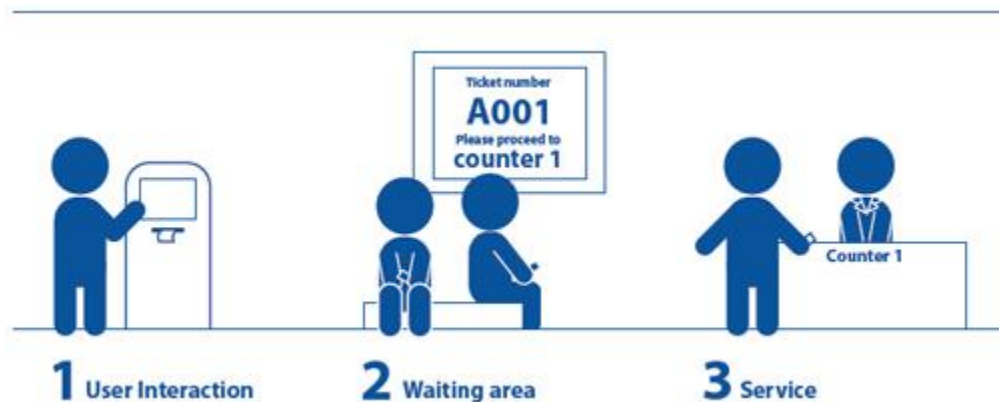


*Proyecto - Sistema de administración de colas*

**Introducción**

Esperar en una cola para ser atendido es parte inseparable de la experiencia de ser cliente de lugares como tiendas, hospitales, bancos e instituciones del gobierno. Mantener un orden de atención justo y acorde con las necesidades del local se convierte en todo un problema que puede resolverse con ayuda de la tecnología. Es por eso que, para este tipo de situaciones, los clientes usualmente esperen la existencia de un mecanismo estructurado de control de colas que permita llevar un orden y eviten la desorganización en la atención.

Para abordar este problema específico existen los sistemas de administración de colas. Comúnmente, este tipo de sistemas se crean mediante el uso de tiquetes numerados que permiten al cliente esperar libremente sin tener que esperar literalmente en una cola uno después del otro. Un sistema de administración de colas ayuda a organizar las colas dando a los clientes una solución virtual y métodos más sofisticados como aplicaciones móviles y mensajes SMS.



**Software a desarrollar**

Su trabajo consiste en implementar en el lenguaje C++ el núcleo de un sistema de administración de colas, el cual va a ser lo suficientemente flexible y configurable como para ser utilizado en diferentes tipos de locales o situaciones. También, elaborar una interfaz de usuario que sirva para simular la utilización del sistema, ver el estado de este, configurar y definir diferentes parámetros y consultar estadísticas de uso del sistema.

El sistema debe administrar lo siguiente:

- Lista de tipos de usuario
- Lista de servicios disponibles
- Lista de áreas

Los tipos de usuario son una clasificación inicial de cada persona que llega al local para ser atendida. Podemos pensar en tipos de usuario como “adulto mayor”, “persona con necesidades especiales”, “usuario regular” o “representante organizacional”. Cada tipo de usuario debe tener una descripción y un número que representa la prioridad para ser atendido, entre menor el número, mayor prioridad. El sistema debe manejar una lista de tipos de usuarios, ordenados según su prioridad.

Los servicios disponibles son las posibles operaciones que un usuario puede realizar en el local. Algunos servicios pueden ser operaciones como “comprar un boleto”, “cambiar un boleto”, “solicitar información”, o “realizar un reclamo”. Cada servicio debe tener una descripción, un número que representa la prioridad del servicio y el área donde el servicio es atendido. El sistema debe manejar una lista de servicios disponibles. Esta lista debe poder ser ordenada según se quiera configurar, no debe ordenarse automáticamente por prioridad.

Las áreas son secciones donde se atienden los diferentes servicios. De igual forma que en un banco existen grupos de ventanillas que realizan transacciones en caja (área de cajas) y grupos de ventanillas de servicio al cliente (área de servicio al cliente), el sistema debe ser capaz de configurar las áreas que se van a tener, con descripción, un código (un *string* que identifica el área) y una serie de ventanillas que posee el área. También, cada área va a tener una cola de prioridad que contiene los tickets que deben atenderse.

Cada área tiene una cantidad de ventanillas configurable. Cada ventanilla tiene un nombre que es el código del área concatenado con un consecutivo. Por ejemplo, si el código de área de cajas es “C” y posee 4 ventanillas, entonces los nombres de estas serían “C1”, “C2”, “C3” y “C4”. La ventanilla debe tener un espacio para mostrar el código del ticket que se está atendiendo actualmente.

