Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2022.

PROYECTO #2

202111563– Jose Pablo Menard Pimentel

Resumen

En este proyecto se presenta un problema en el cuál se necesita un programa, que simula ser tanto una página web como una aplicación móvil, donde una empresa pueda administrar su servicio al cliente de una manera más eficiente y segura en todo aspecto, y esto se logra a través de listas ordenadas y clasificadas que administran los componentes de una respectiva empresa, ya sean los puntos de control, los escritorios de servicio, y las transacciones de los mismos. La aplicación móvil que se menciona anteriormente es principalmente pensada para que los clientes de dichas empresas, igualmente puedan clasificar todo tipo de acciones que deseen realizar referente a las empresas, más específicamente referente a los puntos de control, ya que, en teoría, gracias a las listas ordenadas que tendrán, tanto las empresas como los clientes, se podrá asignar de manera más específica cada acción, en este caso las transacciones, que el cliente desee realizar en los escritorios correspondientes a cada punto de control. La funcionalidad principal es organizar los turnos de los clientes para que sean atendidos de manera ordenada y en un tiempo optimo, el cual también deberá ser mostrado en el programa.

Palabras clave

- Estructuras
- Listas
- Archivos
- Empresas
- Clases

Abstract

In this project a problem is presented in which a program is needed, which simulates being both a web page and a mobile application, where a company can manage its customer service in a more efficient and secure way in all aspects, and this is achieved through ordered and classified lists that manage the components of a respective company, whether they are checkpoints, service desks, and their transactions. The mobile application mentioned above is mainly designed so that the clients of said companies can also classify all kinds of actions that they wish to carry out regarding the companies, more specifically regarding the control points, since, in theory, thanks to the ordered lists that both companies and customers will have, each action, in this case the transactions, that the customer wishes to perform at the desks corresponding to each control point can be more specifically assigned. The main functionality is to organize customer shifts so that they are served in an orderly manner and at an optimal time, which should also be shown in the program.

Key words

- Structures
- Lists
- Files
- Business
- Classes

Introducción

La base primordial de la realización de este proyecto radica en la Programación Orientada a Objetos, debido a que la creación de clases, instancias, atributos y métodos vienen de este paradigma de programación, aunque, por más que la raíz de este programa sea el paradigma POO, realmente no se centra en eso, ni mucho menos es lo que más destaca, ya que, por medio de listas enlazadas, las cuales están directamente relacionadas con las estructuras de datos, y con ayuda de la consola del IDE que se esté utilizando (en mi caso es Visual Studio), se presenta un programa bastante completo, el cual le presenta al usuario, que en este caso debería de ser un empresario o un cliente, dependiendo de las opciones que elija, una serie de opciones bastante concretas y sencillas de ejecutar, las cuales van desde una configuración inicial de los atributos que tienen todas las listas, hasta simples muestras para revisar los datos que están actualmente en el programa. El sentido de administrador se encuentra bastante presente en la parte de empresario del programa, ya que se pueden agregar listas, las cuales son: empresas, puntos de atención, escritorios y transacciones. Y como cliente también se muestran varias opciones útiles, ya que la aplicación debe simular de manera activa la atención al cliente que se recibiría en un lugar físico.

Desarrollo del tema

La primera situación que surge para crear este programa, por lo menos en mi caso al momento de empezar, es que es total y definitivamente necesario tener bien en claro que es lo que pide principalmente en el enunciado del proyecto, ya que al ser un enunciado tan extenso, muchas veces pequeños detalles importantes para la creación del mismo pueden quedar absolutamente de lado, no por nada, sino porque seguramente se pasará algo de largo al momento de leerlo, por lo que está claro que lo primero es analizar a fondo cada requerimiento, instrucción y consideración que se presenta en el enunciado del proyecto.

Luego de encargarse de ese proceso, es absolutamente necesario, con la misma cautela con la que se realizó el análisis del enunciado, es igualmente importante comenzar a estructurar, por lo menos en tu cabeza, la manera en la que se distribuirá el proyecto, específicamente hablando de las opciones que tiene, ya que al ser un enunciado orientado a una problemática real, se espera que el estudiante interprete por su propia cuenta todas las funciones y servicios que la aplicación final debe ofrecer al usuario que la utilice, por esta razón es importante comenzar por la organización de las funciones que tendrá la aplicación, lo cual claramente va a variar dependiendo del estudiante que esté realizando el proyecto.

