

# **Trabajo Final de Programación** **Orientada a Objetos**

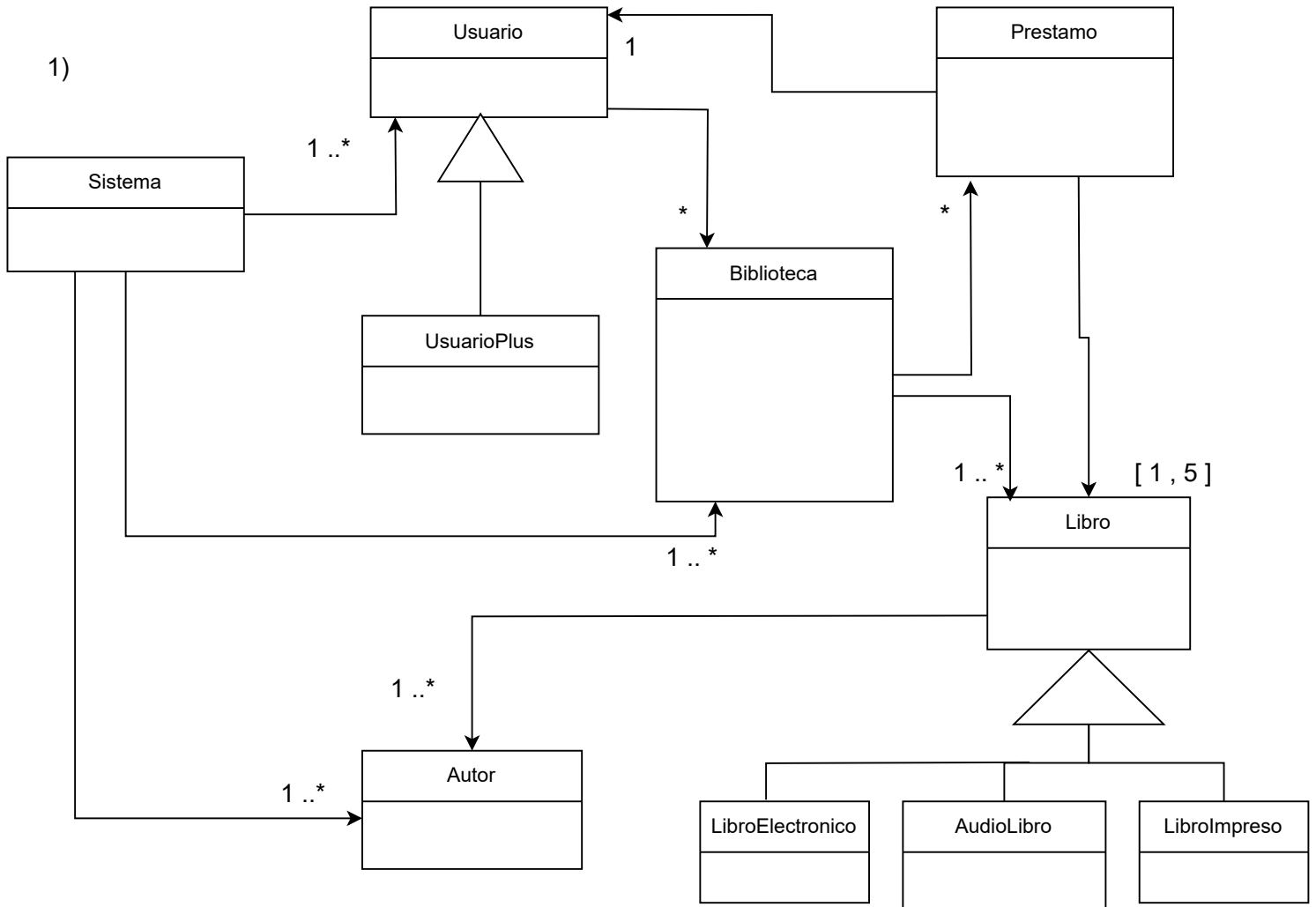
**Alumno: Leiva Julián Agustín**

**Legajo: FAI - 2318**

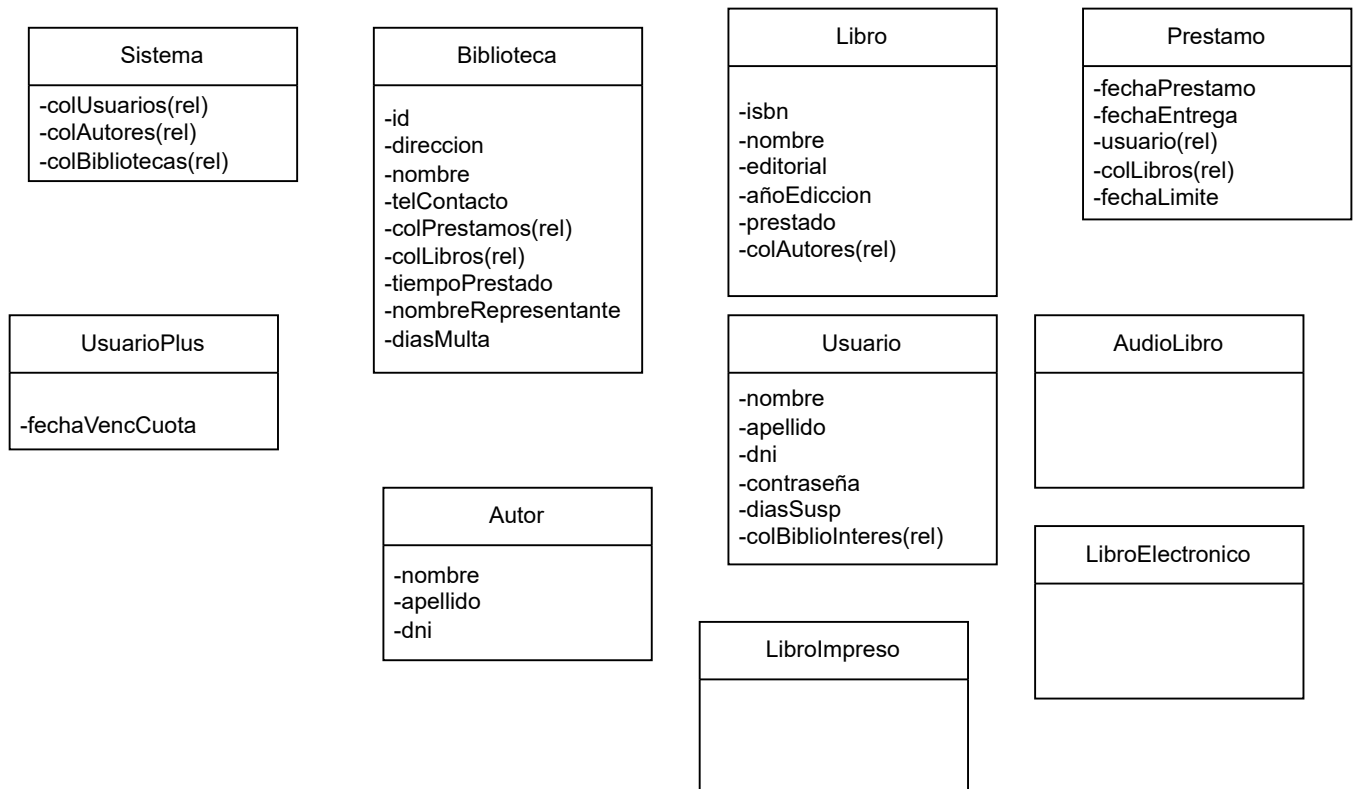
**Año de Cursada: 2022**

## Diagrama de clases

1)



## Atributos de las clases



## Métodos de las clases:

### Sistema

+ generarListadoUsuarioRetrasados

### Biblioteca

+ registrarPrestamo: unUsuario yColLib: unaColLibros

+ verificarCondiciones: unUsuario

+ hayPrestamoActivo:unUsuario

+ buscarUsuarioRetrasados

+ devolverPrestamo: unUsuario

### Prestamo

+ inicializaPrestamo: tiempoPrest yUsuario: unUsuario yColLibros: colLibrosPrest

+ crearPrestConTiempo: tiempoPrest yUsuario: unUsuario yColLibros: colLibrosPrest

