Practica 02 Simulaciones con el TAD Cola

**Introducción**

Esta práctica tiene como fundamento la implementación del TAD Cola en, (estática, estática circular y dinámica) para resolver las simulaciones propuestas.

**2.1 Simulación de la atención de clientes en un supermercado.**

Programa que simula la atención de clientes en un supermercado con n cajeras disponibles para atenderlos, cada una cuenta con su propio tiempo de atención.

Los clientes llegan cada cierto tiempo, se forman al azar en alguna caja y se quedan ahí hasta que son atendidos.

Si ya se atendieron al menos 100 clientes y no hay clientes en las cajas, se cerrará el supermercado.

**2.1.1 Planteamiento del problema**

Simular la atención de clientes en un supermercado con n cajeras mediante el uso del TAD cola.

**2.1.2 Diseño y funcionamiento de la solución (Meter pseudocódigo de las funciones)**

//simulación

Se define una estructura con char para el nombre del supermercado, un entero n para el numero de cajeras disponibles, con entero para el tiempo de atención de cajeras, las filas de la simulación y un entero para el tiempo de llegada de los clientes.

//aleatorio

Se genera un numero aleatorio en un intervalo.

//formar cliente

Con un indicador id del cliente se hace la encolación de un cliente en una cajera que corresponde

