

1. Crear una clase "Persona" con atributos como nombre, edad, altura y peso. Luego, crear un objeto "Persona" y asignar valores a sus atributos. Por último, imprimir los valores de los atributos de la persona.
2. Crear una clase "Rectángulo" con atributos como largo y ancho. Luego, crear métodos para calcular el área y el perímetro del rectángulo. Por último, crear un objeto "Rectángulo" y llamar a los métodos para imprimir el área y el perímetro.
3. Crear una clase "CuentaBancaria" con atributos como número de cuenta, nombre del titular y saldo. Luego, crear métodos para depositar y retirar dinero de la cuenta, y para imprimir el saldo actual. Asegurarse de que el método de retiro no permita retirar más dinero del que hay en la cuenta.
4. Crear una clase "Animal" con atributos como especie y edad. Luego, crear métodos para hacer que el animal hable (imprimir su especie y edad) y para hacer que el animal se mueva (imprimir "El animal se está moviendo"). Luego, crear subclases de "Animal" para diferentes especies (por ejemplo, "Perro", "Gato", "León") y agregar métodos específicos para cada una de estas subclases (por ejemplo, "Ladrar" para "Perro").
5. Crear una clase "Juego" con atributos como número de jugadores y nivel de dificultad. Luego, crear métodos para iniciar el juego, detener el juego y para imprimir el estado actual del juego. Luego, crear subclases de "Juego" para diferentes juegos (por ejemplo, "Ajedrez", "Póker", "Ahorcado") y agregar métodos específicos para cada una de estas subclases (por ejemplo, "Mover pieza" para "Ajedrez").
6. Crear una clase "Circulo" con atributos como radio y centro (que es un objeto "Punto" que representa las coordenadas x e y del centro del círculo). Luego, crear métodos para calcular el área y el perímetro del círculo.
7. Crear una clase "Estudiante" con atributos como nombre, edad, calificación promedio y lista de cursos. Luego, crear métodos para agregar cursos a la lista, calcular el promedio de calificaciones y para imprimir la información del estudiante.
8. Crear una clase "Empleado" con atributos como nombre, cargo y salario. Luego, crear métodos para aumentar el salario del empleado y para imprimir su información. Luego, crear subclases de "Empleado" para diferentes tipos de empleados (por ejemplo, "Gerente", "Desarrollador", "Vendedor") y agregar métodos específicos para cada una de estas subclases (por ejemplo, "Asignar bono" para "Gerente").
9. Crear una clase "Vehiculo" con atributos como marca, modelo y año. Luego, crear métodos para encender el vehículo, apagar el vehículo y para imprimir la información del vehículo. Luego, crear subclases de "Vehiculo" para diferentes tipos de vehículos (por ejemplo, "Automóvil", "Motocicleta", "Camión") y agregar métodos específicos para cada una de estas subclases (por ejemplo, "Activar luces" para "Automóvil").

10. Crear una clase "Biblioteca" con atributos como nombre y lista de libros. Luego, crear métodos para agregar libros a la lista, retirar libros de la lista y para imprimir la información de la biblioteca.
11. Crear una clase "Persona" con atributos como nombre, edad y género. Luego, crear métodos para imprimir la información de la persona y para calcular la edad de la persona en años, meses y días.
12. Crear una clase "CuentaBancaria" con atributos como número de cuenta, titular de la cuenta y saldo. Luego, crear métodos para realizar depósitos y retiros, así como para imprimir la información de la cuenta bancaria.
13. Crear una clase "Animal" con atributos como nombre, edad y especie. Luego, crear métodos para imprimir la información del animal y para calcular la edad del animal en años, meses y días.
14. Crear una clase "FiguraGeometrica" con atributos como tipo de figura y medidas (por ejemplo, si es un triángulo, tendrá medidas de base, altura y lados). Luego, crear métodos para calcular el área y el perímetro de la figura geométrica.
15. Crear una clase "Producto" con atributos como nombre, descripción y precio. Luego, crear métodos para calcular el precio de venta y para imprimir la información del producto

