ACTIVIDAD POO:

• Clase Persona 1. java

```
public class persona {
  private string nombre;
  private int edad;
  private boolean activo;
public persona (string nombre, int edad, boolean activo) {
  this.nombre=nombre;
  this.edad=edad;
  this.activo=activo;
}
public void saludar() {
  System.out.println("Hola, soy" +nombre);
}
public int CalcularAñosRestantes (int edadRetiro) {
  return edadRetiro -edad;
  }
public string getNombre() {return nombre; }
public void setNombre (string nombre) {this.nombre=nombre;}
public int getEdad () {return edad;}
public void setEdad (int edad) {this.edad=edad;}
public boolean isActivo () {return activo;}
public void setActivo (boolean activo) {this.activo=activo;}
public static void main (string [] args) {
  persona p1 =new persona ("Daniela", 18, true);
  p1.saludar ();
  system.out.Println ("Años para Jubilarse:" + p1.Calcularanosrestante());
  }
}
```

• Clase Auto 2. java

```
public class Auto {
  private string marca;
  private string modelo;
  private int velocidad;
public Auto (string marca, string modelo) {
  this.marca = marca;
  this.modelo=modelo;
  this.velocidad=0;
}

public void acelerar (int incremento) {
  velocidad += incremento;
  system.out.println("El auto se ha acelerado:" + velocidad +"Km/h");
  }

public void frenar () {
  velocidad=0;
  System.out.Println ("El auto se ha detenido");
}

public int obtenerVelocidad () {
  return velocidad;
```

```
}
Public static void main (String [] args) {
  Auto a1 = new Auto ("Toyota", "Corolla");
  a1.acelerar (60);
  System.out.println ("Velocidad actual: "+a1.obtenerVelocidad());
  a1.frenar ();
}
}
```

- Clase libro 3 java

```
Public Class libro{
  Private string titulo;
  private string autor;
  private boolean disponible;
Public Libro (String titulo, string autor) {
  this.titulo = titulo;
  this.autor = autor;
  this.disponible = true;
}

Public void prestar () {
  if (disponible) {
  disponible= false;
  system.out.Println ("Libro Prestado: "+ titulo);
  } else {
    system.out.Println ("El libro no esta disponible");
  }
}

Public void devolver () {
  disponible=true;
  system.out.println ("El libro devuelto:"+ titulo);
}

Public boolean estaDisponible () {
  return disponible;
}

Public static void main (String [] args) {
  Libro l1 = new Libro ("Cien años de soledad", "Gabriel Garcia Marquez");
  l1.Prestar ();
  l1.Devolver();
}
  }
```

- Clase Computador 4 java

```
Public class Computador {
  Private string marca;
  Private int ram;
  private boolean encendido;
Public computador (String marca, int ram) {
  this.marca = marca;
  this.ram = ram;
  this.encendido = false;
```

```
}

public void encender() {
  encendido = true;
  System.out.println("El computador está encendido");
}

public void apagar() {
  encendido = false;
  System.out.println("El computador se ha apagado");
}

public boolean verificarEstado() {
  return encendido;
}

public static void main (String[] args) {
  Computador pc=new computador ("Lenovo", 16);
  pc.encender();
  System.out.println("¿Encendido?" + pc.verificarEstado());
  pc.apagar();
  }
}
```

- Clase Animal 5 java

```
abstract class Animal {
  private string nombre;
  private int edad;

public Animal (string nombre, int edad) {
  this.nombre=nombre;
  this.edad=edad;
}

public string getNombre() {return nombre; }
public int getEdad() {return edad; }
public abstract void hacerSonido();
}

Class perro extends Animal {
  public perro (string nombre, int edad) {
  super (nombre, edad);
}

  @override
  public void hacerSonido() {
  System.out.println(getNombre() + "ladra: ¡Guao guao!");
}

  public static void main (String[] args) {
  perro p = new perro ("Rocky", 4);
  p.hacerSonido();
  }
}
```

• Clase Estudiante 1. C#

```
using system;

Public class Estudiante
{
  private string nombre;
  private int nota;
  private bool aprobado;

Public Estudiante (string nombre, int nota)
{
  this.nombre=nombre;
  this.nota=nota;
  this.aprobado=nota >=60;
}

Public void MostrarInfo ()
{
  Console.writeline ($"Estudiante: {nombre}, Nota: {nota}, Aprobado: {aprobado}");
}
  Public bool EstaAprobado()
{
  return aprobado;
}
Public static void Main (string[] args)
{
  Estudiante e1=new Estudiante ("Camila",85);
  e1.MostrarInfo();
  }
}
```

