



MANUAL TÉCNICO

Sopa de Letras

IPC1

CARNET 202407596

NOMBRE: Jefferson Ajcúc

I. INTRODUCCION

Este manual está diseñado para ofrecer una comprensión profunda sobre el funcionamiento del Sistema de Sopa de Letras, un proyecto desarrollado para simular y gestionar un juego clásico de búsqueda de palabras en una matriz de letras. A través de este proyecto, se aplican principios de programación orientada a objetos, estructuras de datos y algoritmos para crear una experiencia interactiva y dinámica para el usuario.

El sistema fue concebido para simular de manera precisa la generación de un tablero de sopa de letras, la inserción de palabras en diferentes direcciones (horizontal y vertical), y la búsqueda de estas palabras por parte del usuario. Mediante una interfaz de consola interactiva, se proporciona una visualización en tiempo real del tablero y del progreso del jugador, incluyendo la cantidad de palabras encontradas, las oportunidades restantes y la puntuación acumulada.

II. OBJETIVOS

El objetivo primordial de este manual es guiar a desarrolladores y usuarios avanzados en la implementación y uso del Sistema de Sopa de Letras, permitiendo su personalización o adaptación a diferentes entornos y necesidades. Además, se pretende que los interesados en programación puedan utilizar este proyecto como una herramienta de aprendizaje para implementar conceptos clave de programación orientada a objetos, manejo de estructuras de datos y algoritmos de búsqueda.

El objetivo funcional del sistema es proporcionar una simulación interactiva que permita la generación dinámica de un tablero de sopa de letras, la inserción y búsqueda de palabras en diferentes direcciones (horizontal y vertical), y la gestión del progreso del jugador a través de una interfaz de consola.

III. Dirigido

Este manual está orientado a desarrolladores de software, ingenieros de sistemas, y profesionales de TI interesados en entender a fondo el diseño y la lógica detrás del sistema. Está diseñado para aquellos que buscan mantener, ampliar o adaptar el sistema, así como para aquellos interesados en aprender más sobre cómo funciona el sistema detrás de el programa sopa de letras

IV. Especificación Técnica

Requisitos de Hardware:

- Computadora de escritorio o portátil.
- Al menos 4 GB de memoria RAM.
- Disco duro con al menos 250 GB de capacidad.
- Procesador Intel Core i3 o superior.
- Resolución gráfica mínima de 1024 x 768 píxeles.

Requisitos de Software:

- Sistema operativo: Windows 7 o superior.
- Java Runtime Environment (JRE) versión 8.2 o superior.
- Java Development Kit (JDK) versión 8.2 o superior.
- NetBeans IDE 8.2 o superior.
- Librerías internas requeridas:
 - java.awt
 - javax.swing
 - java.io
 - java.util

IV. LÓGICA DEL PROGRAMA

Nombre del Paquete	Descripción del Paquete	Clases en el Paquete
sopadeletras	Este paquete contiene la clase principal que inicia la aplicación y contiene todos los metodos del sistema sopa de letras.	Partida SopaDeLetras

CLASE: Partida

Esta clase es la principal del programa y se encarga de inicializar y lanzar la aplicación.

ATRIBUTOS Y VARIABLES

Atributos	Variables
String nombreJugador	nombreJugador
int puntos	puntos
int fallos	fallos
int palabrasEncontradas	palabrasEncontradas

Clase: Partida

La clase Partida es el núcleo de la gestión de datos en el juego de sopa de letras. Su función principal es almacenar y organizar la información relacionada con una partida individual, permitiendo el seguimiento del progreso del jugador y facilitando la visualización de los resultados.

1.

```
public class Partida {
    String nombreJugador;
    int puntos;
    int fallos;
    int palabrasEncontradas;

    public Partida(String nombreJugador, int puntos, int fallos, int palabrasEncontradas) {
        this.nombreJugador = nombreJugador;
        this.puntos = puntos;
        this.fallos = fallos;
        this.palabrasEncontradas = palabrasEncontradas;
    }

    @Override
    public String toString() {
        return "Jugador: " + nombreJugador + ", Puntos: " + puntos + ", Fallos: " + fallos + ", Palabras encontradas: " + palabrasEncontradas;
    }
}
```

CLASE: SopaDeLetras

La clase SopaDeLetras es el núcleo principal del programa que gestiona y controla el funcionamiento del juego de sopa de letras. Actúa como el controlador central que coordina todas las interacciones del usuario, desde la inserción y modificación de palabras hasta la ejecución de las partidas y la visualización de resultados.

