

Informe de Ingeniería

Librería Distrital



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**
Acreditación Institucional de Alta Calidad

Andres Nicolas Mendoza Silva - 20222020152

Daniela Alejandra Gaviria Gonzalez - 20221020091

Tomás Alejandro Delgado Ortiz - 20221020045

Santiago Andrés Guarín Rojas - 20221020101

Valentina Alfonso Vargas - 20222020178

Sergio Porras Rodriguez - 20222020146

Facultad de Ingeniería

Universidad Distrital Francisco José de Caldas

Modelos de Programación I

Helio Henry Ramirez Arevalo

Nancy Yaneth Gelvez Garcia

Junio 2024

Índice.

1. Introducción.....	3
- Descripción general del proyecto.....	3
- Propósito del informe.....	3
2. Decisiones tomadas en el desarrollo del proyecto.....	4
- Distribución de roles.....	4
- Asignación de tareas.....	4
- Metodología de desarrollo.....	4
- Canales de comunicación.....	4
- Implementación de patrones de diseño y modelamiento del sistema.....	5
3. Patrones de diseño empleados.....	5
- Singleton.....	5
- Abstract Factory.....	5
- Prototype.....	5
- Facade.....	5
- Adapter.....	6
- Proxy.....	6
- DAO.....	6
4. Reflexión crítica sobre el desarrollo del proyecto.....	6
- Desempeño Grupal.....	6
- Aplicabilidad de las decisiones tomadas.....	7
- Proceso de análisis.....	7
- Aprendizajes y mejoras continuas.....	7
5. Conclusión.....	8
6. Referencias.....	8

Informe de ingeniería para el proyecto Librería Distrital

1. Introducción.

- Descripción general del proyecto

Este proyecto de Ingeniería de Software tiene como objetivo desarrollar una librería de documentos en formato web que permita a diversos usuarios introducir, consultar y modificar datos de documentos existentes. La herramienta se concibe como un sistema de información dinámico, capaz de gestionar diferentes tipos de documentos, como libros, ponencias y artículos científicos. Cada documento contendrá información detallada, incluyendo título, fecha de publicación, autores, ISBN, SSN, editorial y datos de contacto de autores y editoriales. Además, se registrará quién creó, modificó o eliminó el documento, con la correspondiente fecha de cada acción, garantizando que solo el creador original pueda eliminarlo.

La librería funcionará de manera sencilla y eficiente, permitiendo que los documentos comiencen a existir en el sistema con la introducción de cualquier dato, con la posibilidad de agregar información adicional posteriormente. Los usuarios podrán reservar documentos para consulta física y devolverlos cuando ya no los necesiten. Además, los documentos podrán ser dados de baja y reactivados según sea necesario. El sistema también incorporará autenticación de usuarios para asegurar que todas las acciones de creación, modificación y eliminación de documentos sean seguras y rastreables. El objetivo principal del proyecto es maximizar la calidad del software desarrollado, asegurando que la solución informática sea robusta, fácil de usar y capaz de gestionar de manera efectiva la entrada, modificación y consulta de documentos en la librería.

- Propósito del informe

El propósito de este informe es detallar de manera exhaustiva el desarrollo del proyecto de creación de una librería de documentos en formato web, desde su concepción inicial hasta su implementación final. Este documento pretende ofrecer una visión clara y detallada de las decisiones tomadas durante el proceso, incluyendo la elección de tecnologías y herramientas, la estructura del sistema y las consideraciones de seguridad y usabilidad. Además, se describen los patrones de

diseño empleados para asegurar que el sistema sea modular, escalable y fácil de mantener.

2. Decisiones tomadas en el desarrollo del proyecto.

Durante el desarrollo de este proyecto de ingeniería de software nuestro grupo de desarrollo se vio inmerso en la toma de decisiones que contribuyeron a la optimización de procesos y de tiempo, buena comunicación del equipo y desarrollo del proyecto bajo buenas prácticas, sin dejar de lado el uso de patrones de diseño que garantizan su mantenimiento y uso de manera más sencilla.

- Distribución de roles

Se asignó un rol específico a cada miembro del equipo, como líder, desarrollador, responsable de pruebas, responsable de documentación, identificador de requerimientos, con el fin de asegurar una distribución clara de responsabilidades y áreas de especialización, teniendo en cuenta las habilidades de destrezas de cada integrante del equipo.