• Clase Paciente 2 C#

```
using system;

Public class paciente
{
  private string nombre;
  private double temperatura;
  private bool enTratamiento;
Public paciente(string nombre, double temperatura)
{
  this.nombre = nombre;
  this.temperatura = temperatura;
  this.enTratamiento = false;
}
Public void IniciarTratamiento()
{
  enTratamiento=true;
  Console.writeline ($"{nombre} ha iniciado tratamiento.");
  }

Public void MostrarEstado()
{
  Console.writeline ($"{nombre} {Temperatura} °C - Tratamiento: {entratamiento}
}
```

```
Public static void Main (string [] args)
  {
  Paciente p1=new Paciente ("Juan", 37.8);
  p1. IniciarTratamiento();
  p1. MostrarEstado();
  }
}
```

• Clase Empleado 3 C#

```
using System;

Public Class Empleado
{
  private string nombre;
  private double SalarioBase;
Public Empleado (string nombre, double SalarioBase)
  {
  this.nombre = nombre;
  this.SalarioBase = SalarioBase;
}
Public double CalcularBono (double Porcentaje)
  {
  return SalarioBase * porcentaje / 100;
}
Public void MostrarSalarioTotal (double porcentaje)
  {
  double total=SalarioBase + CalcularBono (porcentaje)
Console.WriteLine ($"{nombre} gana un total de: {total:C}");
}
public static void Main (string [] args)
  {
  Empleado emp1=new Empleado ("Lauro", 1500000);
  emp1. MostrarSalarioTotal (10);
  }
}
```

• Clase Restaurante 4 C#

```
using System;

public class Restaurante
{
  private string nombre;
  private int mesasDisponibles;
public Restaurante (string nombre, int mesas)
}
  Public void ReservarMesa()
{
  if (mesasDisponibles > 0)
  {
  mesasDisponibles--;
  Console.WriteLine ($"Mesa reservada en {nombre}. Quedan {mesasDisponibles}
}
  else
```

```
{
  Console.WriteLine("No hay mesas disponible");
}
  }
Public int MesasRestantes()
  {
  return mesasDisponibles;
}

Public static void Main(String[] args)
{
  Restaurante r1=new Restaurante("Sabor Colombiano", 3);
  r1.ReservaMesa();
  r1.ReservaMesa();
Console.WriteLine($"Mesas Libres: {r1.MesasRestantes()}");
  }
}
```

• Clase Abstracta Dispositivo + clase hija Celular 5 C#

```
using System;

public abstract class Dispositivo
  {
  public string Marca {get;set;}
  public Dispositivo(string marca)
{
  Marca=marca;
}
  public abstract void Encender();
  public abstract void Apagar();
}
  Public class Celular:Dispositivo
  {
  public Celular(string marca):base(marca){}
  public override void Encender
{
  Console.WriteLine($"{Marca} encendido.");
}
  public override void Apagar()
{
  Console.WriteLine($"{Marca} apagado.");
}
  public static void Main(String[] args)
{
  Celular c1=new Celular("Samsung");
  c1.Encender();
  c1.Apagar();
  }
}
```

• Clase Juego 1 Javascript

```
Class Juego {
#nombre;
#nivel;
  this.#nombre = nombre;
  this.#nivel = nivel;
}
IniciarJuego() {
  Console.log(`Iniciando el juego: ${this.#nombre}`);
}
  SubirNivel (puntos) {
  this.#nivel += puntos;
  Console.log(`${this.#nombre} subio al nivel ${this.#nivel}`);
}

  get nivel () {
  return this.#nivel;
}
  Set nivel (nuevoNivel) {
  this.#nivel=nuevoNivel;
  }
}
  Const juego1=new Juego("Aventura Mágica", 3);
  Juego1.iniciarJuego();
  Juego1.SubirNivel (2);
  Console.log("Nivel actual:", Juego1.nivel);
```

• Clase Casa 2 Javascript

```
Class Casa {
Constructor (direccion, habitaciones) {
  this.direccion=direccion;
  this.habitaciones = habitaciones;
  this.ocupada=false;
}
OOupar () {
  this.ocupada=true;
  Console.log(`la casa en ${this.direccion} ahora esta ocupada`);
}

liberal () {
  this.ocupada = false;
  Console.log(`la casa en ${this.direccion} esta libre.`);
}
VerificarEstado ()
  return this.ocupada? "Ocupada":"Dioponible";
  }
}
Const Casa 1=new Casa ("Calle 12 #45", 4);
  Casa 1.Ocupar();
  Console.log (Casa1.VerificarEstado());
  Casa1.liberar ();
```

• Clase Producto 3 Javascript

```
class Producto {
  #precio;
Constructor (nombre, precio) {
  this.nombre = nombre;
  this.precio = precio;
  }

aplicarDescuento (porcentaje) {
  const descuento = (this.#precio * porcentaje) / 100;
  this.#Precio -= descuento;
  console.log (`Descuento aplicado. Nuevo precio: $${this.#precio}`);
}
get Precio () {
  return this.#precio;
}
set precio (nuevoPrecio) {
  if (nuevoPrecio > 0) this.#precio = nuevoPrecio;
}
  }
const prod1 = new Producto("Camisa", 80000);
  prod1.aplicarDescuento (20);
  console.log ("Precio final:", prod1.precio);
```

