

REQUERIMIENTOS FUNCIONALES

R1- Añadir un ítem al sistema de préstamos, bien sea un computador, libro o sala de estudio, estos ítems tienen un código y características específicas de acuerdo con qué tipo de ítem son. los libros tienen un título, un autor, una fecha de publicación, una editorial, y un número de páginas. Asimismo, existen dos tipos de libros: los literarios y los académicos. Estos últimos cuentan con una edición y una lista de cursos que lo usan como parte de su plan académico. Por su parte, los literarios tienen un género y un tipo (Novela, biografía o poesía). Por otro lado, las salas cuentan con un número de sillas y un número de conectores. Existen dos tipos: las salas de estudio, que cuentan con un tamaño de mesa y tablero, y las multimedia, con marca del televisor. Por último, los computadores. Estos tienen una marca, capacidad de disco y memoria ram.

R2- Buscar un ítem en el sistema de préstamos, bien sea un computador, libro o sala de estudio.

R3- Eliminar un ítem en el sistema de préstamos, bien sea un computador, libro o sala de estudio.

R4- Cargar los usuarios a partir de un archivo de texto plano provisto por la universidad. Estos usuarios pueden ser de dos tipos, estudiante o profesor. Cada persona tiene un nombre y un apellido, un número de documento y un estado (suspendido o no). Además, los estudiantes tienen un código, una carrera, y también se desea saber si es becado o no y si es monitor o no. Por otro lado, los profesores tienen una lista de cursos que dictan.

R5- Buscar un usuario registrado en el sistema de préstamos por su número de documento.

R6- Penalizar a un usuario del sistema de préstamos.

R7- Crear un nuevo préstamo teniendo en cuenta la hora inicial del préstamo y según su duración calcular la hora en la que se termina. Además, se debe tener en cuenta que objeto se prestó y a quién se le prestó.

R8- Buscar un préstamo en el sistema.

R9- Extender el tiempo de un préstamo.

R10- Finalizar un préstamo.

R11- Generar un informe sobre los préstamos en curso.

R12- Guardar los cambios efectuados en el sistema.

REQUERIMIENTOS NO FUNCIONALES

RNF1- Los ítems se añaden con un código dependiendo de su tipo y el número de ítem de ese tipo.

RNF2- Los usuarios se almacenan en un ArrayList.

RNF3- Los libros se almacenan en un árbol binario de búsqueda (ABB).

RNF4- Las salas se almacenan en árbol binario de búsqueda (ABB).

RNF5- Los computadores se almacenan en un linked list.

RNF6- Antes de usar los métodos add de la clase Library, toda la información (proporcionada por el usuario a través de la GUI) pasa por la clase InfoHandler, que se encarga de verificar su validez. Esto con el objetivo de no tener la clase Library llena de comprobaciones.

Escenarios

Nombre	Clase	Escenario
Setup_correc tFile	Library	Un archivo de texto con el formato correcto para crear un usuario.
Setup_onlyC ommentsFile	Library	Un archivo de texto que solo contiene comentarios, o sea que no carga ningún usuario.
Setup_incorr ectFormatFil e	Library	Un archivo de texto que contiene información de un estudiante con un formato incorrecto.
Setup_unkno wnClassIdent ifierFile	Library	Un archivo de texto con un identificador que no existe, los identificadores solo son s y p, y en este caso está q.
setup_incorre ctProffesorFo rmatFile	Library	Un archivo de texto con un formato incorrecto, en este caso, contiene mas información de la requerida.
setup_incorre ctStudentFor matFile	Library	Un archivo de texto con un formato incorrecto, en este caso contiene más información de la requerida.

DISEÑO DE PRUEBAS

Objetivo de la prueba: Verificar que el programa agrega los usuarios correctamente				
Clase	Método	Escenario	Valores de entrada	Resultado
Library	loadUsers	Setup_correctFile		Se agregan correctamente los usuarios pasados por el archivo de texto.

