**Proyecto final**

**Integrantes:**

Andres Felipe Cuellar

**Enunciado:**

La tienda más reconocida de la ciudad se ha enterado de que usted es un estudiante de Telemática en la universidad Icesi , con el fin de modernizar y actualizar los sistemas de atención al cliente le ha pedido que realice una aplicación que le permita suplir las necesidades para brindar un servicio óptimo.

El sistema debe permitir registrar personas, cada uno de estos se debe registrar con su respectivo nombre ,identificación y teléfono; además el programa debe permitir reconocer que personas son empleados y cuáles son los clientes.

Cada empleado deben tener el número de personas a las que han atendido previamente, Además con el fin de tener una mayor organización dentro del programa la empresa le ha pedido que estos se agreguen al en un árbol de búsqueda binario, que como criterio de ordenamientos debe tomar los 3 más vistos en ABB (inorden, preorden, postorden), y por identificación.

Por otro lado cada cliente debe tener varios productos que haya adquirido en la tienda, los cuales tendrán nombre y su respectivo precio; en el programa los clientes se deben poder organizar en listas doblemente enlazadas, Con el fin de informarse de que es lo que más se está comprando dentro de la tienda, el programa debe permitir hacer la búsqueda de quienes y que fue lo que compraron dentro del local, la búsqueda se debe realizar de dos maneras la tradicional y la binaria.

En los últimos años la tienda ha optado por tomar una filosofía de ayudar y ser más solidarios con las personas referentes a su círculo, y que mejor opción que empezar por los clientes más fieles dentro del sistema, tomando esto en cuenta se le ha ordenado que el sistema deberá tomar en cuenta todo los comprado dentro de la tienda y si la suma de todos los precios de los productos comprados es igual a $200.000 se le hará un descuento del 20.000 en su cuenta de compra.

Guiados por la filosofía anteriormente mencionada la empresa creo una nueva iniciativa para promover el trabajo entre los empleados, por lo cual se tomó la determinación de que al empleado que posea mayor número de veces atendidas en el sistema se lo nombrara como “El empleado del mes”, lo cual lo hará acreedor a que se publicite su nombre en la página y que aparezca su nombre en la interfaz.

Como última función el programa requiere que se desplieguen dos listados el primero que sea las cuentas de los clientes donde se muestra el historial completo de todos los productos que estos hayan comprado dentro de la tienda, y el segundo es para los empleados donde se evidencien las personas que este ha atendido; posteriormente estos deberán ser dejados como evidencia de lo que sucedió en el mes por lo que el programa deben dejar estos listados también en versión de archivos.

**Nota:** Se debe trabajar con excepciones para evitar la reincidencia de personas dentro del sistema u algún otro dato.

**Requerimientos Funcionales:**

**Rq1:** se debe poder registrar los empleados dentro del programa de la tienda.

**Rq2:** se debe poder registrar los clientes dentro del programa de la tienda.

**Rq3:** Los empleados deben poder organizarse según sus identificaciones.

**Rq4:** Se debe realizar una búsqueda binaria del usuario por el código que diga lo que debe y lo que ha comprado en total en el negocio.

**Rq5:** Se debe realizar una búsqueda tradicional del usuario por el código que diga lo que debe y lo que ha comprado en total en el negocio.

**Rq6:** Se debe poder añadir un producto al cliente.

**Rq7:** Se le debe realizar descuento del 20.000 al cliente que llegue a la suma total de $200.000 en su cuenta de la tienda.

**Rq8:** Se debe poder saber cuál es el empleado del mes (el que haya atendido a mayor número de gente) para poner su nombre en el programa.

**Rq9:** Se debe poder generar el listado de los clientes que muestre los artículos adquiridos.

**Rq10:** Se debe poder generar el listado de los empleados que muestre las personas que este ha atendido dentro del local.

**RNF1:** La estructura correspondiente de los empleados debe ser un árbol binario de búsqueda.

**RNF2:** La estructura correspondiente de los clientes debe ser listas doblemente enlazadas

**RNF3** Los empleados deben poder organizarse según los 3 métodos de ordenamiento más conocidos en los ABB (inorden, preorden, postorden).

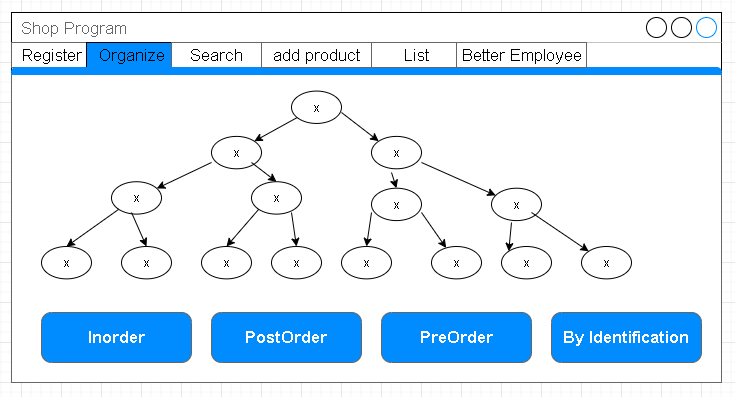
**RNF4:** los listados de clientes y empledos deben ser pasados a archivo.

