

Manual Tecnico

Lenguaje: c++

Versión: 23

Compilador: CMake

Version Compilador: cmake versión 3.28.3

IDE: CLion

IDE versión: 2024.3

Sistema Operativo: Ubuntu 24.04.1 LTS

Estructuras MATRIZ DISPERSA

Métodos

```
MatrixNode *searchDepartment(string department);
```

Busca y devuelve un nodo con el departamento ingresado en el parámetro.

```
MatrixNode *searchCompany(string company);
```

Busca y devuelve un nodo con la empresa ingresada como parámetro

```
MatrixNode *search(string key, MatrixNode *department, MatrixNode  
* company);
```

Busca y retorna un nodo con la llave pasada como parámetro y las cabeceras del departamento y la empresa.

```
void insertAtTheEnd(MatrixNode *user, MatrixNode *department,  
MatrixNode *company);
```

Inserta un nodo al final de las cabeceras ingresadas en los parametros

```
void insertAtTheEndOfDepartment(MatrixNode *user, MatrixNode  
*department);
```

Inserta un nodo al final de la cabecera del departamento ingresado

```
void insertAtTheEndOfCompany(MatrixNode *user, MatrixNode  
*company);
```

Inserta un nodo al final de la cabecera de la empresa ingresada en el parametro

```
void insertAtTheHalfV( MatrixNode *user, MatrixNode *vertical);
```

Inserta un nuevo nodo entre la cabecera vertical indicada

```
void insertAtTheHalfH( MatrixNode *user, MatrixNode *horizontal);
```

Inserta un nuevo nodo entre la cabecera horizontal indicada

```
void insertBack(MatrixNode *user, MatrixNode *forward);
```

Inserta un nodo detras de otro

```
MatrixNode *insertHHeader(string department);
```

Crea un nueva cabecera departamento con el nombre indicado

```
MatrixNode *insertVHeader(string company);
```

Crea una nueva cabecera de empresa con el nombre ingresado

```
MatrixNode *goToDepartment(MatrixNode *node);
```

Devuelve la cabecera departamento del nodo indicado

```
MatrixNode *goToCompany(MatrixNode *node);
```

Devuelve la cabecera empresa del nodo indicado

```
MatrixNode *goToBack(MatrixNode *node, string useName);
```

Devuelve el ultimo nodo detras de un usuario con el nombre de usuario indicado

```
bool isEmpty();
```

Válida si la matriz está vacía

```
bool below(MatrixNode *company, string key);
```

Indica si la empresa buscada se encuentra mas abajo

```
bool futherRight(MatrixNode *department, string key);
```

Indica si el departamento buscado se encuentra mas a la derecha

```
void insertUser(User *user, string key, string department, string company);
```

Inserta un nuevo nodo usuario en las cabeceras indicadas

```
User *getUser(string name, string password, string department, string company);
```

Retorna un usuario en esas cabeceras con esa contraseña y es nombre de usuario

```
User *getUser(string name, string department, string company);
```

Retorna un usuario con ese nombre en esas cabeceras

```
MatrixNode *getHHeader();
```

Retorna la cabecera horizontal de la matriz

```
MatrixNode *getVHeader();
```

Retorna la cabecera vertical de la matriz

Arbol AVL

```
int maxHeight(AvlNode *node);
```

Calcula y retorna la altura maxima de un nodo

```
int calcBalanceFactor(AvlNode *node);
```

Calcula y retorna el factor de equilibrio de un nodo

```
void insert(AvlNode *value , AvlNode* &root);
```

Inserta un nuevo nodo en otro nodo raíz indicado

```
void turnRight(AvlNode *& node);
```

Hace una rotación simple hacia la derecha en un nodo indicado

```
void turnRightLeft(AvlNode *& node);
```

Hace un rotación doble derecha izquierda en un nodo indicado

```
void turnLeft(AvlNode *& node);
```

Hace un rotación simple hacia la derecha en el nodo indicado

```
void turnLeftRight( AvlNode *& node);
```

Hace un rotación doble izquierda derecha en un nodo indicado

```
void deleteAsset(string id, AvlNode *& root);
```

Elimina un activo a partir de su id y nodo en donde buscarlo

```
AvlNode *furtherRight(AvlNode *& node);
```

Devuelve el nodo más a la derecha de un nodo indicado

```
bool isLeaf(AvlNode *node);
```

Indica a través de un booleano si el nodo indicado es una hoja

```
void printTree(AvlNode *root);
```

Imprime la información del activo guardada en el nodo

```
Asset *getAsset(string id, AvlNode *& root);
```

Obtiene un activo con el id buscado en el nodo indicado