Objetivo de la prueba: Ver que si no se pasan los parámetros correctamente y en vez de esto se pasan solo comentarios no se agregan usuarios				
Clase	Método	Escenario	Valores de entrada	Resultado
Library	loadUsers	Setup_onlyCommentsFile		Al no agregarse ningún usuario el tamaño del arraylist debe ser 0

Objetivo de la prueba: Verificar que si el usuario tiene formatos incorrectos no se agrega a la lista de usuarios				
Clase	Método	Escenario	Valores de entrada	Resultado
Library	loadUsers	Setup_loadIncorrectFormatFile		Se espera la excepción UserLoadException

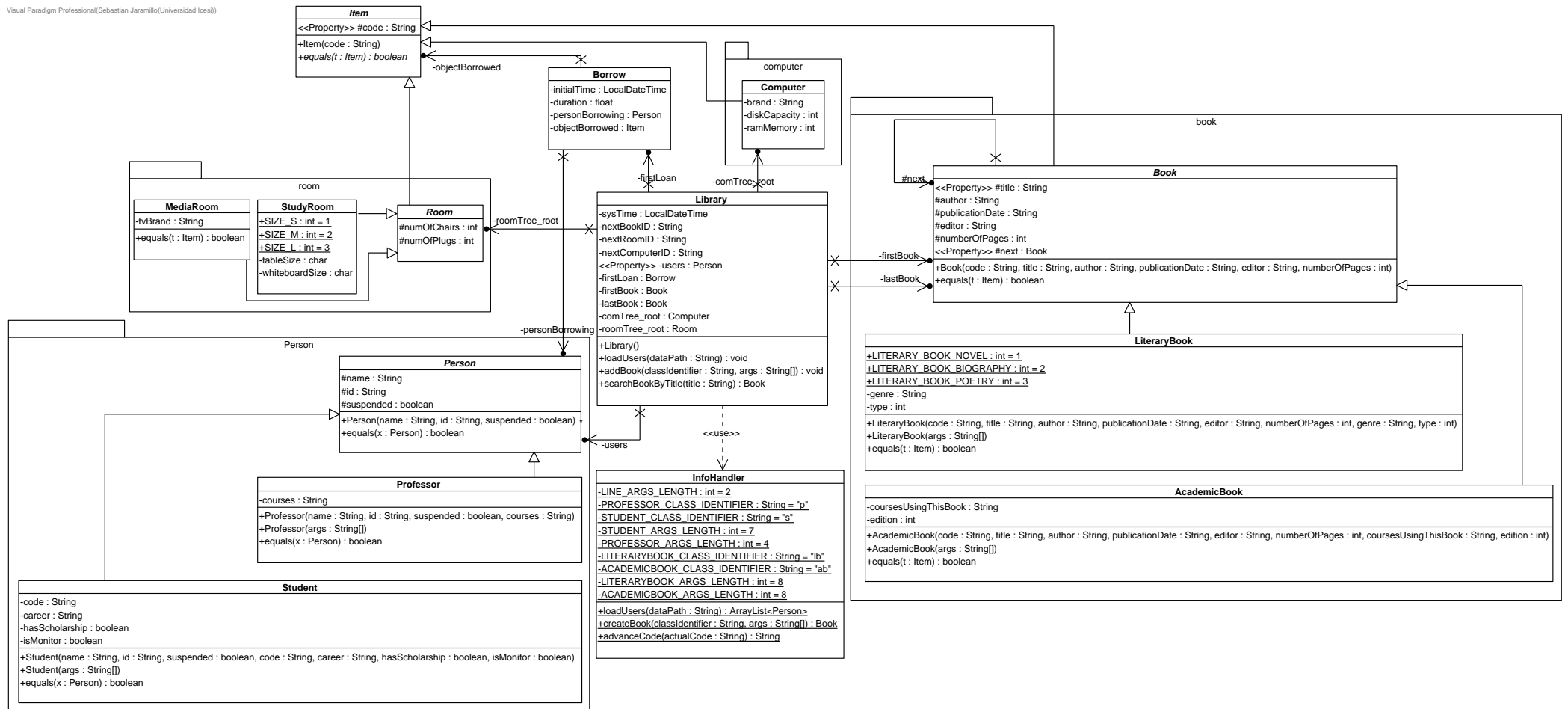
Objetivo de la prueba: Verificar que si el usuario no es profesor o estudiante y tiene otra clase no se agrega				
Clase	Método	Escenario	Valores de entrada	Resultado
Library	loadUsers	Setup_unknownClassId entifierFile		Se espera la excepción UserLoadException

