|  |  |  |
| --- | --- | --- |
| **Objetivos**   1. Realizar un ejercicio que permita incorporar java web 2. Almacenar información en la memoria temporal 3. Utilizar un lista doblemente encadenada 4. Demostrar la comunicación en la web | **Número de participantes**  participante.jpgX 4 | **Materiales**   * Internet * Eclipse o Netbeans * Apache Tomcat * Maven |
| **Duración de la actividad**  **8:00** |
| **Problema: Sistema de Gestión de una biblioteca online**  **Descripción del Problema:**  Crear un sistema de gestión de datos de una biblioteca en línea. El sistema permitirá a los usuarios realizar operaciones como agregar libros, buscar libros por título o autor, y realizar operaciones de préstamo y devolución. Debe implementar esta aplicación utilizando Java web con Maven y Apache Tomcat y utilizar una estructura de datos de **lista doblemente enlazada.**  **Requisitos Funcionales: (completar los requerimientos)**   |  |  | | --- | --- | | Nombre RF1 | Agregar un libro | | Entradas | Titulo  Autor  Año publicación  Foto de la portada | | Descripción | El sistema debe poder agregar un libro nuevo | | Resultado | Libro agregado al sistema |  |  |  | | --- | --- | | Nombre RF2 | Buscar un libro por título o autor. | | Entradas | Título del Libro o Autor: Usuario proporciona el título del libro o el nombre del autor. | | Descripción | Validación de Entrada: Verifica que se haya ingresado información.  Búsqueda en la Base de Datos: Simula la búsqueda utilizando el título o autor proporcionado.  Presentación de Resultados: Muestra la lista de libros coincidentes.  Detalles del Libro: Muestra información adicional del libro. | | Resultado | Muestra detalles del libro.  Informa al usuario y sugiere ajustes en la búsqueda. |  |  |  | | --- | --- | | Nombre RF3 | Mostrar todos los libros disponibles | | Entradas | Solicitud de Mostrar Todos los Libros: El usuario solicita ver la lista completa de libros disponibles. | | Descripción | Acceso a la Base de Datos: Accede a la base de datos de libros.  Recuperación de Información: Recupera la información de todos los libros disponibles.  Presentación de Resultados: Muestra la lista completa de libros al usuario. | | Resultado | Se presenta al usuario una lista de todos los libros disponibles, con detalles como título, autor, año de publicación, y cualquier otra información relevante. |  |  |  | | --- | --- | | Nombre RF4 | Realizar operaciones de préstamo y devolución. | | Entradas | Solicitud de Préstamo o Devolución: El usuario indica si desea realizar una operación de préstamo o devolución | | Descripción | Verificación de Préstamo: Confirma que el libro fue prestado.  Registro de Devolución: Registra la información de la devolución, incluyendo el libro y la fecha de devolución.  Actualización del Estado del Libro: Marca el libro como disponible en la base de datos. | | Resultado | Préstamo Exitoso: Confirma el préstamo y proporciona detalles, incluyendo la fecha de devolución esperada.  Devolución Exitosa: Confirma la devolución y actualiza el estado del libro a disponible.  Préstamo No Exitoso: Informa si el libro no está disponible para préstamo.  Devolución No Exitosa: Informa si el libro no fue prestado previamente. |   **Requisitos Técnicos:**  1. Utilizar Java para la lógica de negocio, use NetBeans como entorno de desarrollo  2. Implementar el proyecto en el marco de estructuras de datos utilizando listas doblemente encadenadas.  3. Implementar una interfaz de usuario web utilizando tecnologías como HTML, CSS y JSP (Java Server Pages).  4. Utilice Bootstrap para darle una mejor presentación a su trabajo  4. Utilizar Maven para la gestión de dependencias y la construcción del proyecto.  5. Desplegar la aplicación en un servidor Apache Tomcat.  6. Implementar la gestión de sesiones para mantener a los usuarios autenticados.  7. Proporcionar una capa de persistencia para almacenar los datos de usuario y las tareas pendientes (manejo de archivos).  Este proyecto te permitirá aplicar tus conocimientos en estructuras de datos de lista doblemente enlazadas para crear un sistema funcional de gestión de tareas. Puedes expandir este proyecto agregando características adicionales, como la capacidad de marcar tareas como completadas o la asignación de tareas a categorías específicas.  **Fases del proyecto**   |  | | --- | | 1. **Análisis de requerimientos y diseño del mundo** | | * Definir los requerimientos funcionales * Crear un diagrama de clases preliminar. * Especificar el funcionamiento de la estructura de las listas doblemente encadenadas y su funcionalidad en el contexto del proyecto. | | 1. **Implementación del modelo de datos** | | * Crear las clases para representar libros, usuarios y otros elementos relevantes. * Implementar la lógica para la manipulación de listas doblemente encadenadas. | | 1. **Desarrollo de la capa de persistencia** | | * Implementar la lógica para escribir y leer datos en archivos txt | | 1. **Desarrollo de la capa de presentación:** | | * Configurar el servidor Apache Tomcat y el entorno web * Crear servlets para manejar las solicitudes de los usuarios y mostrar las respuestas correspondientes. * Diseñar una interfaz web sencilla para interactuar con el sistema de gestión de la biblioteca. | | 1. **Pruebas y depuración** | | * Realizar pruebas exhaustivas para garantizar el funcionamiento correcto del sistema en diferentes escenarios. * Depurar cualquier error encontrado durante las pruebas | | 1. **Documentación y presentación** | | * Preparar documentación detallada sobre el diseño del proyecto, la implementación y las pruebas realizadas. * Presentar el proyecto a los demás estudiantes y profesor, explicando los aspectos técnicos y las decisiones de diseño tomadas durante el desarrollo, así como el manejo de las listas doblemente enlazadas. | | | |