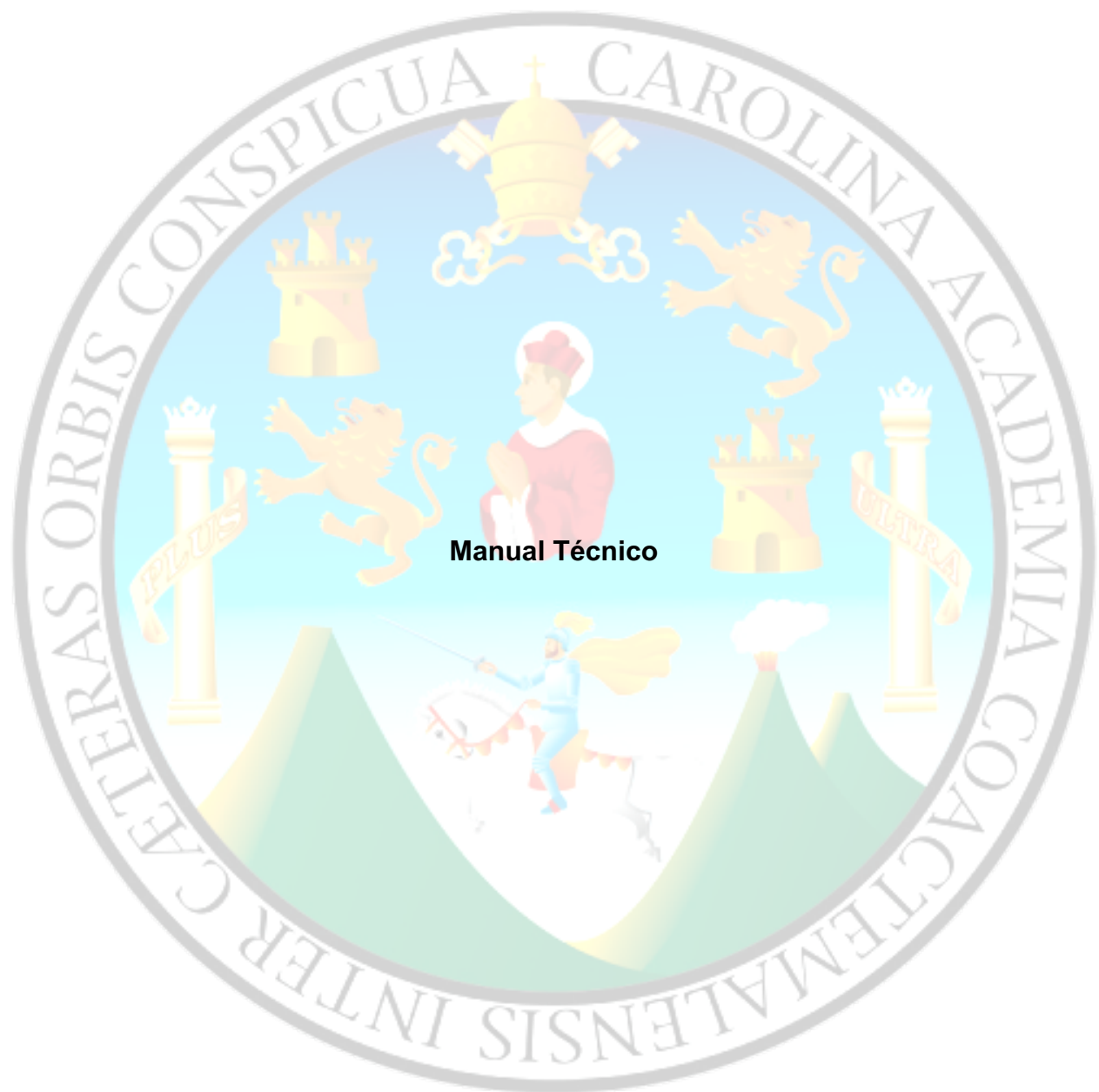


Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Estructura de Datos

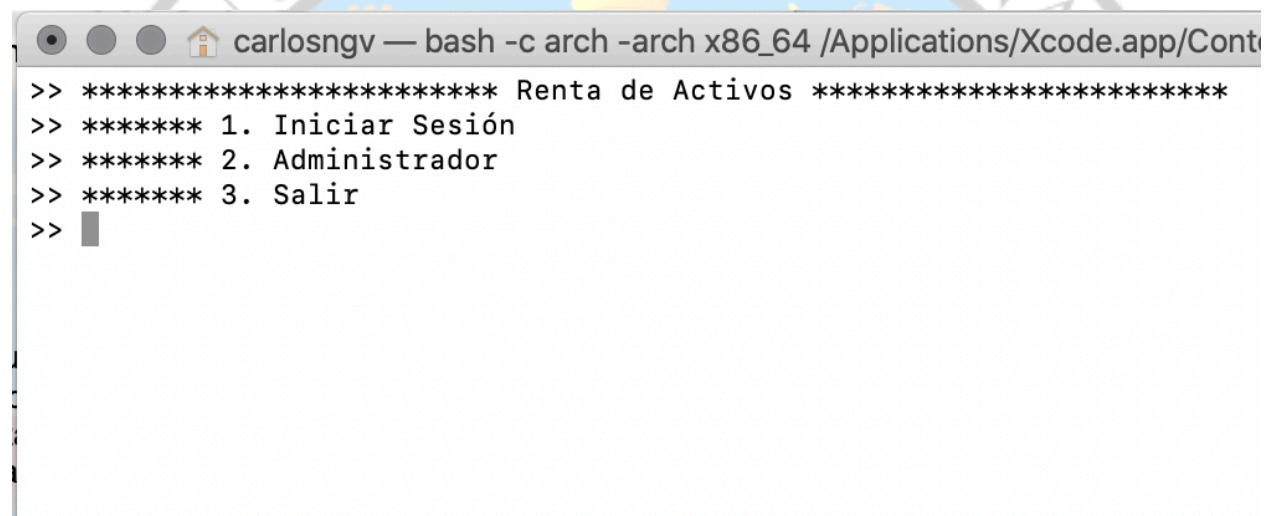


Guatemala 18 de junio de 2020
Carlos Ojani NG Valladares
Carné 201801434

Renta de Activos

Como primer proyecto de Estructuras de Datos, se llevo a cabo un sistema de renta de activos entre usuarios. Se trabajó todo en consola utilizando el lenguaje de programación C++. Se hicieron uso de distintas estructuras de datos. En este caso: matriz dispersa, listas enlazadas y arboles, principalmente. Es con la finalidad de poner en practica los temas vistos en clase y organizar los datos dentro del sistema. A continuación, se describen las estructuras utilizadas, así como su funcionamiento dentro de las distintas funciones que podrían realizarse.

Al iniciar la aplicación se cuenta con un menú amigable para el usuario, como se muestra a continuación:

A screenshot of a terminal window. The title bar shows 'carlosngv — bash -c arch -arch x86_64 /Applications/Xcode.app/Cont'. The terminal content displays a menu for 'Renta de Activos' with three options: '1. Iniciar Sesión', '2. Administrador', and '3. Salir'. The cursor is positioned at the end of the third option.

```
>> ***** Renta de Activos *****  
>> ***** 1. Iniciar Sesión  
>> ***** 2. Administrador  
>> ***** 3. Salir  
>> █
```

Almacenamiento de usuarios.

Para llevar a cabo el almacenamiento de los usuarios que el administrador vaya registrando, se utilizo una matriz dispersa. Cumple con las características que en los encabezados horizontales se colocan los departamentos en los cuales los usuarios pertenecen. En las cabeceras verticales se almacenan las empresas de cada usuario. Siguiendo la teoría de una matriz dispersa, esta va creando sus encabezados conforme se van creando los usuarios, basados en la definición de memoria dinámica.

También la matriz contiene un eje z, por así decirlo. Este se genera cuando se registra un usuario con un departamento y empresa que ya existe. Si ambos existen es que anteriormente se había creado otro usuario allí. Entonces, este nuevo usuario va a dirigirse hacia atrás del primer usuario, formando una lista en el eje z de la matriz. Esto se logra generando otros dos apuntadores, uno frontal y otro trasero. Donde ambos usuarios van enlazados entre sí, cumpliendo con una lista doblemente enlazada en el eje z.