En mi caso particular, al lograr estructurar de la mejor manera posible el desarrollo del proyecto orientado a la funcionalidad que presentará al usuario, decidí empezar de una manera poco ortodoxa, pero en este caso muy efectiva, ya que fue de suma ayuda para localizar tanto las variables como los concepto principales que necesita la aplicación para realizar todas las funcionalidades que el enunciado exige al desarrollador en este caso a mi persona, por lo que decidí empezar de esa forma, guiándome específicamente por la parte del enunciado en la cual indica el funcionamiento principal que se requiere del programa, esto incluye claramente tanto las partes de dicha sección del enunciado, como partes más rebuscadas, que se encuentran en la descripción completa de todo el sistema a desarrollar, por lo que es obligación del desarrollador, en este caso mi persona, encontrar todas y cada una de las pruebas, y no solo eso, sino que también es necesario identificarlas y estructurarlas de manera que sea funcional.

La manera en la que se abordó la resolución del proyecto, luego de aclarar el menú, el cual fue hecho por consola a base de ciclos simples, fue utilizar tanto la programación orientada a objetos como la estructura de datos, en este caso las listas enlazadas dobles y las pilas, para organizar todos los movimientos pertinentes al trabajo requerido, por lo que se debió utilizar mucha lógica relacionada directamente con la estructura de datos, puesto que prácticamente todo el proyecto se resolvía manipulando estas listas mencionadas anteriormente, de las cuales se brindará un poco más de información, primordialmente para contextualizar acerca de la solución que se utilizó.

Una gran ventaja de las listas doblemente enlazadas es que podemos usar un puntero a la celda que contiene el i-ésimo elemento de una lista para representar la posición i, mejor que usar el puntero a la celda anterior, aunque lógicamente, también es posible la implementación similar a la expuesta en las listas simples haciendo uso de la cabecera. El único precio que pagamos por estas características es la presencia de un puntero adicional en cada celda y consecuentemente procedimientos algo más largos para algunas de las operaciones básicas de listas. Si usamos punteros (mejor que cursores) podemos declarar celdas que consisten en un elemento y dos punteros a través de procedimientos para cambiar las estructuras principales de los datos ingresados.

Donde los trazos continuos denotan la situación inicial y los punteados la final. El ejemplo visto se ajusta a la supresión de un elemento o celda de una lista situada en medio de la misma. Para obviar los problemas derivados de los elementos extremos (primero y último) es práctica común hacer que la cabecera de la lista doblemente enlazada sea una celda que efectivamente complete el círculo, es decir, el anterior a la celda de cabecera sea la última celda de la lista y la siguiente la primera. De esta manera no necesitamos chequear para NULL en el anterior procedimiento borrar.

Por consiguiente, podemos realizar una implementación de listas doblemente enlazadas con cabecera tal que tenga una estructura circular en el sentido de que dado un nodo y por medio de los punteros siguiente podemos volver hasta él como se puede observar en la figura (de forma análoga para anterior).

Trabajar con varias posiciones simultáneamente tendrá un comportamiento idéntico al de las listas enlazadas excepto respecto al problema referente al borrado cuando se utilizan posiciones consecutivas. Es posible implementar la función de borrado de tal forma que borrar un elemento de una posición p invalida el valor de dicha posición p y no afecta a ninguna otra posición. Nosotros en nuestra implementación final optaremos por pasar un puntero a la posición para el borrado de forma que la posición usada quede apuntando al elemento siguiente que se va a borrar al igual que ocurría en el caso de las listas simples. Otra posible solución puede ser que la función devuelva la posición del elemento siguiente a ser borrado.

La inserción se debe hacer a la izquierda del nodo apuntado por la posición ofrecida a la función insertar. Esto implica que al contrario que en las listas simples, al insertar un nodo, el puntero utilizado sigue apuntando al mismo elemento al que apuntaba y no al nuevo elemento insertado. Si se desea, es posible modificar la función de forma que se pase un puntero a la posición de inserción para poder modificarla y hacer que apunte al nuevo elemento insertado. En cualquier caso, el comportamiento final de la función deberá quedar reflejado en el conjunto de especificaciones del TDA. (CCIA UGR , 2018)

En una lista doblemente enlazada, es posible, por ejemplo, eliminar un nodo, teniendo únicamente ese nodo, sin necesidad de saber también cuál es el anterior.

Entre las desventajas podemos mencionar que al tener que mantener dos referencias el código se vuelve más complejo, y también que ocupa más espacio en memoria.

Esta información es de suma importancia para entender de una manera realmente técnica, como fue que se realizaron las diferentes listas del proyecto, las cuales contienen tanto los elementos que se mencionan principalmente en el enunciado del proyecto, como los atributos y métodos necesarios para completar de manera satisfactoria todas las necesidades que el usuario necesita realizar.

