

## PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

### FORMATO GUÍA DE APRENDIZAJE

*Estimad@ aprendiz, la identificación de la presente guía está vacía para que usted pueda diligenciarla. Si tiene alguna duda pregúntele a su instructor líder.*

#### 1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Desarrollo de Aplicaciones en JavaScript
- Código del Programa de Formación:
- Nombre del Proyecto: Implementación de Clases y Objetos en JavaScript
- Fase del Proyecto: Creación y uso de Clases y Objetos en JavaScript
- Actividad de Proyecto: Aplicación de principios de Programación Orientada a Objetos (POO) en JavaScript
- Competencia: Definir, instanciar y manipular clases y objetos en JavaScript
- Duración de la Guía: 6 Horas

#### 2. PRESENTACIÓN

Las clases en JavaScript son un tipo especial de función que facilita la creación de objetos y el uso de la Programación Orientada a Objetos (POO). A través de esta guía, aprenderás a definir clases, crear instancias de objetos, trabajar con propiedades y métodos, y aplicar conceptos de herencia y encapsulamiento en tus proyectos. Al finalizar, serás capaz de aplicar estos conceptos en proyectos JavaScript de mayor complejidad.

#### 3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

##### 3.1 ACTIVIDAD DE REFLEXIÓN INICIAL

Reflexiona sobre los siguientes puntos y responde en un documento:

- ¿Por qué crees que la POO es útil en el desarrollo de aplicaciones?
- ¿Cómo te parece que el uso de clases y objetos puede mejorar la estructura y reutilización del código?

### 3.2 ACTIVIDAD DE APROPIACIÓN

#### Paso 1: Crear una Clase en JavaScript

Las clases en JavaScript se definen con la palabra clave `class`, seguida del nombre de la clase y un bloque de código `{}`.

Ejemplo básico de clase:

```
class Persona {  
  constructor(nombre, edad) {  
    this.nombre = nombre;  
    this.edad = edad;  
  }  
  
  saludar() {  
    return `Hola, mi nombre es ${this.nombre} y tengo ${this.edad} años.`;  
  }  
}
```

#### Paso 2: Instanciar un Objeto

Para crear un objeto de la clase `Persona`, usa la palabra clave `new` seguida del nombre de la clase.

Ejemplo de instancia de objeto:

```
const persona1 = new Persona("Juan", 30);  
console.log(persona1.saludar());
```

#### Paso 3: Agregar Métodos y Propiedades

Puedes agregar métodos adicionales o definir propiedades que permitan a los objetos tener comportamientos y características específicas.

Ejemplo con métodos adicionales:

```
const persona1 = new Persona("Juan", 30);
console.log(persona1.saludar());

class Persona {
  constructor(nombre, edad) {
    this.nombre = nombre;
    this.edad = edad;
  }

  saludar() {
    return `Hola, mi nombre es ${this.nombre}`;
  }

  setEdad(nuevaEdad) {
    this.edad = nuevaEdad;
  }
}
```

#### Paso 4: Herencia entre Clases

JavaScript permite que una clase herede de otra utilizando la palabra clave `extends`, lo que permite crear jerarquías de clases.

Ejemplo de herencia:

```
class Estudiante extends Persona {
  constructor(nombre, edad, grado) {
    super(nombre, edad);
    this.grado = grado;
  }

  estudiar() {
    return `${this.nombre} está estudiando en el grado ${this.grado}.`;
  }
}
```

### Paso 5: Encapsulamiento

El encapsulamiento consiste en limitar el acceso a ciertas propiedades o métodos de una clase. En JavaScript, los campos privados se definen con el prefijo #.

Ejemplo de encapsulamiento:

```
class CuentaBancaria {  
    #saldo;  
  
    constructor(titular, saldoInicial) {  
        this.titular = titular;  
        this.#saldo = saldoInicial;  
    }  
  
    depositar(cantidad) {  
        this.#saldo += cantidad;  
    }  
  
    getSaldo() {  
        return this.#saldo;  
    }  
}
```

### Paso 6: Ejercicio Práctico Integrador

- Crea una clase Vehiculo con propiedades como marca y modelo y métodos para encender y apagar el vehículo.
- Crea una clase Coche que herede de Vehiculo y agregue propiedades adicionales como tipoCombustible.
- Implementa un método en Coche que calcule el consumo de combustible en función de la distancia recorrida.

### 3.3. ACTIVIDAD DE TRANSFERENCIA DE CONOCIMIENTO

- **Consulta:** Investiga sobre los conceptos de polimorfismo y composición en la Programación Orientada a Objetos.
- **Evidencia:** Realiza un documento donde expliques estos conceptos e incluye ejemplos en JavaScript.
- 

*Unifica las evidencias en una sólo archivo o carpeta, comprímelo y adjúntalo en el enlace que les comparta la instructora para tal fin.*

***¡Recuerda! No importa que algunas de las actividades sean grupales, las evidencias se montan de forma individual, de tal forma que todos los integrantes de los equipos tengan su trabajo subido en plataforma.***

*Si tienes dudas o inquietudes pregúntale a tu instructora.*

### 3.4. ACTIVIDAD DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
<b>Evidencias de Conocimiento:</b>	Explicación clara y ejemplos de las clases en JavaScript.	Técnica: Evaluación Colaborativa Instrumento: Lista de chequeo
<b>Evidencias de Desempeño:</b>	Implementación de la herencia y encapsulamiento.	Técnica: Observación Instrumento: Lista de chequeo
<b>Evidencias de Producto:</b>	Desarrollo de un archivo JavaScript que implemente las clases y métodos planteados en el ejercicio integrador.	Técnica: Evaluación Colaborativa Instrumento: funcionalidad

#### 4. GLOSARIO DE TÉRMINOS:

- **Clase:** Plantilla o modelo a partir del cual se crean objetos con propiedades y métodos específicos.
- **Objeto:** Instancia de una clase que contiene datos y puede realizar acciones.
- **Encapsulamiento:** Técnica que protege el acceso a los datos de un objeto.
- **Herencia:** Mecanismo que permite crear una clase a partir de otra.
- 

#### 5. REFERENTES BIBLIOGRÁFICOS

- <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes>
- <https://eloquentjavascript.net/>

## 6. CONTROL DEL DOCUMENTO

Nombre Autor (es)	Cargo	Dependencia		Fecha
Vanesa Medina Cuadrado	Instructora	ADSO	02 de noviembre de 2024	

## 7. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

Nombre Autor (es)	Cargo	Dependencia		Fecha
	Instructor	ADSO	actualización	