

Sistema de Administración de Colas

Tomas Apeltauer

Felipe Mora

Índice:

| | |
|---|----|
| Resumen ejecutivo | 3 |
| Introducción | 4 |
| Presentación y análisis del problema | 6 |
| i. Que es lo que hay que resolver | 6 |
| ii. Como se va a resolver el problema | 8 |
| iii. Análisis crítico de la implementación | 9 |
| Conclusiones | 10 |
| Recomendaciones | 11 |
| Referencias | 12 |

Resumen ejecutivo

El proyecto nace con la finalidad de poder utilizar un sistema de tiquetes implementado en colas con la intención de mejorar asuntos de vida cotidiana que por lo general se maneja en diferentes entes gubernamentales y minoritariamente en privados por tediosas y extensas hileras de personas para ser atendidos donde muchas veces su trámite ni siquiera coincide o tiene alguna relación.

El fin de este documento es tratar los puntos más específicos de posible interés tales como el planteamiento detallado del problema que se presenta, sus antecedentes y una manera muy completa de solucionarla realizada por el equipo de trabajo.

El análisis de este problema, la abstracción de conceptos hasta llegar a cada parte del problema y solucionarlo por aparte para formar la solución completa que se presenta y se plantea únicamente con el fin de mejorar el sistema para que sea más flexible y la mayor comodidad para ambas partes (usuarios y funcionarios).

Introducción

En muchas instituciones bancarias no solo estatales se utiliza un sistema donde un funcionario está a cargo de cajas, ventanillas o plataformas de recepción de personas, clientes o usuarios. Estos formados mediante una extensa hilera o un sistema numerado.

El proyecto nace con la finalidad de intentar resolver el problema en uno de los casos mencionados anteriormente; específicamente; cuando este problema es la extensa hilera para que todos lleguen ante el mismo funcionario sin importar el trámite a realizar o servicio a requerir.

Un claro ejemplo de esta problemática se da en el Banco Nacional donde solo se maneja un servicio de máximo cinco cajas (una para buzón nocturno y otra preferencial), tres plataformas y una ventanilla únicamente para entrega de tarjetas a nuevos cuentahabientes. En un día cuando se depositan las pensiones de los adultos mayores (atendidos en la misma ventanilla), se depositan becas donde los encargados deben de firmar un recibido, aunado a esto los dueños de negocios locales realizan depósitos de planillas o cambian efectivo por monedas y viceversa, y por último los usuarios que realizan tramites personales (depósitos, pagos de luz, agua, teléfono, cambios de cheques, y demás) en donde se atienden prácticamente todos los servicios mencionados en la misma área de servicios. Este problema se agrava aún más en sucursales donde el personal no es el suficiente, no es muy eficiente y/o no dispone los recursos tecnológicos para mejorar.

Explorado la problemática en este caso hipotético donde todas las variantes se pueden presentar el mismo día la única manera de hacer este proceso menos tedioso, cansado y con el fin de buscar más comodidad para los usuarios así como ayudar a los funcionarios en el cumplimiento de sus labores.

Una opción desde el punto de vista informático y para efectos de las habilidades de un programador es implementar un sistema de fichas ordenado según caja (ventanilla o plataforma) y servicio que se asocie a esta y que divida las cajas actuales según los servicios requeridos por el ente y por la demanda que requieran y utilicen la mayor cantidad de usuarios.

La principal funcionalidad de este sistema desde el punto de vista del usuario es que este pueda llegar al ente bancario; según el ejemplo; solicitar una ficha según la transacción que este necesite realizar ya sea una o más. En una pantalla o lugar claramente visible se indicara cual es cliente que está siendo atendido por cada caja y otra general donde se haga el llamado principal al usuario. El usuario durante el tiempo de espera podría sentarse donde guste ya que no es necesaria una hilera. Inclusive según el tiempo de espera promedio el usuario podrá calcular e ir a realizar otras diligencias rápidas en otros entes cercanos mientras su lugar está reservado gracias a su ficha (teniendo por sabido que una vez llamado sino se presenta este pierde su lugar y se verá forzado en el mejor de los casos a solicitar otra ficha).

En cuestiones del funcionario estará a cargo de una ventanilla la cual tendrá uno o más servicios que podría atender pero su espacio de trabajo sería el mismo, el usuario llegaría más relajado al momento de ser atendido y por ende sería más el trato a este.

