REPUBLICA BOLIVARIANA DE VENEZUELA.

Ministerio del poder Popular Para la Educación Superior.

Universidad Católica Andrés Bello.

Sede Guayana. Dirección de Pregrado.

Facultad De Ingeniería.

Escuela de Ingeniería Informática.

Cátedra: Algoritmo y Programación II.

**PROYECTO:**

“ESTACIONES DE SERVICIO”

ALUMNOS:

GONZÀLEZ, FELIX.

GONZÀLEZ N., ROGER D.

Prof. Saad, Josè.

Ciudad Guayana, 24 de Enero de 2017.

**INFORME**

**ESTRUCTURA (STRUCT):**

* **Estacion:** Esta cuenta con 8 variables:

1. **id\_estacion:** Es de tipo cadena de caracteres que va a guardar el nombre de la estación.
2. **capacidad:** Es de tipo entero y va a guardar, cuantas personas puede atender simultáneamente la estación de servicios.
3. **tipo:** Es una cadena de caracteres, va a guardar la política que va a seguir la estación de servicios (FIFO, Prioridad o Round Robin).
4. **duracion de pisoturno:** Es de tipo entero, representael tiempo en que será atendido un cliente, si esta estación pertenece a la política Round Robin.
5. **atendiendo:** Es de tipo entero y va a guardar la cantidad de personas atendidas en cada instante de simulación. Esta nunca va a ser mayor a la capacidad.
6. **atendidos:** Es del tipo entero, se va a ir incrementando cada vez que un cliente salga de la estación.
7. **ocio:** Es de tipo entero, se va a ir incrementando si la estación no tiene a nadie en cola ni en atendidos.
8. **col:** Es de tipo entero, guarda la suma de clientes en cola por unidad de tiempo.

* **Cliente:** Cuenta con trece (13) variables:

1. **id\_cliente:** Es una variable del tipo cadena de caracteres, guarda el nombre del cliente.
2. **prioridad:** Es una variable de tipo entero que determina la antelación con que será atendido el cliente que llega a una estación de servicios de este tipo.
3. **tiempo\_llegada:** Es una variable del tipo entero que representa el instante en que llegó el cliente al sistema.
4. **id estacion:** Actúa como un arreglo del tipo entero que permite ubicar a las personas en sus estaciones.
5. **duracion:** Actúa como un arreglo del tipo entero, paralelo a id estación y representa el tiempo que va a estar siendo atendido en la estación.

1. **CantEstaciones:** Es una variable del tipo entero y representa la cantidad de estaciones a las cuales acude el mismo cliente.
2. **EstacionActual:** Es de tipo entero, representa la estación donde se encuentra el cliente en un momento de simulación.
3. **RR:** Es una variable del tipo entero que va a ser un auxiliar si el cliente entra a una estación Round Robin.
4. **total\_sis:** Es una variable del tipo entero que representa el tiempo total que pasó el cliente en el sistema.
5. **total\_col:** Es una variable del tipo flotante (float) y representa el tiempo total que pasó el cliente en cola.
6. **total\_est:** Es una variable del tipo entero y representa el tiempo total que pasó un cliente en ser atendido.
7. **salida:** Es una variable del tipo entero y representa el instante en que el cliente salió del sistema.
8. **Next:** Es una variable del tipo puntero a estructuras cliente; se utiliza para generar las colas.

**FUNCIONES:**

* **ContarArchivo:** (retorna un entero): Recibe un puntero del tipo file (archivo). Esta función cuenta cuantos saltos de línea existen en los archivos, y retorna esa cantidad. Esto sirve para darle tamaño a los arreglos de tipo estaciones y de tipo clientes.
* **LeerArchivo:** (retorna un entero): Recibe un arreglo del tipo estación, dos enteros y un arreglo del tipo cliente. Esta función abre los archivos de clientes y estaciones en un auxiliar guarda línea a línea y con estas líneas va rellenando los respectivos arreglos (estaciones y clientes).
* **PonerenCola:** (retorna un puntero a cliente): Recibe un puntero carácter (tipo de política que lleva la estación), una variable de puntero del tipo cliente (persona que va a entrar en la cola) y un doble puntero a cliente (es la cola de la estación). Primero, va a comprobar si la cola está vacía, si este es el caso, retorna al cliente como primero de la cola; si no, verifica el tipo de política que lleva la estación, si no es prioridad, el cliente se va a ubicar al final de la cola y si es prioridad, se le ubica en la cola.
* **PonerenAtendidos:** (retorna un puntero cliente): Recibe un puntero cliente (el cliente que va a ser atendido), un doble puntero cliente (son los clientes que están atendiéndose en la estación) y un puntero a un entero (cantidad de personas que hay en atendidos). El cliente sale de cola y entra en atendidos, incrementándose en uno el entero que lleva la cantidad de personas atendidas.
* **BuscarEstacion:** (retorna un entero): Recibe un entero (cantidad de estaciones que existen), un arreglo del tipo estación (son las estaciones) y una cadena de caracteres (nombre de la estación a la cual se va a dirigir el cliente). Dentro de un ciclo para (For), desde 0 hasta el número total de estaciones menos uno, se va a ir comparando el nombre de la estación con la estación a la cual va a ir el cliente, y va a retornar su posición en el arreglo.
* **PonerEnTiempo:** (retorna un puntero a cliente): Recibe dos punteros a cliente (uno a la cola y otro a la persona). Esta función se utiliza para organizar en una cola a los clientes por tiempo de llegada, para que de allí, salgan a las colas de sus respectivas estaciones.
* **PasarTiempo:** (vacio, VOID): Recibe dos enteros (cantidad de estaciones y el tiempo de simulación), un arreglo de punteros a clientes (las personas o clientes que están siendo atendidas), un doble puntero a cliente (variable que reubica a los clientes, si estos necesitan ir a otra estación), un arreglo del tipo estaciones. Tiene dos partes: la primera, disminuye en uno el tiempo de duración que le faltan a los clientes para terminar en atendidos; la segunda, verifica qué clientes ya han terminado en esa estación. Pueden salir por dos razones: porque el tiempo en esa estación ya llego a cero o porque se encuentran en una estación de política Round Robin y su variable RR llegó a cero.

