

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Estructuras de Datos

Catedrático:

Ing. Edgar Rene Ornelis Hoil

Tutor académico:

Elian Saúl Estrada Urbina



“MANUAL TECNICO PROYECTO 1”

Melvin Eduardo Ordoñez Sapón RA | 3256215860802

Guatemala, diciembre del 2024

“Id y enseñad a todos”

ACERCA DEL PROGRAMA:

IDE utilizado: Clion

Versión de GCC: 13.3.0

Programado en Sistema Operativo: Ubuntu 24.04 LTS

Terminal Utilizada: Terminal de Linux

Lenguaje de Programación Utilizado: C++

Compatibilidad: Windows, Linux, macOS

Versión Del Programa: 1.0

MÉTODOS IMPORTANTES DEL PROGRAMA

MATRIZ DISPERSA:

Para la matriz dispersa los metodos mas importantes tener en cuenta son los siguientes:

```
NodoMatriz *nodoCabeceraHorizontal;  
// cabecera vertical  
NodoMatriz *nodoCabeceraVertical;
```

Estos dos nodos seran los iniciales y su funcion es representar a los diferentes departamentos ingresados así como las empresas

```
void insertarValor(string valor, string cabezaHorizontal, string cabezaVertical);
```

Este metodo es utilizado para insertar un nuevo nodo ya sea en un departamento y empresa existente o en nuevos nodos

```
NodoMatriz *llegarCabeceraHorizontal(NodoMatriz *nodo);
```

Con este metodo dependiendo de donde este insertado el usuario lo que hara es ir recorriendo su apuntador arriba hasta que este llegue al ultimo y asi poder llegar al departamento que seria el ultimo nodo

```
NodoMatriz *llegarCabeceraVertical(NodoMatriz *nodo);
```

Con este metodo se puede llegar a conocer cual es la empresa asociada al usuario, ya que su funcion es ir recorriendo su apuntador anterior hasta que este llegue al ultimo.

```
void graficarMatriz(const string& nombreArchivo);
```

Este metodo es muy importante ya que su funcion es crear el reporte de la matriz completa, en donde se podran visualizar todas las empresas, departamentos y usuarios existentes.

ARBOL AVL:

Este árbol es importante ya que servirá para almacenar todos los activos de un usuario existente y sus principales metodos son los siguientes.

```
void rotacionIzquierda(NodoAVL *&nodo); // Rotación simple  
void rotacionDerecha(NodoAVL *&nodo); // Rotación simple
```

Estos métodos son para realizar los ordenamientos entre nodos, hacia la derecha e izquierda simple, de igual forma se pueden combinar para obtener las rotaciones dobles

```
//Para recorrer el arbol  
void recorreArbol(NodoAVL *raiz);
```

Este método se utiliza para recorrer todos los nodos de un árbol, esto es importante ya que gracias a el se pueden generar diferentes reportes

```
void graficarArbol();
```

Con este método se gráfica lo que es el árbol AVL que cada nodo representa aun activo de un usuario, se mostrara todos los nodos con su id y nombre del activo

LISTA DOBLE ENLAZADA

```
NodoDoble<T>* inicio;  
NodoDoble<T>* fin;  
int tamano;  
  
// Métodos auxiliares  
NodoDoble<T>* obtenerNodoAvanzando(int index) const;  
NodoDoble<T>* obtenerNodoRetrocediendo(int index) const;
```

Estos métodos son importante ya que por medio de su tamaño y también el atributo siguiente y anterior del nodo se puede recorrer toda la lista, esta lista esta parametrizada para que se puedan crear diferentes listas a utilizar en el programa

CONTROLADOR

Esta es la clase encargada de controlar toda la lógica del sistema, ya que aquí se implementa todo lo antes mencionado, lista doble, árbol avl, matriz dispersa el controlador tiene un método principal que es el siguiente y es utilizado por la clase main que se encarga de llamar a este metodo

```
void iniciarSistema();
```