**PROYECTO FINAL**

**ESTRUCTURA DE DATOS Y LABORATORIO**

**Maria C. Torres Madroñero**

**Objetivos**

* Diseñar una solución algorítmica basada en programación orientada a objetos
* Implementar una solución algorítmica empleando colecciones de objetos, seleccionando la estructura de datos más eficiente computacionalmente, de acuerdo con las especificaciones del problema
* Probar la implementación, realizando operaciones que evidencien el funcionamiento correcto y robustes del desarrollo

**Recursos requeridos**

* PC con NetBeans instalado
* Librería de clases implementadas en los laboratorios
  + Fecha
  + Hora
  + Direccion
  + Usuario
  + Node
  + List
  + DoubleNode
  + DoubleList
  + Queue
  + Stack
  + BinaryTree
  + BST

**DESCRIPCIÓN DEL PROBLEMA – SISTEMA DE MENSAJERIA ELECTRÓNICA**

Una empresa requiere un sistema de mensajería electrónica interna, que solo sea empleada por sus empleados. El sistema de mensajería requerido 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.

2. 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.

3. 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.

4. El “empleado” puede revisar su bandeja de entrada, los mensajes leídos, proyectar mensaje que se mantienen en borradores, y enviar mensajes a usuarios del sistema.

5. La bandeja de entrada de cada empleado mantiene la colección de mensajes electrónicos que aún no se han leído. Al consultar la bandeja de entrada por consola, esta debe presentar la fecha de recepción, el titulo del mensaje y el nombre de la persona que lo envió. Adicionalmente, debe permitir seleccionar cual mensaje se desea leer.

5. Una vez leído un mensaje electrónico se elimina de la “bandeja de entrada” y pasa a la carpeta de mensajes leídos. El empleado puede consultar su carpeta de mensajes leídos similar a la bandeja de entrada.

6. 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. El empleado tiene las opciones de guardar como borrador, descartar o enviar el mensaje.

7. Cuando un empleado decide almacenar como borrador un mensaje de texto, el sistema almacena el mensaje en una colección, donde el empleado solo puede acceder al último borrador. Para acceder a otro mensaje, el empleado debe descartar el mensaje actual o enviarlo.

8. Cuando un empleado decide descartar el mensaje, este se elimina del sistema.

9. Cuando un empleado envía un mensaje, el sistema lo agrega a la bandeja de entrada del destinatario correspondiente.

10. Los mensajes en la bandeja de entrada, mensajes leídos, y borradores de cada usuario se almacena en archivos de texto, denominados por el número de cedula del empleado, seguido por las letras BA para la bandeja de entrada, ML para los mensajes leídos y B para los borradores.

**Actividades**

Problema 1 Diseño de solución algorítmica empleando programación orientada a objetos

Diseñe las clases requeridas para solucionar el problema. Presente los diagramas UML para todas las clases que considere necesario para la solución. Recuerde que puede asumir implementadas las clases desarrolladas durante los laboratorios. Su implementación debe cumplir con las siguientes características:

1. Modularidad: identifique los elementos básicos que requiere el sistema para su diseño
2. Reusabilidad: emplee las clases implementadas en los laboratorios (ejemplo: Usuarios, fecha, hora, listas dobles, etc).
3. Emplee al menos una lista doble
4. Emplee al menos una de estas estructuras: pila, cola o arboles binarios

Entrega 1: Presente un borrador de su diseño en diagramas UML de su solución para el viernes 26 de mayo

Problema 2 Implementación en JAVA

Realice la implementación en JAVA de su diseño. Este debe cumplir con todos los requerimientos mencionados anteriormente. Para la presentación de su implementación cree al menos 5 empleados donde uno de ellos sea el administrador.

Problema 3 Prueba del desarrollo

Realice las pruebas necesarias para verificar las siguientes acciones dentro del sistema

1. Cargue correcto de la información desde Empleados.txt
2. Cargue correcto de la información desde Password.txt
3. Búsqueda de un usuario registrado
4. Creación de un nuevo usuario
5. Eliminación de un nuevo usuario
6. Envió de mensajes entre usuarios
7. Creación de borradores de mensaje de un usuario
8. Lectura de la bandeja de entrada y mensajes leídos de un usuario
9. Escritura correcta de los diferentes archivos de texto

La solución de este proyecto final DEBE presentarse en grupos de hasta 4 estudiantes. Cada grupo deberá presentar su proyecto final y SUSTENTARLO. A continuación, se describen las fechas importantes y la distribución de notas con respecto al Examen Final:

|  |  |  |
| --- | --- | --- |
| Fecha importante | Actividad | Porcentaje Examen Final (20%) |
| Viernes 26 de mayo | Borrador diseño de implementación – diagramas de clase (se revisa en clase) | N/A |
| Martes 6 de junio | Entrega diseño final de diagramas de clase | 5% |
| Entrega proyecto en JAVA | 5% |
| Miércoles 7 de junio | Examen final – teórico | 10% |

Los grupos que obtengan una nota de 5.0 en el proyecto no deberán presentar el examen final teórico.