Universidad Da Vinci De Guatemala

Facultad de Ingeniería

Ingeniería en Sistemas

Logotipo

El contenido generado por IA puede ser incorrecto.

**Implementación de un Diario de Aventuras**

Gustavo David Campos Morales

202400509

Estructuras de datos

Guatemala, marzo de 2025

**Explicación del Problema**

Eres un aventurero que viaja por el mundo completando misiones y quieres llevar un registro eficiente de todas tus hazañas. Para ello, debes implementar un **Tipo de Dato Abstracto (TDA)** en Java que administre un **diario de misiones**.

**Solución**

Vamos a crear una lista enlazada para poder desarrollar el sistema, crearemos un archivo main que será el encargado de inyectar la información para el programa, imprimirá la información que necesitemos. Crearemos la clase DiarioAventurero que llevará la lógica del programa, se creará la lista enlazada que contendrá la información de las misiones registradas y eliminadas, se mostrarán las misiones en orden de finalización, se podrá buscar una misión por su descripción y se editará una misión ya registrada. Tendremos también la clase Misión que manejara el estado de las misiones para mostrar una información u otra.

**Código**

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

**Main**

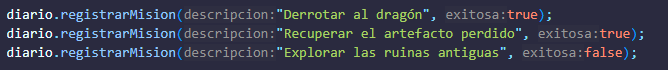
**Texto

El contenido generado por IA puede ser incorrecto.**

Se declara la clase Main y el método principal main() donde comienza la ejecución.



Se crea un objeto llamado diario de la clase DiarioAventurero



Se crean 3 misiones, una de ellas tiene el valor false ya que es la misión fallida.



Muestra todas las misiones

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Se crea una variable de tipo string que almacena el nombre de la misión que queremos buscar en el diario y se crea un condicional. Está condicional se encarga de verificar si la misión “recuperar el artefacto perdida” está registrada, si sí lo está, imprime que la misión ya fue completada, de lo contrario imprime el texto “no ha sido completada”.



Este método sirve para editar la misión y poner una misión nueva.



Imprime todas las misiones registradas



Mira si la misión fue fallida, si fue así, elimina la misión



Vuelve a mostrar todas las misiones pero si la misión anterior fue fallida, entonces fue eliminada, por lo que no se mostrará.

**DiarioAventurero**



Se importa LinkedList para almacenar las misiones

Texto

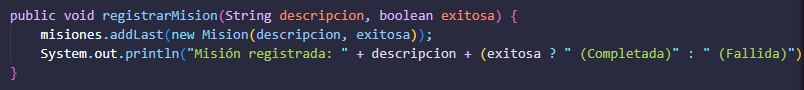
El contenido generado por IA puede ser incorrecto.

Se declara la lista enlazada que almacenará objetos de tipo misión.

Texto

El contenido generado por IA puede ser incorrecto.

Iniciailza la lista misiones como una nueva LinkedList vacia.



Crea una nueva misión y la agrega al final de la lista, se imprime un mensaje confirmando que la misión ya fue registrada, a esta misión se le agrega un valor booleano para manejar si fue completada o no. Dependiendo de este valor se a mostrar el mensaje “completada” o “fallida.

Texto

El contenido generado por IA puede ser incorrecto.

Se crea un condiciones en donde primero verifica si hay misiones en la lista, luego se obtiene la última misión, si está misión falló, entonces la elimina, si no hay misiones fallidas, entonces imprime un mensaje confirmando que la misión fue completada y si la lista está vacía, entonces imprimirá un mensaje diciendo que no hay misiones.

Texto

El contenido generado por IA puede ser incorrecto.

Se crea un condicional que verifica si la lista de misiones está vacía. Sí está vacía muestra el mensaje diciendo que no hay misiones registradas. De lo contrario, muestra un mensaje diciendo que sí hay misiones registradas e imprime las misiones que han sido registradas.

Texto

El contenido generado por IA puede ser incorrecto.

Recorre la lista para verificar y existe alguna misión con la misma descripción que le estamos pasando, si la misión está registrada, retorna true y si no lo está, entonces retorna false.

Texto

El contenido generado por IA puede ser incorrecto.

Se define un método que necesitará 2 parámetros; “descripciónAntigua” y “descripciónNueva”, se crea un bucle for para recorrer la lista de misiones, si existe una misión con la misma descripción entonces significa que encontramos la misión a editar y se le asigna la nueva descripción, luego retornará el nuevo dato anunciando que la modificación se realizó. Si no se encuentra una misión que coincida con la descripción, entonces se mostrará un mensaje diciendo que la misión no fue encontrada.

**Misión**

Una captura de pantalla de un celular con texto e imagen

El contenido generado por IA puede ser incorrecto.

Se declara la clase Misión y se declaran 2 atributos, uno almacena el nombre de la misión y el otro dice si la misión fue completada o no.

Texto

El contenido generado por IA puede ser incorrecto.

Se crea un constructor que inicializa una misión con su descripción y su estado.

Texto

El contenido generado por IA puede ser incorrecto.

Se devuelve la descripción de la misión y devuelve el estado de la misión, si fue completada o no.

Texto

El contenido generado por IA puede ser incorrecto.

Se cambia la descripción de la misión y también el estado de la misión.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Retorna la descripción de la misión y su estado; si la misión fue completada o no.

**Ejemplo de uso con capturas de pantalla o salida esperada**

Texto

El contenido generado por IA puede ser incorrecto.

**Conclusión sobre lo aprendido**

En este caso necesitamos una lista enlazada para poder ir agregando misiones a la misma, esta lista tiene la funcionalidad LIFO que significa que el último en entrar es el primer en salir, con este pudimos ir analizando cada misión que se agregaba y en caso de que la misión no estuviera completada, la misión era eliminada. En este caso la lista enlaza fue muy útil ya que las misiones se agregaban de último y eran analizadas, estas misiones tenían una descripción y un estado que se manejó por medio de booleanos, entonces era más fácil manejar CRUD.