



Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

PRESENTACIÓN DEL PROYECTO Y ESPECIFICACIÓN DE REQUISITOS

Patrones Software

Grado en Ingeniería Informática - Curso 2019/2020

Eduardo Graván Serrano – 03212337L

Adrián Montesinos González – 51139629A

Marcos Barranquero Fernández – 51129104N

ÍNDICE

Introducción	3
Análisis de requisitos	4
Requisitos funcionales	4
Requisitos no funcionales	6
Primer esquema de diseño	9
Usuarios de la aplicación	9
Clases Java	9
Patrones Software	10

INTRODUCCIÓN

Se plantea el desarrollo de un sistema software enfocado a la gestión de libros y usuarios para una biblioteca.

El sistema mantendrá un registro con los datos sobre los usuarios de la biblioteca. En este registro se almacenará la identidad de cada uno de estos usuarios asociado con los libros que han sacado de la biblioteca y la fecha máxima de devolución para estos libros. Los propios empleados de la biblioteca serán capaces de acceder al listado de usuarios de la biblioteca y a sus perfiles para consultar sus datos.

Desde el punto de vista de los clientes, la aplicación permitirá conectarse a través de Internet para que sean capaces de consultar el catálogo de libros disponibles, así como poder acceder a la información sobre los libros que ya hayan sacado de la biblioteca (título del libro, fecha tope de devolución...).

Para mantener la coherencia con los datos que gestiona la aplicación, se propone la serialización de todos los datos que sean necesarios para que el sistema funcione correctamente (lista de usuarios, datos de estos usuarios, catálogo de libros, información del estado de estos libros, etcétera). Esta serialización de los datos sería un primer acercamiento a la gestión de la persistencia de los datos, pudiéndose desarrollar una base de datos en el futuro si se viese necesario.

La aplicación contará con una interfaz gráfica sencilla y limpia para facilitar la navegabilidad por el sistema. Debido a que el lenguaje de programación a utilizar es Java, se plantea el uso de la librería Swing para el desarrollo de la interfaz gráfica. De cara al futuro, se propone el desarrollo de una aplicación web para reemplazar la aplicación de escritorio y facilitar de esta forma el acceso para los usuarios.

El sistema desarrollado deberá hacer un uso eficiente de los recursos de la plataforma en la que se ejecute. Deberá ser ligera y de fácil de utilizar para cualquier usuario, sin necesidad de instrucciones adicionales. Los datos de los usuarios se deben gestionar de forma segura y siguiendo las normas de privacidad especificadas en el Reglamento General de Protección de Datos (RGPD).

Lógicamente, debido a la diferenciación entre los dos tipos de usuarios (clientes de la biblioteca y empleados de la biblioteca) la interfaz gráfica y el acceso a funcionalidades de la aplicación deberán estar separadas para adecuarse a estos tipos de usuario.

Como ya se ha mencionado anteriormente, el lenguaje de programación a utilizar es Java. Lo cual permite que la aplicación desarrollada sea multiplataforma de forma nativa, gracias a la tecnología JVM.

ANÁLISIS DE REQUISITOS

En esta sección de la memoria se trata el análisis de requisitos del sistema software a desarrollar.

Debido a que la aplicación aún no ha entrado en su fase de desarrollo, estos requisitos podrían verse alterados de cara al producto final.

REQUISITOS FUNCIONALES

Se propone la siguiente lista de requisitos funcionales:

RF01

Identificador	RF-01
Nombre	Validación de usuario
Funcionalidad	El sistema comprueba que tanto la contraseña como el usuario coinciden con alguno de los registrados en la base de datos de los usuarios o administradores del Sistema.

RF02

Identificador	RF-02
Nombre	Gestión de libros
Funcionalidad	El Sistema permite dar de alta, baja y modificar los libros guardados.

RF03

Identificador	RF-03
Nombre	Validación de entradas
Funcionalidad	El Sistema debe validar las entradas del usuario para impedir registro de datos erróneos.

RF04

Identificador	RF-04
Nombre	Registro de estadísticas
Funcionalidad	El sistema registra el número de veces que se toman los libros, películas y otros artículos disponibles, de forma que se pueda estudiar las preferencias de los usuarios.