Otra parte fundamental del proceso de desarrollo de este proyecto, que también está a su vez relacionada de manera muy estrecha con las estructuras de datos impartidas durante la clase magistral de curso, son las pilas, que también son listas enlazadas como las que se usaron a través de todo el proyecto, con unas pequeñas diferencias, las cuales tienen origen en la lógica que se emplea para el funcionamiento de las mismas, las cuales funcionan de manera muy similar a la que funcionaría una cola en un banco, ya que se toma el mismo concepto y se aplica a las necesidades del paradigma de programación, para reflejar este comportamiento en el programa, el cual, en este caso, es necesario para la parte tanto de “escritorios” como la de “clientes”, por lo que se brindará información adicional para tener más clara esta técnica, y sus diferencias con una lista normal.

Una pila es una lista ordinal o estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO que permite almacenar y recuperar datos. Esta estructura se aplica en multitud de ocasiones en el área de informática debido a su simplicidad y ordenación implícita de la propia estructura.

Para el manejo de los datos se cuenta con dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, retirar (o desapilar, pop), que retira el último elemento apilado.

En cada momento sólo se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado. La operación retirar permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al siguiente, que pasa a ser el nuevo TOS.

Por analogía con objetos cotidianos, una operación apilar equivaldría a colocar un plato sobre una pila de platos, y una operación retirar a retirarlo.

Una pila típica es un área de la memoria de los computadores con un origen fijo y un tamaño variable. Al principio, el tamaño de la pila es cero. Un puntero de pila, por lo general en forma de un registro de hardware, apunta a la más reciente localización en la pila; cuando la pila tiene un tamaño de cero, el puntero de pila de puntos en el origen de la pila.
Para el manejo de los datos se cuenta con dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, retirar (o desapilar, pop), que retira el último elemento apilado.

La pila de llamadas es un segmento de memoria que utiliza esta estructura de datos para almacenar información sobre las llamadas a subrutinas actualmente en ejecución en un programa en proceso.

Cada vez que una nueva subrutina es llamada, se apila una nueva entrada con información sobre ésta tal como sus variables locales. En especial, se almacena aquí el punto de retorno al que regresar cuando esta subrutina termine (para volver a la subrutina anterior y continuar su ejecución después de esta llamada).

A modo de resumen tipo de datos, la pila es un contenedor de nodos y tiene dos operaciones básicas: push (o apilar) y pop (o desapilar). 'Push' añade un nodo a la parte superior de la pila, dejando por debajo el resto de los nodos. 'Pop' elimina y devuelve el actual nodo superior de la pila. Una metáfora que se utiliza con frecuencia es la idea de una pila de platos en una cafetería con muelle de pila.
En esa serie, sólo la primera placa es visible y accesible para el usuario, todas las demás placas permanecen ocultas.
Como se añaden las nuevas placas, cada nueva placa se convierte en la parte superior de la pila, escondidos debajo de cada plato, empujando a la pila de placas. A medida que la placa superior se elimina de la pila, la segunda placa se convierte en la parte superior de la pila. (Ecu Red, 2011).