+ verificaUsuarioYPrestamotAct: unUsuario

+ verificaUsuarioAtrasado

+ getUsuario

+ devolverLibro: diasMulta

### Usuario

+ seEncuentraSuspendido

+ verificaFechaVencPlus

+ mismoDni: unUsuario

+ getDni

+ suspender: diasMulta

### UsuarioPlus

+ verificaFechaVencPlus

### Libro

+ libroPrestado

+ libroNoPrestado

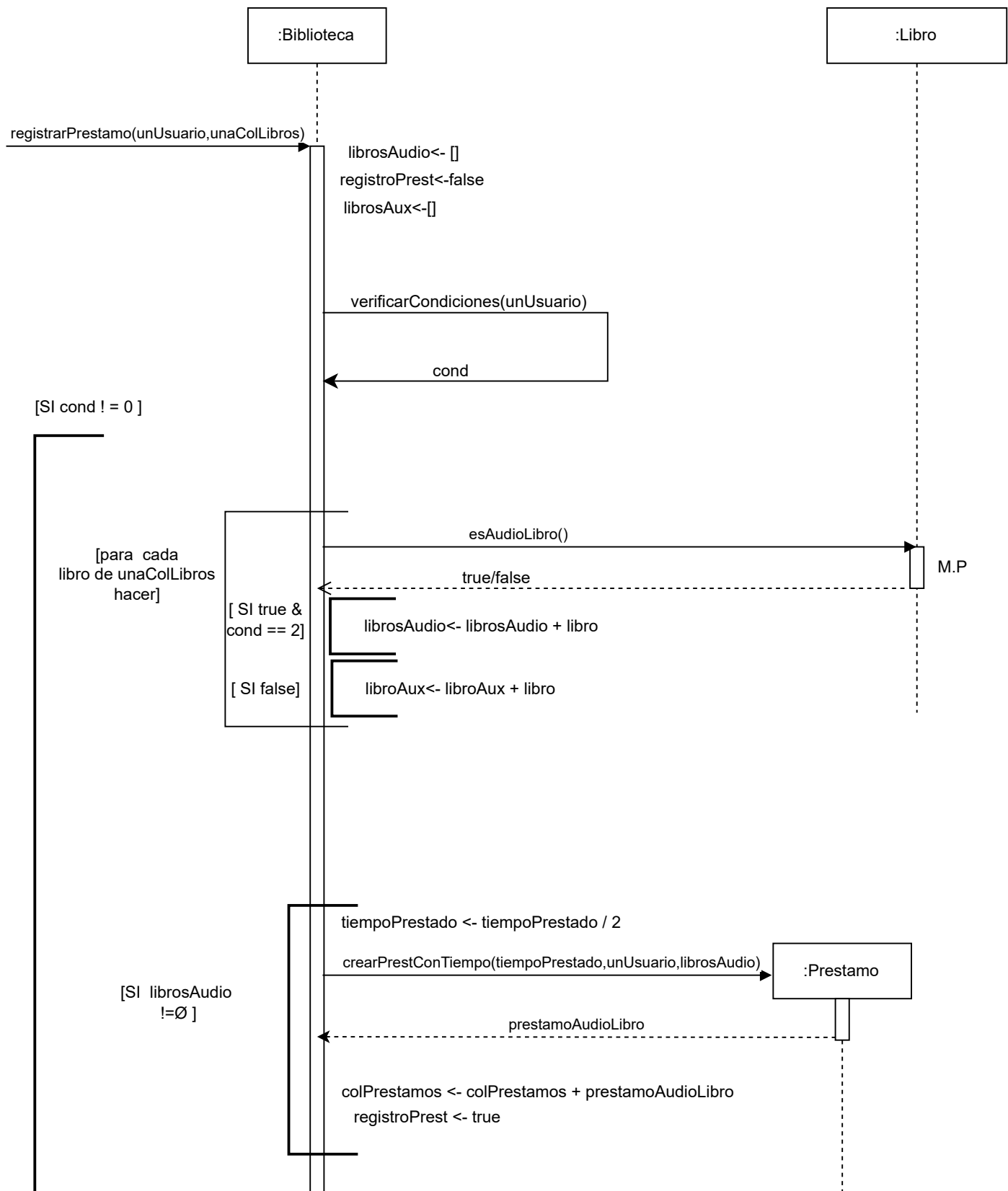
+ esAudioLibro

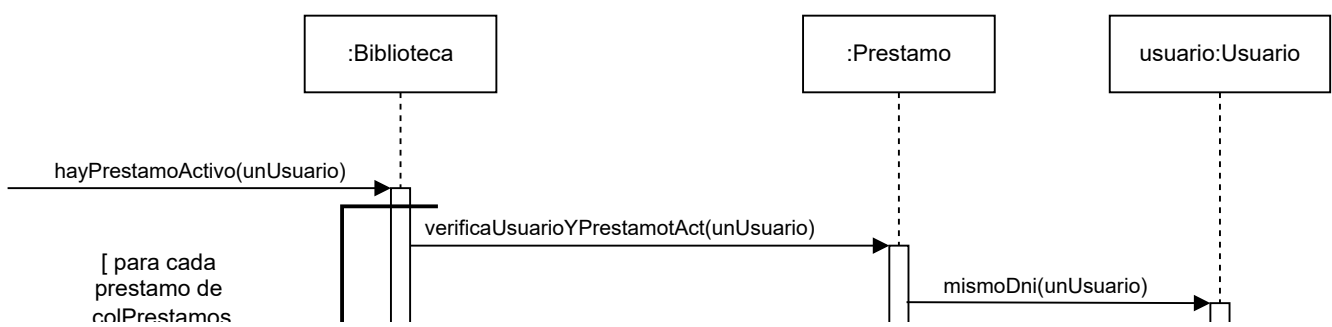
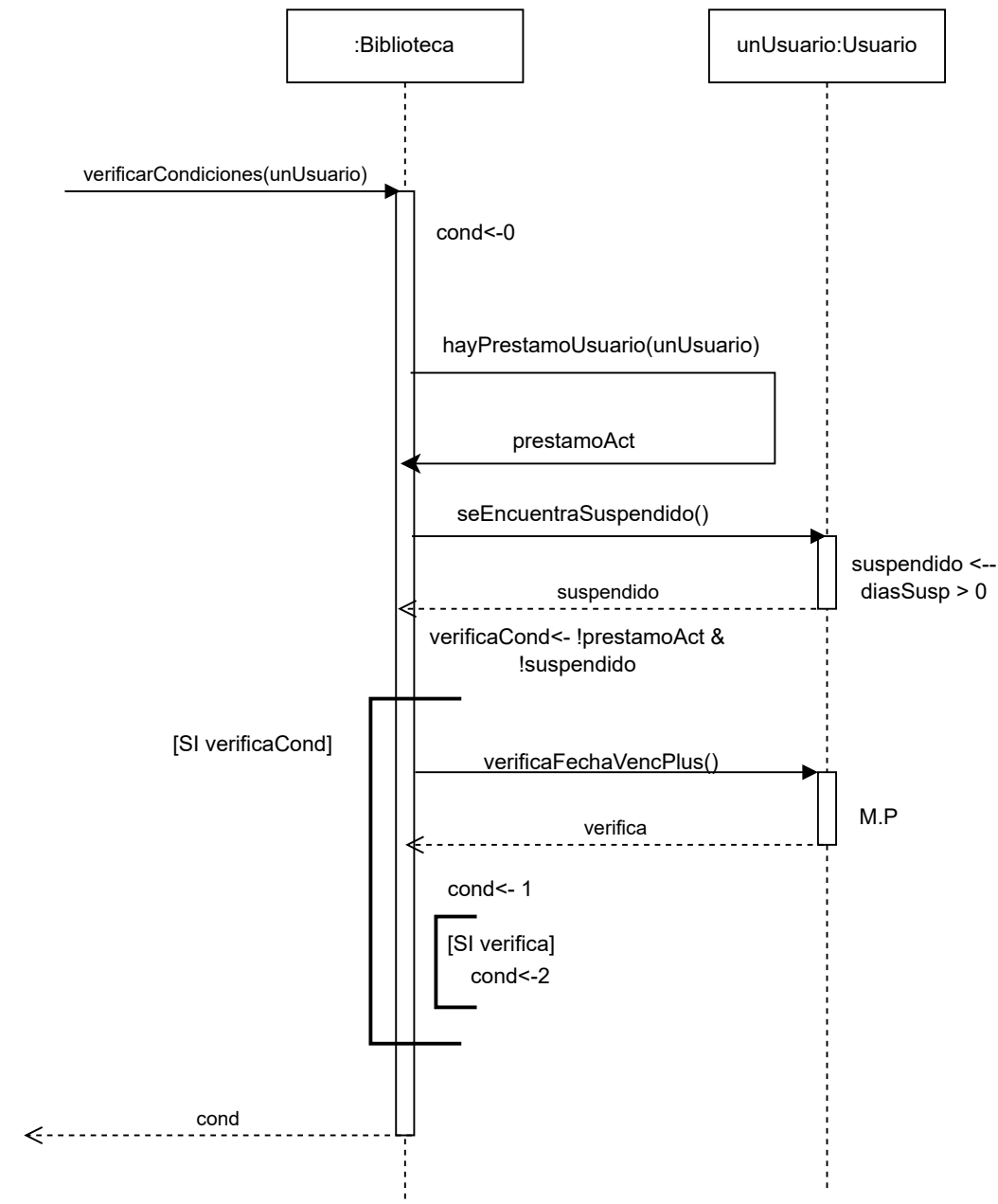
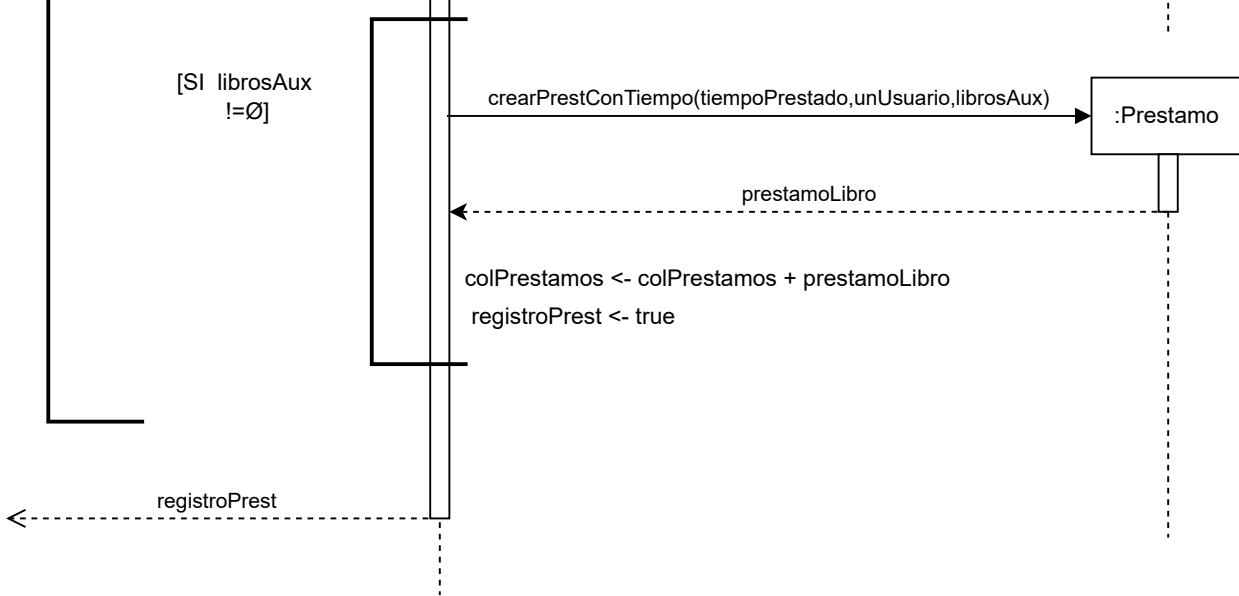
AudioLibro