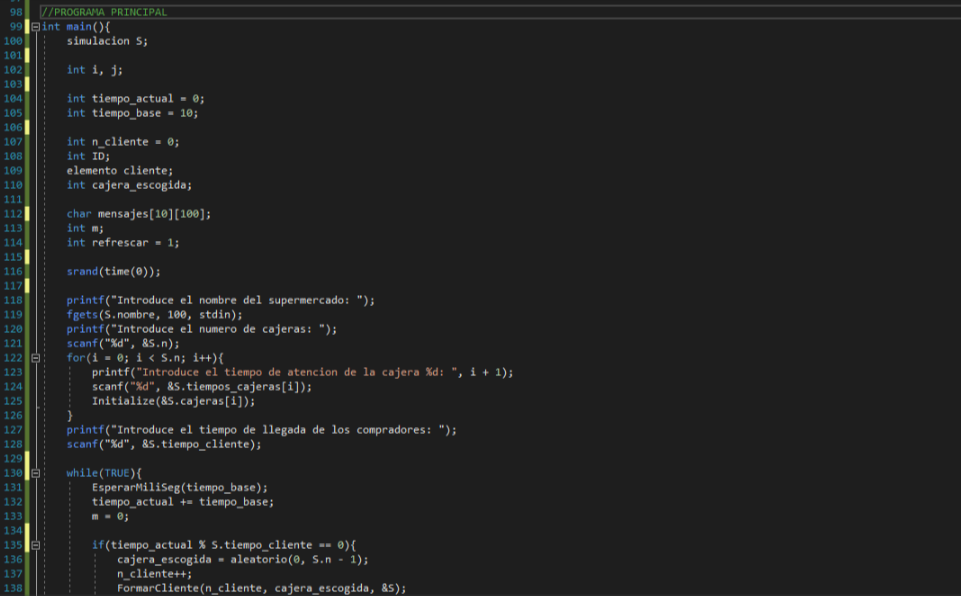
//procesar cajera

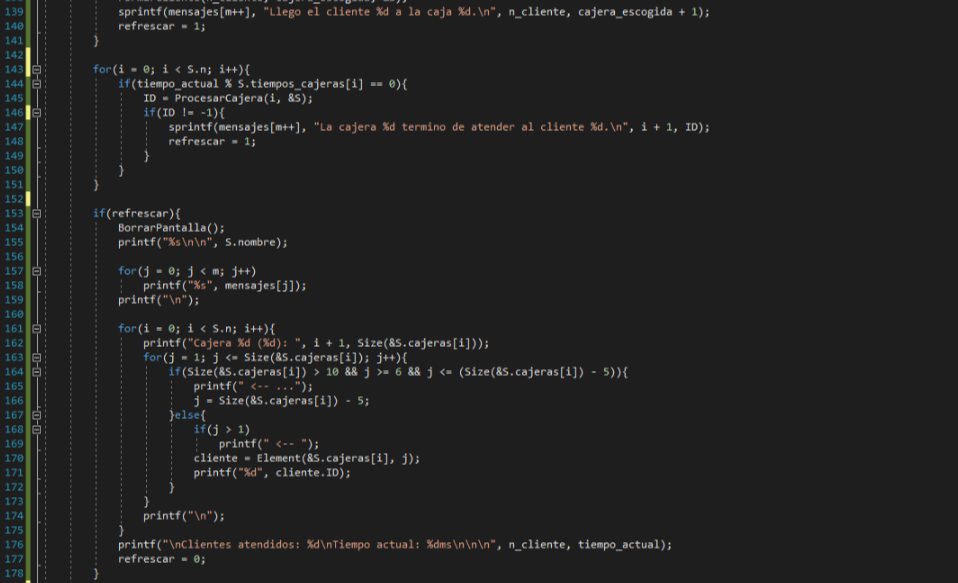
Con el id del cliente se va desencolando el cliente de enfrente para ser atendido si es que la cola no está vacía.

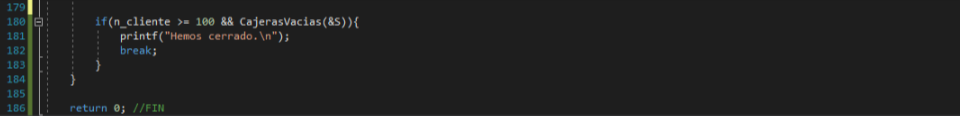
//cajeras vacías

Determina si una cajera está vacía tomando 1 en caso afirmativo y 0 en caso contrario.

**2.1.3 Implementación de la solución**





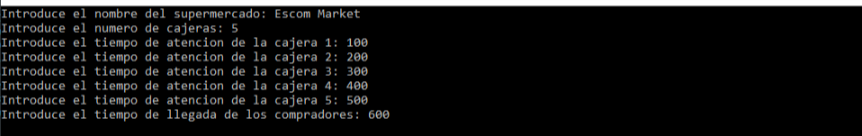


Se hace un contador para identificar cada cliente que va llegando, se generan los números aleatorios y se le pide al usuario el nombre del supermercado, el número de cajeras que va de 1 a 10 con su tiempo de atención de cada cajera y el tiempo de llegada de los clientes.

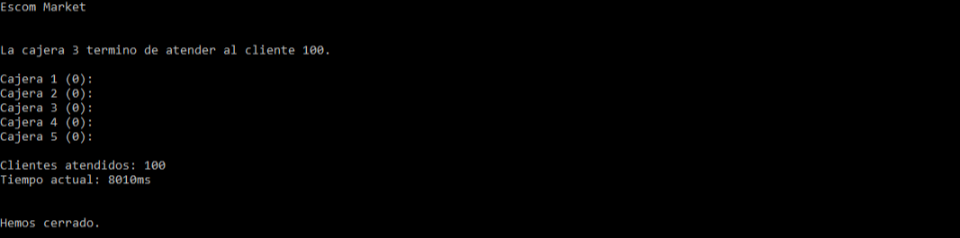
Se ejecuta la simulación esperando el tiempo base, cada que llega un cliente se forma en la caja que elige al azar, se evalúa si alguna de las cajas va a finalizar y acabando se atendió al cliente, se imprime la información con los cambios que se hacen.

Finalmente, se válida si ya se han atendido los 100 clientes y si ya están vacías las cajeras para terminar la simulación.