16. Crear una clase "Vehiculo" con atributos como marca, modelo, año de fabricación y kilometraje. Luego, crear métodos para realizar mantenimientos, para calcular el consumo de combustible y para imprimir la información del vehículo.
17. Crear una clase "Empleado" con atributos como nombre, edad, salario y fecha de contratación. Luego, crear métodos para calcular el salario anual del empleado, para imprimir la información del empleado y para calcular la antigüedad del empleado en años, meses y días.
18. Crear una clase "Libro" con atributos como título, autor, editorial y año de publicación. Luego, crear métodos para calcular el número de páginas del libro, para imprimir la información del libro y para buscar una palabra clave dentro del contenido del libro.
19. Crear una clase "Universidad" con atributos como nombre, dirección y lista de estudiantes. Luego, crear métodos para agregar estudiantes, para eliminar estudiantes y para imprimir la información de la universidad.
20. Crear una clase "Pelicula" con atributos como título, género, director y lista de actores. Luego, crear métodos para agregar actores, para eliminar actores y para imprimir la información de la película.

21. Crear una clase "Empresa" con atributos como nombre, dirección, lista de empleados y lista de productos/servicios. Luego, crear métodos para

- agregar empleados, para eliminar empleados, para agregar productos/servicios y para eliminar productos/servicios.
22. Crear una clase "CuentaBancaria" con atributos como titular, saldo y lista de transacciones. Luego, crear métodos para agregar transacciones, para imprimir el historial de transacciones y para hacer depósitos y retiros.
 23. Crear una clase "Juego" con atributos como nombre, descripción, género y lista de jugadores. Luego, crear métodos para agregar jugadores, para eliminar jugadores, para iniciar el juego y para finalizar el juego.
 24. Crear una clase "RedSocial" con atributos como nombre, lista de usuarios y lista de publicaciones. Luego, crear métodos para agregar usuarios, para eliminar usuarios, para agregar publicaciones y para eliminar publicaciones.
 25. Crear una clase "Tienda" con atributos como nombre, dirección, lista de productos y lista de clientes. Luego, crear métodos para agregar productos, para eliminar productos, para agregar clientes y para eliminar clientes.
 26. Crear una clase "Sistema de Reservas" que permita a los usuarios reservar habitaciones en un hotel. La clase debe tener atributos como la lista de habitaciones disponibles, la lista de reservaciones y el número máximo de habitaciones que se pueden reservar a la vez. Luego, crear métodos para agregar habitaciones, para eliminar habitaciones, para agregar reservaciones y para eliminar reservaciones.
 27. Crear una clase "Banco" con atributos como nombre, dirección, lista de cuentas bancarias y lista de transacciones. Luego, crear métodos para agregar cuentas bancarias, para eliminar cuentas bancarias, para hacer depósitos, para hacer retiros y para imprimir el historial de transacciones.
 28. Crear una clase "Sistema de Ventas" que permita a los usuarios comprar productos en una tienda en línea. La clase debe tener atributos como la lista de productos disponibles, la lista de compras y el número máximo de productos que se pueden comprar a la vez. Luego, crear métodos para agregar productos, para eliminar productos, para agregar compras y para eliminar compras.
 29. Crear una clase "Sistema de Biblioteca" que permita a los usuarios pedir prestados libros de una biblioteca. La clase debe tener atributos como la lista de libros disponibles, la lista de préstamos y el número máximo de libros que se pueden pedir prestados a la vez. Luego, crear métodos para agregar libros, para eliminar libros, para agregar préstamos y para eliminar préstamos.
 30. Crear una clase "Sistema de Restaurante" que permita a los usuarios hacer reservaciones en un restaurante y hacer pedidos de comida. La clase debe tener atributos como la lista de mesas disponibles, la lista de reservaciones, la lista de platos y la lista de órdenes. Luego, crear métodos para agregar mesas, para eliminar mesas, para agregar

reservaciones, para eliminar reservaciones, para agregar platos, para eliminar platos, para agregar órdenes y para eliminar órdenes.

