Diagramas UML

(Unified Modeling Language)

Manuel Liébana Rosas

Índice

- Dominio del problema
- Diagrama de componentes
  - Casos de uso
  - Narrativa Casos de uso
  - Diagrama de estados
  - Diagrama de secuencia
- Diagrama de estructura
  - Diagrama de clases
- Código

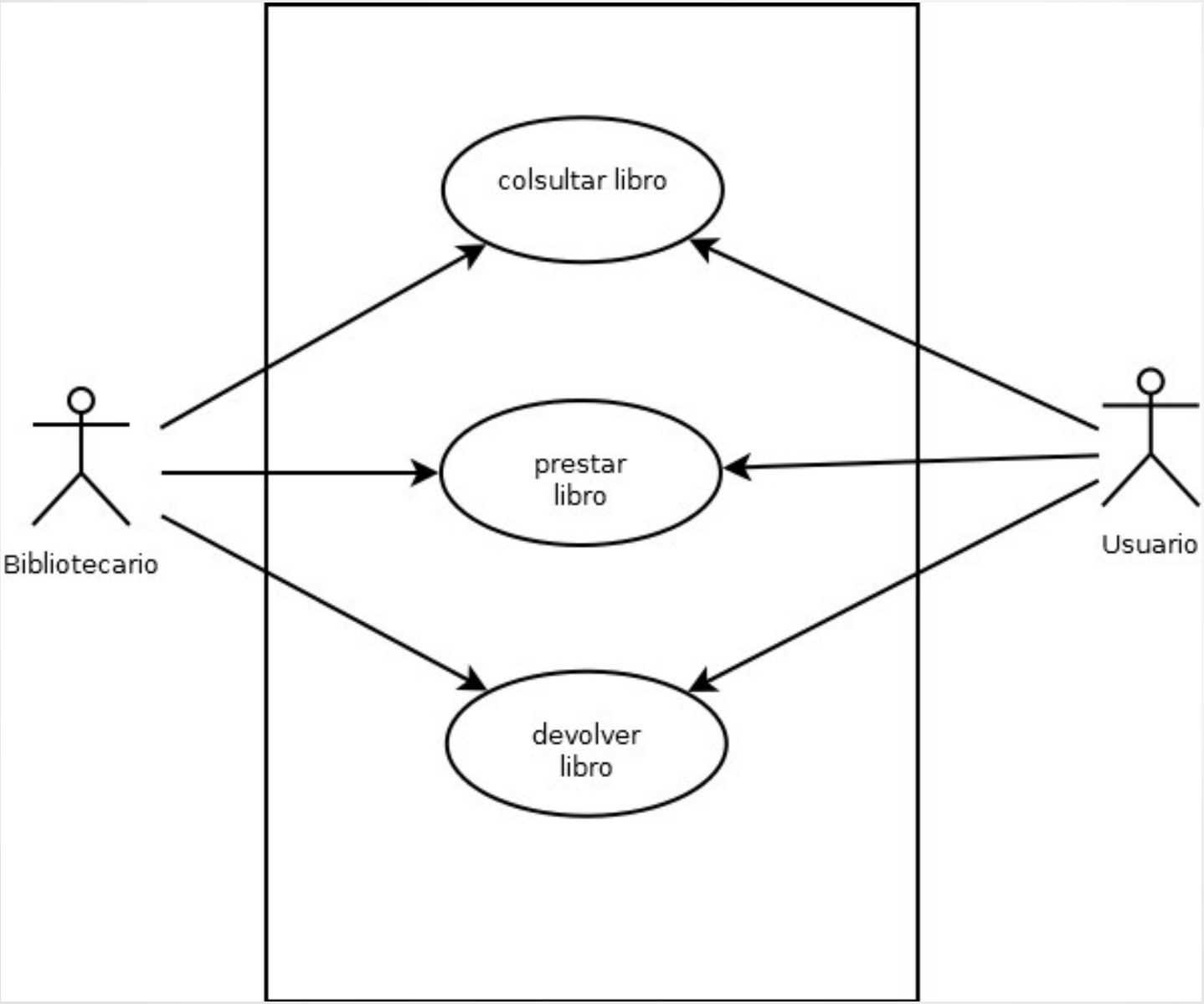
Dominio del problema

- Proyecto: Biblioteca
- Se va a desarrollar una aplicación para la gestión del préstamo y devolución de libros en una biblioteca
- El funcionamiento general será:
  - El usuario elige un libro y se lo indica al bibliotecario
  - El bibliotecario comprueba que el usuario pueda sacar el libro y que el libro esté disponible
  - Una vez que todo es correcto se realiza el préstamo