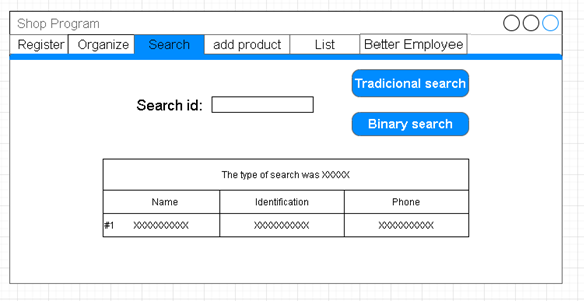
**Justificación:**

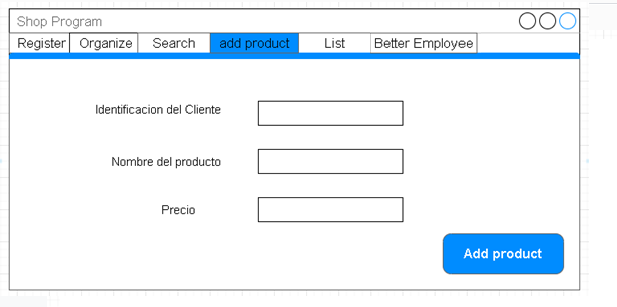
Este trabajo se acopla a la mayor de las parte de lo visto dentro del curso; este trabajo contiene Excepciones, Arboles binarios, listas, los diversos tipos de ordenamientos, búsquedas e interfaz gráfica .Dentro del programa de estudios de Apo II se nos ha recalcado la importancia de los mismos. Por esta misma razón es que se ha decidido la realización del proyecto de manera individual, dado que con este proyecto se busca el medir que tanto ha aprendido el estudiante de estos temas durante el semestre.

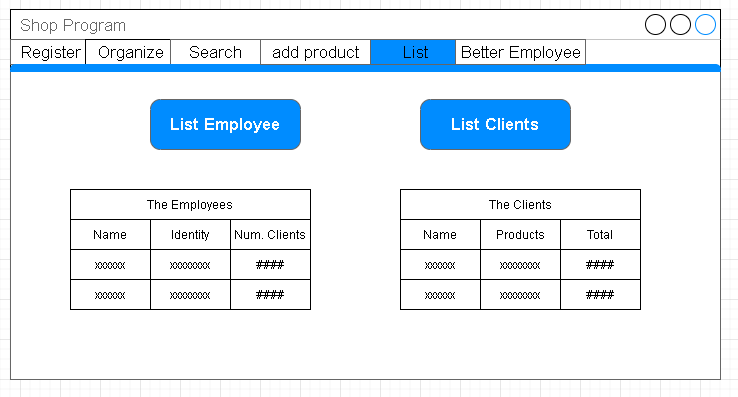
**Mockups:**

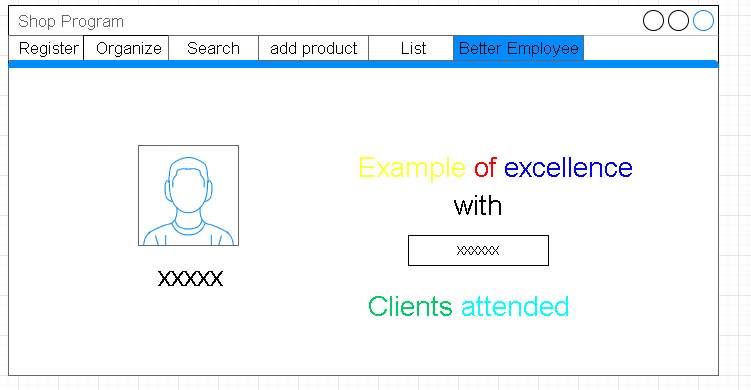
****

****

****

****

****

****

**public** Client findBinaryClient(String id){

ArrayList<Client>clients = orderByIdClient();

**boolean** finded = **false**;

Client client=**null**;

**int** start = 0;

**int** end = clients.size()-1;

**while**(start <= end && !finded) {

**int** middle = (start + end)/2;

**if**(clients.get(middle).getId().equals(id)) {

client = clients.get(middle);

finded = **true**;

}

**else** **if**(clients.get(middle).getId().compareTo(id) < 0) {

end = middle - 1;

}

**else** {

start = middle +1;

}

}

**return** client;

}

**public** ArrayList<Client> orderByIdClient(){

ArrayList<Client> arr = na;

**for**(**int** i = 1; i < arr.size(); i++) {

**for**(**int** j = i; j >0; j--) {

**if**(arr.get(j-1).compareTo(arr.get(j)) > 0) {

Client aux = arr.get(j);

arr.set(j, arr.get(j-1));

arr.set(j-1, aux);

}

}

}

**return** arr;

}