NIVEL 2

1. Crea una clase **Baraja** que represente una baraja de cartas españolas. La baraja debe contener 40 cartas (10 cartas de cada palo). La clase debe tener un método **mezclar** que mezcle las cartas de la baraja y otro método **siguienteCarta** que devuelva la siguiente carta de la baraja. Cuando se llegue a la última carta, se debe reiniciar la baraja y comenzar de nuevo.
2. Crea una clase **Mochila** que represente una mochila que puede contener varios objetos. La mochila debe tener una capacidad máxima y se debe poder añadir y quitar objetos. La clase debe tener un método **pesoTotal** que devuelva el peso total de los objetos que hay en la mochila y otro método **valorTotal** que devuelva el valor total de los objetos que hay en la mochila.
3. Crea una clase **Agenda** que permita gestionar los contactos de una agenda. La agenda debe permitir añadir, eliminar y buscar contactos por nombre. Cada contacto debe tener un nombre, un número de teléfono y una dirección de correo electrónico.
4. Crea una clase **Fraccion** que represente una fracción matemática. La clase debe tener métodos para sumar, restar, multiplicar y dividir fracciones, y para convertir la fracción a un número decimal. También se deben poder comparar fracciones.
5. Crea una clase **Lista** que represente una lista de elementos. La lista debe permitir añadir, eliminar y buscar elementos por posición. También se deben poder obtener el número de elementos de la lista y vaciar la lista.
6. Crea una clase **Calculadora** que tenga métodos para realizar operaciones aritméticas básicas: suma, resta, multiplicación y división. La clase debe permitir operar con números enteros y decimales. También se deben poder realizar operaciones encadenadas.
7. Crea una clase **Ahorcado** que implemente el juego del ahorcado. El juego debe permitir introducir una palabra y luego ir pidiendo letras. Si la letra introducida está en la palabra, se deben mostrar todas las apariciones de la letra en la palabra. Si la letra no está en la palabra, se debe ir dibujando un muñeco ahorcado. El juego termina cuando se adivina la palabra o se completa el dibujo del muñeco.
8. Crea una clase **Laberinto** que represente un laberinto. El laberinto debe estar formado por celdas que pueden estar libres o ser obstáculos. La clase debe tener un método para moverse por el laberinto, teniendo en

- cuenta que no se puede atravesar los obstáculos. El objetivo del juego es llegar de una posición inicial a una posición final del laberinto.
9. Crea una clase **GestorBiblioteca** que permita gestionar los préstamos de una biblioteca. La biblioteca tiene un conjunto de libros que se pueden prestar a los usuarios. La clase debe permitir añadir, eliminar y buscar libros. También debe permitir prestar y devolver libros a los usuarios, y mantener un registro de los préstamos.
 10. Crea una clase **Partida** que represente una partida de un juego de mesa. La partida debe tener un tablero de juego y varios jugadores que pueden moverse por el tablero. La clase debe permitir realizar movimientos en el tablero y llevar un registro de los puntos de cada jugador. También se deben poder guardar y cargar partidas.
 11. Crea una clase **Persona** con los atributos **nombre**, **edad** y **documento**. Implementa un método **esMayorDeEdad()** que devuelve **true** si la edad de la persona es mayor o igual a 18 y **false** en caso contrario. Además, crea una clase **Cliente** que herede de la clase **Persona** y que tenga un atributo **saldo** y un método **comprar(double monto)** que disminuya el saldo en el monto de la compra realizada. Si el saldo es insuficiente, el método deberá imprimir un mensaje indicando que la operación no se puede realizar.
 12. Crea una clase **Banco** que tenga como atributos una lista de clientes y un saldo total. La clase deberá tener un método **agregarCliente(Cliente cliente)** que añada un cliente a la lista, y un método **calcularSaldoTotal()** que calcule y devuelva el saldo total del banco sumando el saldo de todos sus clientes. Además, implementa un método **buscarCliente(String documento)** que busque y devuelva el cliente con el documento indicado. Si el cliente no existe, el método deberá devolver **null**.
 13. Crea una clase **CuentaBancaria** con los atributos **numero**, **saldo** y **cliente**. La clase deberá tener un método **realizarDeposito(double monto)** que aumente el saldo en el monto indicado, y un método **realizarRetiro(double monto)** que disminuya el saldo en el monto indicado. Además, implementa un método **transferir(double monto, CuentaBancaria cuentaDestino)** que transfiera el monto indicado desde la cuenta actual a la cuenta de destino. Si la cuenta actual no tiene suficiente saldo, el método deberá imprimir un mensaje indicando que la operación no se puede realizar.
 14. Crea una clase **Vehiculo** con los atributos **marca**, **modelo**, **año** y **precio**. Implementa un método **getPrecioVenta()** que devuelva el precio de venta del vehículo, el cual será el precio original disminuido en un porcentaje que dependerá del año de fabricación. Si el vehículo tiene más de 5 años, el descuento será del 10%. Si tiene más de 10 años, el descuento será del 20%.