Diagrama de componentes
Casos de uso

- Un modelo de casos de uso captura el comportamiento observable de un sistema o de una clase
- El objetivo del diagrama de casos de uso es describir los requerimientos y la interacción entre el sistema con el exterior
- Los usuarios del sistema se denominan **actores** y las funciones y responsabilidades se conocen con el nombre de **casos de uso**.
- En nuestro caso, los actores son el bibliotecario y el usuario que va a sacar o devolver un libro.
- Y tenemos 3 casos de uso: consultar libro, prestar libro y devolver libro.

Casos de uso


colsultar libro
prestar libro
devolver libro
Bibliotecario
Usuario

Narrativa Casos de uso

- **Nombre**: Prestar libro
- **Precondiciones**:

  El usuario debe haber seleccionado el libro a sacar
- **Postcondiciones**:

  El sistema registrará al usuario que se lleva el libro
- **Escenario principal**:
  - 1. El bibliotecario introduce en el sistema el usuario al que se le presta el libro
  - 2. El sistema comprueba que el usuario no tenga ninguna sanción
  - 3. El usuario indica el libro elegido al bibliotecario
  - 4. El bibliotecario introduce en el sistema el libro a prestar
  - 5. El sistema comprueba que el libro está disponible
  - 6. El sistema registra el libro como prestado y devuelve la fecha de devolución
- **Escenarios alternativos**:
  - 2.a. El sistema detecta que el usuario tiene una sanción vigente
    - 1. El sistema le comunica al bibliotecario que el usuario tiene una sanción
    - 2. Vuelve al paso 1
  - 5.a. El sistema detecta que el libro no esta disponible
    - 1. El sistema le comunica al bibliotecario el libro no está disponible
    - 2. Vuelve al paso 3

Diagrama de estados

- Un diagrama de estado muestra la secuencia de estados que un objeto o una interacción pueden atravesar durante su existencia en respuesta a los estímulos que vayan recibiendo, junto con las correspondientes respuestas y acciones
- Cada objeto está en un estado en cierto instante
- Son deterministas (El comportamiento depende del estado actual)

