

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Estructura de Datos B

Ing. Carlos Gustavo Alonzo

Aux. Josué David Itzep Salvador



PRACTICA 1

DESCRIPCION GENERAL

La modelación y simulación de sistemas, es en la actualidad una de las técnicas más utilizadas por grandes empresas, ya que ayuda a analizar y poner a prueba distintos sistemas sin necesidad de implementarlos, lo que les ayuda a reducir costos y maximizar ganancias, con la posibilidad de someter los distintos sistemas a las pruebas necesarias.

La empresa Wall-E-Mart es una cadena de supermercados multinacional, que vende todo tipo de productos del hogar, perecederos, etc. y sus sucursales se encuentran ubicadas en puestos estratégicos de las distintas ciudades del mundo, y está evaluando distintos escenarios para la construcción y reestructuración de las distintas sucursales, esta empresa ha solicitado la ayuda de los estudiantes de Modelación y Simulación de Sistemas 1 de la facultad de Ingeniería, sin embargo el presupuesto es limitado y las licencias de software de simulación son altas, por lo que para la realización de este proyecto han solicitado que los estudiantes de Estructura de Datos desarrollen el software que sea capaz de simular el sistema.

Dicho software a desarrollar es una aplicación de consola en el lenguaje C, en la cual se pueda ingresar distintos valores que permitirán el establecimiento de distintas configuraciones del sistema, así mismo debe ser capaz de mostrar los distintos subsistemas de una forma visual, agradable y amena, mediante representaciones graficas (imágenes).

MENU PRINCIPAL

La aplicación de consola mostrara como pantalla principal un menú de opciones en la cual se podrá elegir las configuraciones de los distintos subsistemas.

- Carretera
- Parqueo
- Supermercado, Estanterías y Productos
- Cajas
- Ventas
- Simular
- Salir

CARRETERA

Ya que la ubicación de las distintas sucursales es transitada por miles de carros a diario, se simulara el tráfico de la carretera principal, por lo que se podrá ingresar el número de carros que se encuentran en el flujo vehicular, además se puede elegir los distintos clientes/carros que entraran al parqueo de Wall-E-Mart, a través de un nodo “Entrada”, el cual podrá especificarse, y será una posición en el rango de 0 a los N carros que haya en la carretera. La carretera consistirá de una **lista simplemente enlazada**, donde cada nodo de la lista será un carro y por ende el número de carros será igual al número de nodos más el nodo entrada, así mismo cada carro podrá avanzar hacia la “Entrada” del parqueo mediante el ingreso de su id, se desplazaran una posición a través de la carretera. En cualquier momento podrán ingresar más carros a la carretera. Se debe de visualizar el estado de la carretera principal, en cualquier momento, carros, y su información respectiva. Los carros que entran al parqueo deben de desaparecer de la carretera y estacionarse en un espacio del parqueo que se especificara más adelante, los clientes que terminen sus compras también podrán regresar a las carreteras, y aparecerán después de la “Entrada”. En cada nodo solo puede haber un carro.

