

DOCUMENTACIÓN DEL CODIGO

By Alexis Geovanni Sánchez Hernández

Introducción

Desarrollo de una Aplicación de Gestión de Usuarios y Ordenamiento de Listas.

En este proyecto, se llevado a cabo el desarrollo de una aplicación de gestión de usuarios con una funcionalidad adicional de ordenamiento de listas. Nuestra aplicación combina aspectos de la programación de aplicaciones, la interfaz de usuario y la educación en algoritmos de ordenamiento. Este sistema se destaca por su versatilidad y capacidad para brindar a los usuarios la oportunidad de aprender y experimentar con diferentes algoritmos de ordenamiento, al mismo tiempo que proporciona una solución práctica para la gestión de datos de usuario.

Características

Características Clave de la Aplicación:

Gestión de Usuarios: La aplicación permite a los usuarios registrarse, iniciar sesión y gestionar su información personal, como su nombre de usuario, contraseña, número favorito y lugar de nacimiento.

Ordenamiento de Listas: Un componente fundamental de mi aplicación es la capacidad de ordenar listas de números utilizando varios algoritmos de ordenamiento. Los usuarios pueden comparar el rendimiento y la eficiencia de algoritmos como Burbuja, Burbuja Mejorada, Inserción, Quicksort y Mergesort.

Interfaz de Usuario Intuitiva: He diseñado una interfaz de usuario amigable que permite a los usuarios ingresar datos, realizar operaciones de ordenamiento y visualizar resultados de manera clara y efectiva.

Beneficios y Objetivos:

La aplicación proporciona a los usuarios una experiencia práctica para aprender sobre algoritmos de ordenamiento y cómo afectan el rendimiento en situaciones del mundo real.

Fomenta la educación y la comprensión de los conceptos de ordenamiento y eficiencia algorítmica.

Ofrece una solución práctica para la gestión de usuarios y datos personales.

ESTRUCTURA DE DATOS

Matrices

matriz_user_in_code: Esta matriz almacena los nombres de usuario (hasta 10 caracteres) de los usuarios registrados.

matriz_pass_in_code: En esta matriz se almacenan las contraseñas de los usuarios (exactamente 8 caracteres).

Listas

respuestas_numero: Almacena las respuestas numéricas a las preguntas de seguridad de los usuarios.

respuestas_hogar: Guarda las respuestas relacionadas con el lugar de nacimiento de los usuarios.

Variables

ultimo_indice: Un contador que rastrea el último índice utilizado en las matrices de usuarios y contraseñas.

username_entry: Un objeto para la entrada del nombre de usuario.

password_entry: Un objeto para la entrada de la contraseña.

numero_entry: Un objeto para la entrada de la respuesta numérica.

hogar_entry: Un objeto para la entrada de la respuesta de lugar de nacimiento.

FUNCIONES

Funciones Principales

Registrar(): Esta función registra a un nuevo usuario verificando varias condiciones, como el límite de usuarios, campos vacíos, duplicación de usuarios, longitud del usuario y validez de la contraseña. Luego, almacena la información en las matrices y en archivos JSON.

login(): Verifica si las credenciales ingresadas coinciden con las almacenadas en las matrices de usuarios y contraseñas. En caso de éxito, inicia sesión en la aplicación.

eliminar_usuario(): Permite eliminar un usuario específico después de verificar las respuestas de seguridad. Compacta las matrices y las listas para mantener la estructura de datos libre de espacios en blanco.

actualizar_usuario(): Permite actualizar la información de un usuario específico, como la contraseña, la respuesta numérica o la respuesta de lugar de nacimiento.

Funciones Secundarias

cargar_datos(): Carga los datos previamente almacenados en archivos JSON en las matrices y listas correspondientes. Se llama al inicio de la aplicación para cargar datos previos.

guardar_datos(): Guarda los datos actuales de las matrices y listas en archivos JSON para mantener la persistencia de la información entre sesiones.

fondo(): Esta función establece una imagen como fondo de la ventana de la aplicación. Requiere la biblioteca PIL para funcionar.

cargar_datos_desde_json(): Carga datos previamente guardados desde archivos JSON en las matrices `matriz_user_in_code` y `matriz_pass_in_code`. Esto se utiliza al inicio de la aplicación para cargar datos previamente registrados.

MANEJO DE ERRORES

El código implementa manejo de errores para las siguientes situaciones:

Registro:

Límite de usuarios alcanzado.

Campos vacíos en el registro.

Duplicación de nombres de usuario.

Longitud excesiva del nombre de usuario.

Contraseña incorrecta (debe contener 8 caracteres, al menos 1 mayúscula, 1 símbolo y 1 número).

Numero incorrecto, el numero debe ser un numero valido.

Lugar de Nacimiento incorrecto, el lugar de nacimiento debe contener solo letras.

Inicio de sesión:

Usuario no encontrado o contraseña incorrecta.

Eliminar Usuario:

Respuesta incorrecta, las respuestas de seguridad no coinciden con el usuario.