- Asignación de tareas

Teniendo en cuenta la distribución de roles dentro del equipo, se asignaron tareas y fechas límite para el cumplimiento de estas tareas, cada integrante, según su rol, tuvo que mostrar y compartir evidencia de su avance en las reuniones grupales de desarrollo.

- Metodología de desarrollo

El grupo decidió el uso de una metodología de desarrollo ágil, teniendo en cuenta el tiempo del que se disponía para el desarrollo del proyecto, por lo que se usó la metodología Kanban, para facilitar la colaboración y la adaptación a los cambios en el proyecto. Esto permitió dividir el trabajo en iteraciones manejables y responder rápidamente a los cambios de percepción frente a los requisitos propuestos en el documento del enunciado del proyecto.

- Canales de comunicación

El grupo de desarrollo decidió establecer canales de comunicación claros y accesibles, como correos electrónicos, chats grupales (Whatsapp) y herramientas de colaboración

en línea (Github y Notion), para facilitar la comunicación entre los miembros del equipo y garantizar que todos estuvieran al tanto de los cambios y actualizaciones en el proyecto.

- **Implementación de patrones de diseño y modelamiento del sistema**

En una reunión grupal, el equipo decidió usar los patrones de diseño que encontró pertinentes para el desarrollo del proyecto tales como prototype, singleton, facade, DAO, proxy, factory, adapter, de igual manera se hizo un reconocimiento y abstracción de clases según el documento de requerimientos.

3. Patrones de diseño empleados.

- **Singleton**

El patrón Singleton se implementó en el sistema de login para garantizar que solo una instancia de la sesión de usuario esté activa en un momento dado. Este enfoque es crucial por motivos de seguridad, ya que previene la posibilidad de que múltiples instancias de sesión sean creadas simultáneamente, lo que podría conducir a inconsistencias y vulnerabilidades en el manejo de las sesiones.

- **Abstract Factory**

El patrón Abstract Factory se utilizó para la creación de archivos según su tipo y atributos. Dado que nuestro sistema maneja diferentes tipos de documentos, como libros, ponencias y artículos científicos, cada uno con sus propios atributos específicos, se necesitaba una forma flexible y extensible de instanciar estos objetos.

- **Prototype**

El patrón Prototype se utilizó para la creación de copias de cada registro original de los documentos. En nuestro sistema, existe la necesidad de duplicar registros de documentos de manera eficiente para mantener versiones o realizar pruebas sin alterar los datos originales.

- **Facade**

El patrón Facade se implementó para simplificar el acceso a los métodos modulares a través de la interfaz gráfica de usuario (GUI). Dado que el sistema incluye múltiples

funcionalidades y operaciones complejas, como la gestión de documentos, usuarios y registros de actividades, era necesario proporcionar una interfaz sencilla y coherente para los usuarios.

- **Adapter**

El patrón Adapter se implementó para la conversión del formato de datos de cada uno de los documentos. Dado que los documentos en la librería pueden tener diferentes formatos y estructuras de datos, era necesario contar con una forma flexible y eficiente de convertir estos formatos para asegurar la interoperabilidad dentro del sistema.

- **Proxy**

El patrón Proxy se implementó para la sobreescritura del estado de los libros y para servir como puente entre la interfaz de usuario y el DAO (Data Access Object). El Proxy actúa como intermediario, controlando el acceso a los libros y asegurando que las operaciones de modificación y consulta se manejen adecuadamente.

- **DAO**

El patrón DAO (Data Access Object) se utilizó para el manejo de la persistencia de datos en el sistema, abarcando las operaciones CRUD (Create, Read, Update, Delete) de los documentos. El DAO proporciona una capa de abstracción entre la lógica de negocio y la base de datos, facilitando el acceso y la manipulación de los datos almacenados.

