

PRÁCTICA 0

Requisitos previos

- Tener instalado [Eclipse](#) para desarrollo en Java.
- Puede resultarte de ayuda [Umbrello](#).
- Tener cuenta de git**b en algún remoto.
- Crea un proyecto en eclipse e integra dentro de paquete diferente cada ejercicio.

Entregables:

- Un zip con la carpeta src con tu código
- Un enlace a tu cuenta de git**b

Fecha de Entrega:

La práctica deberá entregarse antes del 16/09 a las 20:30.

Ejercicios

Ejercicio 0: Sistema de Gestión de Biblioteca

El sistema debe permitir la gestión de libros, usuarios, y Préstamos de libros. Aquí están los detalles y requisitos para el sistema:

1. Clase **Libro**:

○ **Atributos:**

- **isbn** (String): Identificador único del libro.
- **titulo** (String): Título del libro.
- **autor** (String): Autor del libro.
- **anioPublicacion** (int): Año de publicación del libro.
- **disponible** (boolean): Indica si el libro está disponible para préstamo.

○ **Métodos:**

- **Préstamo()**: Marca el libro como no disponible.
- **devolucion()**: Marca el libro como disponible.
- **informacion()**: Devuelve una descripción del libro.

2. Clase **Usuario**:

- **Atributos:**
 - **idUsuario** (String): Identificador único del usuario.
 - **nombre** (String): Nombre del usuario.
 - **telefono** (String): Número de teléfono del usuario.
 - **email** (String): Correo electrónico del usuario.
- **Métodos:**
 - **registrar()**: Registra un nuevo usuario.
 - **actualizarInfo()**: Actualiza la información del usuario.
 - **consultarPréstamos()**: Devuelve una lista de libros prestados por el usuario.

3. Clase **Préstamo**:

- **Atributos:**
 - **idPréstamo** (String): Identificador único del Préstamo.
 - **fechaPréstamo** (Date): Fecha en que se realizó el Préstamo.
 - **fechaDevolucion** (Date): Fecha en que se devolvió el libro (puede ser nula si no ha sido devuelto).
- **Métodos:**
 - **realizarPréstamo()**: Asocia un libro con un usuario y establece la fecha de Préstamo.
 - **finalizarPréstamo()**: Marca el Préstamo como finalizado y establece la fecha de devolución.

4. Clases **Biblioteca**: Con el main para poder hacer pruebas. Tienes 2 mains adjuntos para comprobar que tu programa es correcto, por lo que deberás crear 2 Bibliotecas, **Biblioteca1** y **Biblioteca2**

5. Relaciones:

- Un **Libro** puede estar asociado con varios **Préstamos** (un libro puede ser prestado múltiples veces).
- Un **Usuario** puede tener varios **Préstamos** (un usuario puede prestar varios libros).
- Un **Préstamo** está asociado con un **Libro** y un **Usuario**.

Instrucciones:

1. Crea un proyecto en Eclipse para programar este sistema de gestión de bibliotecas. Crea 2 clases

Ejercicio 1: Herencia y Polimorfismo

- **Ejercicio:** Crea una clase base llamada `Animal` con un método `hacerSonido()`. Luego, define dos clases derivadas: `Perro` y `Gato`, cada una con su propia implementación del método `hacerSonido()`. Demuestra el polimorfismo llamando al método `hacerSonido()` desde una referencia de tipo `Animal`.

El sonido que hace un animal consiste en imprimir por pantalla su onomatopeya.

- **Conceptos Clave:** Herencia, polimorfismo, métodos sobrescritos.
-

Ejercicio 2: Interfaz y Clases Implementadoras

- **Ejercicio:** Define una interfaz `Imprimible` con un método `imprimir()`. Luego, crea dos clases: `Documento` y `Reporte`, que implementen esta interfaz. Cada clase debe proporcionar su propia implementación del método `imprimir()`. Escribe un programa principal para demostrar cómo se pueden usar estas clases.
- **Conceptos Clave:** Interfaces, implementación de interfaces, clases que implementan interfaces.