Diagrama de estados

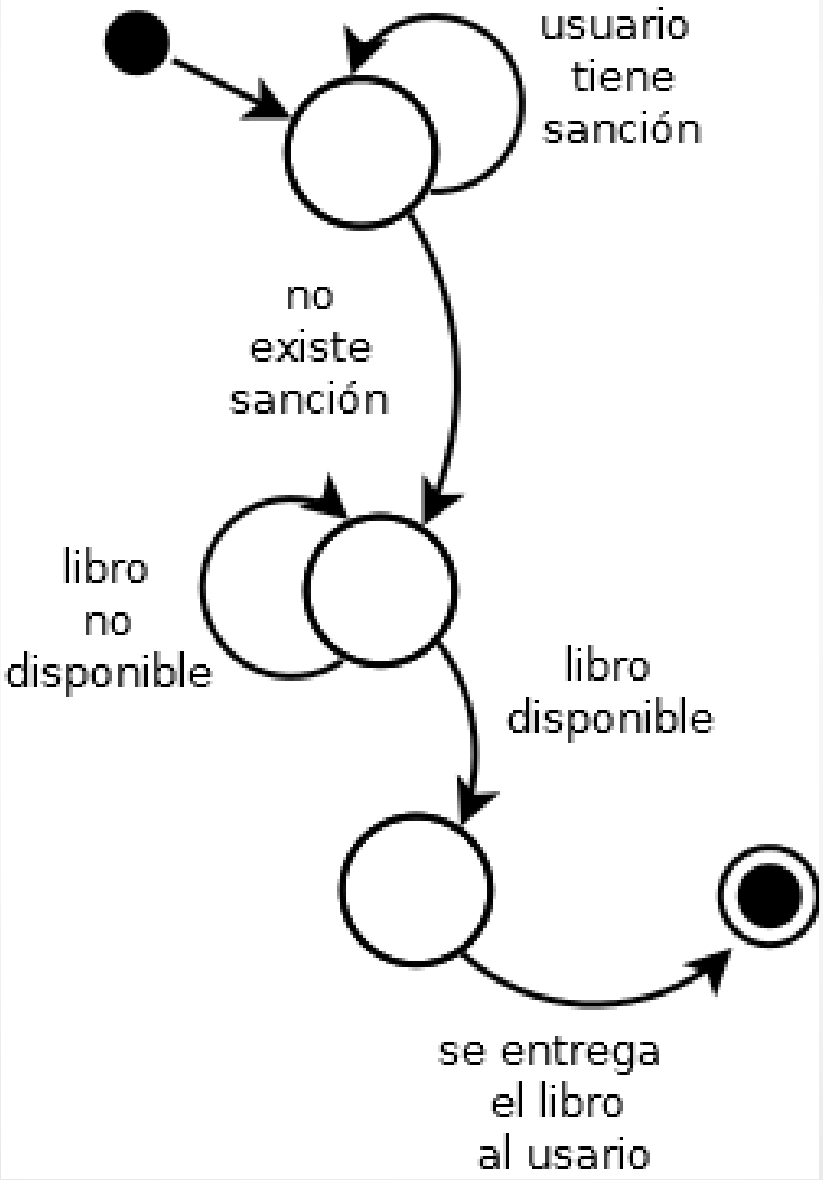


Diagrama de secuencia

- Los diagramas de secuencia muestran la interacción (los mensajes que se intercambian entre objetos) ordenada en una secuencia de tiempos
- El diagrama de secuencia enfatiza el ordenamiento temporal de los mensajes en una interacción
- Debe hacer un diagrama de secuencia por cada caso de uso