Lo anterior se logra, primeramente, se verifica si los encabezados ya sea habían creado. Esta verificación se realiza cada vez que se desea insertar un nuevo usuario. Su método recibe como parámetro un nodo actual, el valor de la empresa y el nuevo valor a insertar, de donde el nodo actual va recorriendo. Luego recorre hasta la izquierda donde se topa con las cabeceras de empresas y dependiendo de donde halle la coincidencia recorre hacia arriba o abajo. Cuando encuentra la coincidencia entre empresas, significa que si existe un usuario ahí. Entonces, se generan los enlaces de atrás y frente. A continuación se muestra un fragmento de código de este método, cabe mencionar que es lo mismo para las cabeceras de departamento, únicamente cambia la orientación de recorrido.

```
bool Matrix::verifyY(string value, Node* start, Node* nodeValue)
{
    Node* aux_y = start -> getLeft();
    while(aux_y -> getLeft() != NULL)
    {
        aux_y = aux_y -> getLeft();
    }
    if(aux_y -> getData() == value)
    {
        while(start -> getBack() != NULL)
        {
            start = start -> getBack();
        }
        start -> setBack(nodeValue);
        nodeValue -> setFront(start);

        return true;
    }
    return false;
}
```

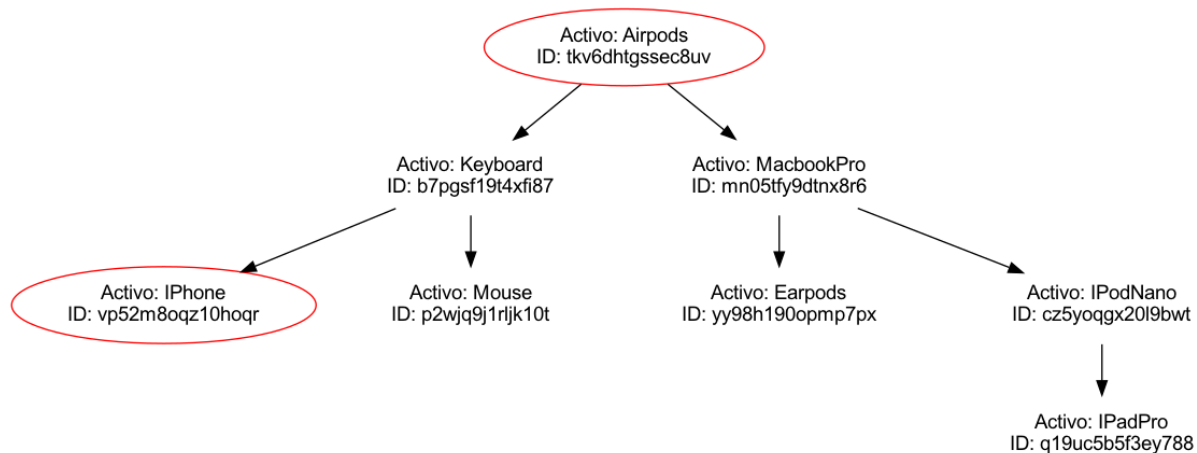
Si no se encuentra coincidencia, se procede a insertar en medio.

Agregar Activos

Cada usuario contiene un árbol AVL, un árbol balanceado. En donde se almacenan los activos que cada usuario va ingresando. Los activos cuentan con los siguientes atributos:

- ID (Alfanumerico de longitud 15, este se toma como referencia para ingresar al AVL).
- Nombre
- Descripción

La inserción en el AVL se basa en la inserción del AST, se toman los ID de cada activo para verificar cuál es mayor o, en su caso, menor. Dependiendo de la altura o si se forma un subárbol degenerado, se aplican rotaciones en el árbol. Rotaciones simples o compuestas, dependiendo sea el caso. A continuación un ejemplo de activos de un usuario:



Los nodos en rojo significan que esos activos se encuentran rentados por otros usuarios. Esto explicará posteriormente. Dentro del AVL, también se puede modificar o eliminar. Para modificar se hace una búsqueda del nodo, siguiendo las convenciones del AST, al llegar al nodo (activo) se podrá modificar el nombre y descripción. Accediendo a sus atributos mediante *getters* y *setters*. Para eliminar se le solicita al usuario el ID del activo que desea eliminar. Dependiendo en donde se encuentre el nodo a eliminar, se reestructurará el árbol. Esto se logra recorriendo el árbol y guiándose de los factores de equilibrio de cada nodo al momento de eliminar. Si es una hoja el nodo a eliminar, se elimina directamente. De lo contrario, si afecta la altura de algún otro nodo, se aplican rotaciones.