15. Crea una clase **Zoo** que tenga como atributos una lista de animales y un nombre. La clase deberá tener un método **agregarAnimal(Animal animal)** que añada un animal a la lista, y un método **listarAnimales()** que imprima por pantalla el nombre y la especie de todos los animales en la lista. Además, implementa un método **buscarAnimal(String nombre)** que busque y devuelva el animal con el nombre indicado. Si el animal no existe, el método deberá devolver **null**.

NIVEL 3

1. Crea una clase llamada "Figura" con los siguientes atributos: nombre (String) y color (String). La clase debe tener un constructor que reciba ambos atributos y un método llamado "dibujar" que imprima en consola "Dibujando [nombre de la figura] de color [color]".
2. Crea una clase llamada "Cuadrado" que herede de la clase "Figura". La clase "Cuadrado" debe tener un atributo adicional llamado "lado" (double) y un constructor que reciba nombre, color y lado. Sobrescribe el método "dibujar" para que imprima en consola "Dibujando un cuadrado de color [color] y de lado [lado]".
3. Crea una clase llamada "Rectangulo" que herede de la clase "Figura". La clase "Rectangulo" debe tener dos atributos adicionales llamados "base" (double) y "altura" (double) y un constructor que reciba nombre, color, base y altura. Sobrescribe el método "dibujar" para que imprima en consola "Dibujando un rectángulo de color [color], base [base] y altura [altura]".
4. Crea una clase llamada "Circulo" que herede de la clase "Figura". La clase "Circulo" debe tener un atributo adicional llamado "radio" (double) y un constructor que reciba nombre, color y radio. Sobrescribe el método "dibujar" para que imprima en consola "Dibujando un círculo de color [color] y radio [radio]".
5. Crea una clase llamada "Figuras" que tenga un atributo llamado "figuras" que sea un arreglo de objetos de la clase "Figura". La clase debe tener un constructor que reciba como parámetro el arreglo de figuras y un método llamado "dibujarTodo" que llame al método "dibujar" de cada una de las figuras en el arreglo.
6. Crear una clase abstracta llamada **FiguraGeometrica** que contenga un método abstracto llamado **calcularArea()**. Crear dos clases que hereden de esta clase abstracta llamadas **Circulo** y **Triangulo** que implementen este método para calcular el área de cada figura. Luego, crear una clase **Main** que instancie un objeto de cada clase y muestre en consola el área calculada.
7. Crear una clase **Auto** con los atributos **marca**, **modelo**, **color** y **precio**. Crear una clase **Concesionaria** que contenga un **ArrayList** de Autos y los