4. Reflexión crítica sobre el desarrollo del proyecto.

- **Desempeño Grupal**

El desempeño grupal en el desarrollo del proyecto fue notablemente positivo, caracterizado por una colaboración efectiva, comunicación clara y una gestión del tiempo eficiente. Desde el inicio, establecimos canales de comunicación abiertos y regulares, utilizando herramientas como WhatsApp, Notion y reuniones semanales para asegurarnos de que todos los miembros del equipo estuvieran al tanto del progreso y de cualquier desafío emergente. La distribución de tareas se realizó de manera equitativa, aprovechando las fortalezas individuales de cada miembro, lo que

contribuyó a un buen flujo de trabajo para alcanzar los objetivos del proyecto a tiempo

- Aplicabilidad de las decisiones tomadas

A lo largo del proyecto, las decisiones que tomamos resultaron efectivas y adecuadas para alcanzar nuestros objetivos. Optamos por utilizar varios patrones de diseño, como Singleton, Abstract Factory, Prototype, Facade, Adapter, Proxy y DAO, para asegurar que el sistema fuera escalable, fácil de mantener y seguro. Esto nos permitió organizar el sistema de manera clara y fácil de entender, lo que hizo que el desarrollo, la integración y cualquier expansión futura fueran más simples. Además, nos enfocamos en diseñar la interfaz de usuario pensando en el usuario, lo que resultó en una experiencia intuitiva y satisfactoria para los usuarios.

- Proceso de análisis

Aunque usamos herramientas más simples como Notion para la gestión de tareas y Git para el control de versiones en lugar de las tradicionales, nuestro proceso de análisis fue sólido y efectivo. Empleamos Notion para organizar los requisitos del proyecto, asignar tareas y seguir el progreso. Git nos permitió colaborar en el código, con cada miembro del equipo trabajando en sus propias ramas y fusionando cambios de manera ordenada. A pesar de la simplicidad de nuestras herramientas, pudimos mantener un seguimiento claro del avance del proyecto y tomar decisiones informadas en cada etapa del desarrollo

- Aprendizajes y mejoras continuas

Este proyecto nos mostró lo importante que es adaptarnos y aprovechar al máximo las herramientas disponibles. Aprendimos a ser más eficientes utilizando herramientas simples pero poderosas como Notion y Git, y vimos la importancia de una comunicación clara y una planificación detallada en todo momento. Como área para mejorar, reconocimos que podríamos haber integrado más de cerca las metodologías ágiles en nuestro proceso de análisis, lo que podría habernos brindado más flexibilidad y adaptabilidad frente a nuevos requisitos o desafíos. En proyectos futuros, nos esforzaremos por explorar y utilizar más herramientas que nos ayuden a mejorar nuestra colaboración y productividad.

5. Conclusión.

El desarrollo del proyecto de la librería de documentos destacó por la aplicación efectiva de diversos patrones de diseño, una colaboración grupal sólida y un análisis detallado utilizando herramientas como Notion y Git. Los patrones de diseño como Singleton, Abstract Factory y Proxy fueron fundamentales para garantizar la escalabilidad, mantenibilidad y seguridad del sistema. La comunicación clara, la distribución equitativa de tareas y la gestión eficiente del tiempo fueron aspectos clave del desempeño grupal. Esta experiencia ha sido invaluable para nuestro desarrollo profesional como ingenieros de sistemas, proporcionándonos una comprensión más profunda de la aplicación práctica de los patrones de diseño y familiarizándonos con herramientas esenciales para la gestión de proyectos de desarrollo de software. Además, la colaboración en equipo nos ha enseñado la importancia de la comunicación efectiva y la distribución equitativa de responsabilidades, mientras que hemos aprendido a adaptarnos y sacar el máximo provecho de las herramientas disponibles, lo que seguramente nos beneficiará en futuros proyectos profesionales.

6. Referencias.

Perfil, V. T. mi. (s/f). Mi granito de java. Blogspot.com. Recuperado el 3 de junio de 2024, de <https://migranitodejava.blogspot.com/2011/05/singleton.html?m=1>

Refactoring Guru. (s.f.). Design Patterns. Recuperado de <https://refactoring.guru/es/design-patterns>

Data Access Object (DAO). (s/f). Reactiveprogramming.io. Recuperado el 3 de junio de 2024, de <https://reactiveprogramming.io/blog/es/patrones-arquitectonicos/dao>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley

Debrauwer, L. (2018). Patrones de diseño en Java: los 23 modelos de diseño: descripciones y soluciones ilustradas en UML 2 et Java. Ediciones ENI..