**2.1.4 Funcionamiento**

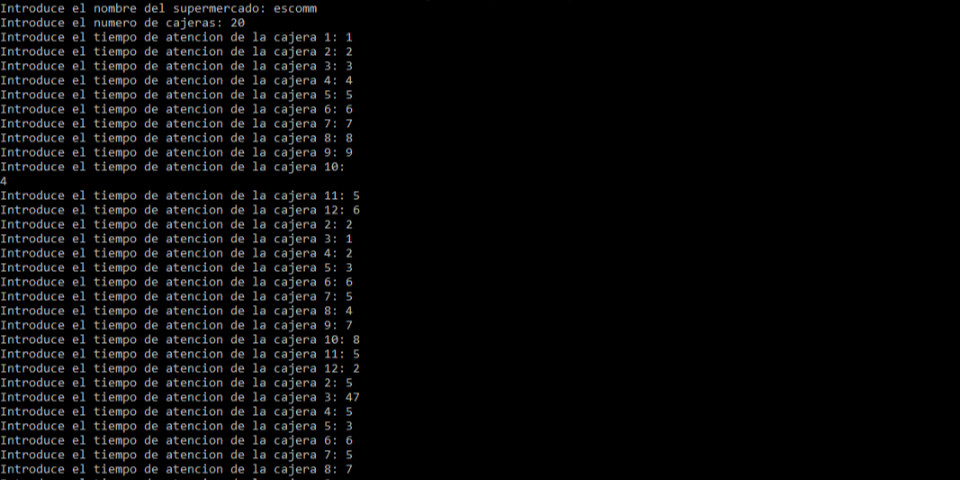


Ejecutando el programa se ingresan los datos conforme se van pidiendo ingresando el nombre del supermercado el numero de cajeras y los tiempos de atencion de llegada de cada cliente tambien el tiempo de llegada de los clientes.



Una vez haber cumplido las condiciones de atender a los 100 clientes y no quedando ninguno en ninguna cajera el programa finaliza.

**2.1.5 Errores detectados**

Si se intentan meter más de 10 cajeras el programa tendrá errores y ni siquiera funcionará como debe de ser.

**2.1.6 Posibles mejoras**

Como la posible mejora, la solución viable es validar que tenga a lo más 10 cajeras.

**2.2 Simulación de la ejecución de procesos en el sistema operativo.**

El programa simula la ejecución de procesos de un sistema operativo con las condiciones de cada quantum es de 1 segundo.

**2.2.1 Planteamiento del problema**

Simular la ejecución de procesos de un sistema operativo implementando el TAD cola.

**2.2.2 Diseño y funcionamiento de la solución (Meter pseudocódigo de las funciones)**

//mostrar actual

Se imprimen los procesos o programas que se está ejecutando y el tiempo que lleva en total de ejecución.

//mostrar ultimo

Muestra el id, el nombre y el tiempo restante de ejecución del último programa en la cola antes de iniciar la ejecución de otro.

//mostrar siguiente

Muestra el programa que está enfrente de la cola con su id, nombre y tiempo restante de ejecución.

//mostrar teminado

Se imprime el nombre del programa que ha terminado su ejecucion por completo.

//mostrar finalizados

De la cola donde estaban los programas conforme terminaron de ejecutarse imprime su id nombre y el tiempo que le tomo ejecutar completamente el programa.

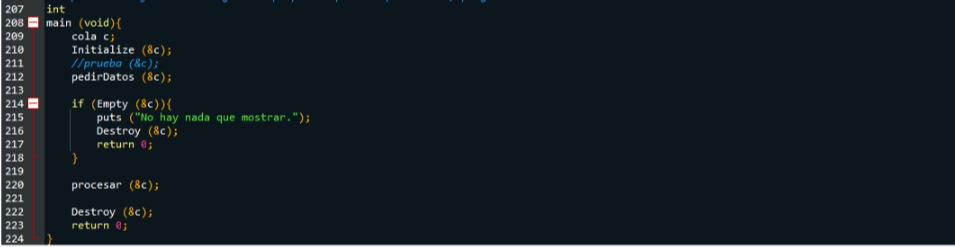
//procesar

Simula el procesamiento de una cola mediante el criterio de que se desencola el programa y se incrementa su contador y si el contador es menor al tiempo definido se vuelve a encolar, valga la redundancia a la cola, en caso de finalizar su contador se encola a la cola de nombre acabados.