- métodos `agregarAuto()`, `eliminarAuto()` y `buscarAuto()` que permitan agregar, eliminar y buscar autos por marca o modelo. Crear una clase `Main` que permita al usuario agregar, eliminar y buscar autos en la concesionaria.
8. Crear una clase `Persona` con los atributos `nombre`, `edad` y `dni`. Crear una clase `Cliente` que herede de `Persona` y contenga un `ArrayList` de `CuentaBancaria`. Crear una clase `CuentaBancaria` con los atributos `numero`, `saldo` y `tipo` y los métodos `depositar()`, `extraer()` y `mostrarSaldo()`. Crear una clase `Main` que permita al usuario crear un cliente, agregar cuentas bancarias y realizar operaciones bancarias como depósitos y extracciones.
 9. Crear una clase `Vuelo` con los atributos `numero`, `origen`, `destino`, `horaSalida`, `horaLlegada` y `precio`. Crear una clase `Pasajero` con los atributos `nombre`, `apellido`, `dni`, `fechaNacimiento` y `nacionalidad`. Crear una clase `Reserva` con los atributos `vuelo`, `pasajero`, `asiento` y `precioReserva`. Crear una clase `Main` que permita al usuario reservar un vuelo ingresando los datos del pasajero y el vuelo y asignando un asiento disponible.
 10. Crear una clase abstracta `Animal` con los atributos `nombre`, `edad` y `tipo` y los métodos `comer()`, `dormir()` y `hacerSonido()`. Crear dos clases que hereden de esta clase abstracta llamadas `Perro` y `Gato` que implementen estos métodos. Luego, crear una clase `Main` que instancie un objeto de cada clase y muestre en consola los sonidos que hacen cada uno.
 11. Crea una clase abstracta "FiguraGeometrica" con los métodos abstractos "calcularArea" y "calcularPerimetro". Luego, crea las clases "Rectangulo", "Triangulo" y "Circulo", todas ellas heredando de "FiguraGeometrica" y sobrescribiendo los métodos para calcular su área y perímetro.
 12. Crea una clase "Mascota" con las propiedades "nombre", "edad" y "especie". Luego, crea las clases "Perro" y "Gato", ambas heredando de "Mascota". Cada clase debe tener su propio método "hacerSonido" que imprima un sonido característico de cada animal.
 13. Crea una clase "Banco" que tenga una lista de "CuentaBancaria". La clase "CuentaBancaria" debe tener las propiedades "numeroCuenta", "saldo" y "titular". La clase "Banco" debe tener métodos para agregar cuentas, eliminar cuentas, transferir dinero entre cuentas y buscar una cuenta por su número.
 14. Crea una clase "Juego" con una propiedad "tablero" que será un arreglo bidimensional de tamaño $n \times n$. La clase debe tener métodos para iniciar el juego, hacer un movimiento y verificar si hay un ganador.
 15. Crea una clase "Tienda" que tenga una lista de "Producto". La clase "Producto" debe tener las propiedades "nombre", "precio" y "cantidad". La clase "Tienda" debe tener métodos para agregar productos, eliminar productos, buscar productos por nombre y obtener el valor total de los productos en la tienda.

16. Crea una clase "Banco" que tenga una lista de clientes y una lista de cuentas bancarias. Cada cuenta bancaria debe estar asociada a un cliente y tener un saldo y un número de cuenta. Los clientes deben tener un nombre, un número de identificación y una lista de cuentas bancarias. La clase Banco debe tener métodos para agregar clientes, agregar cuentas bancarias a un cliente, transferir dinero entre cuentas y obtener el saldo total del banco.
17. Crea una clase "Empresa" que tenga una lista de empleados y una lista de proyectos. Cada proyecto debe estar asociado a uno o más empleados y tener un nombre y una fecha de inicio. Los empleados deben tener un nombre, un número de identificación y una lista de proyectos. La clase Empresa debe tener métodos para agregar empleados, agregar proyectos a un empleado, asignar un empleado a un proyecto y obtener la lista de empleados que trabajan en un proyecto en particular.
18. Crea una clase "Zoo" que tenga una lista de animales y una lista de empleados. Cada animal debe tener un nombre, una especie y una edad. Cada empleado debe tener un nombre, un número de identificación y un puesto (como veterinario, cuidador, etc.). La clase Zoo debe tener métodos para agregar animales, agregar empleados, asignar un empleado a un animal y obtener la lista de empleados que trabajan con un tipo de animal en particular.
19. Crea una clase "Universidad" que tenga una lista de estudiantes, una lista de profesores y una lista de cursos. Cada curso debe estar asociado a un profesor y tener un nombre y un horario. Cada estudiante debe tener un nombre, un número de identificación y una lista de cursos. Cada profesor debe tener un nombre, un número de identificación y una lista de cursos que enseña. La clase Universidad debe tener métodos para agregar estudiantes, agregar profesores, agregar cursos, asignar un estudiante a un curso y obtener la lista de estudiantes que toman un curso en particular.
20. Crea una clase "Hospital" que tenga una lista de pacientes, una lista de médicos y una lista de enfermeros. Cada paciente debe tener un nombre, un número de identificación y una lista de médicos y enfermeros que lo atienden. Cada médico debe tener un nombre, un número de identificación y una lista de pacientes a los que atiende. Cada enfermero debe tener un nombre, un número de identificación y una lista de pacientes a los que atiende. La clase Hospital debe tener métodos para agregar pacientes, agregar médicos, agregar enfermeros, asignar un médico a un paciente, asignar un enfermero a un paciente y obtener la lista de pacientes que atiende un médico o un enfermero en particular.