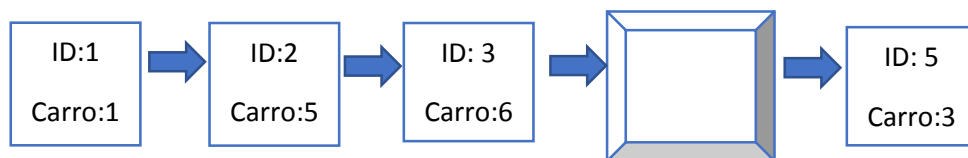


Fig. 1 Carretera, transitada por 4 automóviles, con entrada en la posición 4

PARQUEO

El área de parqueo consistirá en una **lista circular simplemente enlazada**, el número de carros que puede haber en el parqueo deberá ser ingresado, estos serán el número de nodos de la lista, en un nodo solo puede haber un carro, el parqueo contara con los espacios disponibles para estacionarse, vacíos al inicio. Los carros que salen de la carretera para entrar en el parqueo elegirán su posición mediante la generación de un numero aleatorio que estará entre el rango de ID's que posee el parqueo, así será hasta que encuentre un espacio disponible, de no encontrar un espacio disponible saldrá del parqueo para incorporarse después de la “Entrada” del parqueo, la cual se encuentra en la carretera. Se debe de visualizar los parqueos ocupados y disponibles en cualquier momento, así como la información del espacio, los carros en parqueo, etc. El tamaño del parqueo puede ser aumentado o disminuido en cualquier momento, si se encuentran carros en una posición del parqueo eliminada, deberán de incorporarse al tráfico de la carretera después de la “Entrada” del parqueo.

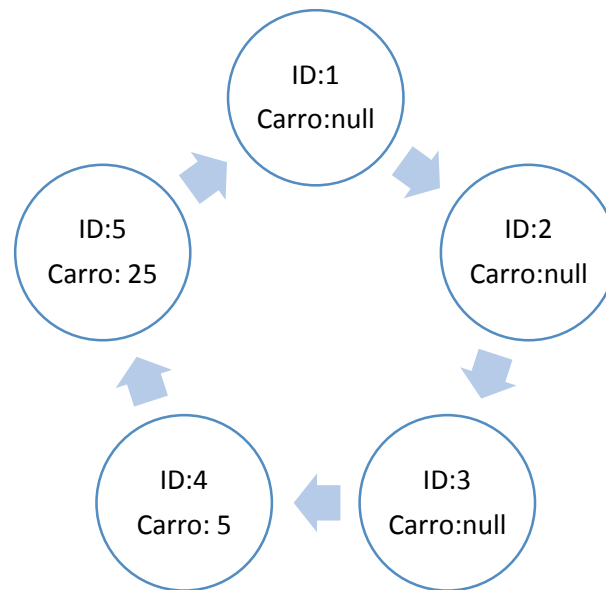


Figura 2. Parqueo

SUPERMERCADO, ESTANTERIAS y PRODUCTOS

Los pasillos del supermercado y estanterías serán representados por medio de una **lista circular doblemente enlazada**, cada nodo de esta lista representara un pasillo del supermercado, cada pasillo poseerá una **pila** de clientes y otra de productos, cada nodo de la pila de productos será una unidad de producto (e.g. 100 productos es igual a 100 nodos en pila), ya que los nuevos sistemas de ventas utilizan el sistema LIFO (Last In First Out) o UEPS (Ultimo en Entrar Primero en Salir), debido a que los productos que se ofrecen no son perecederos y la inflación de la economía es constante en los países donde se ubican las sucursales. El número de productos en cada estantería deberá de ingresarse por medio de consola, dicho número de unidades será el mismo para cada pasillo. Se podrá visualizar el estado del supermercado, y sus estanterías con productos, además su información, en cualquier momento. Así mismo será posible mover a los clientes alrededor de los pasillos en ambas direcciones utilizando su id, y realizar compras de los distintos productos que se encuentran en los estantes especificando el id del producto y su cantidad, los productos deberán de desaparecer de los estantes. Los clientes se podrán desplazar a lo largo de los pasillos e ingresaran a la pila de clientes de cada pasillo del supermercado, por lo que puede haber varios clientes en un pasillo.

