



## Taller #2 Java Globant – Programación Orientada a Objetos (POO)

### Nivel 1 – Conceptos Fundamentales de POO

1. **Clase `Producto`:**

Crea una clase `Producto` con atributos `nombre`, `precio`, y `cantidadEnStock`. Implementa métodos para aumentar o disminuir el stock (sin permitir cantidades negativas) y un método para calcular el valor total del stock.

2. **Clase `Libro` y Método de Impresión:**

Diseña una clase `Libro` con atributos `titulo`, `autor`, y `numPaginas`. Implementa un método `imprimirResumen()` que muestre el título y el autor, y otro método que determine si el libro tiene más de 300 páginas.

3. **Sistema Básico de vehículos:**

Crea una clase `Vehiculo` con atributos `marca`, `modelo`, y `velocidadMaxima`. Implementa métodos para *acelerar* y *frenar*, modificando la velocidad actual, y asegúrate de que no exceda la velocidad máxima ni sea negativa.

4. **Clase `Persona` y Comparación de Edades:**

Crea una clase `Persona` con atributos `nombre`, `edad`, y `sexo`. Añade un método que permita comparar la edad entre dos personas, mostrando cuál es mayor.

5. **Clase `CuentaBancaria` con Historial de Transacciones:**

Implementa una clase `CuentaBancaria` con atributos `numeroCuenta`, `titular`, y `saldo`. Añade métodos para depositar y retirar dinero. Guarda un historial de transacciones (tipo y monto) y un método para mostrarlo.

---

6. **Sistema de Agenda de Contactos:**

Diseña una clase `Contacto` con atributos `nombre`, `telefono`, y `email`. Luego, crea una clase `Agenda` que permita agregar, eliminar y buscar contactos por nombre.

7. **Sistema de Gestión de Notas de Estudiantes:**

Crea una clase `Estudiante` con atributos `nombre`, `listaNotas` y un método para agregar notas. Incluye métodos para calcular el promedio y determinar si el estudiante aprobó (promedio  $\geq 60$ ).

8. **Inventario de Productos con Categorías:**

Crea una clase `Producto` con atributos `nombre`, `precio`, y `categoria`. Luego, diseña una clase `Inventario` que gestione una lista de productos y tenga métodos para buscar productos por categoría y calcular el valor total del inventario.

9. **Sistema de Biblioteca con Préstamos de Libros:**

Crea una clase `Libro` con `titulo`, `autor` y un booleano `disponible`. Luego,



implementa una clase `Biblioteca` que permita prestar y devolver libros, y que no permita prestar un libro si ya está prestado.

**10. Sistema de Empleados con Gestión de Bonos:**

Crea una clase `Empleado` con atributos `nombre`, `salarioBase` y `aniosExperiencia`. Implementa métodos para calcular un bono de fin de año en base a su salario y experiencia, siguiendo criterios como: +5% de bono por cada año trabajado.

---

**11. Sistema de vehículos con Herencia:**

Partiendo de la clase `Vehiculo`, crea subclases como `Auto` y `Moto` que agreguen atributos específicos como `numPuertas` o `tipoCasco`. Añade métodos en cada clase para describir sus características.

**12. Sistema de Personajes en un Videojuego con Herencia y Polimorfismo:**

Diseña una clase `Personaje` con atributos `nombre` y `nivel`. Luego, crea subclases como `Guerrero` y `Mago` que tengan habilidades específicas y métodos para atacar. Usa polimorfismo para que cada personaje ataque de manera distinta.

**13. Sistema de Animales en un Zoológico:**

Crea una clase base `Animal` con un método `hacerSonido()`. Luego, crea subclases como `Perro`, `Gato`, y `Pajaro`, cada una implementando su propio sonido. Usa polimorfismo para llamar `hacerSonido()` sin importar el tipo de animal.

**14. Sistema de Pagos con Diferentes Métodos:**

Diseña una clase `Pago` con un método `procesarPago()`. Crea subclases `PagoTarjeta` y `PagoEfectivo` que sobrescriban el método para procesar el pago de diferentes maneras.

**15. Sistema de Mantenimiento de Vehículos en una Flota:**

Diseña una clase `Vehiculo` con atributos `marca`, `modelo`, y `kilometraje`. Luego, crea una clase `Flota` que tenga una lista de vehículos y métodos para registrar mantenimientos basados en el kilometraje de cada vehículo.

---

**16. Sistema de Gestión de Inventario en una Tienda:**

Crea una clase `Producto` y una clase `Inventario` que permita agregar, eliminar y buscar productos por nombre o categoría. Implementa un método para actualizar el stock automáticamente al realizar ventas.

**17. Sistema de Registro de Ventas para una Tienda en Línea:**

Implementa una clase `Cliente` y `Pedido`. Cada pedido debe contener una lista de productos comprados y calcular el total. Crea métodos para realizar la compra y actualizar el inventario de productos en cada pedido.

**18. Aplicación de Control de Horas para Empleados:**

Crea una clase `Empleado` con un método para registrar horas trabajadas. Implementa



una clase `ControlDeHoras` que calcule el salario semanal basándose en las horas trabajadas y el pago por hora. Considera pago extra por horas adicionales.

19. **Sistema de Reservas de Vuelos:**

Diseña clases `Vuelo`, `Pasajero` y `Reserva`. Los pasajeros deben poder reservar y cancelar vuelos, y el sistema debe verificar la disponibilidad de asientos. Lleva un registro de cada reserva con la fecha, el número de asiento y el pasajero.

20. **Sistema de Cuentas Bancarias con Transferencias:**

Crea una clase `Banco` con una lista de `CuentaBancaria`. Cada cuenta debe tener un saldo y un método para realizar transferencias entre cuentas, verificando que no exceda el saldo disponible. Lleva un registro de las transacciones y permite consultar el historial