|  |  |  |
| --- | --- | --- |
| **Objetivos**   1. Realizar una revisión de los elementos fundamentales de la lógica de programación 2. Establece una guía de construcción a partir de la documentación de un proyecto de software 3. Construir un prototipo funcional de software en un IDE | **Número de participantes**  participante.jpgparticipante.jpg | **Materiales**   * Internet * Eclipse o NetBeans * Procesador de texto |
| **Duración de la actividad**  **4:00** |
| **Problema: Gestión de un Diario Personal**  **Descripción del Problema:**  Se desea desarrollar un programa en Java para crear un diario personal. El diario permitirá al usuario agregar nuevas entradas, cada una con una fecha automática del sistema y una descripción proporcionada por el usuario. Además, el usuario podrá consultar una entrada específica proporcionando la fecha correspondiente. El usuario podrá modificar una entrada agregando una nueva descripción y al final podrá eliminar una entrada, como condición el usuario solo podrá ingresar una sola entrada por día. Para realizar el ejercicio es necesario utilizar conceptos de Programación Orientada a Objetos (POO), ciclos, condicionales y manejo de contenedoras variables para el manejo de una persistencia simple basada en memoria.  **Requerimientos Funcionales:**   |  |  | | --- | --- | | Nombre RF1 | AgregarEntrada | | Entradas | idEntrada,  fecha,  descripcion | | Descripción | Permite agregar una nueva entrada al diario personal | | Resultado | Entrada agregada al diario personal |  |  |  | | --- | --- | | Nombre RF2 | ConsultarEntrada | | Entradas | fecha | | Descripción | Arroja la entrada que el sistema encontró con la fecha ingresada por el usuario | | Resultado | Entrada listada |  |  |  | | --- | --- | | Nombre RF3 | EliminarEntrada | | Entradas | idEntrada | | Descripción | Permite eliminar la entrada teniendo en cuenta el idEntrada ingresado por el usuario | | Resultado | Entrada del diario eliminada |  |  |  | | --- | --- | | Nombre RF4 | ModificarEntrada | | Entradas | idEntrada | | Descripción | Permite ingresar una nueva descripción en la entrada encontrada a partir del idEntrada ingresada por el usuario | | Resultado | Entrada modificada en su descripción |  |  |  | | --- | --- | | Nombre RF5 | MostrarMenu | | Entradas | N/A | | Descripción | Permite mostrar el menú de opciones al usuario | | Resultado | Menú de opciones listado en la consola |   **Requerimientos No Funcionales:**   1. Usabilidad: la interfaz de usuario debe ser intuitiva y fácil de usar. El proceso de agregar y consultar entradas debe ser claro y directo para el usuario. 2. Seguridad: las entradas del diario deben estar protegidas y no accesibles para otros usuarios. 3. Rendimiento: el programa debe ser eficiente en términos de uso de recursos y respuesta del sistema. 4. Persistencia: el programa debe permitir que la información perdure en la memoria RAM mientras la aplicación este activa   **Diagrama de clases:**    **Requisitos Técnicos:**  1. Utilizar Java para la lógica de negocio, utilice NetBeans como entorno de desarrollo  2. Implementar una interfaz de usuario basada en la consola del programa  3. Permita una mensajería adecuada con el usuario  4. Utilizar Maven para la gestión de dependencias y la construcción del proyecto.  7. Proporcionar una capa de persistencia sencilla con contenedoras para almacenar los datos de usuario.  Este proyecto permitirá poner en práctica tus conocimientos previos en algoritmos y programación para crear un sistema funcional de gestión de tareas. Puedes expandir este proyecto agregando características adicionales.  **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 las contenedoras y su funcionalidad en el contexto del proyecto. | | 1. **Implementación del modelo de datos** | | * Crear las clases necesarias para el mundo * Implementar la lógica para la manipulación de las contenedoras ya sean de tamaño fijo o variable | | 1. **Desarrollo de la capa de persistencia** | | * Implementar la lógica para almacenar datos en la memoria RAM | | 1. **Desarrollo de la capa de presentación:** | | * Diseñar una interfaz sencilla para interactuar con el usuario | | 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 al profesor, explicando los aspectos técnicos y las decisiones de diseño tomadas durante el desarrollo, así como las dificultades que se presentaron. |   **Entregable**  El estudiante debe entregar una aplicación perfecta (sin errores), debe subirla a GIT y en la plataforma subir un enlace en un archivo de texto para poder acceder a el. El programa debe ser Orientado a Objetos. | | |