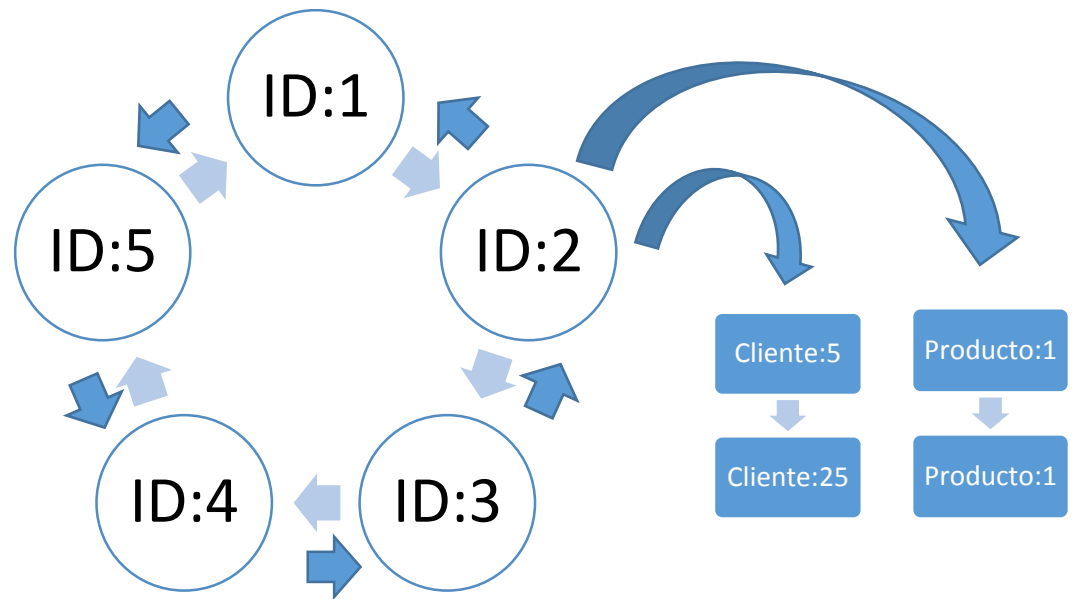


Figura 3. Pasillos, con pila de clientes y productos

CAJEROS

El sistema de cajeros constara de una **lista simplemente enlazada**, el número de cajeros se ingresara por medio de consola, y cada cajero contara con una **cola** donde se ubicaran los clientes, cada nodo de las colas será un cliente, y deberá de poderse visualizar en cualquier momento. Se deberá de atender a un cliente mediante el ingreso del id del cajero.

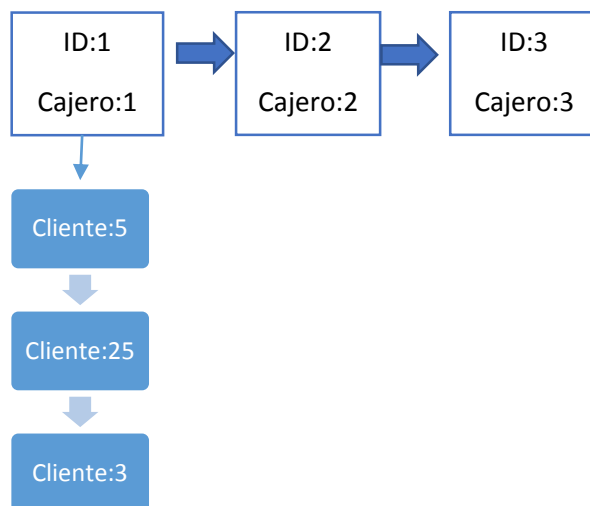


Figura 4. Cajeros, con cola de espera de clientes

CLIENTES

Los clientes tendrán un nombre y un id, que será igual al del carro en el que se encuentran, cada cliente debe de ser capaz de poder salir de su vehículo, y trasladarse hacia el supermercado y ocupar un lugar dentro del supermercado, la posición inicial del cliente en el supermercado será calculada aleatoriamente, el numero calculado será el id del pasillo en donde se posicionara el cliente. Las operaciones de mover al cliente, realizar compra, etc. deberán de realizarse mediante el ingreso del id del cliente.

