|  |  |  |
| --- | --- | --- |
| **Resultados de Aprendizaje**  Aplicar todos los conocimientos adquiridos en el semestre frente a la construcción de una aplicación de software usando estructuras de datos lineales | **Número de participantes**  X 3 | **Materiales**   * Internet * Eclipse o Netbeans * Apache Tomcat * Maven |
| **Duración de la actividad**  Proyecto final |
| 1. **Planteamiento del Proyecto:**   El estudiante debe plantear un ejercicio al profesor, de manera libre y autónoma, de tal forma que pueda aplicar los conocimientos de listas simples y doblemente enlazadas  Para ello se establecerán las siguientes condiciones para la entrega.   1. Se motiva a los estudiantes a conformar grupos de trabajo de hasta tres personas. 2. El trabajo debe estar orientado a java web JEE8 con Maven como biblioteca de repositorios y apache tomcat 9 3. El sistema debe contar con una persistencia de grabación en documentos de texto (.TXT) 4. El código debe tener la funcionalidad de un CRUD que permita Ingresar datos, ver detalle de los registros, eliminar un registro, modificar un registro, listar los datos. 5. Es importante definir que se valorará la documentación de la aplicación y con ella se revisará el software. 6. El estudiante puede solicitar el acompañamiento del docente frente a dudas que tenga en la realización del proyecto, en las horas destinadas para clase o en las horas definidas para el acompañamiento de los cursos. 7. Para la documentación tener en cuenta lo siguiente: 8. **Diseño de software**   **Listado de Requerimientos** (Liste aquí los requerimientos funcionales que la aplicación tendrá)   |  |  | | --- | --- | | Nombre | R1-Persistencia en archivos txt | | Resumen | El sistema debe almacenar y recuperar los contactos utilizando un archivo de texto plano (.TXT) para asegurar la persistencia de los datos. | | Entradas | Datos del contacto (nombre, email, teléfono) y las acciones del usuario (crear, editar, eliminar). | |  |  | | Resultados | El archivo contacts.txt es actualizado y refleja el estado actual de la lista de contactos tras cada operación. | |  | |  |  |  | | --- | --- | | Nombre | R2-Creacion de contactos | | Resumen | El aplicativo debe permitir al usuario registrar nuevos contactos atraves de un formularios en la interfaz web | | Entradas | Nombre , correo electronico y telefono | |  |  | | Resultados | El contacto es agregado a la lista y guardado en el archivo contacts.txt | |  | |  |  |  | | --- | --- | | Nombre | R3-Listado de contactos | | Resumen | El sistema debe mostrar al usuario la lista completa de contactos almacenados en el archivo de texto | | Entradas | Solicitud de visualización de contactos (por ejemplo, al acceder a la página principal). | |  |  | | Resultados | Se muestra una tabla o lista con todos los contactos registrados, obtenidos desde el archivo de texto. | |  | |  |  |  | | --- | --- | | Nombre | R4-Visualización de detalles de un contacto | | Resumen | El sistema debe permitir al usuario consultar los detalles de un contacto específico. | | Entradas | Selección de un contacto de la lista (por ID). | |  |  | | Resultados | Se despliega una pantalla con los datos completos del contacto seleccionado. | |  | |  |  |  | | --- | --- | | Nombre | R5-Edición de contactos | | Resumen | El sistema debe permitir modificar los datos de un contacto existente. | | Entradas | Selección de un contacto y datos modificados (nombre, email, teléfono). | |  |  | | Resultados | El contacto es actualizado en la lista y el archivo de texto refleja los nuevos valores. | |  | |  |  |  | | --- | --- | | Nombre | R6-Eliminacion de contactos | | Resumen | El sistema debe permitir al usuario eliminar un contacto registrado. | | Entradas | Selección de contacto a eliminar (por ID). | |  |  | | Resultados | El contacto es removido de la lista y el archivo contacts.txt se actualiza para reflejar la eliminación. | |  | |  |  |  | | --- | --- | | Nombre | R7-Validación de datos | | Resumen | El sistema debe validar que los datos ingresados por el usuario sean correctos antes de procesar la operación (por ejemplo, que el email tenga formato válido y el teléfono no esté vacío). | | Entradas | Datos del formulario al crear o editar un contacto. | |  |  | | Resultados | Si hay errores de validación, se informa al usuario y no se realiza la operación; si son válidos, se procesa normalmente. | |  | |   **Modelo conceptual**  Realice el diagrama de clases de la aplicación  **Requisitos Técnicos del Proyecto “Contact Manager”**   * **Java JDK 17**   El sistema está desarrollado y probado con Java 17, versión compatible con Maven y Apache Tomcat   * **Apache Maven**   Se utilizó Apache Maven como herramienta de gestión y construcción del proyecto. Versión mínima recomendada: 3.6+.   * **Apache Tomcat 9**   El servidor de aplicaciones utilizado para desplegar y ejecutar la aplicación web es Apache Tomcat 9.   * **NetBeans IDE**   El desarrollo del proyecto se realizó usando el entorno de desarrollo integrado (IDE) NetBeans, que facilita la gestión de proyectos Maven y la integración con Tomcat.   * **JSP (JavaServer Pages)**   Para la construcción de las vistas dinámicas, el sistema utiliza JSP.   * **Servlets**   La lógica de negocio y el manejo de peticiones HTTP se realiza a través de servlets Java.   * **CSS y HTML5**   La interfaz gráfica del sistema se diseñó utilizando HTML5 y hojas de estilo CSS para asegurar una experiencia visual adecuada.   * **JSTL (JavaServer Pages Standard Tag Library)**   Se emplea JSTL para facilitar la manipulación de datos y lógica de presentación en las JSP.   * **Git**   El control de versiones y la colaboración en el desarrollo del proyecto se gestionó a través de Git  **Prototipo de interfaz**    **Promp usado con chatgpt**  Guiame paso a paso para poder desarrollar una aplicación web para la gestión de contactos personales, utilizando Java con servlets y JSP, que permita registrar, listar, consultar, editar y eliminar contactos. El sistema debe garantizar la persistencia de la información mediante el uso de archivos de texto plano (.TXT) como medio de almacenamiento, sin emplear bases de datos tradicionales,quiero que me expliques que hace cada cosa de manera detallada.   1. **Evaluación** 2. Para evaluar el programa el estudiante debe solicitar la reunión con el profesor y su grupo de trabajo 3. Una valoración será por la documentación anteriormente descrita 4. Otra Valoración será del aplicativo el cual se revisará de acuerdo con la documentación 5. **Criterios de evaluación** 6. El estudiante debe demostrar que ha construido un aplicativo funcional, sin errores con las condiciones que establece el profesor 7. El estudiante debe demostrar que conoce el manejo de la persistencia simple y la sabe integrar con el aplicativo de software 8. El estudiante debe demostrar que conoce el desarrollo web y sus pilares como lo son html, css y javascript, de igual forma los frameworks de apoyo en su construcción como lo son fontawesome, Bootstrap, tailwind 9. El estudiante es consiente que las aplicaciones de software deben ser fáciles de usar, es decir, propender por la funcionalidad 10. El estudiante debe demostrar que tiene tanto el conocimiento de la parte funcional de la construcción como de su construcción 11. El estudiante debe demostrar que hace uso de la IA de manera ética y responsable, no colocando código que no entiende, puesto que será puesto a prueba, en caso de usar la IA al final de este documento colocar el prompt que utilizó y la IA donde lo aplicó. | | |