

## Unidad 5 – Programación orientada a Objetos

**5.1** Crear la siguiente clase:

- **Nombre de la clase:** Empleado
- **Atributos:** nombre, apellido y sueldo
- **Métodos:** mostrar\_informacion()

El método mostrar\_informacion() debe mostrar por pantalla los 3 atributos.

**5.2** Simular como se incrementa la edad de una persona cada vez que cumple años. Crear la siguiente clase:

- **Nombre de la clase:** Persona
- **Atributos:** nombre y edad
- **Métodos:** cumplir\_años() y mostrar\_edad()

El método mostrar\_edad() debe mostrar la edad y el método cumplir\_años() debe incrementar la edad cada vez que se invoque.

**5.3** Simular el cálculo del stock de un producto. Crear la siguiente clase:

- **Nombre de la clase:** Producto
- **Atributos:** nombre, precio y stock.
- **Métodos:** vender(cantidad)

El método vender debe restar del stock una cantidad recibida, validar que el stock no sea menor que cero.

**5.4** Simular la interacción entre una persona y una cuenta bancaria. Crear las siguientes clases:

- **Nombre de la clase:** CuentaBancaria
- **Atributos:** saldo
- **Métodos:** ver\_saldo() y depositar(monto)

El método ver\_saldo() muestra el saldo actual por pantalla.

El método depositar(monto) almacena el monto recibido en el saldo, validar que el monto recibido no sea negativo. Mostrar el mensaje “deposito correcto” o “error en el monto ingresado” dependiendo del caso.

- **Nombre de clase:** Persona
- **Atributos:** nombre, cuenta (cuenta es una instancia de CuentaBancaria)
- **Métodos:** depositar\_dinero(monto) y consultar\_saldo()

El método depositar\_dinero(monto) debe enviar un mensaje al método depositar(monto) con el monto recibido.

El método consultar\_saldo() debe enviar un mensaje al método ver\_saldo() de la cuenta bancaria.

**5.5** Simular como interactúa un usuario para tomar un libro prestado de una biblioteca.

Crear las siguientes clases:

- **Nombre de la clase:** Libro
- **Atributos:** titulo, autor y disponible (Booleano).
- **Métodos:** mostrar\_info()

El método mostrar\_info() muestra información sobre los atributos del libro.

- **Nombre de la clase:** Usuario
- **Atributos:** nombre
- **Métodos:** tomar\_libro(libro)

El método tomar\_libro(libro) debe mostrar por pantalla el nombre del usuario y el título del libro que tomo. Validar que el libro este disponible caso contrario mostrar un mensaje indicando que el libro no está disponible.

- **Nombre de la clase:** Biblioteca
- **Atributos:** libros (es una lista de objetos de tipo libro)
- **Métodos:** agregar\_libro(libro) y mostrar\_libros()

El método agregar\_libro(libro) debe agregar el libro recibido a la lista de libros y mostrar por pantalla el título del libro agregado.

El método mostrar\_libros() debe mostrar por pantalla el título y si esta disponible cada uno de los libros de la lista de libros.

### Encapsulamiento

**5.6** Simular la creación de un usuario para un sistema y proteger el cambio de contraseña (atributo oculto).

- **Nombre de la clase:** Usuario
- **Atributos:** nombre y contraseña (String y oculto)
- **Métodos:** verificar\_contraseña(contraseña), cambiar\_contraseña(actual, nueva)

El método verificar\_contraseña(contraseña) valida si la contraseña ingresada es la misma que la del usuario, retorna True si es correcta y False si es incorrecta.

El método cambiar\_contraseña(actual, nueva) valida si la contraseña actual coincide con la del usuario, si coinciden se actualiza por la nueva.

**5.7** Agregar a la clase Usuario del ejercicio anterior los atributos email.

- Crear los métodos getters y setters para el email
  - **mostrar\_mail()** : muestra el mail del usuario
  - **modificar\_mail(mail)**: modificar el mail del usuario, validar que el nuevo mail sea un correo valido (debe contener el carácter "@")