Cuando un cliente llega al local, primero se acerca a una terminal donde el sistema le muestra los tipos de usuarios que hay disponibles para que elija el suyo. Luego le muestra la lista de servicios disponibles en el local. Al seleccionar el servicio que va a utilizar, el sistema crea el ticket y lo agrega en la cola correspondiente al área que tiene asociada ese servicio.

El ticket tiene un código que se genera concatenando el código del área donde será atendido y un número consecutivo global que aumenta con cada ticket que se solicita en el sistema. Este consecutivo debe iniciar en el valor 100. Por ejemplo, “C100”, “S102” y “E103” serían los códigos de tres tickets que se solicitaron para un servicio que se atiende en el área con código “C”, otro para el área con código “S” y otro para el área con código “E”. En ningún momento deberían generarse tickets con diferente código de área pero con mismo el consecutivo. Los tickets también deben tener asociada la hora en que fueron solicitados, ya que posteriormente se quiere saber el tiempo de espera para ser atendido. Otra cosa importante que debe asociarse al ticket es la prioridad final de este. La prioridad del ticket (PT) se calcula a partir de la prioridad del usuario que lo solicita (PU) y la prioridad del servicio que quiere solicitar (PS), mediante la siguiente fórmula  $PT = PU * 10 + PS$ .

En general, las funcionalidades principales del sistema son:

- Usuario solicita un ticket: se selecciona el tipo de usuario, se selecciona el servicio que desea usar el usuario, se genera un ticket con el código respectivo, la hora actual, la prioridad final y se inserta en la cola de prioridad del área asociada al servicio.
- Ventanilla atiende un ticket: se ingresa el código de la ventanilla que va a atender un ticket, se extrae un ticket de la cola de prioridad del área al que pertenece la ventanilla, se actualiza el código del ticket siendo atendido en la ventanilla y se registra la hora en que fue atendido.

- Administración del sistema: se pueden agregar o eliminar elementos de las listas de tipos de usuario, áreas disponibles y sus ventanillas y servicios disponibles. Al hacer cualquier tipo de modificación deben eliminarse los datos relacionados para que no haya inconsistencias en el sistema. En caso de que se borre un tipo de usuario o un servicio, deben eliminarse todos los tickets de las diferentes colas. Si se borra un área, deben eliminarse los servicios que son atendidos en esa área y las ventanillas que tenga asociadas. Siempre debe darse un mensaje de advertencia al usuario y solicitar confirmación antes de realizar cualquier borrado que implique eliminar otros datos.
- Consulta de estadísticas: se pueden consultar una serie de estadísticas de utilización del sistema:
  1. Tiempo promedio de espera por cada área. Esto hace necesario llevar control del tiempo que le toma a cada ticket en ser atendido y la cantidad de tickets que se han atendido en un área.
  2. Cantidad de tickets dispensados por área. Debe llevarse control de la cantidad de tickets dispensados para cada área.
  3. Cantidad de tickets atendidos por ventanilla. Esto implica llevar control de la cantidad de veces que una ventanilla específica ha atendido clientes.
  4. Cantidad de tickets solicitados por servicio. Debe llevarse control de la cantidad de tickets solicitados por cada servicio disponible.
  5. Cantidad de tickets emitidos por cada tipo de usuario.

A continuación, se presenta la estructura del menú sugerida para la interfaz del sistema. Cada opción debe facilitar la interacción con el usuario, haciendo que su utilización sea lo más sencilla posible. Pueden utilizarse números o caracteres sencillos para elegir diferentes opciones. El usuario no tiene porqué escribir códigos o nombres de elementos de memoria, ni recordar algo que ya no se observa en la pantalla. Si hay que elegir algún elemento, debe mostrarse una lista con los elementos para saber cuál es su código o posición. Obsérvese que diferentes opciones del menú tienen opción para regresar sin realizar ninguna acción.

Menú principal:

1. Estado de las colas  
Debe mostrar las áreas existentes, la cantidad de ventanillas definidas para cada área y los códigos de los tickets presentes en las diferentes colas. Cada ventanilla debe mostrar el código del último ticket atendido. Espera a que el usuario presione alguna tecla para regresar al menú principal.
2. Tickets
  - 2.1. Seleccionar tipo de cliente y servicio  
Esta opción muestra la lista de tipos de usuario para que se elija uno. Esta lista no debe mostrar las prioridades internas. El usuario selecciona el tipo de cliente que es. Posteriormente se muestra la lista de servicios disponibles en el local. Esta lista debe ser mostrada en el orden en que se configuró en el sistema, no debe ser fija, ni por áreas. El usuario selecciona uno de los servicios y el sistema genera el ticket según se explicó anteriormente y lo inserta en la cola de prioridad respectiva. El ticket se imprime en pantalla: código del ticket, hora de llegada, prioridad final del ticket, cualquier otro dato relevante. Luego de realizar la acción se mantiene en el menú de tickets.
  - 2.2. Regresar  
Regresar al menú principal sin realizar ninguna acción.
3. Atender  
Esta opción solicita al usuario el área y el número de ventanilla y busca en la cola respectiva el siguiente ticket a atender. Lo elimina de la cola y lo asigna como el ticket que se está atendiendo actualmente en la caja. Actualiza los datos necesarios para que se refleje en las estadísticas del

sistema. Si no hay ningún elemento en la cola respectiva, debe indicar que no hay usuarios en espera. Luego de realizar la acción, regresa al menú principal.

