RENTA DE ACTIVOS

2016-02516 BRYAN CRISTHOPHER IGNACIO XOY

INTRODUCCION:

Este manual tiene como propósito describir los aspectos técnicos informáticos de la aplicación. Tiene la información del funcionamiento y comportamiento de las clases utilizadas como también los diagramas de clase.

Entorno Utilizado:

IDE CLion

Lenguaje de Programación:

C++

FLUJO DEL PROGRAMA:

El programa comienza con un menú donde iniciara sesión ya sea administrador o usuario los cuales tendrán un menú. Las estructuras utilizadas en este programa son las siguientes.

- Matriz Dispersa – Contendra los usuarios ingresados al sistema.
- ArbolAVL – Contendra los Activos de cada Usuario.

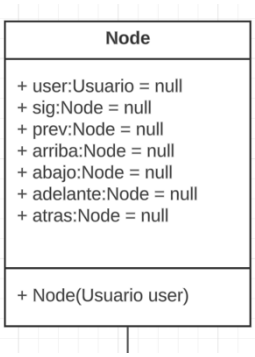
Menú Administrador: Tendrá opciones en las cuales podrá modificar los Nodos Usuarios dentro de la matriz, que son:

- Registrar Usuario
- Reporte Matriz Dispersa
- Reporte Activos Disponibles de un Departamento
- Reporte Activos Disponibles de una Empresa
- Reporte Transacciones
- Reporte Activos de un Usuario
- Activos rentados por un Usuario

Menú Usuario:

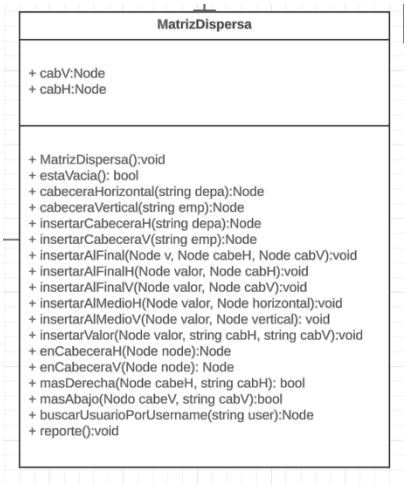
- Agregar Activo
- Eliminar Activo
- Modificar Activo
- Rentar Activo
- Activos Rentados
- Mis Activos Rentados
- Cerrar Sesión

CLASES:



Clase Node:

Esta clase sirve para crear Nodes que contengan punteros estos por defecto tienen el valor de nullptr se enlazaran con otro Node. Tiene a su vez una variable user de tipo Usuario que contendrá al usuario.

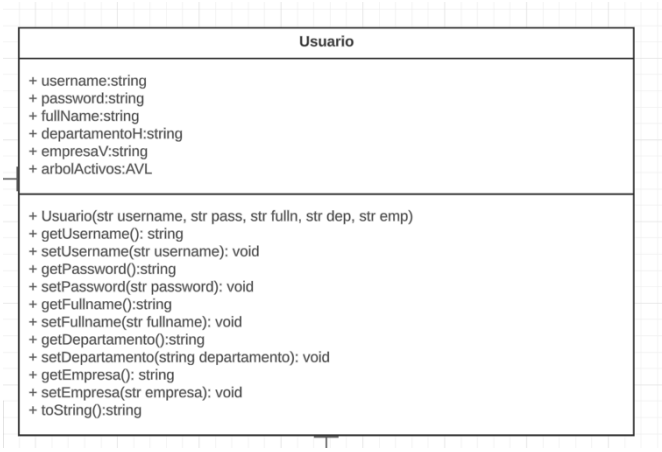


Clase MatrizDispersa:

Esta clase contiene dos apuntadores que se utilizaran para diferentes casos de la matriz y también cuenta con varios métodos para modificar la matriz, como también un método para imprimir la grafica de la matriz.

