**Marcar con una X la opción que considere correcta.**

**1. ¿Qué es un algoritmo?**

a) Un lenguaje de programación.

~~b) Un conjunto de instrucciones para resolver un problema.~~

c) Un tipo de dato.

d) Un concepto de programación orientada a objetos.

**2. ¿Qué es una variable en programación?**

a) Un valor constante.

~~b) Un espacio de almacenamiento para un valor.~~

c) Un tipo de dato.

d) Un operador.

**3. ¿Cuál de los siguientes es un principio de la Programación Orientada a Objetos?**

~~a) Herencia~~  
b) Recursividad  
c) Iteración  
d) Variables globales

**4. ¿Qué es un constructor en POO?**

a) Un tipo de método que crea una copia de un objeto.  
b) Un método que permite modificar las propiedades de un objeto.  
~~c) Un método especial que se ejecuta cuando se crea una nueva instancia de una clase.~~  
d) Una propiedad que define el valor de una clase.

**5. ¿Qué significa "encapsulamiento" en la POO?**

a) Crear instancias de clases.  
~~b) Ocultar los detalles internos de un objeto y exponer solo lo necesario.~~  
c) Hacer que las clases hereden características de otras clases.  
d) Modificar los valores de las propiedades de un objeto sin restricciones.

**6. ¿Qué tipo de acceso tiene una propiedad definida como private en una clase de JavaScript?**

a) Público, accesible desde cualquier parte.  
~~b) Solo accesible desde la clase en la que se define.~~  
c) Accesible desde cualquier clase hija.  
d) No se puede modificar ni acceder.

**Preguntas de verdadero y falso.**

7. La herencia permite que una clase herede atributos y métodos de otra clase.

Verdadero

8. En POO, un objeto es una instancia de una clase, que contiene solo métodos, pero no atributos.

Falso

9. El polimorfismo permite que diferentes clases respondan al mismo mensaje o método de manera diferente.

Verdadero

10.En POO, un constructor es un tipo de método especial que se utiliza para destruir un objeto cuando ya no es necesario.

Falso

11. La encapsulación es un principio de POO que implica ocultar los detalles internos de una clase y solo exponer una interfaz pública para interactuar con ella.

Verdadero

**Preguntas de redacción**

12. Escribir un programa en **JavaScript** que contenga una clase mensaje, un atributo nombre y un método para saludar el usuario con el nombre.

class Mensaje{

let nombre;

constructor(nombre){

this.nombre = nombre;

}

Saludar(){

Console.log("Hola, mi nombre es: + this.nombre");

}

}

let mensaje = new Mensaje('Isabel');

mensaje.Saludar();

13. Escribir un programa en **JavaScript** que aplique el concepto de **encapsulamiento**.

class Persona{

let nombre;

setNombre(nombreP){

this.nombre = nombreP;

}

getNombre(){

return this.nombre;

}

}

let persona = new Persona();

persona.setNombre('María');

console.log(getNombre());

14. Escribir un programa en **JavaScript** de un sistema de varios vehículos: moto, carro, avión que aplique el concepto de **herencia**.

class Vehiculo{

let modelo;

MostrarVehiculo(modelo){

this.modelo = modelo;

Console.log("El vehículo es modelo +modelo");

};

Acelerar(){

Console.log("El vehículo está acelerando");

;

Frenar(){

Console.log("El vehículo está frenando");

};

}

class Moto extends Vehiculo{

AcelerarMoto(){

Console.log("La moto está acelerando.");

}

FrenarMoto(){

Console.log("La moto está frenando.");

}

}

class Carro extends Vehiculo{

AcelerarCarro(){

Console.log("El carro está acelerando.");

}

FrenarCarro(){

Console.log("El carro está frenando.");

}

}

class Avion extends Vehiculo{

Volar(){

Console.log("El avión está volando.");

}

}

let moto = new Moto();

moto.MostrarVehiculo(2023);

moto.Acelerar();

moto.AcelerarMoto();

let carro = new Carro();

carro.MostrarVehiculo(2025);

carro.FrenarCarro();

let avion = new Avion();

avion.Volar();

15. Escribir un programa en **JavaScript** que aplique el concepto de **polimorfismo**.

abstract class Animal{

abstract Moverse();

abstract Comer();

}

class Perro extends Animal{

Moverse(){

Console.log("El perro puede caminar o correr.");

};

Comer(){

Console.log("EL perro se alimenta de purina y toma agua.");

};

}

class Pajaro extends Animal{

Moverse(){

Console.log("El pájaro vuela.");

};

Comer(){

Console.log("El pájaro come algunas lombrices.");

};

}

let perro = new Perro();

perro.Moverse();

perro.Comer();

let pajaro = new Pajaro();

pajaro.Moverse();

pajaro.Comer();

**16. Pregunta:**

Texto

El contenido generado por IA puede ser incorrecto.

**¿Cuál es el resultado al ejecutar este código?**

~~a) Uncaught TypeError: perro.saludar is not a function~~  
b) Hola, soy Rex  
c) undefined  
d) Mi nombre es Rex

**17. Pregunta:**

Texto

El contenido generado por IA puede ser incorrecto.

**¿Cuál es el resultado al ejecutar este código?**

1. undefined  
   ~~b) Uncaught TypeError: Cannot read property 'nombre' of undefined~~  
   c) Mi nombre es undefined  
   d) error

**18. Pregunta:**

Texto

El contenido generado por IA puede ser incorrecto.

**¿Cuál es el resultado al ejecutar este código?**

**a) Uncaught TypeError: Perro is not a constructor  
~~b) Woof! Soy Rex~~c) undefined  
d) error**

**19. Pregunta:**

Texto

El contenido generado por IA puede ser incorrecto.

**¿Cuál es el resultado al ejecutar este código?**

**~~a) Uncaught TypeError: persona1.saludar is not a function~~  
b) Hola, soy Juan  
c) Uncaught TypeError: Cannot read property 'nombre' of undefined  
d) error**

**20. Pregunta:**

Texto

El contenido generado por IA puede ser incorrecto.

**¿Qué error ocurre al ejecutar este código?**

**~~a) Uncaught ReferenceError: this.nombre is not defined~~  
b) Soy un undefined  
c) Uncaught TypeError: Cannot read property 'saludar' of undefined  
d) Hola, soy Rex**