//pedir datos

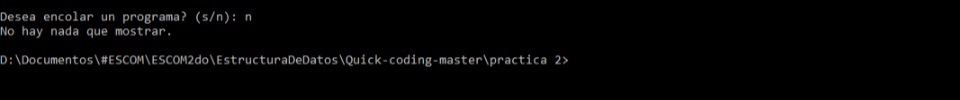
Da la opcion de encolar o no un progrma teniendo por defecto la cola de los programas, en caso afirmativo se piden los datos de programa, los cuales son el nombre, la actividad, el id, el tiempo de ejecucion y despues se pregunta si se quiere encolar otro programa.

**2.2.3 Implementación de la solución**

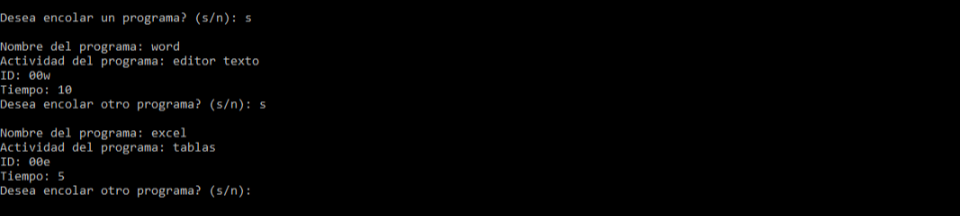


Se inicializan las colas ypiden los datos de los programas a encolar y en dado caso de que no haya ningún programa muestra el mensaje, en caso de que si se hace el procesamiento de los programas para después finalizar.

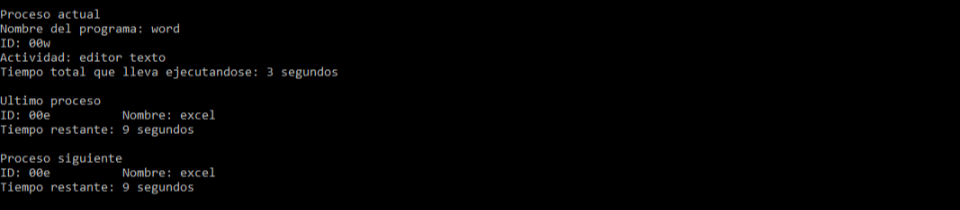
**2.2.4 Funcionamiento**

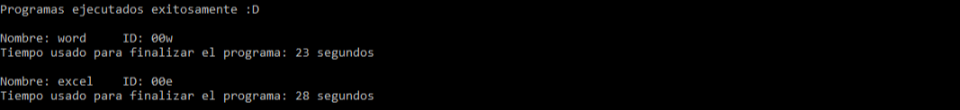


Se pregunta si se desea encolar el programa, si la respuesta es no muestra que no hay nada.



En caso contrario se piden los datos del programa y empieza a ejecutar la simulación.





Y al final se muestran los tiempos de ejecución.

**2.2.5 Errores detectados**

Debido a que no pide el tiempo de ejecución el sistema si se dan tiempos altos el proceso tarda demasiado.

**2.2.6 Posibles mejoras**

Como posible mejora es pedir al usuario un tiempo de ejecución para agilizar el tiempo de espera.

**2.3 Simulación de un banco.**

El programa simula la atención de clientes en un banco.

**2.3.1 Planteamiento del problema**

Simular la atención de un banco implementado el TAD cola con condiciones.

**2.3.2 Diseño y funcionamiento de la solución (Meter pseudocódigo de las funciones)**

Se define una estructura cajero con un booleano para saber si está ocupado o no y un elemento el cual va a ser persona.

Y una estructura simulación la cual va a tener el número de cajeros que van a atender junto con su tiempo de atención y el de los clientes, usuarios, los preferentes, y el número de clientes y preferentes desde el último usuario.

Una estructura “cuantos” pone por defecto false en la propiedad de los cajeros para regresar los cajeros disponibles.