#### 4. Administración

##### 4.1. Tipos de usuario

###### 4.1.1. Agregar

Agrega un nuevo tipo de usuario con su nombre y su prioridad a la lista de tipos de usuario del sistema. La lista se mantiene ordenada por la prioridad de cada tipo de usuario. Luego de realizar la acción se mantiene en el menú de tipos de usuario.

###### 4.1.2. Eliminar

Elimina un tipo de usuario del sistema. Luego de realizar la acción se mantiene en el menú de tipos de usuario.

###### 4.1.3. Regresar

Regresa al menú de administración sin realizar ninguna acción.

##### 4.2. Áreas

###### 4.2.1. Agregar

Agrega una nueva área con su respectiva descripción, código y cantidad de ventanillas y la agrega a la lista de áreas del sistema. Crea automáticamente la lista de ventanillas con sus códigos. Luego de realizar la acción se mantiene en el menú de áreas.

###### 4.2.2. Modificar cantidad de ventanillas.

Se solicita un área existente dentro del sistema, se muestra la cantidad de ventanillas actuales y solicita un nuevo número de ventanillas. Borra las anteriores y las crea de nuevo según la solicitud del usuario. Luego de realizar la acción se mantiene en el menú de áreas.

###### 4.2.3. Eliminar

Elimina un área y todas sus ventanillas asociadas. Si hay servicios relacionados al área que se borra, estos deben eliminarse del sistema. Debe pedirse confirmación al usuario. Luego de realizar la acción se mantiene en el menú de áreas.

###### 4.2.4. Regresar

Regresa al menú de administración sin realizar ninguna acción.

##### 4.3. Servicios disponibles

###### 4.3.1. Agregar

Agrega un nuevo servicio, con su respectiva descripción, el área asignada para su atención y la prioridad del servicio. Luego de realizar la acción se mantiene en el menú de servicios disponibles.

###### 4.3.2. Eliminar

Se muestra la lista de servicios disponibles en el sistema y se elige uno para eliminarlo. Luego de realizar la acción se mantiene en el menú de servicios disponibles.

###### 4.3.3. Reordenar

Permite al usuario reordenar los elementos en la lista de servicios. Solicita el número de servicio a reubicar y la posición destino. Sólo hay una lista de servicios en todo el sistema, no se debe agrupar por área, ni por otro criterio. Luego de realizar la acción se mantiene en el menú de servicios disponibles.

###### 4.3.4. Regresar.

Regresa al menú de administración sin realizar ninguna acción.

##### 4.4. Limpiar colas y estadísticas.

Elimina los contenidos de todas las colas de prioridad, los tiquetes atendidos en ventanillas y los datos de las estadísticas que se hayan registrado hasta el momento. Es una reinicialización

del sistema sin alterar tipos de usuario, servicios, ni áreas. Luego de realizar la acción regresa al menú de administración.

#### 4.5. Regresar

Regresa al menú principal sin realizar ninguna acción.

#### 5. Estadísticas del sistema

Se muestran todas las estadísticas explicadas anteriormente de una forma clara y concisa. Se espera a que el usuario presione alguna tecla para regresar al menú principal.

#### 6. Salir

Abandonar el programa.

## Recomendaciones para la implementación

Es importante que se abstraigan las diferentes funcionalidades que se necesitan, para programarlas separadamente y poder reutilizarlas. Básese en la descripción del menú anterior para determinar cuáles son las funcionalidades que deben implementarse en el sistema.

Identifique qué entradas, salidas y restricciones tienen cada una de ellas. Identifique otras funcionalidades que le puedan ayudar a cumplir el objetivo de programar el sistema.

Aplique sus conocimientos de programación orientada a objetos para crear tipos de datos que pueda utilizar en las estructuras desarrolladas en el curso. No olvide la trinidad de C++: si escribe un constructor para una clase, entonces es necesario especificar la asignación y el constructor de copia.

Si necesita que los objetos de sus clases personalizadas sean ordenables, no olvide implementar los operadores de comparación.

Si necesita que los objetos de sus clases personalizadas se puedan imprimir, no olvide implementar el operador "<<".

Utilice subrutinas en el programa principal para organizar las diferentes secciones de la interfaz.