• Clase Profesor 4 JavaScript

```
class profesor {
  Constructor (nombre, materia, años Experencia) {
  this.nombre = nombre;
  this.materia = materia;
  this.añosExperencia = años Experencia;
}
Presentarse () {
  console.log (`soy ${this.nombre}, profesor de ${this.materia}.`);
}
CalcularSalario (base, bono) {
  return base + bono * this.años Experencia;
  }
}
const prof1 = new Profesor ("Carlos", "Matematicas", 5);
  prof1.Presentarse ();
  console.log ("Salario total:", prof1.CalcularSalario (2000000, 150000));
```

• Clase Planta 5 Javascript

```
class Planta {
  Constructor (nombre) {
  if (new.target == Planta) {
  throw new Error ("No se puede instanciar una clase abstracta");
}
  this.nombre = nombre;
}
  Crecer () {
  Console.log (`${this.nombre} está creciendo...`);
}
```

```
florecer () {
  throw new Error ("Este metodo debe ser implementado subclase.");
  }
}
  Class flor extends planta {
  Constructor (nombre, color) {
  Super (nombre);
  this.Color = Color;
}
  florecer () {
Console.log ("La flor ${this.nombre} de color ${this.color} ha florecido);
  }
}
  Const flor1 = new flor ("Rosa", "roja");
  flor1.crecer();
}flor1.florecer();
```

• Clase CuentaBancaria go 1

```
package main
  import "fmt"
type CuentaBancaria struct {
  titular string
  Saldo float64
}
fun (c* CuentaBancaria) Depositar (monto float64) {
    c.Saldo t= monto
    fmt.Println ("Deposito realizado. Saldo actual:", c.Saldo)
}
fun (c* CuentaBancaria) Retirar (monto float64) {
  if monto <= c.Salado {
  c.Saldo -= monto
  fmt.Println ("Retiro exitoso. Saldo restante:", c.Saldo)
} else {
  fmt.Println ("fondos  insuficientes.")
}
  }
fun (c CuentaBancaria) ConsultarSaldo () float64 {
  return c.Saldo
}
func main () {
  Cuenta := CuentaBancaria {titular: "Daniela", Saldo: 500000}
  Cuenta.Depositar (200000)
  Cuenta.Retirar (300000)
  fmt.Println ("Saldo final:", Cuenta.ConsultarSaldo())
}
```

• Struct Bicicleta 2 go

```
package main

import "fmt"

type Bicicleta struct {
  Marca string
```

```
velocidad int
}
func (b* Bicicleta) Acelerar (incremento int) {
  b.velocidad += incremento
  fmt.Println("la bicicleta aceleró a", b.velocidad, "km/h")
}
func (b* Bicicleta) frenar () {
  b.Velocidad = 0
  fmt.Println("la bicicleta se ha detenido.")
}

func main() {
  bici := Bicicleta{Marca: "GW", velocidad: 10}
  bici.Acelerar(15)
  bici.frenar()
}
```

- Struct Chef 3 go

```
package main
import "fmt"
type chef struct {
  Nombre string
  plato string
  Experencia int
  }
func (c chef) prepararplato() {
  fmt.Println(c.Nombre, "esta preparando", c.plato)
}
fun (c chef) calcularpago (base float64) float64 {
  return base + float64 (c.Experencia) * 50000
}
func main () {
  Chef := chef{"Camilo", "paella", 5}
  chef.Preparar plato()
  fmt.Println("pago total:", chef.calcularPago(1000000))
}
```

- Struct Hotel 4 go

```
Package main
import "fmt"
type Hotel struct {
  Nombre string
  Habitaciones int
  ocupadas int
}
func (h* Hotel) Reservar Habitaciones () {
  if h.ocupadas < h.Habitaciones {
  h.ocupadas ++
  fmt.Println("Reserva exitosa. Ocupadas:", h.ocupadas)
} else {
  fmt.Println("No hay habitaciones disponibles.")
  }
}
func (h* Hotel) Disponibles () int {
return h.Habitaciones - h.ocupadas
```

```
}
func main() {
    hotel := Hotel{"Sol caribe", 10, 8}
    hotel.ReservarHabitaciones()
    fmt.Printlh("Habitaciones disponibles", habitacion.Disponibles())
}
```

- Celular y Tablet 5 go

```
package main
import "fmt"
type Dispositivo interface {
    Encender()
    Apagar()
}
type Celular struct {
    Marca string
}
fun (c celular) encender() {
    fmt.Println(c.Marca, "apagado.")
}
type Tablet struct {
    Modelo string
}
func (t Tablet) Encender() {
    fmt.Println(t.Modelo, "apagada.")
}
func main() {
    var d1 Dispositivo = Celular{"Samsung"}
    var d2 Dispositivo = Tablet{"iPad Air"}
    d1.Encender()
    d2.Encender()
    d1.Apagar()
    d2.Apagar()
}
```