VENTAS

Las ventas realizadas por todos y cada uno de los cajeros deberá ser almacenada en una **cola**, se podrá obtener una sub-lista de las compras que ha realizado un cliente, de cada producto en específico, y de un cajero en específico, dicha cola y sub-lista podrán ser visualizadas de manera gráfica en cualquier momento, la cola original de transacciones **NO** deberá ser alterada en ningún momento. Luego de realizar la compra el cliente regresara a su carro automáticamente

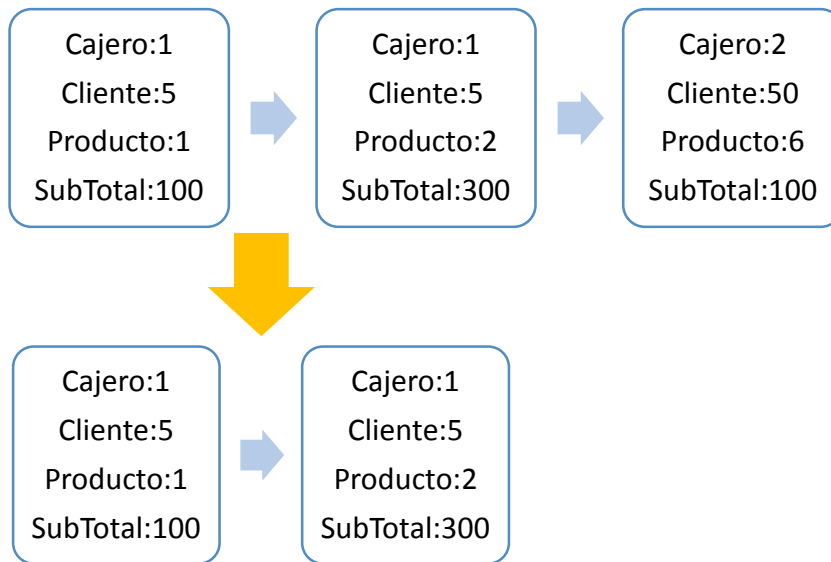


Figura 5. Registro de ventas, y sub-lista por cajero 1

VENTAS-PRODUCTO

Se tendrá una **lista simple enlazada "L1"** (naranja) de los productos que están o estuvieron a la venta en el supermercado, cada nodo de esta, contendrá una **lista simple enlazada "L2"** (verde) y sus nodos almacenaran la información de las ventas, como cajero, cliente, etc. Además cada nodo de esta lista apuntara al nodo vecino de la lista L1, al inicio los nodos de L2 tendrán un valor NULL, simulando una matriz dispersa, la lista L2 tendrá un tamaño estático de 4 columnas (sin incluir la columna de productos), la cual podrá ser mostrada en su forma matricial, y en su forma lineal (mapeo lexicográfico), tanto por filas, como por columnas.