Diagrama de secuencia

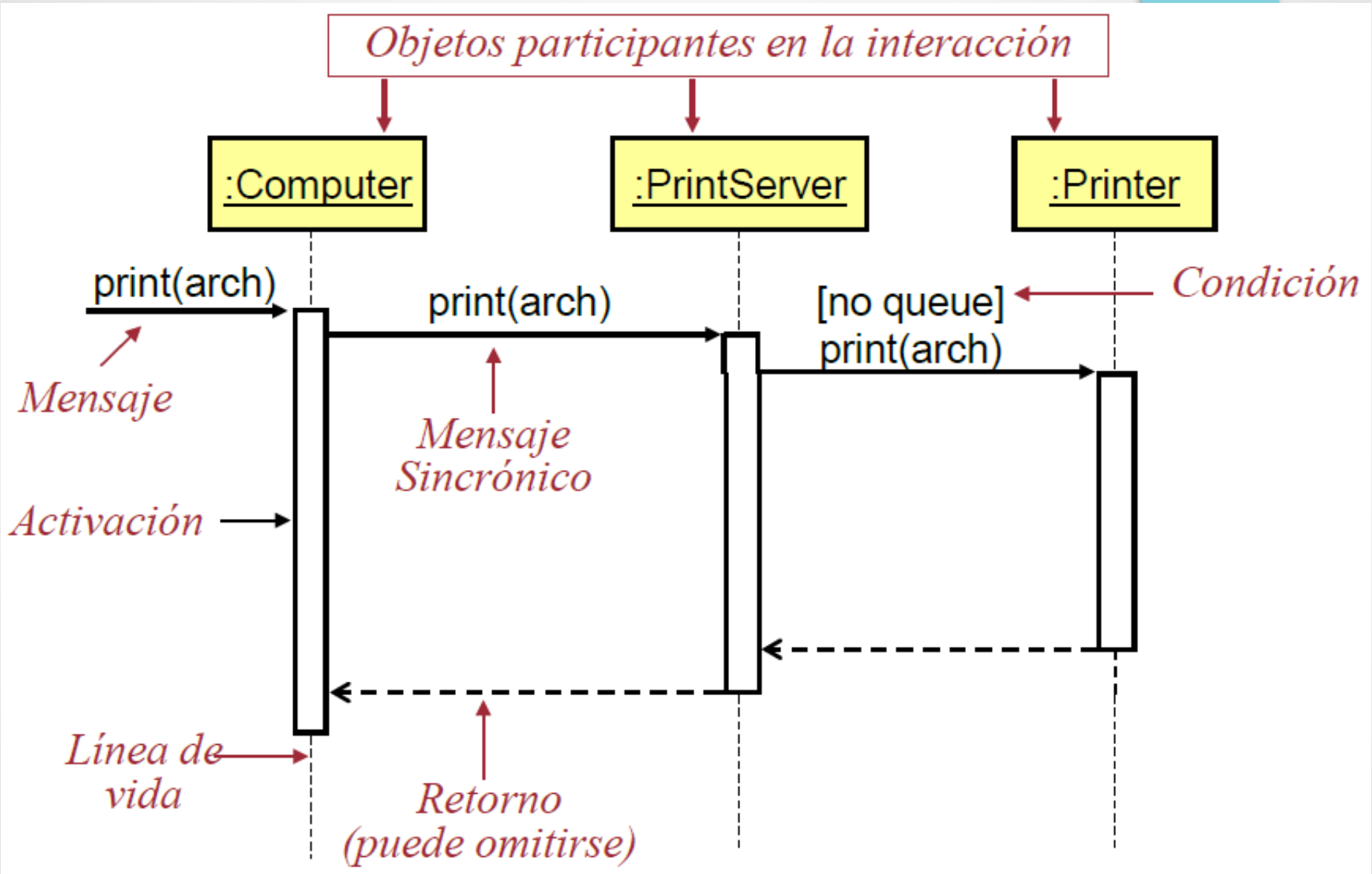


Diagrama de secuencia

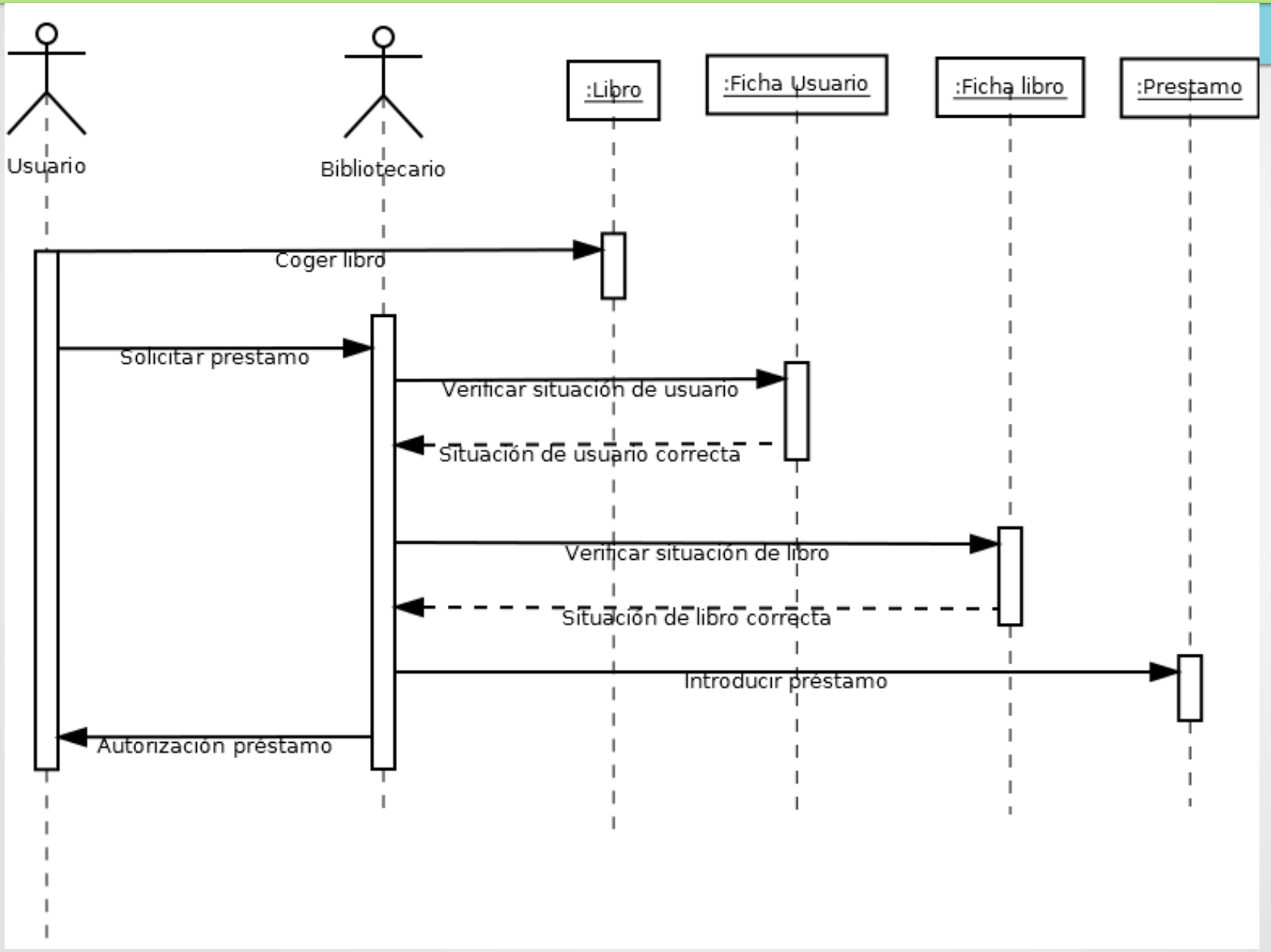


Diagrama de estructura
Diagrama de clases

- Un diagrama de clases de diseño muestra la especificación para las clases software de una aplicación
- Son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones
- Deben incluir:
  - Clases, asociaciones y atributos
  - Interfaces, con sus operaciones y constantes
  - Métodos
  - Navegabilidad
  - Dependencias

Diagrama de clases

- Pasos para la creación del Diagrama de Clases de Diseño:
  - Identificar todas las clases participantes en la solución software. Esto se lleva a cabo analizando los Diagramas de secuencia
  - Representarlas en un diagrama de clases
  - Duplicar los atributos que aparezcan en los conceptos asociados del Modelo Conceptual
  - Añadir los métodos, según aparecen en los Diagramas de secuencia
  - Añadir información de tipo a los atributos y métodos
  - Añadir las asociaciones
  - Añadir flechas de navegabilidad
  - Añadir relaciones de dependencia

Diagrama de clases

- Identificar y representar las clases de diseño
- Añadir atributos

Diagrama de clases

- Añadir los métodos, según aparecen en los Diagramas de secuencia
- Añadir información de tipo a los atributos y métodos

