

Laboratorio

Práctica 6. Tipos de datos estructurados en Python

Unidad 5. Tipos de datos estructurados

1. Objetivos

- Aprender a hacer programas que utilicen los diferentes objetos de datos estructurados que ofrece Python: cadenas, tuplas, listas y diccionarios.
- Conocer y utilizar los métodos (funciones) asociados a los datos estructurados en Python.
- Consolidar el desarrollo de programas en Python siguiendo las directrices de la programación modular utilizando funciones.
- Practicar la interacción de los programas con archivos de texto para leer y escribir datos.

2. Introducción

Esta práctica consiste en el desarrollo de un programa que implemente una red social. El programa debe cargar las bases de datos suministradas y, a partir de estas, implementar la funcionalidad básica de una red social y contar con persistencia de datos.

2.1. Requerimientos del programa

Requisitos funcionales

El programa a desarrollar debe ofrecer al usuario un menú con las siguientes opciones:

Registrarse en la red social: aquí se le debe pedir al usuario que ingrese su nombre y contraseña. Se debe verificar que el nombre sea solo texto y que no haya sido registrado previamente en la red social.

Iniciar sesión: para iniciar sesión el usuario debe suministrar su nombre y contraseña. Se debe verificar que el nombre y la contraseña correspondan a un usuario registrado; si no corresponden, se le debe indicar al usuario que sus datos están errados. Una vez el usuario inicie sesión, se le debe informar sobre la cantidad de amigos, solicitudes de amistad y mensajes que tiene; además, se le debe presentar un menú con las siguientes opciones:

- **Ver usuarios registrados:** el usuario puede ver los nombres de los usuarios registrados en la red social.

- **Enviar una solicitud de amistad:** el usuario puede enviar una solicitud de amistad a alguno de los usuarios registrados. Para preservar la integridad de las bases de datos, se debe tener en cuenta las siguientes restricciones en la funcionalidad:
 - a. No se puede enviar una solicitud de amistad a usuarios que no están registrados en la red social.
 - b. No se puede una solicitud de amistad a usuarios que estén en su lista de amigos.
 - c. No se puede enviar una solicitud de amistad a otros usuarios más de una vez.
 - d. No se puede enviar una solicitud de amistad a un usuario que haya solicitado ser su amigo. Lo más lógico es aceptar la solicitud que le han hecho.
 - e. El usuario no se puede enviar una solicitud de amistad a sí mismo.
- **Ver solicitudes de amistad pendientes:** con esta opción se puede ver qué usuarios le han enviado una solicitud de amistad. También se debe ofrecer la funcionalidad de aceptar alguna de las solicitudes disponibles; en este sentido, si se acepta una solicitud, la misma debe ser eliminada del listado de solicitudes pendientes, y los dos usuarios (el que envió la solicitud y el que la aceptó) deben ser catalogados como amigos.
- **Ver mensajes:** esta funcionalidad le permite al usuario ver los mensajes que le han enviado sus amigos ordenados cronológicamente a partir del más reciente.
- **Enviar mensaje a un amigo:** con esta opción el usuario puede escribirle un mensaje a alguno de sus amigos. Se debe tener en cuenta que solo se puede enviar mensajes a usuarios registrados en la red social y que sean amigos de quien pretende enviar el mensaje. El mensaje debe contener fecha y hora (dia/mes/año, hora:minuto:segundo), además del nombre del usuario que envió el mensaje.
Ayuda: investigar el módulo datetime de Python.
- **Cerrar sesión:** al seleccionar esta opción, se debe cerrar la sesión iniciada y el programa debe regresar al menú principal.

Salir del programa: al elegir esta opción el usuario termina la ejecución de la aplicación.

Importante: cualquier cambio en la información de los usuarios debe quedar registrada en las bases de datos respetando los formatos establecidos.

Bases de datos

Se le entregan los siguientes archivos: users.txt y userData.txt

El archivo users.txt contiene información sobre los nombres de los usuarios registrados y sus contraseñas en el siguiente formato.

nombre;contraseña

En el archivo userData.txt se encuentra la información sobre amigos, solicitudes de amistad, intereses y mensajes de cada usuario registrado en la red social en el siguiente formato:

```
*nombre:amigo1,amigo2,...,amigon,<solicitud1,solicitud2,...,solicitudn>
{gusto1,gusto2,...,guston}
mensaje1
mensaje2
.
.
.
mensajen
```

Nota: Inspeccione los archivos manualmente para familiarizarse con su formato.

Requisitos no-funcionales

El programa desarrollado debe cumplir con los siguientes:

- Utilizar funciones en el desarrollo del programa, a criterio del programador, con sus respectivas especificaciones a manera de docstrings.
- Crear dos módulos (archivos fuente), uno para el programa principal que implementará los menús para la interacción con el usuario, y otro para las funciones que se encargarán de la administración de la información de las bases de datos.
- Utilizar strings, tuplas, listas y diccionarios.

2.2. Etapas de desarrollo

Sesión 1:

Al finalizar la primera sesión de laboratorio, cada grupo debe entregar al profesor una hoja con un diagrama libre que indique la organización que le desea dar a los datos en el programa: tipos de variables, jerarquías, contenidos, etc. Este es el insumo principal para poder iniciar con el desarrollo del código; además se debe entregar un diagrama de flujo que muestre la funcionalidad general del programa.

Sesión 2:

Al finalizar la segunda sesión de laboratorio, cada grupo debe tener listas las funciones necesarias para cargar en las variables los datos de los archivos y para el proceso inverso, es decir, poder guardar en los archivos todo lo que esté en las variables respetando el formato definido.

Sesión 3:

Durante la tercera sesión se completan las funcionalidades restantes del programa.

Opcional

Agregue una opción al menú de inicio de sesión en donde el usuario pueda solicitar un listado de usuarios que tengan intereses similares a los suyos (dos o más). De esta manera, el usuario puede enviarle solicitudes de amistad a usuarios con intereses afines; para ello, se debe tener en cuenta que:

1. Se debe agregar una nueva opción donde el usuario pueda ingresar sus gustos, estos se deben guardar en la base de datos userData.txt respetando el formato establecido.
2. Esta opción solo debe mostrar otros usuarios compatibles que aún no son amigos del usuario.

3. Evaluación

La evaluación se basará en los códigos enviados y una sustentación oral sobre los temas de la práctica. Además, se tendrá en cuenta una bonificación para quienes haga la parte opcional.