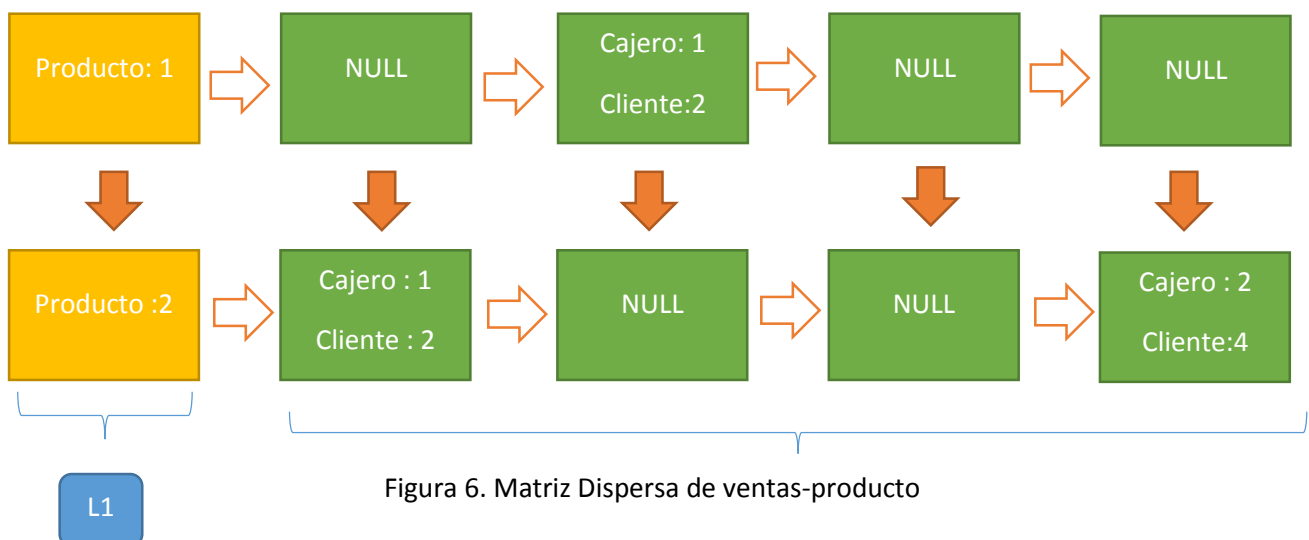


Figura 6. Matriz Dispersa de ventas-producto

El orden en que se ingresaran las ventas a la “matriz” será mediante la generación de un número aleatorio. Que tomara un valor de 1 – 4 y será la posición en L2, si ya existe un valor este será eliminado y se agregara la última venta realizada.

La forma lineal de la matriz tanto de filas y de columnas será realizada mediante una **lista doblemente enlazada**, por lo que habrán dos listas, una para filas y otra para columnas, y se podrá acceder a los datos de ellas por medio de una coordenadas i,j y j,i respectivamente las cuales se ingresaran por medio de consola, y se deberá demostrar la información correspondiente a cada nodo, de no poseer información deberá mostrar el siguiente mensaje: “Mensaje de ” + #carnet + “: Es un nodo NULO, no posee información”.



Figura 7. Forma Lineal de Matriz de Ventas-Producto, por filas

GENERALIDADES

- Se deberá mostrar el estado inicial de los sistemas (por defecto será nulo, o habrá un nodo cabecera) de la siguiente manera



- Las posiciones generadas aleatoriamente y los movimientos de los clientes deberán de ser registradas/mostradas en consola, y se deberá de especificar el id, la posición, etc. de la misma manera para los carros, ventas realizadas por cajeros, etc.

“El cliente #ID se ha movido al pasillo #3”

“El cliente #ID se ha movido del parqueo #2 hacia el pasillo #2”

“El carro #ID se ha movido al pasillo #3”

- Se debe de poder visualizar el sistema como un todo

ENTREGABLES

- Código Fuente
- Ejecutable
- Manual Técnico
- Manual Usuario

OBSERVACIONES

- Lenguaje a utilizar: C
- Sistema Operativo: Linux (Recomendado: Ubuntu 14.04)
- Queda terminantemente prohibido el uso de librerías propias del lenguaje, uso de arreglos, vectores, de detectarse el uso de estas se anulara la practica
- Las estructuras deben ser implementadas mediante el uso de **struct y punteros**
- La aplicación será compilada y ejecutada en el momento de la calificación con el compilador del sistema gcc
- Las gráficas deben de mostrarse dentro de la aplicación, sin necesidad de buscarse dentro del S.O.
- Se debe de utilizar una herramienta para generar las imágenes que contendrán las gráficas de las estructuras (Recomendado: Graphviz)

PENALIZACIONES

- Si existe un “null pointer exception”, o “segmentation fault(core dumped)”, se penalizara, a criterio del auxiliar
- Si la aplicación se cierra repentinamente existirá una penalización a criterio del auxiliar

FECHA Y MODO DE ENTREGA

- Miércoles 21 de Febrero de 2018 antes de las 23:59, entregas tarde serán penalizadas
- Se deben de enviar los entregables vía Dropbox, en un comprimido con el nombre: [EDD]Practica1_#carnet (e.g. [EDD]Practica1_201318613)
- Dropbox: <https://www.dropbox.com/request/ekFaYkdLs2sj7fKL6t2D>

REQUERIMIENTOS MINIMOS

- Las estructuras deben de ser mostradas mediante imágenes, no mostrarse por medio de texto en consola.
- De no subir los entregables vía Dropbox NO se calificara.

***Copias totales o parciales, tendrán nota de 0 puntos, serán reportadas al catedrático y a la Escuela de Ciencias y Sistemas FI-USAC**