//Revisar cajero libre

Verifica si el cajero está disponible para atender, si son más de 1 devuelve el primero que hay y devuelve los cajeros que están desocupados.

//formar persona

Toma en cuenta una persona con su tipo de jerarquía para atender el banco, se encola de acuerdo al tipo de cliente y como estén los demás cajeros.

//pasar persona

Toma en cuenta el tipo de cliente con su prioridad y la posición del cajero valido, se pasa al cliente cambiando el estado del cajero a ocupado.

//persona puede pasar

Toma el tipo de cliente que es y determina si puede pasar al cajero en base a las políticas del banco, si puede pasar devuelve la posición del cajero.

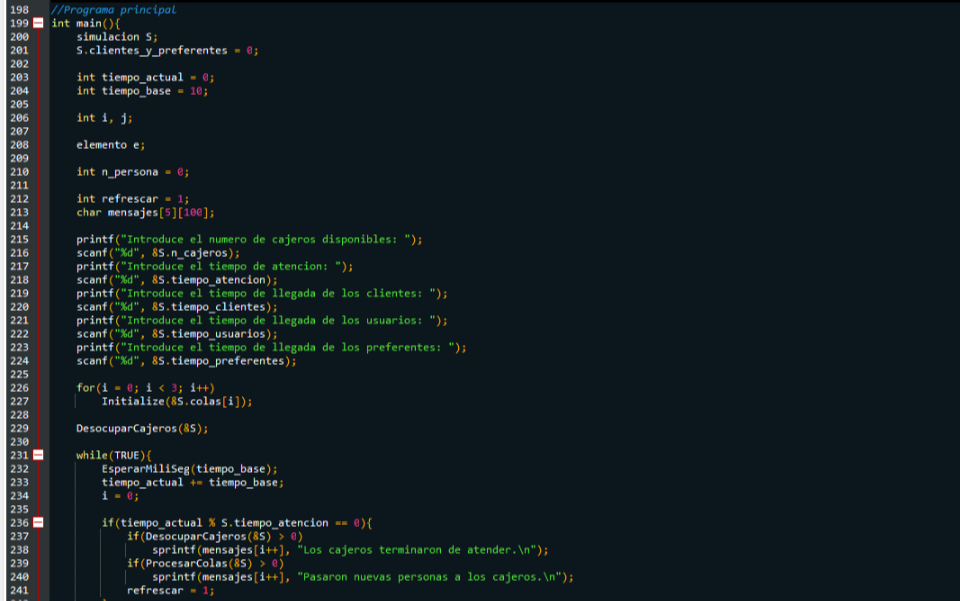
//Procesar Llegada Persona

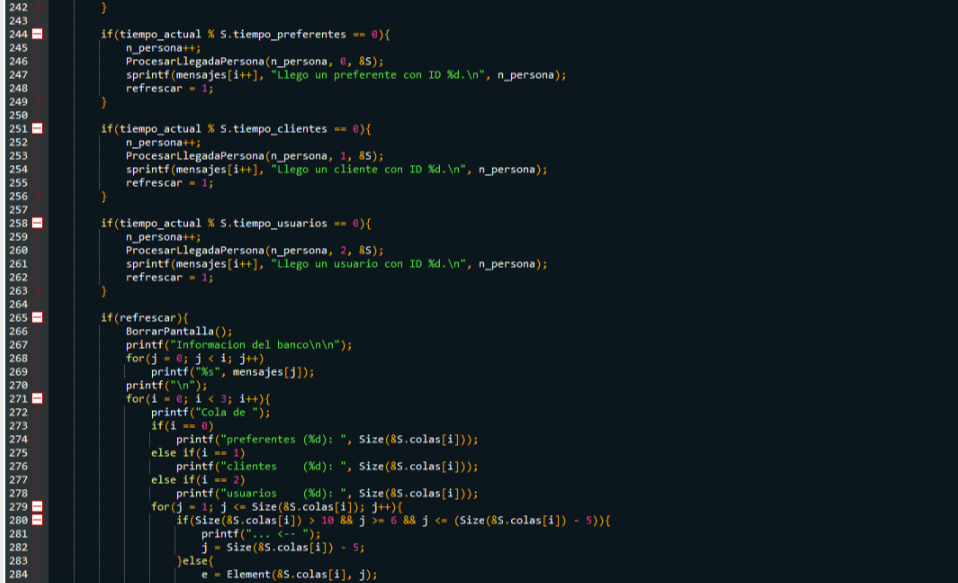
Dependiendo del tipo del cliente determina si pude pasar tan pronto se desocupe un cajero o si se tiene que esperar en base al tipo de cliente que es e irse a formar a una cola.