En resumen se aleja a ambas partes de gran parte de estrés, se trata de dar un servicio más personalizado hacia el usuario, se evita la mala costumbre de “guardar” los campos a conocidos que apenas vienen llegando y se debe procurar darle al cliente un espacio cómodo donde pueda pasar el rato de manera agradable, lejos de las tediosas filas, inclusive llevando sol cuando la hilera es demasiado grande y la gente tiene que esperar afuera.

En fin para efectos más teóricos y con lo que respecta a este curso a continuación se detallaran esos aspectos en función del proyecto.

Presentación y análisis del problema

i. Que es lo que hay que resolver

Durante la etapa de análisis previo y la de montar la estructura base del programa el equipo de trabajo identifico diferentes sub problemas a trabajar definidos de la siguiente manera:

- ❖ Utilizar un menú que guie de forma efectiva, robusta, flexible y ordenada dentro de las diferentes funciones del programa.
- ❖ Crear un servicio a base de entradas del usuario. Donde estos vendrían a ser las posibles transacciones que realiza el usuario.
- ❖ Crear una lista de servicios que contenga los servicios creados por el usuario.
- ❖ Crear un tipo de ventanilla asociado a un servicio inicializando sus ventanillas de atención.
- ❖ Obtener el código del servicio y asociarlo al tipo de ventanilla.
- ❖ Crear una lista de tipos de ventanillas donde se guarden en memoria según se vayan creando.
- ❖ Inicializar la ventanilla con un código cargado según el tipo de ventanilla al que se le asocia.
- ❖ Inicializar un ticket al solicitar un ticket para encolarlo en la cola correspondiente.
- ❖ Imprimir las colas que existen dentro del sistema.
- ❖ Eliminar un servicio de la lista según su información.
- ❖ Eliminar un tipo de ventanilla de la lista tipo de ventanillas; eliminando así la cantidad de ventanillas asociadas al sistema.
- ❖ Reordenar los servicios existentes según las especificaciones que brinde el administrador.

- ❖ Atender a un cliente implica desencolar de la cola correspondiente según las indicaciones que dio el usuario.
- ❖ Atender a un cliente preferencial de la misma manera que se atiende un cliente común solo que desencolarlo de la cola prioritaria de cada tipo de ventanilla.
- ❖ Retornar al menú del sistema desde cada submenú.
- ❖ Retornar al submenú principal desde cada rama especializada.
- ❖ Hacer una pausa en el sistema para ver los cambios realizados y el resultado de las operaciones realizadas.
- ❖ Presentar estadísticas del sistema en total de tiquetes dispensados por ventanilla, atendidos, dispensados por tipo de servicio así como los preferenciales.
- ❖ Además el llevar una estadística del tiempo de espera promedio por ventanilla.

ii. Como se va a resolver el problema

El equipo en la búsqueda de la solución más eficiente inicio en el desarrollo de la plantilla del menú sugerida por el profesor para luego crear desde ahí en papel las posibles clases y el posible manejo de datos que se realizara.

Se define una jerarquía interna entre los servicios, tipo ventanilla y ventanilla para definir una dependencia planteándonos que para que exista una ventanilla debe de existir un tipo de ventanilla para asociársele, para crear un tipo de ventanilla se debe de crear un tipo de servicio para que se asocie a este tipo de ventanilla y por último para que todo esto exista debe de existir un servicio.

Luego de esto se implementó crear tipo ventanilla el cual su constructor crea un dos colas una para tickets preferenciales y el otro para tickets de usuarios comunes y corrientes, crea una lista para crear las ventanillas, le llama una función auxiliar al constructor para que este cree la cantidad de ventanillas necesarias las cuales se pasan por parámetros dentro del constructor y de la función de crear ventanillas.

A las ventanillas se les asocia el código del servicio para que sea un identificador cuando se impriman las colas.

La clase servicio se implementa únicamente para el manejo de la lista de servicios.

La clase servicios las listas que se crean en servicio.

La clase menú principal recibe, envía y revisa las opciones que ingresa el administrador y envía a banco controller de ser válidas las opciones.