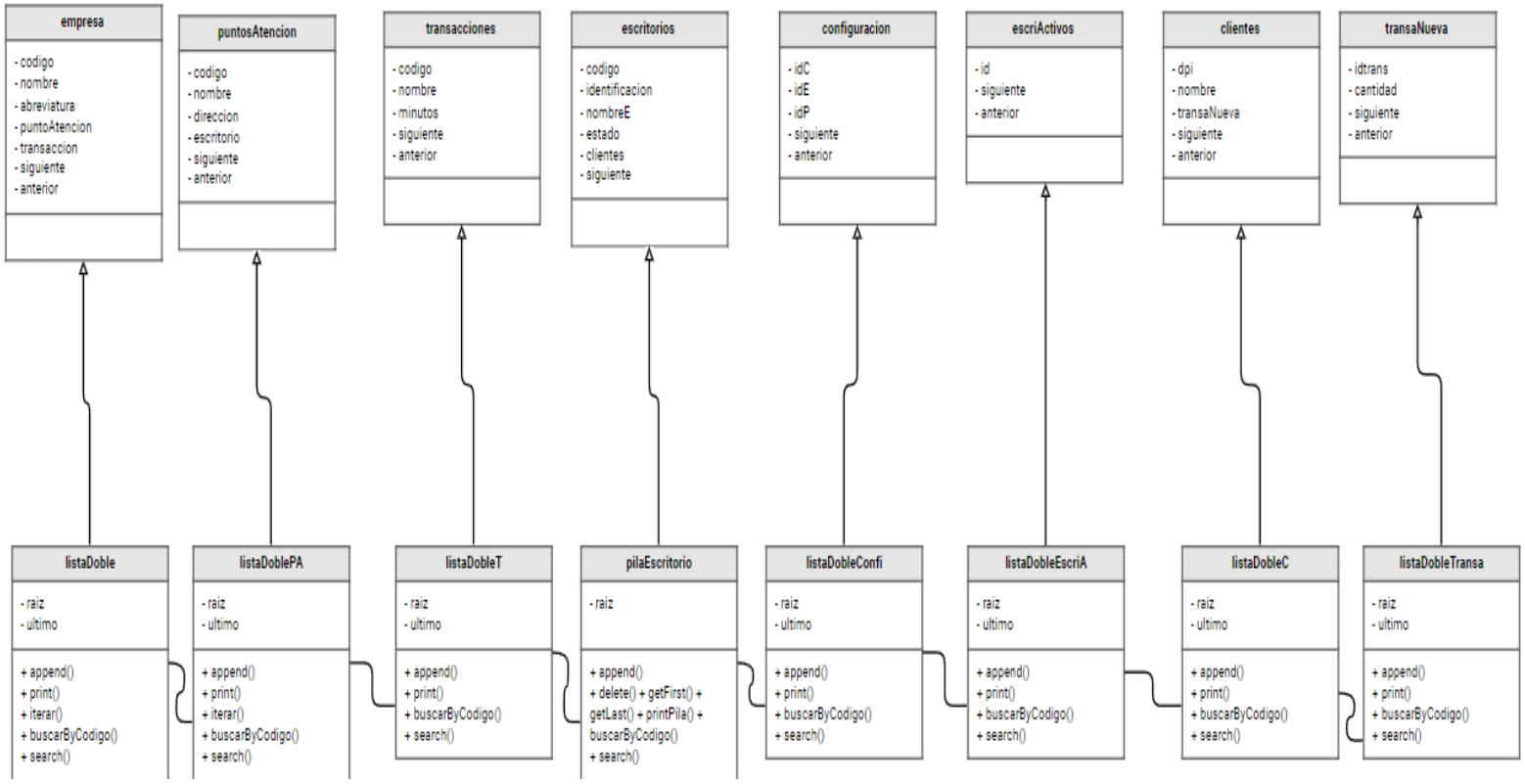
Un requisito típico de almacenamiento de una pila de n elementos es O(n). El requisito típico de tiempo de O(1) las operaciones también son fáciles de satisfacer con un array o con listas enlazadas simples.

Hay muchas variaciones en el principio básico de las operaciones de pila. Cada pila tiene un lugar fijo en la memoria en la que comienza. Como los datos se añadirán a la pila, el puntero de pila es desplazado para indicar el estado actual de la pila, que se expande lejos del origen (ya sea hacia arriba o hacia abajo, dependiendo de la aplicación concreta).

La pila es visualizada ya sea creciente de abajo hacia arriba (como pilas del mundo real), o, con el máximo elemento de la pila en una posición fija, o creciente, de izquierda a derecha, por lo que el máximo elemento se convierte en el máximo a "la derecha".

Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2022.

Diagrama de clases:



Conclusiones

Luego de realizar un análisis a profundidad de todo el desarrollo del proyecto, indagando también en los elementos básicos del mismo, siendo estos las listas enlazadas doblemente y las pilas, por medio de investigaciones más estructuradas, la primera conclusión contundente que personalmente puedo sacar de este proyecto es la importancia de la comprensión del funcionamiento básico de las estructuras de datos, que al final se remontan a la comprensión en sí misma de la programación orientada a objetos, la cual claramente es la raíz de todo el proyecto, lo cual da a entender, que para la resolución general de problemas los cuales requieren soluciones más específicas, es fundamental simplemente tener bien claros los principales conceptos básicos de los pilares de la programación y, en este caso en específico, el conocimiento bien entendido de las estructuras de datos también son importantes, pero el mensaje es que ciertamente el factor de la comprensión del desarrollador, mi persona en este caso, para llegar a organizar de una manera adecuada todas las soluciones que se le pretenden dar al usuario, por lo que concluyo que lo más importante de este proyecto, es la capacidad de análisis que se le exige al estudiante, para lograr una correcta comprensión del tema, con la intensión de finalmente, darle resolución al mismo.

Referencias bibliográficas

CCIA UGR . (23 de 06 de 2018). *Listas Doblemente Enlazadas*. Obtenido de https://ccia.ugr.es/~jfv/ed1/tedi/cdrom/docs/ldoble.html

Ecu Red. (07 de 03 de 2011). *Pila (Estructura de datos)*. Obtenido de https://www.ecured.cu/Pila_(Estructura_de_datos)