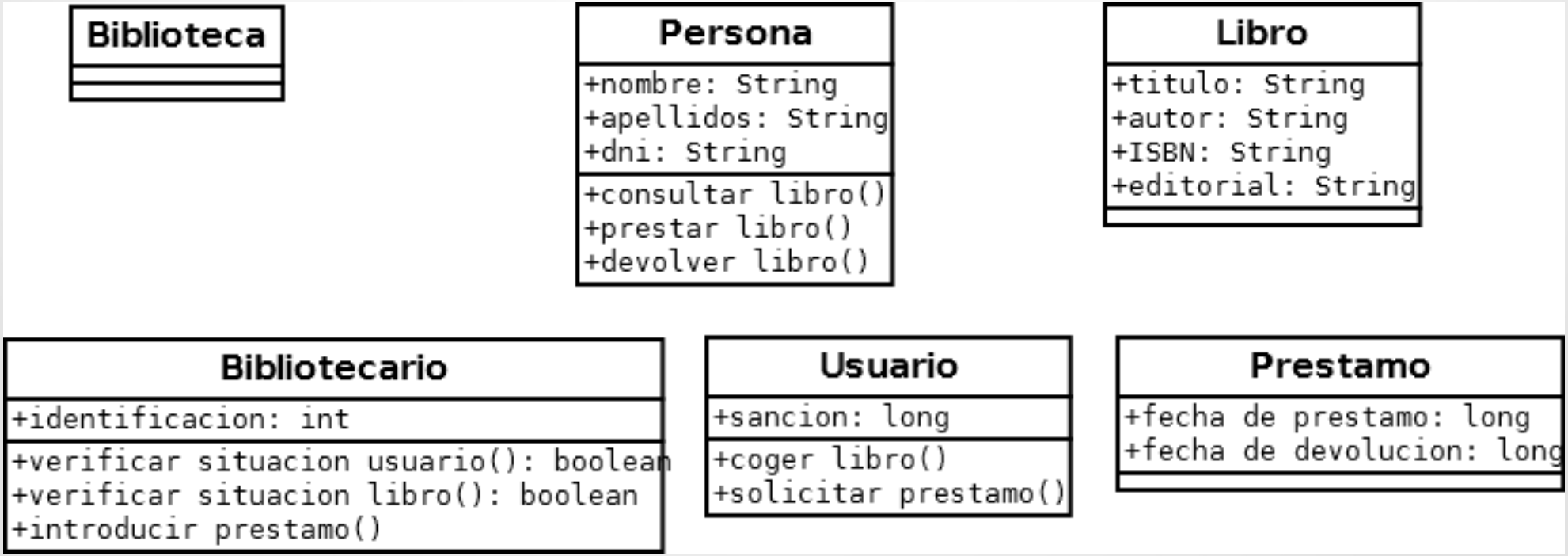


Diagrama de clases

- Añadir las asociaciones necesarias
- Añadir flechas de navegabilidad
- Añadir dependencias