La clase banco controller se encarga de manejar las instrucciones desde menú, crea los tipos necesarios y maneja todo lo que refiere al sistema en sí. Cabe resaltar que se utilizó el patrón singleton para el desarrollo de esta clase.

La clase tiquetes instancia a los tiquetes q se solicitan cuando se utiliza la opción solicitar tiquete.

Las estadísticas por falta de tiempo y coordinación no se pudo concluir de manera efectiva con un algoritmo plenamente desarrollado.

iii. Análisis crítico de la implementación

- Agregar estadísticas: falta de implementar todo el punto 5 que compete a estadísticas donde se debe hacer una variable de tiempo en cada tiquete que se inicialice con la hora del sistema en el momento en el que se solicita el tiquete y que se compare con el tiempo de atención. También las estadísticas pendientes en el trabajo.
- Realizar una interfaz gráfica agradable y fácil de usar para el usuario y para los funcionarios incluyendo para el administrador así como escalable para que no sucumba ante los errores que pueden ingresar los usuarios.
- Crear nueva clase que maneje la lista de tipo ventanilla para quitarle esa función al banco controler y hacerlo aún más abstracto.
- Utilizar archivos de texto para guardar el estado del sistema salvando así los tipos de servicios y las ventanillas para evitar el tedioso proceso de inicializarlas cada vez que se inicia el sistema.

Conclusiones

- El equipo de trabajo concluyo que se realizó una buena labor en cuestiones de principios de programación la abstracción de las funciones. El software se trató de hacer con el código lo más claro posible y escalable en caso de una segunda versión para el mejoramiento de detalles que otra persona fuera del proyecto tal vez pueda apreciar mejor.
- El utilizar el patrón singleton ayuda a manejar la lógica en un lugar y accesarla desde otra clase.
- El tema del proyecto es muy útil ya que lo podemos utilizar como un simulacro de lo que se puede realizar en la vida real en caso de los que gustamos optar por el software bancario.
- El proyecto ayudo a terminar de aplicar y conocer conceptos de pilas, colas y demás estructuras que se dieron en clases.

Recomendaciones

- Buscar abstraer lo máximo posible los diferentes sub problemas que este proyecto presenta.
- El diseño en papel de las clases es muy importante ya que nos da un esqueleto para el posible programa.
- Basarse en la plantilla del menú que brinda el profesor para abstraer los conceptos de tipo ventanilla y el de ventanilla por aparte.
- Tratar de iniciar todas las estructuras con el template <typename E> ya que nos brinda la misma estructura para cualquier tipo de elemento ya sean objetos o tipos nativos.
- Tratar de tener una buena comunicación de equipo con el compañero para entre los dos unir ideas y buscar lo más eficiente intercambiando conocimiento.

Referencias

- <http://superuser.com/questions/429693/git-list-all-files-currently-under-source-control>
- <http://stackoverflow.com/questions/2047465/how-can-i-delete-a-file-from-git-repo>
- <http://stackoverflow.com/questions/1143796/git-remove-a-file-from-the-repositorywithout-deleting-it-from-the-local-filesy>
- <http://www.daniweb.com/software-development/cpp/threads/196899/exit-not-declared-in-scope>
- <http://stackoverflow.com/questions/5973427/error-passing-xxx-as-this-argument-of-xxx-discards-qualifiers>
- <http://stackoverflow.com/questions/1549930/c-equivalent-of-java-tostring>
- <http://runnable.com/Us5yYlzcivFWAAYA/how-to-create-a-new-linked-list-for-c%2B%2B>
- <http://stackoverflow.com/questions/270947/can-any-one-provide-me-a-sample-of-singleton-in-c/271104#271104>
- <http://www.cplusplus.com/reference/cstdlib/exit/>
- <http://stackoverflow.com/questions/12691316/getline-does-not-work-if-used-after-some-inputs>
- <http://stackoverflow.com/questions/4506750/how-do-i-use-the-enum-value-from-a-class-in-another-part-of-code>
- <http://www.zeuscmd.com/tutorials/cplusplus/50-gettersandsetters.php>
- <http://stackoverflow.com/questions/1596432/getter-and-setter-pointers-or-references-and-good-syntax-to-use-in-c>
- <http://stackoverflow.com/questions/15604127/do-a-getter-for-an-object>
- <http://blog.slickedit.com/2007/11/c-tips-pointers-and-memory-management/>