Verifique las entradas del usuario y utilice ciclos para volver a solicitar cualquier dato incorrecto.

Evite combinar el uso de cin y getline. Se recomienda getline ya que permite leer strings completos con espacios.

No descuide la administración de memoria. Libere la memoria de los objetos que se hayan creado en memoria dinámica y que ya no se van a utilizar más.

## Documentación

Como documentación interna, todo archivo (sí, todo archivo) incluido en el proyecto debe tener como documentación al menos un descripción general de sus funcionalidades, y el nombre las personas que la escribieron. Escriba código claro y conciso, trate de apegarse a los principios de código limpio para el código que escriba. Recuerde que es mucho más importante que su código sea claro y fácil de entender que si tiene pocas líneas.

En cuanto a la documentación externa, debe entregarse un documento en formato PDF con las siguientes secciones. Por favor, no copiar partes de este enunciado dentro de la documentación. En caso de que eso suceda, la sección evaluada tendrá una nota de cero.

1. Portada
2. Introducción. ¿Por qué se hace el proyecto y qué es lo que abarca? (1-2 páginas.)
3. Presentación y análisis del problema.
  - a. Descripción del problema. ¿Qué es lo que hay que resolver? Identificar subproblemas que deben resolverse en el proyecto y explicar cada uno de ellos. Por favor, no copiar y pegar partes de este enunciado (1 o 2 páginas).
  - b. Metodología. ¿Cómo resolvió el problema? Explicar la forma en que se resolvieron los diferentes subproblemas. Describir de forma técnica y detallada. Utilice diagramas para representar la estructura del proyecto (1 o 2 páginas).
  - c. Análisis crítico. ¿Qué se logró implementar? ¿Qué faltó? ¿Qué cosas se podrían mejorar de lo que se implementó? Desarrollar cada uno de estos aspectos (1 o 2 páginas).
4. Conclusiones: resoluciones puntuales tras el proyecto. Estas deben ser relacionadas con los aspectos técnicos del trabajo únicamente. Mínimo 10 conclusiones. (1-2 págs.)
5. Recomendaciones: consejos o advertencias que se derivan de las conclusiones. Lecciones aprendidas durante el desarrollo del proyecto. Recomendaciones para personas que tengan que hacer el mismo trabajo. También deben estar orientadas con aspectos técnicos de la tarea programada. Hacer una o más recomendaciones por cada conclusión. Mínimo 10 recomendaciones. (1-2 págs.)
6. Referencias. Deben incluirse en formato APA.

### ***Forma de trabajo***

El proyecto se desarrollará en parejas o tríos.

### ***Entrega***

El tiempo asignado para la tarea programada es de 2 semanas.

### ***Evaluación***

La tarea tiene un valor de 20% de la nota final, en el rubro de Proyectos Programados.

Desglose de la evaluación de la tarea programada:

Documentación:      30%

Programación:      70%

### ***Recomendaciones adicionales***

Pruebe cada funcionalidad individualmente. No implemente todo el programa sin verificar el funcionamiento por separado de cada una de sus partes. Esto dirige a errores que son más difíciles de encontrar.

Recuerde que el trabajo es en equipos, es indispensable la comunicación y la coordinación entre los miembros del subgrupo.

No dude en consultar diferentes fuentes para satisfacer las dudas. Aparte de las búsquedas en internet, asegúrese de exponer sus dudas a sus compañeros, profesor y conocidos que estudien la carrera; en la mayoría de las ocasiones es más provechosa conversación de 10 minutos entre personas que están trabajando en lo mismo que pasar horas buscando la respuesta a una duda de forma individual.

No deje la documentación para el final, es buena práctica ir desarrollándola durante todo el transcurso del proyecto. Recuerde que la documentación debe ser concisa y puntual, por lo que en realidad no toma mucho tiempo al realizarla de esta forma.

Plagios no serán tolerados en ninguna circunstancia. Cualquier intento de fraude será evaluado con una nota de cero y se enviará una carta al expediente del estudiante. Siempre escriba su propio código y documentación.

## ***Referencias***

Queue management system. (2014, March 21). In Wikipedia, The Free Encyclopedia. Retrieved 00:37, March 31, 2014, from [http://en.wikipedia.org/w/index.php?title=Queue\\_management\\_system&oldid=600570892](http://en.wikipedia.org/w/index.php?title=Queue_management_system&oldid=600570892)

Queue management systems. (2013). In QMatic, Customer experience management. Retrieved March 30, 2014, from <http://www.qmatic.com/en/Solutions/Business-Solutions/Queue-Management-Systems/>

SmartQueue. (2014). In SmartQueue, Integrated multimedia queue management system. Retrieved March 30, 2014, from <http://www.smartqueue.com.au/overview/smartqueue-integrated-multimedia-queue-management-system.html>