ATRIBUTOS Y VARIABLES

Nombres de Variables	Tipo de Variable
vista	EstacionDeCarga
listaProductos	LinkedList<Producto>
controladorProduccion	ControladorProduccion

MÉTODO: mostrarMenuPrincipal

El método mostrarMenuPrincipal muestra el menú principal del juego, presentando opciones para iniciar una nueva partida, ver el historial de partidas, mostrar puntuaciones altas, ver información del estudiante o salir del programa. Luego, espera la selección del usuario, valida la entrada y ejecuta la opción correspondiente. Si la opción no es válida, solicita una nueva entrada. El proceso se repite hasta que el usuario elige salir.

```
private static void mostrarMenuPrincipal() {
    while (true) {
        System.out.println("=== MENÚ PRINCIPAL ===");
        System.out.println("1. Nueva Partida");
        System.out.println("2. Historial de Partidas");
        System.out.println("3. Mostrar Puntuaciones más Altas");
        System.out.println("4. Mostrar Información del Estudiante");
        System.out.println("5. Salir");
        System.out.print("Seleccione una opción: ");

        try {
            int opcion = scanner.nextInt();
            scanner.nextLine(); // Limpiar el buffer

            switch (opcion) {
                case 1:
                    nuevaPartida();
                    break;
                case 2:
                    mostrarHistorial();
                    break;
                case 3:
                    mostrarPuntuacionesAltas();
                    break;
                case 4:
                    mostrarInformacionEstudiante();
                    break;
                case 5:
                    System.out.println("Saliendo del programa...");
                    return;
                default:
                    System.out.println("Opción no válida. Intente de nuevo.");
            } catch (java.util.InputMismatchException e) {
                System.out.println("Error: Debes ingresar un número entero. Intente de nuevo.");
                scanner.nextLine(); // Limpiar el buffer para evitar un bucle infinito
            }
        }
    }
}
```

MÉTODO: nuevaPartida

El método nuevaPartida gestiona el inicio de una nueva partida, permitiendo al usuario ingresar su nombre y luego presenta un submenú con opciones para insertar palabras, jugar o terminar la partida. Dependiendo de la opción seleccionada, se redirige al usuario al menú de palabras para gestionar las palabras del juego o al inicio del juego. Si el usuario elige terminar la partida, se cierra el proceso y regresa al menú principal.

```
private static void nuevaPartida() {
    System.out.print("Ingrese su nombre: ");
    nombreUsuario = scanner.nextLine();
    System.out.println("Bienvenido, " + nombreUsuario + "!");

    while (true) {
        System.out.println("=== NUEVA PARTIDA ===");
        System.out.println("1. Menú Palabras");
        System.out.println("2. Jugar");
        System.out.println("3. Terminar Partida");
        System.out.print("Seleccione una opción: ");

        try {
            int opcion = scanner.nextInt();
            scanner.nextLine(); // Limpiar el buffer

            switch (opcion) {
                case 1:
                    menuPalabras();
                    break;
                case 2:
                    jugar();
                    break;
                case 3:
                    System.out.println("Terminando partida...");
                    return;
                default:
                    System.out.println("Opción no válida. Intente de nuevo.");
            }
        } catch (java.util.InputMismatchException e) {
            System.out.println("Error: Debes ingresar un número entero. Intente de nuevo.");
            scanner.nextLine();
        }
    }
}
```

MÉTODO: menuPalabras

El método menuPalabras presenta un submenú que permite al usuario gestionar las palabras del juego. Ofrece opciones para insertar, modificar, eliminar o mostrar las palabras ya ingresadas. Si elige la opción de insertar palabras, el usuario puede agregar nuevas palabras al banco. Si selecciona modificar, puede cambiar una palabra existente, y si opta por eliminar, puede borrar una palabra del banco. También puede ver todas las palabras almacenadas

```
private static void menuPalabras() {
    while (true) {
        System.out.println("=== MENÚ PALABRAS ===");
        System.out.println("1. Insertar Palabras");
        System.out.println("2. Modificar Palabras");
        System.out.println("3. Eliminar Palabras");
        System.out.println("4. Mostrar Palabras");
        System.out.println("5. Salir");
        System.out.print("Seleccione una opción: ");

        try {
            int opcion = scanner.nextInt();
            scanner.nextLine(); // Limpiar el buffer

            switch (opcion) {
                case 1:
                    insertarPalabras();
                    break;
                case 2:
                    modificarPalabras();
                    break;
                case 3:
                    eliminarPalabras();
                    break;
                case 4:
                    mostrarPalabras();
                    break;
                case 5:
                    System.out.println("Saliendo del menú de palabras...");
                    return;
                default:
                    System.out.println("Opción no válida. Intente de nuevo.");
            }
        } catch (java.util.InputMismatchException e) {
            System.out.println("Error: Debes ingresar un número entero. Intente de nuevo.");
            scanner.nextLine(); // Limpiar el buffer para evitar un bucle infinito
        }
    }
}
```