// procesar colas

Toma en cuenta el tiempo de atención de los cajeros y determina su las personas que están al frente de las colas pueden pasar o no a algún cajero, si se puede se desencola el cliente.

**2.3.3 Implementación de la solución**







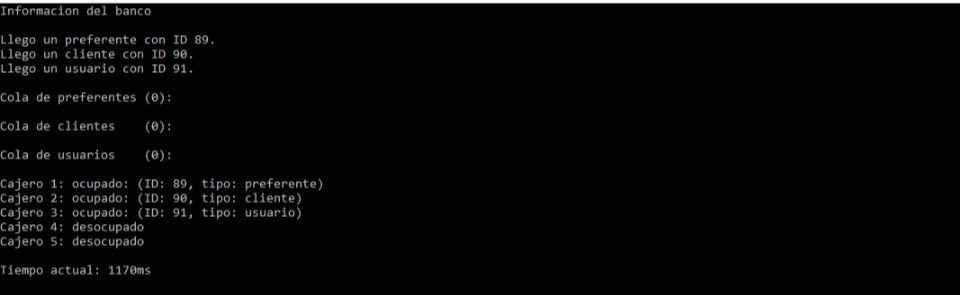
se inicializa la simulación declarando las variables, y se piden los datos al usuario y se inicializan las 3 colas que se van a utilizar con las propiedades de los cajeros desocupados.

Se ejecuta la simulación de manera no definida y se desocupan los cajeros, llegando los clientes con sus respectivos tiempos y se muestra la simulación en tiempo real mostrando la información del banco y el estatus de las colas de preferentes, clientes y usuarios, corriendo en base al tiempo de llegada de cada cliente y el número de personas que haya en las colas tomando en cuenta su tipo.

**2.3.4 Funcionamiento**



Se le piden los datos desde el número de cajeros hasta los tiempos de cada tipo de persona.



Se hace la simulación mostrando la información de lo que está pasando en el banco.

**2.3.5 Errores detectados**

Debido a que está limitado a un número de cajeros no se pueden poner más y si los tiempos de llegada se acortan se hacen colas cada vez mas largas.

**2.3.6 Posibles mejoras**

La mejora adecuada sería dejar poner más cajeros tomando en cuenta el tiempo de llegada de las personas para que no se hagan colas tan largas.

2.4 Conclusiones

Pavel

En conclusión, el uso del TAD cola y TAD pila es de suma importancia en la resolución de cierto tipo de problemas que, si pueden ser resueltos por otros métodos, pero con estos tipos de estructuras agiliza la programación e implementación, llegando a la solución de una manera simple y eficaz.

Fuentes

Introduction to Algorithms, 3rd Edition (en inglés) (3rd edition edición). The MIT Press. 31 de julio de 2009. ISBN 9780262033848. Consultado el 9 de enero de 2016.

The Algorithm Design Manual (en inglés) (2nd edition edición). Springer. 26 de julio de 2008. ISBN 9781848000698. Consultado el 9 de enero de 2016.

Thorup, Mikkel (1 de diciembre de 2007). «Equivalence Between Priority Queues and Sorting». J. ACM 54 (6). doi:10.1145/1314690.1314692. ISSN 0004-5411. Consultado el 9 de enero de 2016.