Usuario no encontrado.

Actualizar usuario:

Contraseña incorrecta (debe contener 8 caracteres, al menos 1 mayúscula, 1 símbolo y 1 número).

Numero incorrecto, el numero debe ser un numero valido.

Lugar de Nacimiento incorrecto, el lugar de nacimiento debe contener solo letras.

Usuario no encontrado.

Ordenamientos:

Ingresar solo números, ingresar números separados por comas.

Fondo:

Si no se encuentra la imagen o la librería se usara un color en especifico para el fondo.

VENTANAS

Menú Principal (menu_principal)

La ventana del menú principal es la pantalla de inicio de la aplicación y presenta opciones para iniciar sesión con una cuenta existente o registrar una nueva cuenta. A continuación, se detalla su funcionamiento:

Entradas: Nombre de usuario y contraseña (para iniciar sesión).

Salidas: Redirección a la ventana de registro o a la ventana de submenú, dependiendo de la opción seleccionada por el usuario.

Elementos de la Ventana del Menú Principal:

Etiquetas de Usuario y Contraseña:

Etiquetas que indican los campos de entrada de nombre de usuario y contraseña.

Campos de Entrada de Usuario y Contraseña:

Campos de entrada de texto donde el usuario ingresa su nombre de usuario y contraseña.

Botón "Iniciar Sesión":

Permite al usuario iniciar sesión con su cuenta existente llamando a la función login.

Botón "Registrar":

Redirige al usuario a la ventana de registro (ventana_registro).

Ventana de Fondo:

Se utiliza para establecer una imagen de fondo en la ventana principal.

Ventana de Registro (ventana_registro)

La ventana de registro permite a los usuarios crear una nueva cuenta proporcionando un nombre de usuario, una contraseña, un número favorito y un lugar de nacimiento. A continuación, se detalla su funcionamiento:

Entradas: Nombre de usuario, contraseña, número favorito y lugar de nacimiento.

Salidas: Registro de un nuevo usuario y redirección a la ventana de inicio de sesión (menu_principal).

Elementos de la Ventana de Registro:

Etiquetas de Usuario, Contraseña, Número Favorito y Lugar de Nacimiento:

Etiquetas que indican qué información debe proporcionar el usuario.

Campos de Entrada de Usuario, Contraseña, Número Favorito y Lugar de Nacimiento:

Campos de entrada de texto donde el usuario ingresa la información requerida.

Botón "Registrar":

Al hacer clic en este botón, se recopilan los datos ingresados por el usuario y se realiza el proceso de registro. Llama a la función registrar.

Botón "Volver al Menú Principal":

Permite al usuario regresar al menú principal sin registrarse.

Ventana Submenú (ventana_submenu)

Esta función crea una ventana de submenú en la interfaz de usuario. La ventana muestra botones que permiten al usuario acceder a diferentes secciones de la aplicación.

Entradas: Ninguna

Salidas: Ventana del submenú en la interfaz gráfica.

Elementos de la Ventana Submenú

Botones:

"Datos": Abre la ventana de datos (función ventana_datos).

"Ordenamientos": Permite acceder a una sección para realizar ordenamientos (función `ventana_ordenamientos`).

"Arboles": Permite acceder a una sección para los arboles (función `ventana_arboles`).

"Volver": Regresa al menú principal de la aplicación (función `menu_principal`).

Ventana Datos (`ventana_datos`)

La ventana de datos muestra información de los usuarios registrados en la aplicación, como los nombres de usuario, contraseñas y respuestas a preguntas de seguridad. También proporciona botones para eliminar, actualizar y cargar datos desde archivos JSON.

Entradas: Ninguna

Salidas: Ventana de datos en la interfaz gráfica.

Elementos de la Ventana Datos

Matrices de Usuarios y Contraseñas:

Muestra las matrices `matriz_user_in_code` y `matriz_pass_in_code` en una tabla, donde cada fila corresponde a un usuario registrado. Las contraseñas se muestran en forma oculta.

Respuestas de Seguridad:

Muestra las respuestas de seguridad de los usuarios en columnas separadas para respuestas numéricas y respuestas de lugar de nacimiento.

Botones:

"Eliminar Usuario": Abre la ventana para eliminar un usuario específico (función `ventana_eliminar`).

"Actualizar Usuario": Abre la ventana para actualizar los datos de un usuario (función `ventana_actualizar`).

"Cargar Datos": Permite cargar datos desde archivos JSON (función `cargar_datos_desde_json`).

"Volver": Regresa a la ventana de submenú (función `ventana_submenu`).

Ventana Eliminar Usuario (`ventana_eliminar`)

Esta ventana permite al usuario eliminar un usuario específico proporcionando el nombre de usuario y respuestas de seguridad.

Entradas: Nombre de usuario, número favorito y lugar de nacimiento.

Salidas: Eliminación del usuario y actualización de datos en la interfaz gráfica.

Elementos de la Ventana Eliminar Usuario

Entradas de Usuario:

Campos de entrada de texto para el nombre de usuario, número favorito y lugar de nacimiento.