Ejercicio 1: Sistema de gestión de una biblioteca

Crea un sistema de gestión de una biblioteca que contenga las siguientes clases:

- **Libro**: una clase que represente a un libro, con los atributos **título**, **autor**, **isbn**, **editorial** y **numPaginas**.
- **Usuario**: una clase que represente a un usuario de la biblioteca, con los atributos **nombre**, **apellido**, **dni** y **fechaNacimiento**.
- **Prestamo**: una clase que represente el préstamo de un libro a un usuario, con los atributos **libro**, **usuario**, **fechaInicio** y **fechaFin**.
- **Biblioteca**: una clase que contenga un listado de libros y un listado de usuarios, y permita prestar y devolver libros.

La clase **Biblioteca** debe tener los siguientes métodos:

- **prestarLibro(libro: Libro, usuario: Usuario, fechaInicio: Date, fechaFin: Date): boolean**: este método debe permitir prestar un libro a un usuario. Debe verificar que el libro esté disponible y que el usuario no tenga más de 3 libros prestados. Si se puede prestar el libro, debe crear un objeto de tipo **Prestamo** y agregarlo a una lista de préstamos de la biblioteca. El método debe retornar **true** si el préstamo se realizó exitosamente, y **false** en caso contrario.
- **devolverLibro(libro: Libro, usuario: Usuario): boolean**: este método debe permitir devolver un libro a la biblioteca. Debe verificar que el libro esté prestado por el usuario, y que la fecha actual sea anterior a la fecha de fin del préstamo. Si se puede devolver el libro, debe eliminar el objeto de tipo **Prestamo** correspondiente de la lista de préstamos de la biblioteca. El método debe retornar **true** si la devolución se realizó exitosamente, y **false** en caso contrario.
- **buscarLibro(isbn: string): Libro**: este método debe buscar un libro en el listado de libros de la biblioteca, a partir de su ISBN. Si el libro existe, debe retornar el objeto de tipo **Libro**. En caso contrario, debe retornar **null**.
- **buscarUsuario(dni: string): Usuario**: este método debe buscar un usuario en el listado de usuarios de la biblioteca, a partir de su DNI. Si el usuario existe, debe retornar el objeto de tipo **Usuario**. En caso contrario, debe retornar **null**.

Además, la clase **Prestamo** debe tener el siguiente método:

- **estaEnCurso(): boolean:** este método debe retornar **true** si el préstamo está en curso (es decir, si la fecha actual es posterior a la fecha de inicio y anterior a la fecha de fin), y **false** en caso contrario.

Ten en cuenta que la fecha de inicio y la fecha de fin de un préstamo deben ser objetos de tipo **Date**.

COMPLEMENTARIO

Construye un sistema de gestión de una biblioteca en Java, que permita gestionar el registro de libros y usuarios de la biblioteca.

Los usuarios deberán poder registrarse en la biblioteca y buscar y reservar libros en el catálogo de la biblioteca. Cada libro deberá tener un título, autor, género y número de copias disponibles. Los usuarios deberán poder reservar una copia de un libro y ver su historial de reservas.

El sistema debe ser capaz de verificar si un usuario tiene multas pendientes y no permitir la reserva de nuevos libros hasta que se hayan pagado las multas.

Además, el sistema debe ser capaz de proporcionar estadísticas sobre los libros más populares y los usuarios más activos.

Este ejercicio requiere de múltiples clases, uso de herencia, polimorfismo, encapsulamiento, manejo de excepciones.