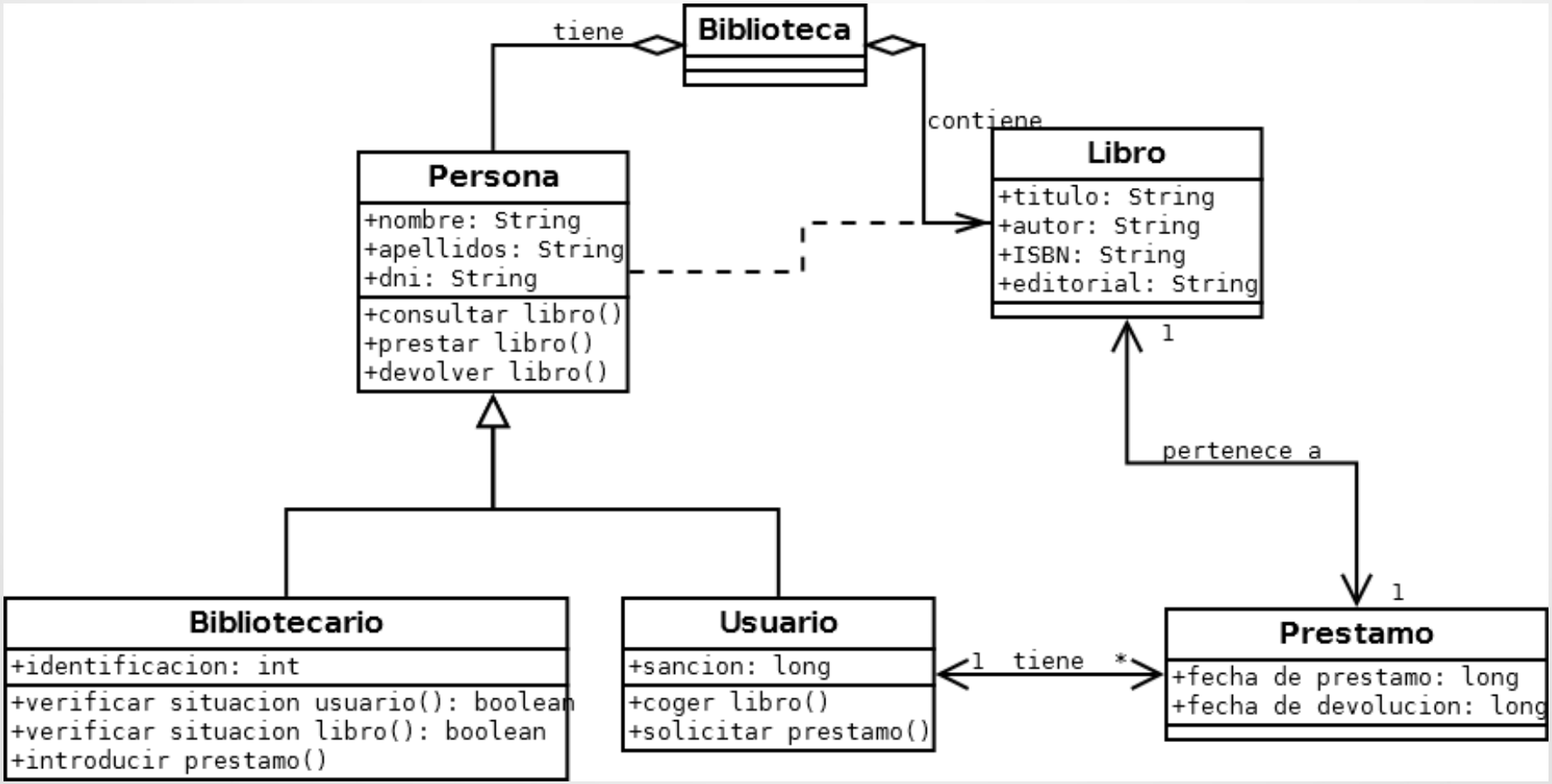
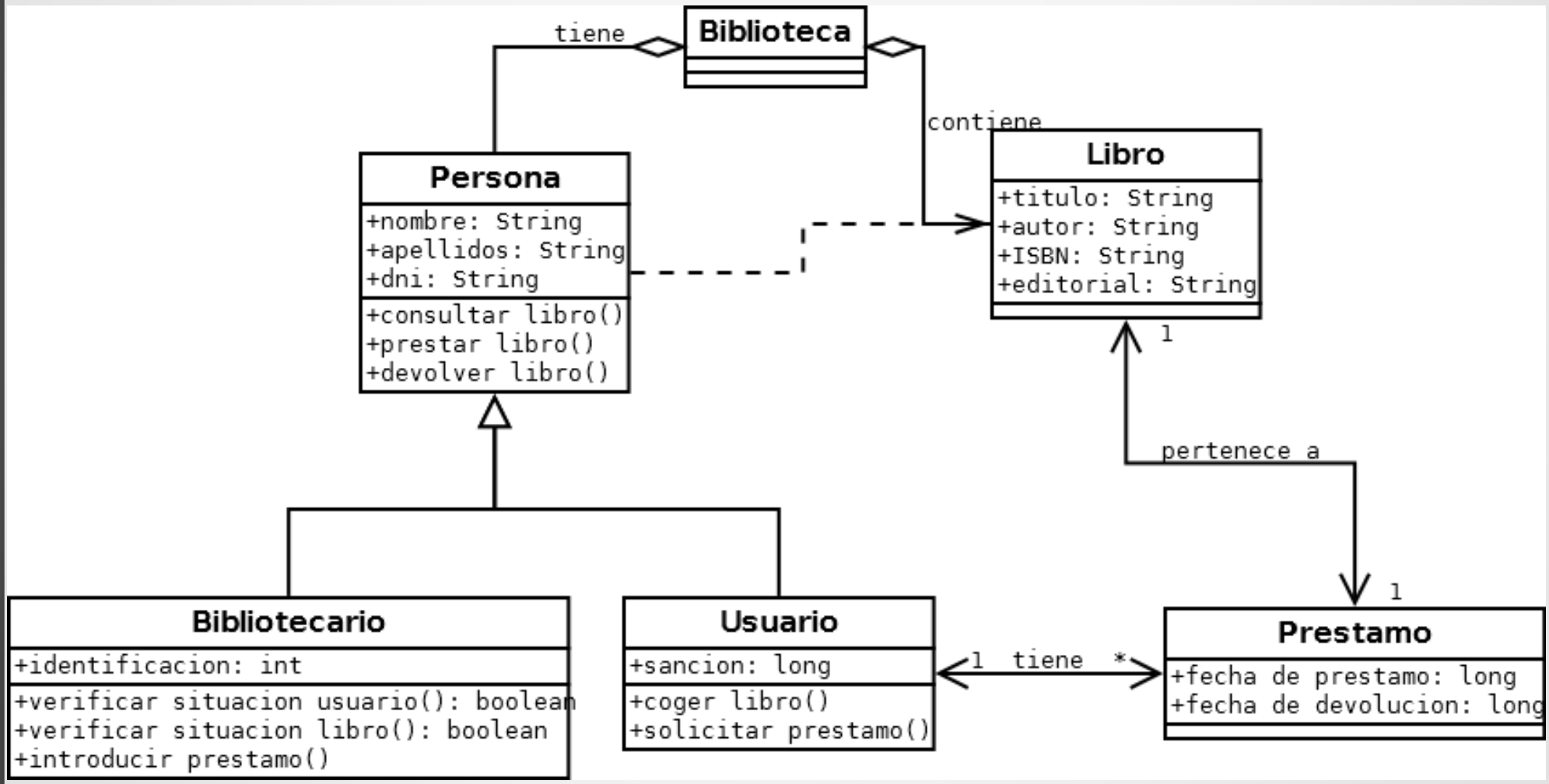