Botón "Eliminar": Permite al usuario eliminar el usuario específico proporcionado.

Uso de Tkinter

La función utiliza la biblioteca Tkinter para crear y gestionar la interfaz gráfica.

Campos de entrada de texto se crean utilizando la clase Entry de Tkinter.

El botón de eliminación se crea utilizando la clase Button de Tkinter y está vinculado a la función eliminar_usuario.

Ventana Actualizar Usuario (ventana_actualizar)

La ventana permite al usuario actualizar la información de un usuario proporcionando un nuevo nombre de usuario, nueva contraseña, nuevo número favorito y nuevo lugar de nacimiento.

Entradas: Nombre de usuario, nueva contraseña, nuevo número favorito y nuevo lugar de nacimiento.

Salidas: Actualización de los datos del usuario y actualización de datos en la interfaz gráfica.

Elementos de la Ventana Actualizar Usuario

Entradas de Usuario:

Campos de entrada de texto para el nombre de usuario, nueva contraseña, nuevo número favorito y nuevo lugar de nacimiento.

Botón "Actualizar": Permite al usuario actualizar la información del usuario proporcionado.

Ventana Ordenamientos(ventana_ordenamientos)

Crea una ventana que permite a los usuarios ordenar listas de números utilizando diferentes métodos de ordenamiento, como Burbuja, Burbuja Mejorada, Inserción, Quicksort y Mergesort. Aquí se proporciona una descripción de cada sección de esta ventana:

Sección Burbuja:

Etiqueta "Ingresa los números a ordenar separados por comas": Proporciona instrucciones para ingresar la lista de números.

Campo de entrada: Permite al usuario ingresar la lista de números separados por comas.

Etiquetas de resultados: Muestran la lista ordenada, la lista original y la información sobre el proceso de ordenamiento.

Botón "Ordenar": Inicia el proceso de ordenamiento utilizando el algoritmo de Burbuja.

Botón "Volver": Permite volver al menú principal.

Sección Burbuja Mejorada:

Similar a la sección de Burbuja, pero utiliza el algoritmo de Burbuja Mejorada para el ordenamiento, utilizando la bandera para optimizar la iteración y conteo de consultas mejorando al código de burbuja.

Sección Inserción:

Etiqueta "Ingresa los números a ordenar separados por comas": Proporciona instrucciones para ingresar la lista de números.

Campo de entrada: Permite al usuario ingresar la lista de números separados por comas.

Etiquetas de resultados: Muestran la lista ordenada, la lista original y la información sobre el proceso de ordenamiento.

Botón "Ordenar": Inicia el proceso de ordenamiento utilizando el algoritmo de Inserción.

Botón "Volver": Permite volver al menú principal.

Sección Quicksort:

Etiqueta "Ingresa los números a ordenar separados por comas": Proporciona instrucciones para ingresar la lista de números.

Campo de entrada: Permite al usuario ingresar la lista de números separados por comas.

Etiquetas de resultados: Muestran el tiempo de ejecución, la lista ordenada y la lista original.

Botón "Ordenar": Inicia el proceso de ordenamiento utilizando el algoritmo de Quicksort.

Botón "Volver": Permite volver al menú principal.

Sección Mergesort:

Etiqueta "Ingresa los números a ordenar separados por comas": Proporciona instrucciones para ingresar la lista de números.

Campo de entrada: Permite al usuario ingresar la lista de números separados por comas.

Etiquetas de resultados: Muestran la lista ordenada, la lista original y la información sobre el proceso de ordenamiento.

Botón "Ordenar": Inicia el proceso de ordenamiento utilizando el algoritmo de Mergesort.

Botón "Volver": Permite volver al menú principal.

Cada sección contiene campos de entrada para que el usuario ingrese la lista de números a ordenar. Cuando se hace clic en el botón "Ordenar", se ejecuta el algoritmo de ordenamiento correspondiente y se muestran los resultados en las etiquetas de resultados.

La funcionalidad y la interfaz de estas secciones son similares, con la diferencia de que cada una utiliza un algoritmo de ordenamiento diferente. El código también maneja casos de entrada incorrecta y muestra mensajes de error cuando es necesario.

En resumen, esta ventana proporciona una forma interactiva de probar diferentes algoritmos de ordenamiento y ver cómo funcionan en una lista de números específica.

Requisitos Adicionales

Uso de PIL

Para que la función `fondo()` funcione correctamente, es necesario instalar la biblioteca PIL (Python Imaging Library) mediante `from PIL import Image, ImageTk`. Asegúrate de que esta biblioteca esté instalada en tu entorno de desarrollo.

Uso de Tkinter

La aplicación utiliza la biblioteca Tkinter para crear y gestionar la interfaz gráfica.

Campos de entrada de texto se crean utilizando la clase `Entry` de Tkinter.

Los botones se crean utilizando la clase `Button` de Tkinter y está vinculado a sus respectivas funciones.