MÉTODO: insertarPalabras

El método insertarPalabras permite al usuario ingresar nuevas palabras al banco de palabras del juego. Primero solicita al usuario el número de palabras que desea agregar, y verifica si el número total de palabras no excede el límite máximo. Luego, para cada palabra, pide al usuario que la ingrese, valida que su longitud esté entre 3 y 8 caracteres, y la agrega al arreglo de palabras si es válida. Si una palabra no cumple con la longitud permitida, el método le pide al usuario que reintente.

```
private static void insertarPalabras() {
    System.out.print("Ingrese el número de palabras que desea insertar: ");
    try {
        int cantidad = scanner.nextInt();
        scanner.nextLine(); // Limpiar el buffer

        if (cantidadPalabras + cantidad > MAX_PALABRAS) {
            System.out.println("No se pueden agregar más palabras. Límite alcanzado.");
            return;
        }

        for (int i = 0; i < cantidad; i++) {
            System.out.print("Ingrese la palabra " + (i + 1) + ": ");
            String palabra = scanner.nextLine();
            if (validarLongitudPalabra(palabra)) {
                palabras[cantidadPalabras] = palabra.toUpperCase();
                cantidadPalabras++;
                System.out.println("Palabra agregada: " + palabra);
            } else {
                System.out.println("La palabra no cumple con la longitud permitida. Intente de nuevo.");
                i--; // Reintentar la misma posición
            }
        }
    } catch (java.util.InputMismatchException e) {
        System.out.println("Error: Debes ingresar un número entero. Intente de nuevo.");
        scanner.nextLine(); // Limpiar el buffer para evitar un bucle infinito
    }
}
```

MÉTODO: modificarPalabras

El método modificarPalabras permite al usuario modificar una palabra existente en el banco de palabras del juego. Primero, solicita al usuario que ingrese la palabra que desea modificar. Luego, busca esa palabra en el banco y, si la encuentra, pide al usuario que ingrese la nueva palabra. La nueva palabra se valida para asegurarse de que cumpla con la longitud permitida (entre 3 y 8 caracteres). Si es válida, la palabra se reemplaza en el banco de palabras.

```
private static void modificarPalabras() {

    System.out.print("Ingrese la palabra que desea modificar: ");
    String palabraVieja = scanner.nextLine().toUpperCase();
    int indice = buscarPalabra(palabraVieja);
    for (int i = 0; i <= indice; i++) {
        if (indice != -1) {
            System.out.print("Ingrese la nueva palabra: ");
            String palabraNueva = scanner.nextLine();

            if (validarLongitudPalabra(palabraNueva)) {
                palabras[indice] = palabraNueva.toUpperCase();
                System.out.println("Palabra modificada.");
                i++;
            } else {
                System.out.println("La nueva palabra no cumple con la longitud permitida.");
                i--;
            }
        } else {
            System.out.println("La palabra no se encuentra en el banco de palabras.");
            i--;
        }
    }
}
```


MÉTODO: eliminarPalabras

El método `eliminarPalabras` permite al usuario eliminar una palabra existente del banco de palabras del juego. Primero, solicita al usuario que ingrese la palabra que desea eliminar. Luego, busca esa palabra en el banco de palabras. Si la palabra se encuentra, se elimina desplazando las palabras siguientes hacia la izquierda para llenar el espacio vacío, y se actualiza el contador de palabras. Si la palabra no se encuentra en el banco, muestra un mensaje de error indicándolo.

```
private static void eliminarPalabras() {
    System.out.print("Ingrese la palabra que desea eliminar: ");
    String palabra = scanner.nextLine().toUpperCase();
    int indice = buscarPalabra(palabra);

    if (indice != -1) {
        // Eliminar la palabra moviendo las palabras restantes
        for (int i = indice; i < cantidadPalabras - 1; i++) {
            palabras[i] = palabras[i + 1];
        }
        cantidadPalabras--;
        System.out.println("Palabra eliminada.");
    } else {
        System.out.println("La palabra no se encuentra en el banco de palabras.");
    }
}
```

MÉTODO: mostrarPalabras

El método `mostrarPalabras` muestra todas las palabras almacenadas en el banco de palabras del juego. Si no hay palabras en el banco, muestra un mensaje indicando que está vacío. Si hay palabras, las imprime en la consola una por una. El método recorre el arreglo de palabras y las despliega, facilitando al usuario la visualización de las palabras actuales en el banco. Si el banco de palabras está vacío, el método muestra el mensaje: "No hay palabras en el banco."

```
private static void mostrarPalabras() {
    if (cantidadPalabras == 0) {
        System.out.println("No hay palabras en el banco.");
    } else {
        System.out.println("=== PALABRAS ===");
        for (int i = 0; i < cantidadPalabras; i++) {
            System.out.println(palabras[i]);
        }
    }
}
```

DIAGRAMA DE FLUJO