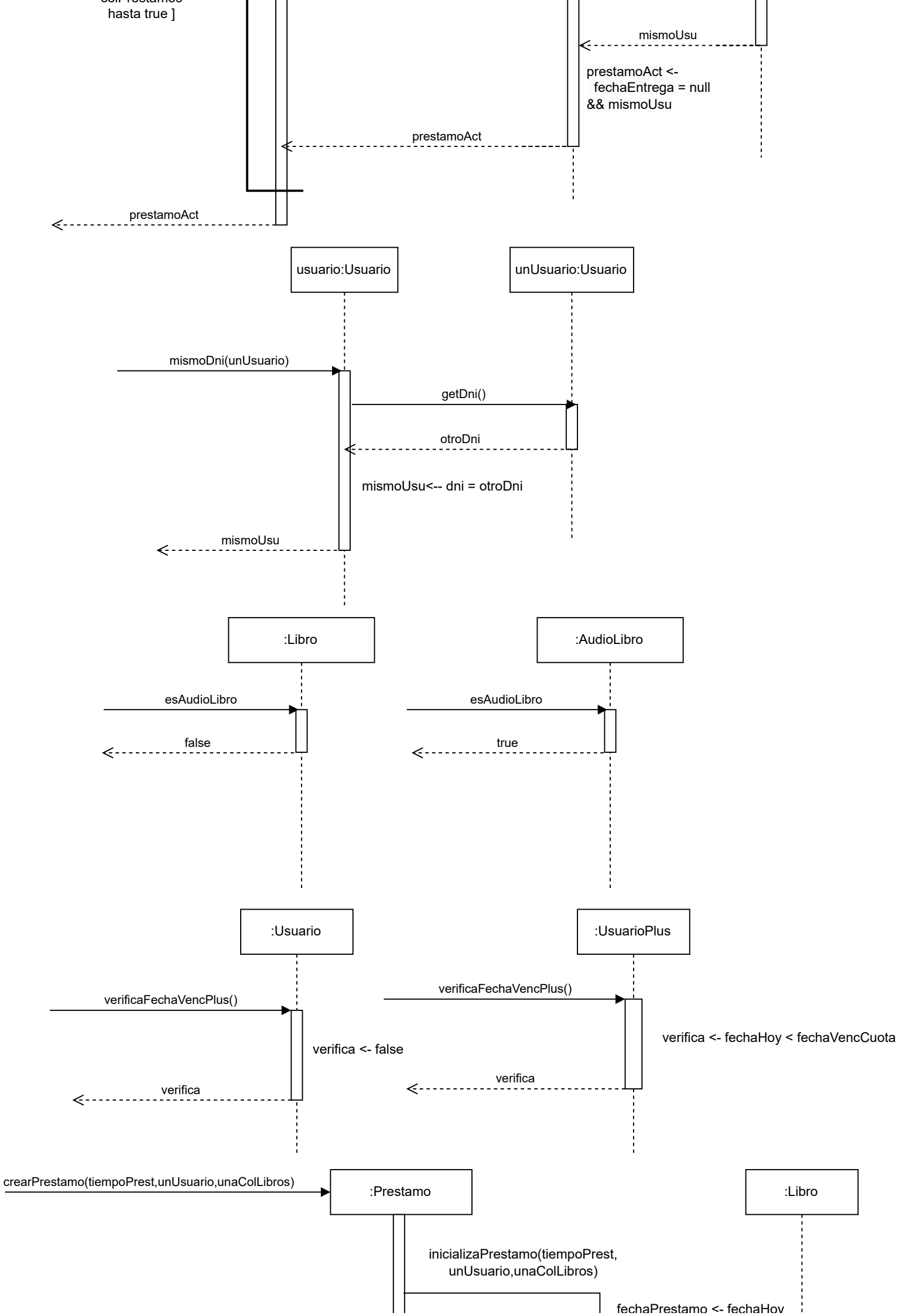
+ esAudioLibro

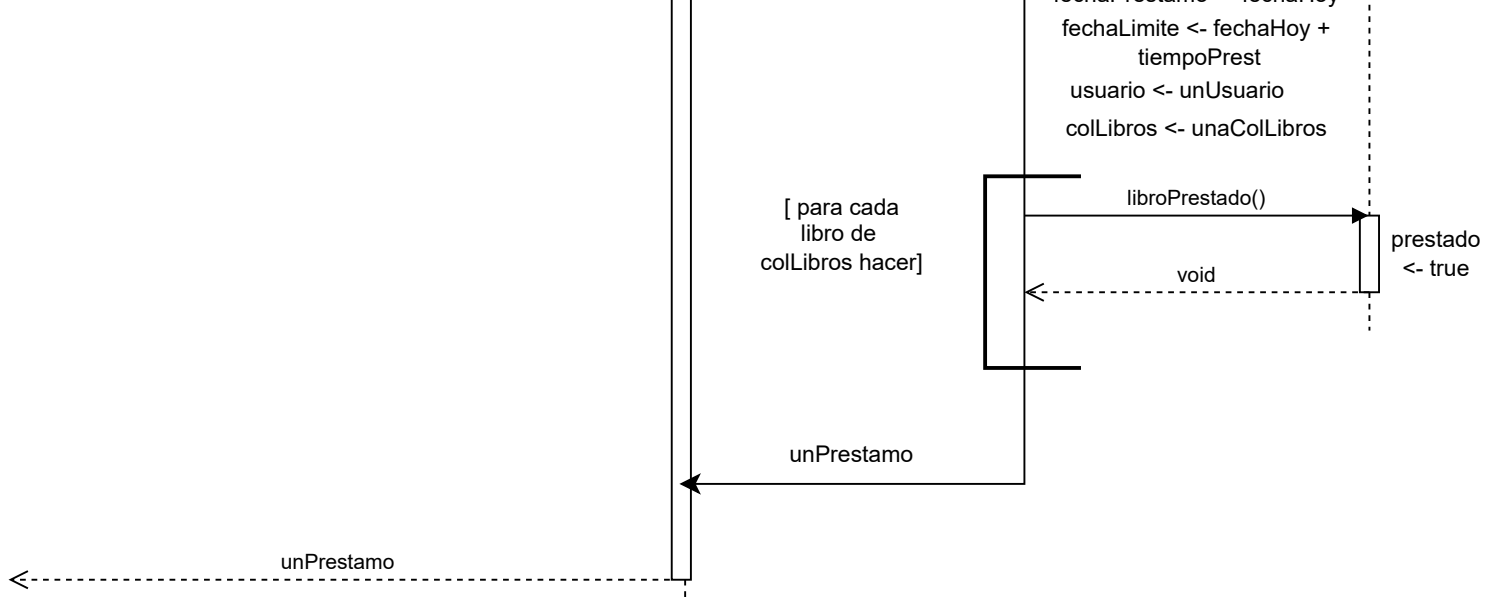
Diagrama de secuencias

2) -Registrar préstamo (unUsuario, unaColLibros) para una biblioteca. Tenga en cuenta que en unaColLibros puede haber libros de distinto tipo por lo que, si corresponde, tendrán que generarse 2 préstamos