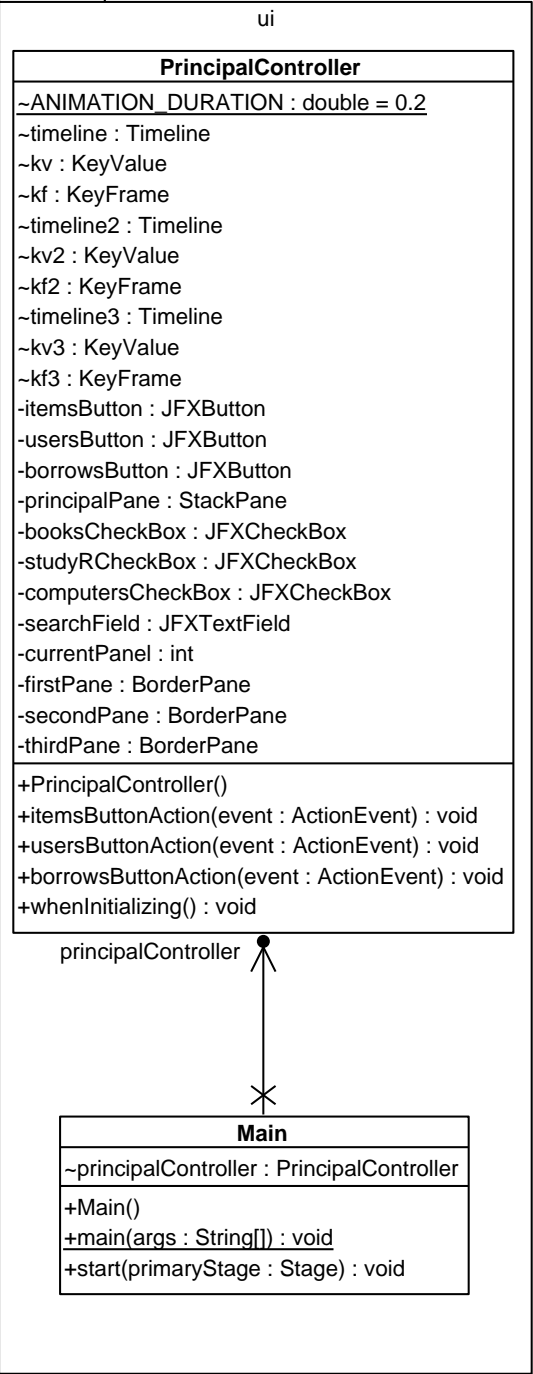
Objetivo de la prueba: Verificar que el programa no agrega a un profesor si tiene parámetros de mas

Clase	Método	Escenario	Valores de entrada	Resultado
Library	loadUsers	setup_incorrectProfessorFormatFile		Se espera la excepción UserLoadException

Objetivo de la prueba: Verificar que el programa no agrega a un estudiante si tiene parámetros de mas

Clase	Método	Escenario	Valores de entrada	Resultado
Library	loadUsers	Setup_incorrectStudentFormatFile		Se espera la excepción UserLoadException





model

LibraryTest

-path : String

-library : Library

-setup_correctFile() : void

-setup_onlyCommentsFile() : void

-setup_incorrectFormatFile() : void

-setup_unknownClassIdentifierFile() : void

-setup_incorrectProffesorFormatFile() : void

-setup_incorrectStudentFormatFile() : void

~loadUsers_correctFile() : void

~loadUsers_onlyCommentsFile() : void

~loadUsers_incorrectFormatFile() : void

~loadUsers_unknownClassIdentifierFile() : void

~loadUsers_incorrectProffesorFormatFile() : void

~loadUsers_incorrectStudentFormatFile() : void