En la primera salida, verifica si el cliente debe ir a otra estación, si es así, se le reubica (pasa a la cola de clientes para reubicar). Si no debe ir a otra estación, sale el cliente del sistema. La segunda forma de salida, si su variable RR llegó a cero, el cliente se va a reubicar en la cola de esa estación.

* **ImprimirCola:** (Vacio, VOID): Recibe un entero (cantidad de estaciones), un arreglo del tipo estación (es la información de cada una de las estaciones) y doble puntero a cliente (las colas y los atendidos de cada estación). Esta función imprime o muestra por pantalla, la dinámica del sistema en cada estación y por cada cliente.
* **FinSimulacion:** (retorna un entero): Recibe un arreglo tipo cliente (información de cada cliente), un entero (representa el número total de clientes). La función es un FOR desde 0 hasta el número total de clientes menos uno, donde se evidencia si el cliente ha hecho el recorrido por las estaciones asignadas.
* **Archivo:** (retorna Vacio, VOID): Recibe un puntero a File, tres (3) enteros, un flotante, un arreglo del tipo estación y un arreglo del tipo cliente. Esta función con los datos enviados crea el archivo reporte.

**FUNCIONAMIENTO DEL PROGRAMA:**

1. El Programa solicita que se ingrese el nombre de los archivos de estaciones y de clientes que se van a utilizar para la simulación.
2. Solicita el tiempo, expresado en milisegundos, que durará cada una unidad de tiempo de la simulación.
3. El programa, con la función contar archivo, determina cuántas estaciones hay y cuántos clientes se deben atender en total.
4. Con esos datos conocidos, se crean los arreglos de clientes y de estaciones.
5. Con la función leer archivos, se van guardando en los arreglos las respectivas informaciones que hay en los archivos.
6. Se crea el puntero a cliente, llamado orden de llegada, el cual con la función PonerEnTiempo va a crear una cola de los clientes dependiendo de su tiempo de llegada, ordenando de menor a mayor.
7. Se crean tres (3) variables puntero cliente; dos son un arreglo de punteros a clientes (cola y atendidos) y el otro, la cola de reubicar, donde estarán los clientes que han sido atendidos en una estación y necesitan ir a otra. A todas estas, se les inicializa con NULL.
8. Se iguala el tiempo de simulación a cero (0) y dentro de un ciclo repetir hasta (DO WHILE), se limpia la pantalla, se coloca el tiempo de simulación asignado, se llama a la función PasarTiempo; luego dentro de un WHILE verifica que los primeros clientes de la cola orden de llegada sean igual al tiempo de simulación, si es así, el cliente será ubicado en su respectiva cola. Este ciclo termina, cuando no encuentre ningún cliente que vaya a entrar en esa unidad de tiempo o que, ya todos los clientes hayan entrado al sistema por lo menos una vez.
9. Cola Reubicar: Los clientes que terminaron en su estación son dirigidos a la cola de su nueva estación, siempre y cuando tengan que ir a otra.
10. Se revisan todas las colas de cada una de las estaciones, se pasa de cola a atendidos, siempre y cuando exista una cola y los atendidos actuales no superen a la capacidad de la estación.
11. Se muestra en pantalla, ImprimirCola, la situación momentánea de la simulación y el programa se detiene en el tiempo pedido. Esto se repite, hasta que la función FinSimulacion haya decretado que todos los clientes fueron a todas las estaciones asignadas.
12. Se crea el archivo reporte con todos los datos que se han ido recogiendo durante la ejecución del programa o simulación.

ANEXO:

* **ESQUEMA DE FUNCIONAMIENTO DEL PROGRAMA ESTACIONES DE SERVICIOS**