

SISTEMA DE MENSAJERIA: LISTAS ENLAZADAS Y ORDENAMIENTO

Maria C. Torres Madroñero

Profesora asociada

Departamento de Ciencias de la Computación y la Información

Objetivos

- Implementar una solución algorítmica empleando colecciones de objetos, haciendo uso de la estructura de datos lista simple y lista doble

Recursos requeridos

- PC
- IDE para JAVA o Python – el estudiante deberá seleccionar un lenguaje de programación para el desarrollo de las prácticas de laboratorio

Actividades preliminares al laboratorio

- Lectura de la guía

Marco teórico

- Revisar la documentación de la guía del laboratorio 4

Descripción del Problema

Una empresa requiere un sistema de mensajería electrónica interna, que solo sea empleada por sus empleados. El sistema de mensajería debe cumplir las siguientes especificaciones técnicas:

1. Los empleados de la empresa se encuentran almacenados en un archivo de texto denominado "Empleados.txt"; en este archivo, se almacena la información personal (nombre, cedula, fecha de nacimiento, ciudad de nacimiento) y de contacto (teléfono, correo electrónico y dirección (calle, nomenclatura, barrio, ciudad, urbanización, número de apartamento)). Al iniciar el sistema debe cargar los empleados de la empresa desde el archivo de texto (ver archivo empleados.txt de ejemplo).
2. Los empleados permanecen ordenados en la colección de usuarios de acuerdo con su número de identificación de menor a mayor. El sistema debe permitir organizar los usuarios de acuerdo con el número de identificación cada vez que se ingresa un nuevo usuario.
3. Para acceder al sistema, cada empleado tiene una contraseña, conformada por número y letras. Las contraseñas de cada usuario se almacenan en un archivo de texto "Password.txt". Este archivo incluye en cada línea: la cedula de ciudadanía de un usuario, su contraseña, y una descripción si el usuario es "empleado" o "administrador". Para acceder al sistema, el usuario debe digitar su número de identificación y contraseña, y el sistema debe verificar con la información de este archivo si los datos son correctos (ver archivo Password.txt de ejemplo).
4. El "administrador" del sistema tiene la posibilidad de realizar las mismas funcionalidades que los empleados, pero además puede, registrar nuevos usuarios al sistema, cambiar contraseñas, y eliminar usuarios. Cuando se realicen este tipo de cambios en la información de los usuarios, se debe actualizar el archivo Empleados.txt y Password.txt, manteniendo la consistencia entre ambos.
5. El "empleado" puede revisar su bandeja de entrada y enviar mensajes a usuarios del sistema.
6. La bandeja de entrada de cada empleado mantiene la colección de mensajes electrónicos que ha recibido un usuario. Al consultar la bandeja de entrada por consola, esta debe presentar la fecha de recepción, el título del mensaje y el nombre de la persona que lo envió. Adicionalmente, debe permitir seleccionar cual mensaje se desea leer (tip: se recomienda implementar la bandeja de entrada como una lista doble).
7. Cuando un usuario quiere redactar un nuevo mensaje, por consola, se debe pedir:
 - Cedula del usuario al que se envía el mensaje

- Título del mensaje
- Mensaje
- Además de la información que redacta el usuario, cada mensaje de forma automática se le asigna la fecha y hora de envió.
- 8. Cuando un empleado envía un mensaje, el sistema lo agrega a la bandeja de entrada del destinatario correspondiente.
- 9. Los mensajes en la bandeja de entrada se almacena en archivos de texto, denominados por el número de cedula del empleado seguido por las letras BA para la bandeja de entrada.

Sugerencia: la lectura y escritura de los archivos de texto se sugiere solo se realice una vez, al principio del programa y una vez se vaya a finalizar la ejecución.

Actividades

- Identifique, diseñe e implemente las clases que requiere para resolver el problema. Debe hacer uso de las clases DoubleNode, DoubleList, Usuario, Fecha, y Direccion desarrolladas en los laboratorios anteriores. El propósito de este laboratorio es que no modifique ninguna de estas clases, sino que las utilice para solucionar el problema.
- Cree una colección de usuarios con una lista doble, que le permita realizar las operaciones de buscar usuario, eliminar usuario, editar usuario (cambio de contraseña), y agregar nuevos usuarios. La clase que maneja los usuarios debe permitir ordenar por número de identificación de menor a mayor; para esto implemente uno de los algoritmos de ordenamiento estudiados en clase.
- Para cada usuario se debe permitir crear una bandeja de entrada, es decir, una colección de correos electrónicos. Esta colección debe implementarse como una lista doble, en al cual se podrá agregar, eliminar y leer un correo.

Pruebas sugeridas para la presentación de la implementación:

- Ingresar como administrador al sistema, crear un nuevo usuario (se verifica que queden en orden los usuarios en la colección), eliminar un usuario, cambiar contraseña
- Ingresar como empleado al sistema solo permite enviar y leer mensajes
- Crear un nuevo mensaje desde un empleado y enviar el mensaje a otro empleado
- Consultar la bandeja de entrada de un empleado
- Leer un mensaje
- Revisar los archivos de textos generados por el programa

Instrucciones de entrega

- La solución de los problemas debe desarrollarse en JAVA o Python. Los estudiantes tendrán la libertad de seleccionar el lenguaje de programación y plataforma para presentar la solución de los problemas.
- La solución debe emplear librerías nativas y se invita a los estudiantes a no usar código descargado de internet. Los laboratorios están diseñados para practicar los fundamentos teóricos; entre más código escriba el estudiante más fácil será su comprensión de los temas de clase.
- La solución se puede presentar en grupos de hasta 3 estudiantes.
- La solución de los problemas debe entregarse y sustentarse en el aula de clase o en la hora de asesoría a estudiantes. No se reciben soluciones por correo electrónico.
- La fecha máxima de entrega es el 3 de abril 2024 durante la sesión práctica.