RF05

Identificador	RF-05
Nombre	Devolución de préstamos
Funcionalidad	El sistema permite gestionar devoluciones de libros para los usuarios de la administración.

RF06

Identificador	RF-06
Nombre	Gestión de castigo de usuarios
Funcionalidad	El sistema permite penalizar a los usuarios cuyos préstamos hayan vencido.

RF07

Identificador	RF-07
Nombre	Logueo de actividad
Funcionalidad	El sistema permite monitorizar la actividad de los usuarios.

RF08

Identificador	RF-08
Nombre	Clasificación de usuarios
Funcionalidad	El sistema distingue entre usuarios de la administración de la biblioteca y clientes.

RF09

Identificador	RF-09
Nombre	Serialización de los datos
Funcionalidad	El sistema almacena los datos utilizando serialización.

RF10

Identificador	RF-10
Nombre	Límite de prestamos
Funcionalidad	Cada usuario tiene un límite de libros y artículos asignados máximos que puede tener prestados a la vez.

RF11

Identificador	RF-11
Nombre	Atomicidad de datos
Funcionalidad	Las operaciones con datos y asignaciones deben de ser atómicas respecto a la serialización.

REQUISITOS NO FUNCIONALES

RNF01

Identificador	RNF-01
Nombre	Rendimiento
Descripción	La aplicación debe ser ligera para hacer un buen uso de los recursos hardware del terminal en el que se ejecute. Requisitos mínimos: Pentium II, 512MB de RAM, 500MB de almacenamiento en disco disponible.

RNF02

Identificador	RNF-02
Nombre	Disponibilidad
Descripción	El sistema debe ser accesible a cualquier hora del día, todos los días de la semana.

RNF03

Identificador	RNF-03
Nombre	Accesibilidad
Descripción	La interfaz gráfica de usuario debe ser clara y limpia para facilitar la navegabilidad de los usuarios. La aplicación no debería necesitar ningún tipo de documentación complementaria para su uso.

RNF04

Identificador	RNF-04
Nombre	Seguridad
Descripción	El sistema realizará copias de seguridad frecuentemente. Se debe seguir el RGPD en el tratamiento de los datos de los usuarios de la aplicación.

RNF05

Identificador	RNF-05
Nombre	Disponibilidad multiplataforma
Descripción	El sistema se podrá ejecutar sobre los principales sistemas operativos (Windows 7 en adelante, Linux, macOS).

RNF06

Identificador	RNF-06
Nombre	Lenguaje de programación
Descripción	El lenguaje de programación a utilizar para el desarrollo de la aplicación será Java en su versión 8.

RNF07

Identificador	RNF-07
Nombre	Interfaz gráfica
Descripción	El sistema tendrá una interfaz gráfica desarrollada utilizando la librería Swing.

RNF08

Identificador	RNF-08
Nombre	Manual de usuario
Descripción	Se entregará un manual de usuario que detalle el uso de la aplicación y los aspectos básicos del funcionamiento del sistema que los usuarios deban conocer.

RNF09

Identificador	RNF-09
Nombre	Empleo de patrones de diseño
Descripción	La aplicación debe desarrollarse haciendo uso de patrones software.

RNF10

Identificador	RNF-10
Nombre	Escalabilidad y modularidad de la aplicación
Descripción	El sistema debe ser escalable y modular, para poder permitir mantenimiento e implementaciones de funcionalidad futuras.

RNF11

Identificador	RNF-11
Nombre	Uso de métodos adecuados en relación a seguridad y estabilidad
Descripción	El sistema no debe utilizar métodos “deprecated” que puedan comprometer el rendimiento, la seguridad o estabilidad de la aplicación.

RNF12

Identificador	RNF-12
Nombre	Estándares de buenas prácticas
Descripción	El sistema debe emplear buenas prácticas relacionadas con las tecnologías empleadas en su fase de desarrollo.

RNF13

Identificador	RNF-13
Nombre	Interacción del texto
Descripción	El sistema permite las opciones de selección, copiado y pegado de texto.