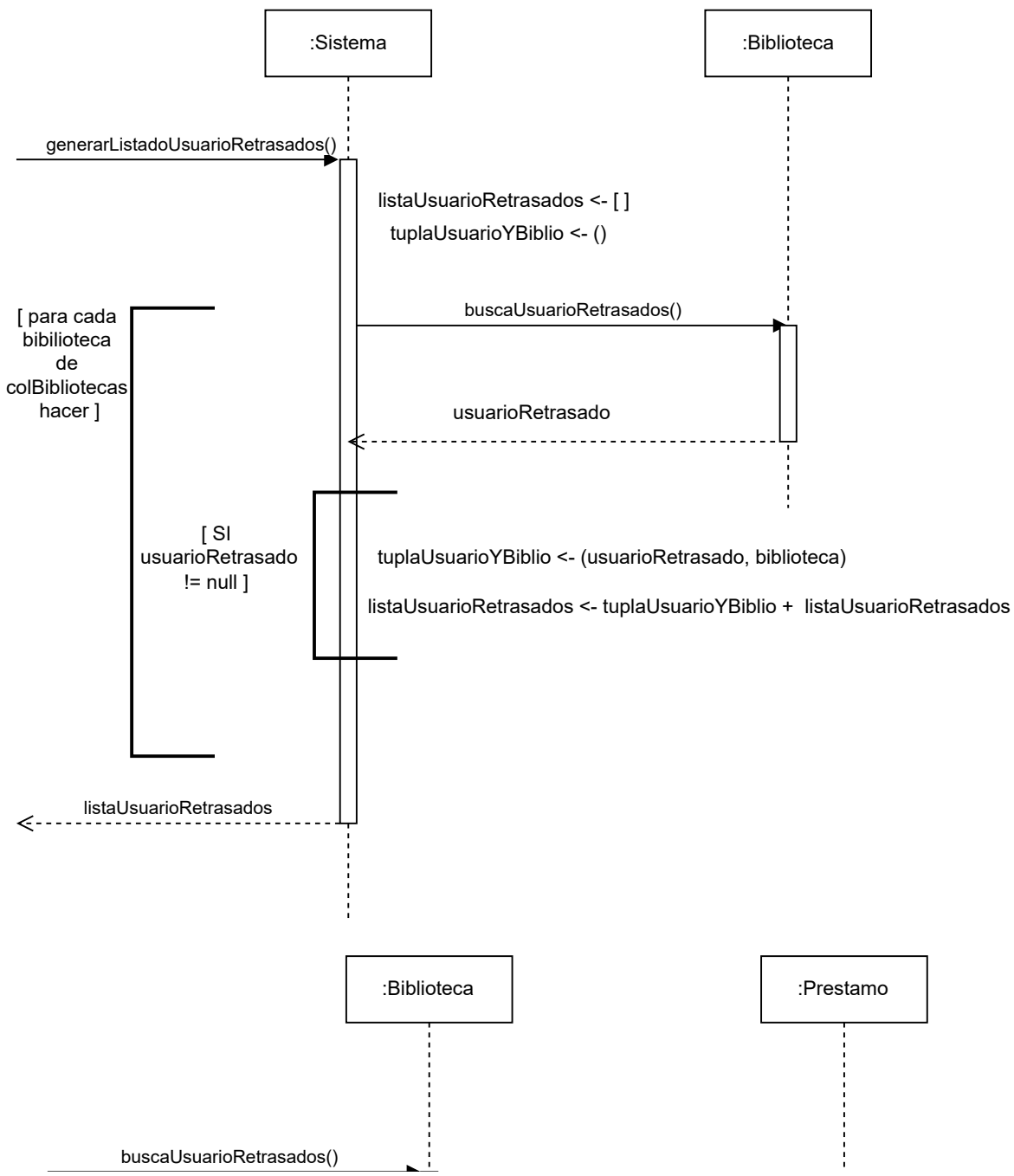






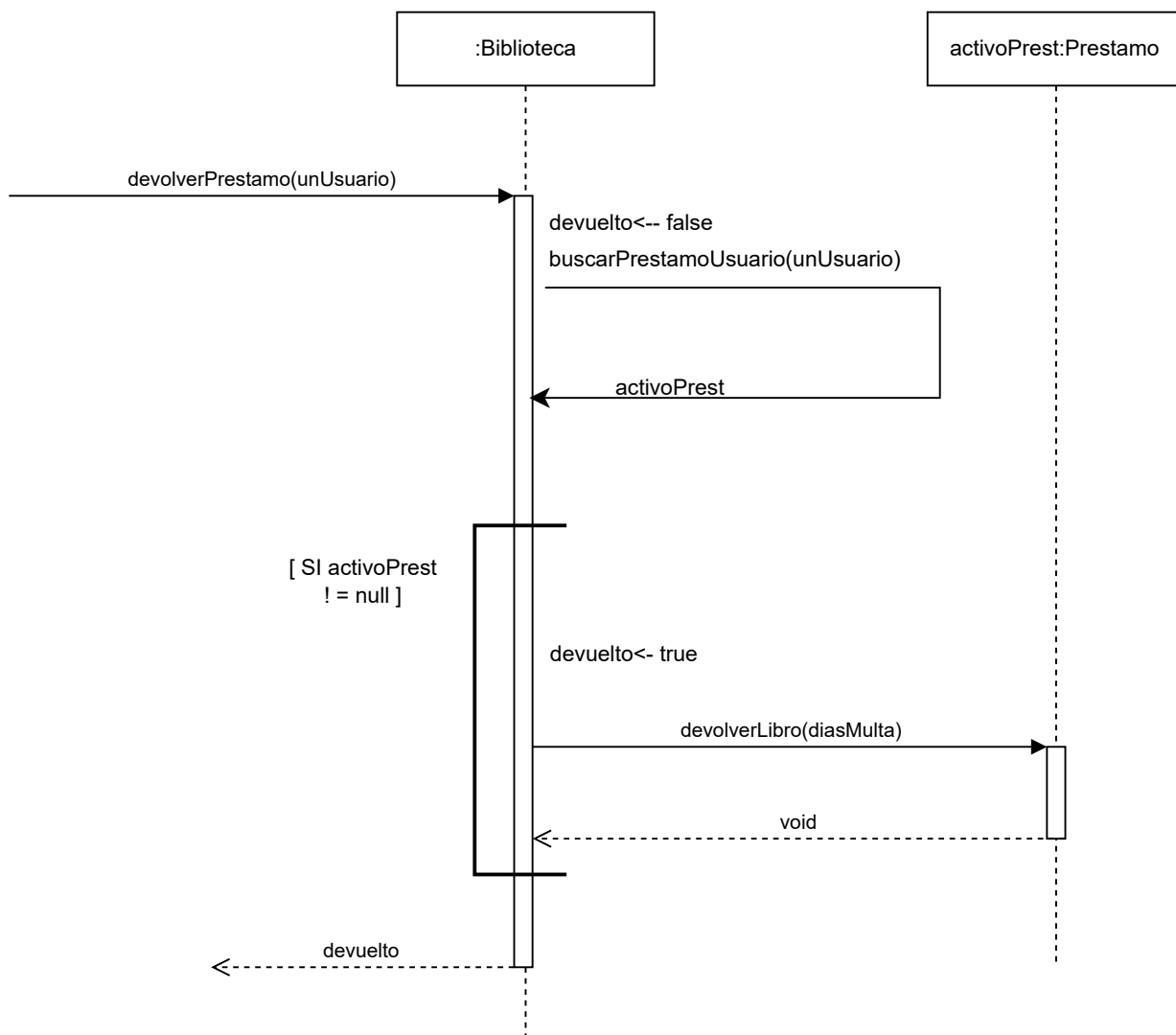


- **GenerarListadoUsuariosRetrasados ()** que devuelva una lista con los usuarios que están retrasados en la devolución de un préstamo, junto con la indicación de la biblioteca correspondiente