Métodos:

- **estaVacia():** este método retorna True si los nodos no apunta a ningún Node.
- **cabeceraHorizontal(string depa):** este método encuentra la cabecera horizontal con el departamento mandado.
- **cabeceraVertical(string emp):** este método encuentra la cabecera vertical con la empresa.
- **insertarCabeceraH(string depa):** este método inserta el nuevo nodo cabecera pasado por un str departamento.
- **insertarCabeceraV(string emp):** este método inserta el nuevo nodo cabecera pasado por un str empresa.
- **insertarAlFinal(Node v, Node cabeH, Node cabV):** este método inserta al final de una cabecera H y V el nodo.
- **insertarAlFinalH(Node valor, Node cabH):** este método inserta al final de la lista cabecera horizontal.
- **insertarAlFinalV(Node valor, Node cabV):** este método inserta al final de la lista cabecera vertical.
- **insertarAlMedioH(Node valor, Node horizontal):** este método inserta en medio de la lista de cabecera horizontal.
- **insertarAlMedioV(Node valor, Node vertical):** este método inserta en medio de la lista de cabecera vertical.
- **insertarValor(Node valor, string cabH, string cabV):** este método inserta un valor en una cabeceraHyV el nodo.
- **enCabeceraH(Node node):** este método encuentra el nodo en la cabecera horizontal.
- **enCabeceraV(Node node):** este método encuenta el nodo en la cabecera vertical.
- **masDerecha(Node cabeH, string cabH):** este método recorre a la derecha para ver si esta o no el nodo.
- **masAbajo(Nodo cabeV, string cabV):** este método recorre hacia abajo para ver si esta o no el nodo.
- **buscarUsuarioPorUsername(string user):** este método busca el nodo entre las cabeceras y retorna el nodo.
- **reporte():** este método sirve para generar la grafica en graphviz de la matriz.



Clase Usuario:

Esta clase crea instancias de tipo Usuario las cuales tendrá un árbol de activos.

+ Usuario(str username, str pass, str fulln, str dep, str emp)
Este es el constructor de la clase.

+ getUsername(): Este método retorna un string usurname.

+ setUsername(str username): Este método modifica el username.

+ getPassword(): Este metodo retorna la contraseña.

+ setPassword(str password): Este método modifica la contraseña.

+ getFullname():Este metodo retorna el nombre completo.

+ setFullname(str fullname): Este metodo modifica el nombre completo.

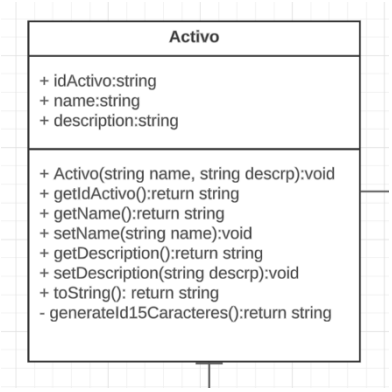
+ getDepartamento(): Este metodo retorna el departamento.

+ setDepartamento(string departamento): Este metodo modifica el departamento.

+ getEmpresa(): Este método retorna la empresa.

+ setEmpresa(str empresa): Este metodo modifica la empresa.

+ toString():Este método retorna un string con todos los atributos del usuario creado.



Clase Activo

Esta clase crea instancias de tipo Activo.

+ Activo(string name, string descrp): Este es el constructor del Activo.

+ getIdActivo():Este método retorna el id del activo.

+ getName(): Este método retorna el nombre.

+ setName(string name): Este método modifica el nombre.

+ getDescription():Este método retorna la descripción del activo.

+ setDescription(string descrp): Este método modifica la descripción del activo.

+ toString(): Este método retorna un string con todos los datos de los atributos del objeto.

- generateId15Caracteres(): Este método genera un id con 15 caracteres para cada activo creado.

Clase NodeAVL

Esta clase crea instancias de este tipo las cuales se utilizaran en el ArbolAVL.

+ NodeAVL(Activo activo): Este es el constructor de la clase que necesitara un activo mandado como parámetro.

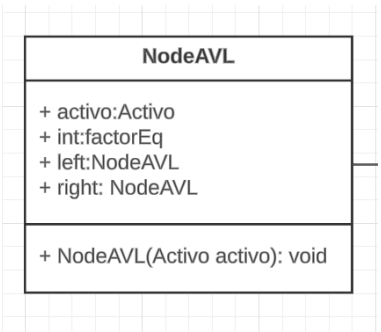


DIAGRAMA UML: