Trabajo practico segundo parcial de Agustín Parrondo y Federico San Pedro, Comisión 3 año, carrera TSP, Año 2017.

Diario de trabajo:

Seguimiento de las actividades que realizamos como grupo (realmente no hay tareas hechas en su totalidad por uno solo y nos ayudamos mutuamente a resolver los conflictos).

Semana 1:

Creación de diagramas de estructuras.

Idealización del menú.

(tarea puramente conjunta)

Semana 2:

Comienzo librería árbol(Agustín).

Comienzo y fin librería lista(Federico).

Implementación Git hud (propuesto por Agustín).

Semana 3:

Comienzo librería fila(Federico).

Comienzo librería cajas (Agustín y Federico).

Fin librería árbol(Agustín).

Semana 4:

Fin librería cajas (Agustín funciones básicas, Federico algoritmos de planificación y procesamiento de las cajas).

Comienzo menú(Agustín).

Semana 5:

Fin del menú(Agustín).

Implementación algoritmos para guardar personas y cajas en archivos(Agustín).

Implementar todo en el main (Federico y Agustín).

Semana 6:

Comienzo y fin parte de usuarios (Federico y Agustín).

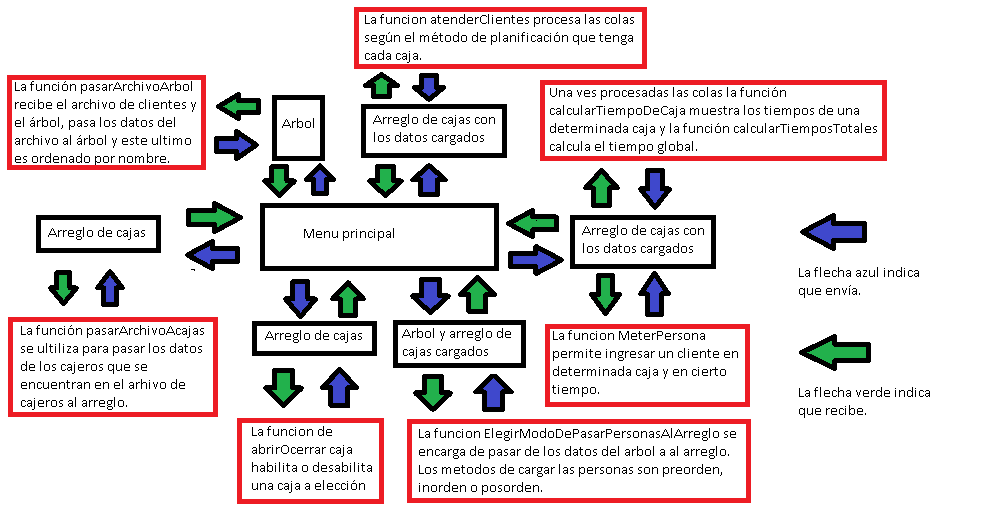
Solución problemas varios, hacer invulnerable al programa(Agustín).

Matriz de soluciones:

|  |  |  |
| --- | --- | --- |
| Problema | Causa | Solución |
| El árbol era borrado en el menú cada vez que volvía al while | ¿? | La función “menú” ahora devuelve el árbol |
| Al juntar todas las librerías en un solo programa no reconocía muchas funciones. | ¿Caprichos del Code Blocks? | Eliminar todas las librerías y crear nuevos archivos con el contenido copiado. |
| Las cajas no recibían clientes. | Accidentalmente pasaba el archivo donde se guardan las cajas al arreglo dos veces. | Eliminar la parte del código innecesaria. |

Diagramas de estructuras:

Funciones utilizadas y sus relaciones:



Para poder pasar los clientes que están en el árbol al arreglo se utiliza Filas. Las filas son estructura que contiene una cabecera y una cola (el programa está definido como último), estos dos últimos mencionados son listas doblemente enlazadas.

Para que una persona pueda ser almacenada en una caja se tiene que cumplir dos condiciones:

1. La caja tiene que estar abierta.
2. El tiempo de pago del cliente tiene que coincidir con la de la caja.

Al tener el árbol cargado y con tener por lo menos una caja abierta, con el uso de la función ElegirModoDePasarPersonasAlArreglo podemos realizar el pasaje y en qué modo. Los modos de pasaje son 3:

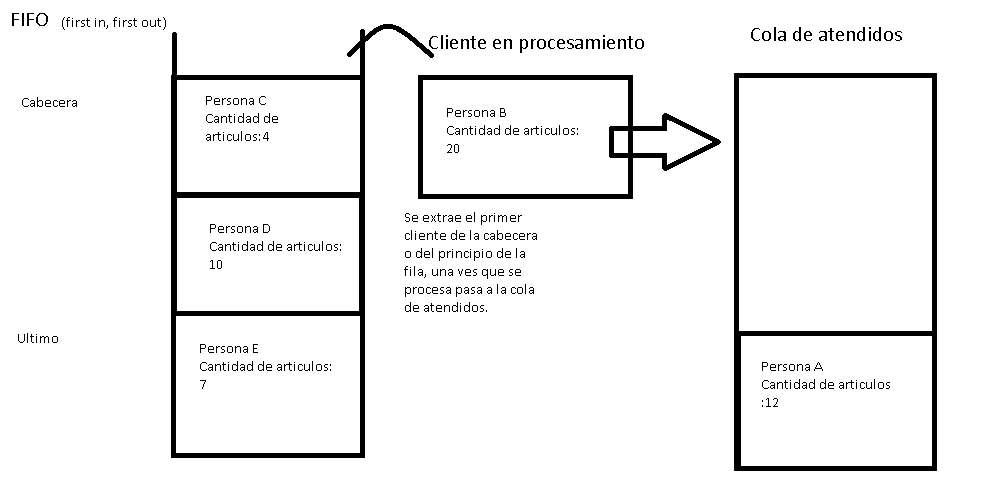
1. Preorden.
2. Inorden.
3. PosOrden.

La función es de tipo entero ya que devuelve la cantidad de personas que se pasaron al arreglo, la cual dicho retorno se utiliza para calcular los tiempos totales de procesado.

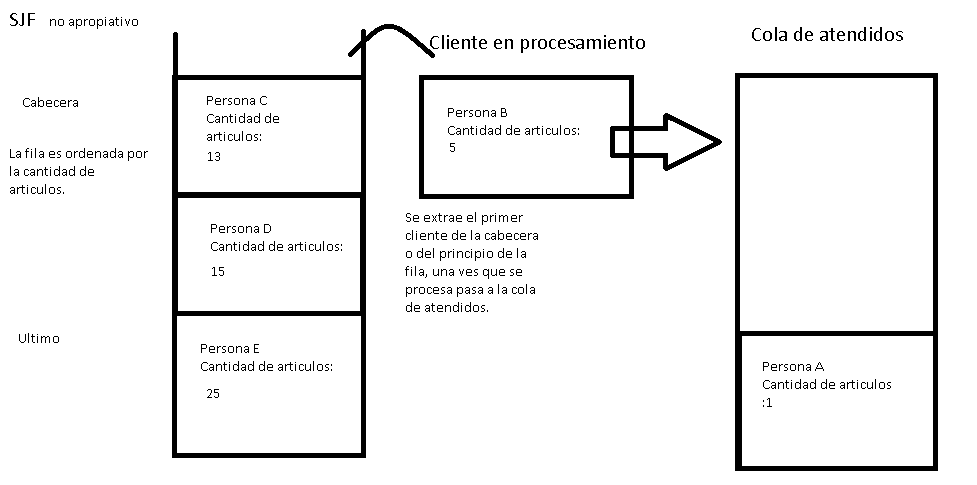
El programa también permite mostrar las estructuras con las siguientes instrucciones:

1. Archivo de clientes: CargarPersonas (char nombre[]);. La función recibe por parámetro el nombre del archivo;
2. Árbol: modosDeMuestraDelArbol(nodoArbol \* arbol); .Muestra el árbol según el modo que se le indique, recibe como parámetro el árbol;
3. Arreglo de cajeros: con esta instrucción mostrarCajas(caja cajero[],int dimension) muestra todas las cajas, y con MostrarDeterminadaCaja(caja cajita[]) muestra una caja a elección.

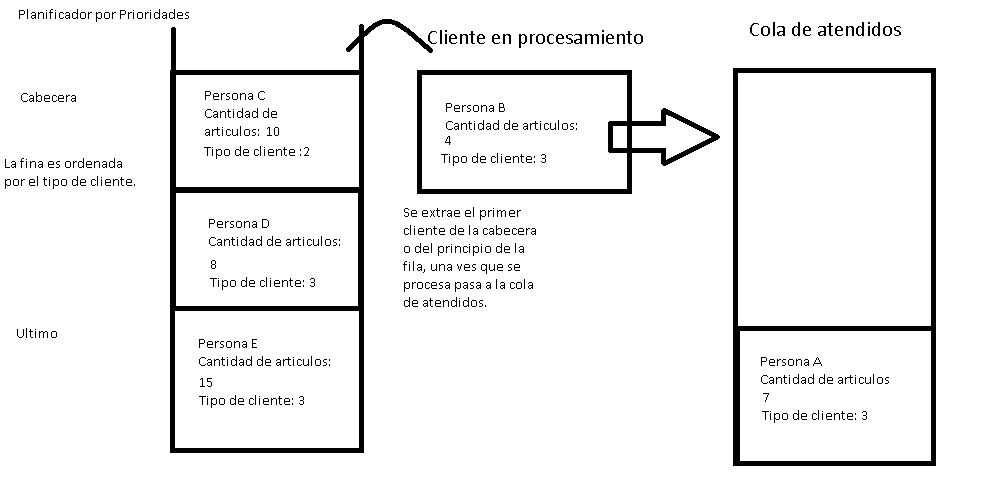
Después de cargar a todas las personas en las cajas, antes de que sean procesados, todas las filas que contenga clientes son ordenados según el tipo de algoritmo de cada caja. En el FIFO queda tal cual llegan a la cola.



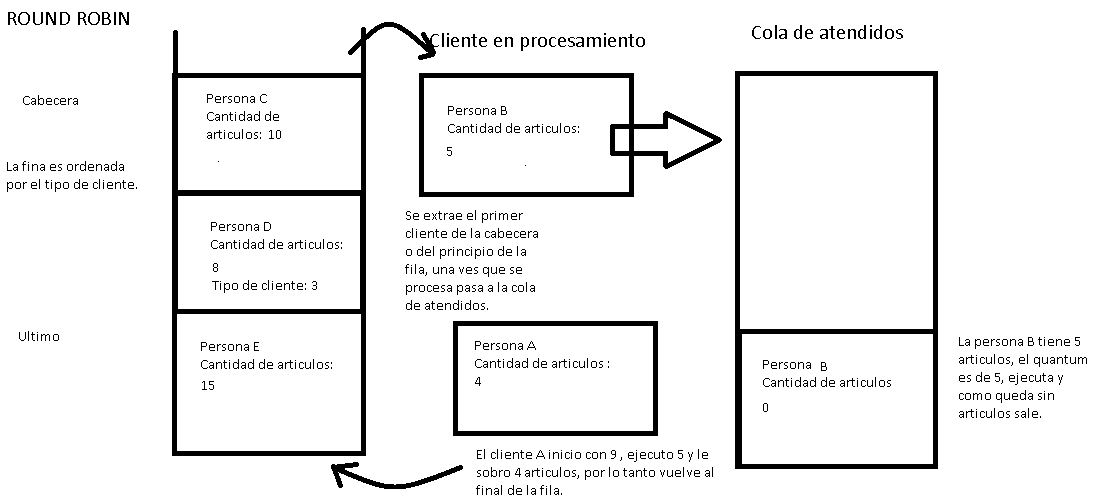
Con el SJF no apropiativo son ordenados por cantidad de artículos (de menor a mayor).



El planificador por prioridades ordena la cola de la siguiente manera; en primer lugar van las embarazadas, luego les sigue los jubilados y por último se procesan las personas normales.



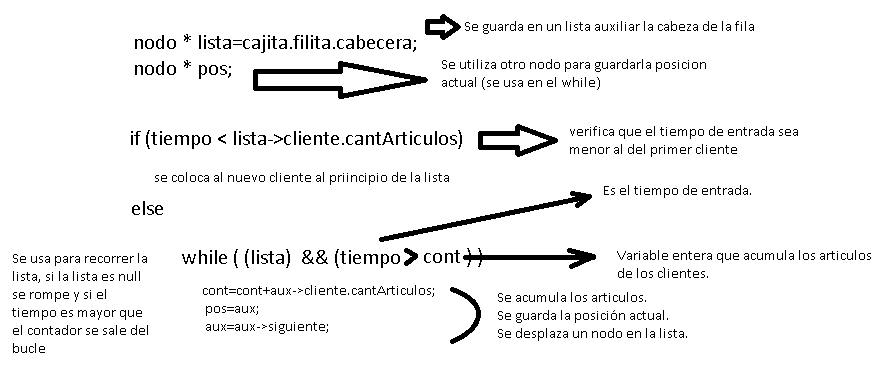
El último algoritmo es el round robin, en la cual tiene un quantum (en nuestro caso es 5), que indica la cantidad de productos que va a procesar cada cliente a la hora de llegar a la cola. Cuando la cantidad de artículos sea 0 sale de la fila, en caso contrario vuelve a la fila original.



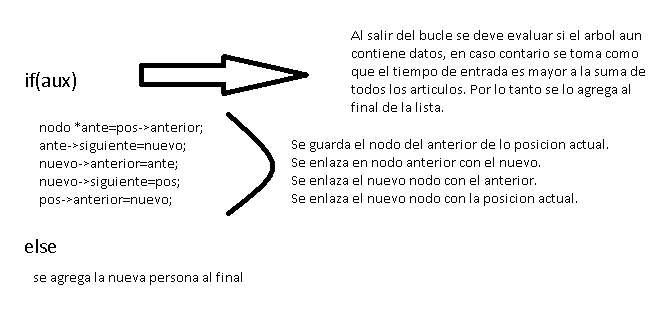
La función que está encargada de seleccionar el algoritmo de planificación es atenderClientes(cajita,dim) , esta función recibe el arreglo de caja y la dimensión.

Si se desea ingresar una persona en un tiempo x, se llama a la función AtenderEnXTiempo(cajita[pos],p,tiempo) , recibe por parámetro el arreglo en que se lo desea insertar (cajita[pos]), la persona nueva (p) y el tiempo de entrada. La función primero debe asegurarse que la caja este abierta y que el tipo de pago de la misma coincida con la de la persona.

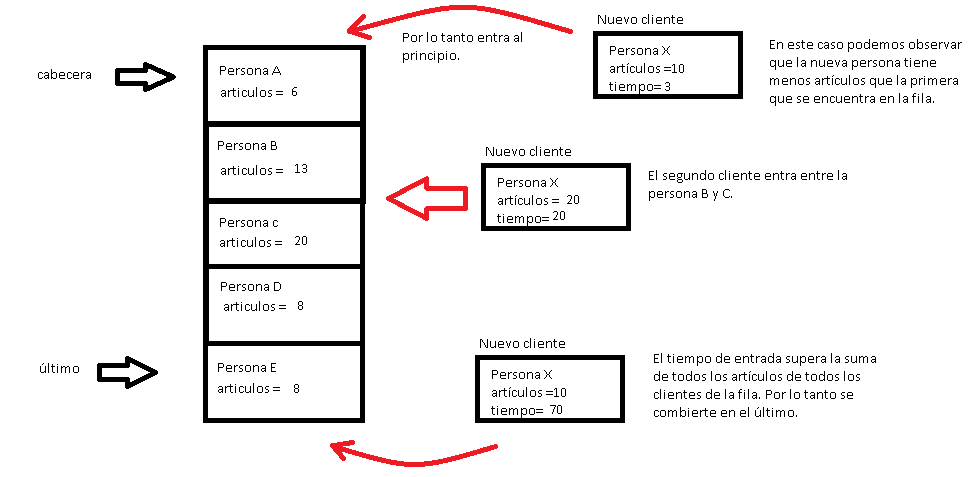
Se planteó el siguiente algoritmo:



A continuación se inserta el nuevo cliente:



Ejemplo gráfico de la función del algoritmo



El programa contiene las siguientes librerías:

* lista: contiene la estructura y todas las funciones con respecto al uso de listas doblemente enlazadas.
* persona: almacena la estructura de personas y el uso de archivos.
* fila: acá se diseñó la estructura de filas con sus respectivas funciones para su uso.
* arbol: tiene la estructura del árbol y sus respectivas funciones para su uso.
* caja: contiene la estructura de caja con sus funciones.
* Librería madre y opciones: esta librería fue agregada para armar la lógica del menú.