película
private id → int private titulo → string private genero → string private duración → int private clase → char
public consultar()

```
// instanciando la clase peliculas
pelicula_1 = new pelicula_1
{
    id= 001,
    titulo= "La Fuga",
    genero= "Acción",
    duracion="2:15",
    clase="B"
};

// instanciando para modificar un valor de atributos
pelicula_1 => duracion = "5:00"; // NO TIENE ACCESO A MODIFICAR, ERROR

// instanciando metodo publico
pelicula_1 =>consultar(); // SIN NOVEDAD

```

Herencia

PROFE VIDERMID

Se refiere a reutilizar código sin necesidad de hacer copias redundantes en el mismo script, consiste en relacionar clases para que hereden propiedades y métodos de otras.



Herencia e Instanciación

PROFE VIDERMID

Es la creación de un Objeto a partir de una clase, se asignan los valores a cada atributo. Esto trae como beneficio la reutilización de código, es decir si se construyen N objetos de un mismo tipo, todos podrán instanciar a la misma clase ya existente.

serie_1 = nueva **Serie**

id= 001
titulo= “Exploradores del Hielo”
genero= “Suspenso”
duración=”1:32”
clase=”B”
temporada=“01”
episodio=“01”

Clase

Serie

private temporada → string
private episodio → string

serie_1

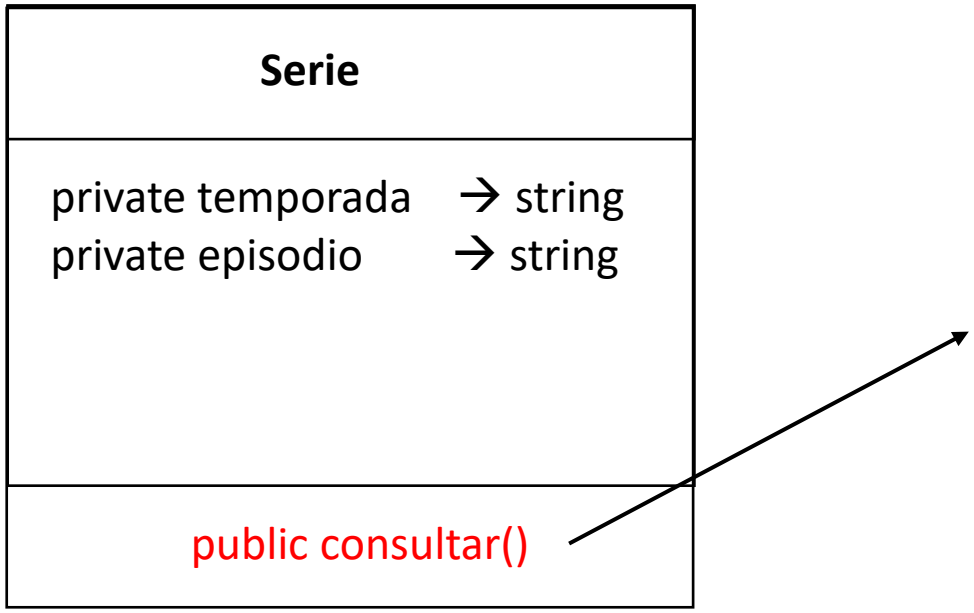
Ingeniería Digital

Polimorfismo

PROFE VIDERMID

Es la capacidad de que un mismo método obtenga diferente resultado dependiendo de la sub clase (clase extendida ó hija) en la que se ejecute. En el ejemplo observamos que el método “consultar()” no puede mostrar la temporada ni el episodio de la clase “Serie”, entonces toca modificarlo pero en la clase extendida o hija.

Clase extendida o hija




```
1   //SALIDA CON LA MODIFICACIÓN DEL METODO consultar en la clase hija
2   serie_1 => consultar()
3   /*
4       id= 001
5       titulo= “Exploradores del Hielo”
6       genero= “Suspenso”
7       duración=”1:32”
8       clase=”B”
9       temporada=“01”
10      episodio=“01”
11  */
```