Rentar Activo

Para llevar a cabo la renta de activos se agregó un atributo denominado *rented*, de tipo booleano. Que al ser false, significa que no ha sido rentado. De lo contrario, sí. Esto sirve para mostrar o no los activos en el catalogo, dependiendo del estado de la variable *rented*.

Al momento de realizar una renta de activos, se genera una nueva transacción. Las transacciones servirán para llevar un historial de quiénes han rentado activos, adjuntando por cuantos días, el nombre del activo y su identificador. Estas transacciones son almacenadas dentro de una lista doblemente enlazada circular.

El usuario que haya rentado algún activo podrá devolverlo sin ningún problema. Al momento que lo haga, el estado *rented* del activo rentado será vuelto a false y podrá

visualizarse nuevamente en el catalogo, listo para ser rentado por otro usuario. Cabe mencionar que las transacciones quedan guardadas en la lista, no importando que se haya devuelto el activo.

Los usuarios también podrán ver sus activos rentados por otros usuarios. Mostrando un catalogo de cuales de sus activos fueron rentados. Al ser devueltos por el usuario ajeno, saldrán de ese catalogo.

Administrador.

El administrador podrá generar reportes gráficos con respecto a la información almacenada.

Reporte Matriz Dispersa.

En esta opción el administrador genera un reporte grafico de la matriz dispersa que contiene los usuarios con sus respectivos departamentos y empresas.

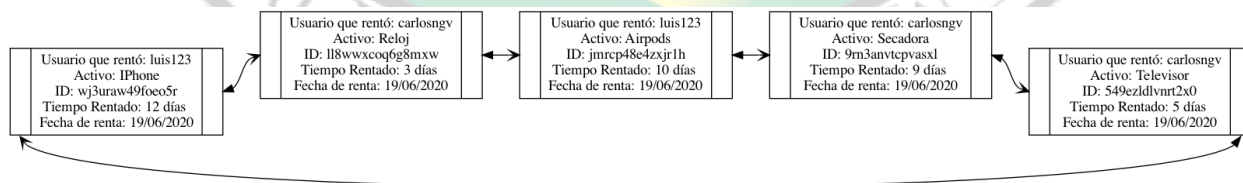
Reporte Activos Disponibles de una Empresa o Departamento.

En este reporte, se realiza un simple recorrido en la matriz y se imprimen los arboles de los usuarios dentro de ese rango. Al administrador se le pide el departamento o empresa, esto indica que se generarán los reportes de todo un departamento o empresa. Al momento de recorrer la matriz y si se encuentra el departamento/empresa se van graficando, ya sea para la izquierda o abajo. Dependiendo del dato ingresado. Estas graficas serán visibles dentro de una carpeta y tendrán el nombre del usuario de los activos.

Reporte Transacciones

Acá se genera la grafica, de la ya antes mencionada, lista circular que contiene las transacciones. Cada nodo almacena el nombre de quién rentó el activo, el ID del activo, descripción y por cuantos días.

A continuación se muestra un ejemplo de las transacciones:

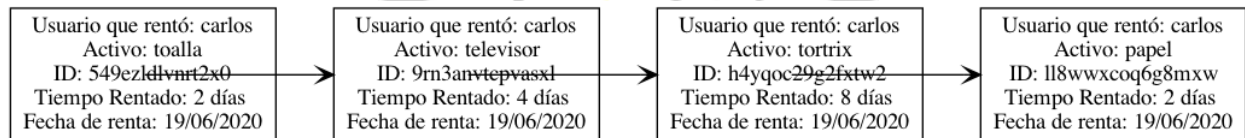


Activos de un Usuario.

Este reporte genera únicamente un árbol. El del usuario a ingresar. Se genera una búsqueda del usuario y se extrae su árbol AVL.

Activos Rentados por un Usuario.

Esta deriva del reporte de transacciones, a diferencia de que solamente se genera la lista con las transacciones de un usuario en específico. Es decir, un filtro. Cada transacción encontrada de ese usuario fue almacenada en una lista enlazada simple, para mayor facilidad. A continuación se muestra un ejemplo:



Ordenar Transacciones.

Este reporte también deriva de la lista circular de transacciones. Como cada transacción fue ingresada de manera descendente, puede quedar así la grafica o el administrador tiene la opción de generar una lista circular ascendente. Esto se logró cambiando la orientación del recorrido, ya que es una lista enlazada doble circular.