PRIMER ESQUEMA DE DISEÑO

En este apartado se describe un primer acercamiento teórico al desarrollo del sistema software.

Como ya se menciona en secciones anteriores, debido a que la aplicación aún no se ha empezado a desarrollar, es posible que haya cambios significativos en la estructura propuesta.

Se plantean las clases Java que se prevé serán necesarias, así como la enumeración de una serie de patrones software que se cree podrán ser utilizados para el desarrollo del sistema.

USUARIOS DE LA APLICACIÓN

Se definen, a priori, los siguientes usuarios de la aplicación:

Administrador de biblioteca	
Rol	Gestión de artículos y usuarios <ul style="list-style-type: none">Alta y baja de usuariosGestión de libros y artículosMantenimiento de backups
Atributos	<ul style="list-style-type: none">Hereda de clienteEl correo debe acabar en @biblioteca.uah.es

Usuario	
Rol	Alquiler de libros y artículos
Responsabilidades	<ul style="list-style-type: none">Alta y baja de usuariosGestión de libros y artículosMantenimiento de backups
Atributos	<ul style="list-style-type: none">NombreMailContraseñaDNITeléfonoCorreo

CLASES JAVA

Se definen, a priori, las siguientes clases:

Libro	
Descripción	Abstracción de un libro
Atributos	<ul style="list-style-type: none">NombreISBNAutorEdiciónFechaCategoría

Otros artículos	
Descripción	Se estudian implementar otros artículos prestables tales como portátiles, películas, regletas, etc.
Atributos	<ul style="list-style-type: none"> • Los necesarios para ese artículo.

Logueador	
Descripción	Debe monitorizar y escribir en archivos la actividad del usuario en la aplicación.
Atributos	<ul style="list-style-type: none"> • Archivo donde se loguea • Serializador

Estadísticas	
Descripción	Debe estudiar el volume de artículos prestados.
Atributos	<ul style="list-style-type: none"> • Estructuras de datos apropiadas para dicha implemetación.

Serializador	
Descripción	Debe serializar y almacenar los datos apropiadamente.
Atributos	<ul style="list-style-type: none"> • Estructuras de datos apropiadas para dicha implemetación.

PATRONES SOFTWARE

Se estudia el empleo de los siguientes patrones:

PATRONES DE CREACIÓN

- **Patrón Singleton:** proporcionamos un punto de acceso global a una clase que solo puede tener una instancia.
- **Patrón Factory:** permite la instanciación de objetos delegada en una clase.
- **Patrón Builder:** permite separar la construcción de un objeto en distintas fases, validando y asegurando que se actualiza la base de datos, el estado, etc.

PATRONES DE COMPORTAMIENTO

- **Patrón Observer:** permite establecer una clase en la cual los cambios que sufra se notificarán al resto de clases.
- **Patrón Iterator:** proporciona una interfaz que permite recorrer colecciones elemento a elemento.
- **Patrón Command:** proporciona una clase que permite encapsular una acción y los datos necesarios para ejecutarlo. Lo empleamos para realizar la acción de deshacer.

- **Patrón Strategy:** proporciona una interfaz para configurar el comportamiento de otros componentes. Podemos emplearlo para ordenar y filtrar listas de artículos, libros, etc.
- **Patrón State:** permite definir diferentes comportamientos para un objeto en función de su estado. Utilizable para definir las operaciones entre libros y artículos en función de si están disponibles, prestados, etc.

PATRONES ESTRUCTURALES

- **Fachada:** permite realizar un acceso más simplificado a las clases o subsistemas a través de la clase fachada.
- **Patrón Bridge:** desacopla un componente complejo en dos clases separadas en la abstracción funcional y la implementación interna.
- **Patrón Proxy:** permite añadir lógica adicional entre la interfaz de un objeto y su cliente, como controlar el acceso a diferentes recursos según una lógica de validación. Usable, por ejemplo, a la hora de visualizar información de usuarios y empleados.

FRAMEWORK

Emplearemos el Framework de Swing de Java para realizar la implementación de la interfaz gráfica.