Diagrama de clases



Código

```java
public abstract class Persona {

    protected String nombre;
    protected String apellidos;
    protected String DNI;

    public Persona() {
        nombre = "";
        apellidos = "";
        DNI = "";
    }
    public Persona(String nombre,
            String apellidos, String DNI){
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.DNI = DNI;
    }
    public abstract String prestarLibro();

    public String getNombre() {
        return nombre;
    }
    public String getApellidos() {
        return apellidos;
    }
```

```java
public class Bibliotecario extends Persona{

    int identificacion;

    public Bibliotecario(int identificacion,
            String nombre, String apellidos, String DNI) {
        super(nombre, apellidos, DNI);
        this.identificacion = identificacion;
    }

    @Override
    public String prestarLibro() {
        return "El biliotecario presta el libro";
    }
```

```java
public class Usuario extends Persona{
    long sancion;
    LinkedList<Prestamo> prestamos;

    public Usuario(long sancion, String nombre,
            String apellidos, String DNI) {
        super(nombre, apellidos, DNI);
        this.sancion = sancion;
        prestamos = new LinkedList<>();
    }
    @Override
    public String prestarLibro() {
        return "El usuario tiene un libro prestado";
    }
```

Código

```
public static void main(String[] args) {
    ArrayList<Persona> lista = new ArrayList<>();

    Bibliotecario b = new Bibliotecario(0, "Paco", "Molina Martínez", "12345678a");
    Usuario c = new Usuario(0, "Pepe", "Ortega Jimenez", "87654321a");

    lista.add(b);
    lista.add(c);

    for(int i = 0;i<lista.size();i++){
        System.out.println(
                "Nombre: "+lista.get(i).getNombre()+
                "\nApellidos: "+lista.get(i).getApellidos()+
                "\nDNI: "+lista.get(i).getDNI());
        System.out.println(lista.get(i).prestarLibro());
    }
}
```

```
Nombre: Paco
Apellidos: Molina Martínez
DNI: 12345678a
El biliotecario presta el libro
Nombre: Pepe
Apellidos: Ortega Jimenez
DNI: 87654321a
El cliente tiene un libro prestado
```