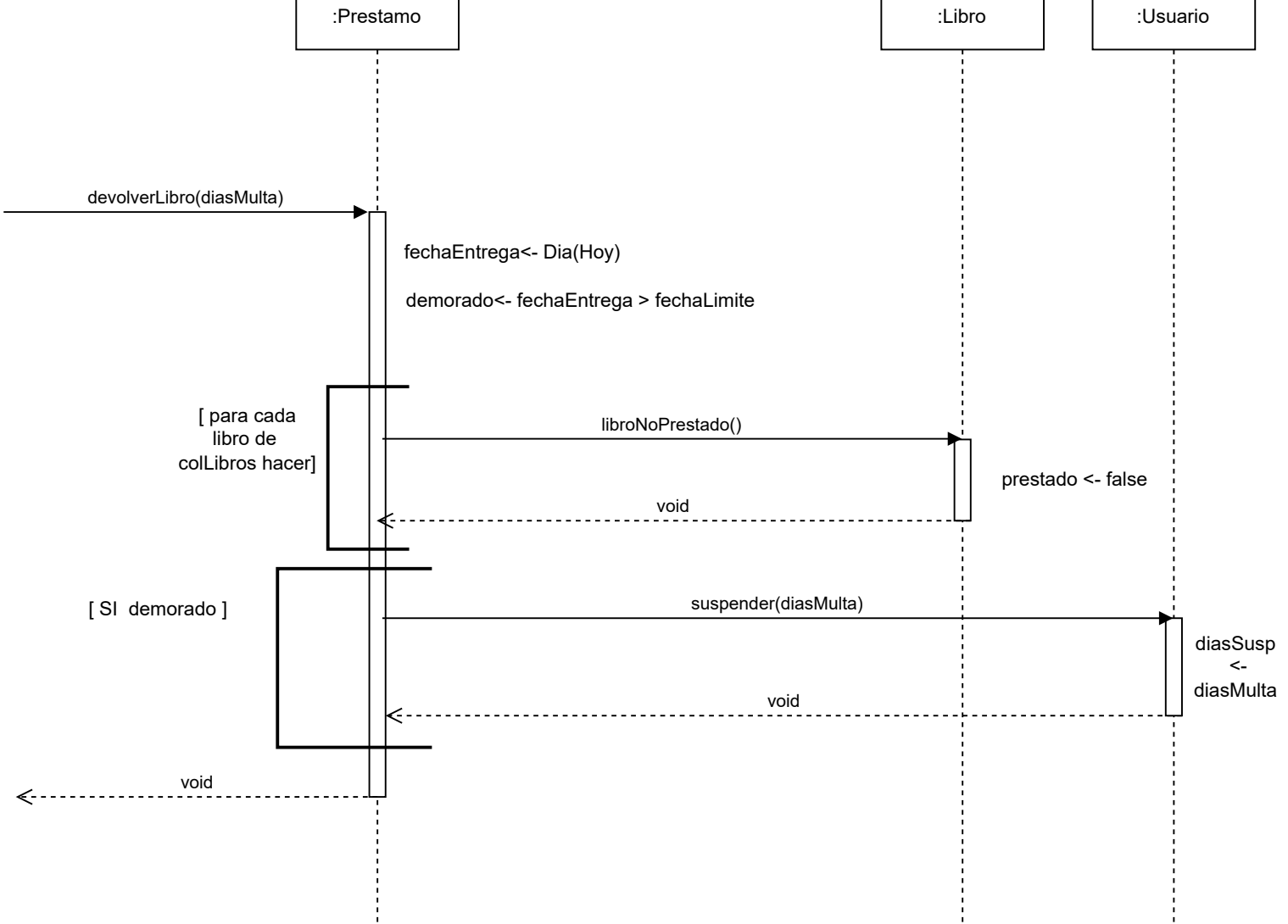


### - Devolver préstamo (unUsuario) para la biblioteca



El método buscarPrestamoUsuario(unUsuario) ya se encuentra definido





## MANUAL DE USUARIO:

Una vez descomprimido el archivo zip, abre la carpeta y copia la carpeta fotos y el archivo TrabajoFinalPoo.st a la carpeta de Pharo, esto es básicamente porque necesito la ruta actual en donde se va a ejecutar el programa con sus respectivas imágenes.

-----

### **Playground:**

```
| archivo |  
archivo := FileSystem workingDirectory  
/ 'TrabajoFinalPOO.st'.  
"Verifica si el archivo existe y luego lo ejecuta"  
archivo exists ifTrue: [  
  archivo fileIn.  
  "Inicia el programa después de cargar el archivo"  
  Inicio crear.  
]  
ifFalse: [ self inform: 'El archivo no se encontró.' ].
```

-----

Una vez colocado el código del playground y ejecute, tardará unos segundos.

- Hay 15 libros, los cuales hay 5 de tipo impreso, 5 de tipo electrónico y 5 de tipo audio.
- Hay 5 bibliotecas, todas tienen los mismos libros pero diferentes para cada biblioteca.
- Hay 4 usuarios los cuales hay 2 usuarios plus y 2 usuarios normales.
  - El usuario normal con **dni: 1 y contraseña: 1**, tiene 3 bibliotecas de interés agregadas.
  - El usuario plus con **dni: 4 y contraseña: 4**, tiene 1 biblioteca de interés agregada.
  - EL usuario normal con **dni: 2 y contraseña: 2** y el usuario plus con **dni: 3 y contraseña: 3** no tienen bibliotecas de interés agregadas.

- Hay 3 préstamos registrados los cuales corresponden al usuario normal con dni: 1 y contraseña: 1 (1 préstamo para la biblioteca con id: 1) y el usuario plus con dni: 4 y contraseña: 4 (2 préstamos para la biblioteca con id: 2).
- Para acceder al botón “Acceso Administrador”, te pide el nombre de usuario y la contraseña del administrador.

**nombre de usuario: julian**

**contraseña: julian**

- Una vez puesto los datos y apretando el botón de “Acceder”, se abre la ventana del menú administrador, la cual tiene 3 botones: agregar biblioteca, agregar libro/s y quitar libro/s.
- Para acceder al botón “Acceso Usuario”, te pide el dni y la contraseña del usuario.
  - Una vez puesto los datos y apretando el botón de “Iniciar Sesión” se abre la ventana del menú usuario, la cual tiene 6 botones: agregar biblioteca de interés, cambiar tipo de usuario, solicitar préstamo, devolver préstamo, información de biblioteca y mi información (datos personas, bibliotecas de interés y préstamos activos).