Instituto Tecnológico de Costa Rica

Administración de Tecnologías de Información

TI-2402 Algoritmos y Estructuras de Datos

Prof: Andréi Fuentes Leiva.

**Segunda Tarea Programada en Java**

Daniel Navarro Romero

Carné: 2014044199

Ernesto Brenes Calderón

Carné: 2014044135

Steven Rojas Cortes

Carné: 2014071312

Contenido

[Descripción del problema 3](#_Toc400997998)

[Diseño del programa 4](#_Toc400997999)

[Diseño de interfaz: 4](#_Toc400998000)

[Ingresar Datos del Cliente 5](#_Toc400998001)

[Liberar Cajas Ocupadas 6](#_Toc400998002)

[Uso de estructuras de datos 6](#_Toc400998003)

[Algoritmos de ordenamiento 6](#_Toc400998004)

[Librerías utilizadas 7](#_Toc400998005)

[Librería de interfaz gráfica java swing y awt 7](#_Toc400998006)

[Librería de generar Gráficos JFreechart. 8](#_Toc400998007)

[Librería de utilidad JCommon. 8](#_Toc400998008)

[Análisis de Resultados 9](#_Toc400998009)

[Objetivos Alcanzados 9](#_Toc400998010)

[Manual de usuario 10](#_Toc400998011)

[Compilación: 10](#_Toc400998012)

[Interfaz y funcionalidades 10](#_Toc400998013)

[Apartado 1: 13](#_Toc400998014)

[Apartado 2: 14](#_Toc400998015)

[Apartado 3: 14](#_Toc400998016)

[Apartado 4: 15](#_Toc400998017)

[Apartado 5: 15](#_Toc400998018)

# Descripción del problema

Partiendo del objetivo dado por el profesor para la tarea programada: "implementar un sistema de atención a clientes para el BAC San José. La idea es que el sistema les asigne citas de atención a los clientes, para evitar que tengan que hacer largas filas en las sucursales”, se puede deducir que las principales dificultades van a ser la manipulación de las citas, el control de las colas de prioridad, el uso de librerías requeridas para implementar cada funcionalidad, entre otras menores.

Una gran ventaja para esta tarea programada es la experiencia adquirida en la primera tarea programa, sobre el lenguaje java, Github y Ant. Otro punto muy beneficioso para este trabajo es el cambio de perspectiva con respecto a Github y Ant. Este cambio implicó dejar de ver a Github y Ant como requisitos de la tarea programada y empezar a percibirlos como grandes herramientas de apoyo para el desarrollo de este trabajo. Esta nueva perspectiva fue un gran beneficio, ya que el uso de Ant permitió cortar cualquier dependencia de los IDE’s de java. Por su parte Github suplió la falta de reuniones para programar, debido a la gran cantidad de trabajos de cada integrante del grupo. Sin embargo se realizaron un par de reuniones, por considerarlas de vital importancia por parte del equipo de trabajo.

Aunque este trabajo genera diversas tareas por realizar, algunas tareas ya conocidas permitirán la retroalimentación de dichas herramientas en trabajos realizados anteriormente, con el fin de ejecutar este trabajo de manera más dinámica y eficiente. Aunque estas herramientas mencionadas anteriormente faciliten la elaboración de nuestro trabajo, uno de los principales problemas, se trata en la implementación de nuevos métodos que permitirán ciertas funcionalidades requeridas para la tarea. El uso de nuevas estructuras tales como de cola de prioridad, en la cual se deben asignar la posición a cada cliente según sea el tipo correspondiente, ya sea discapacitado, que abarca el tipo de persona que será atendida primeramente, seguido por adultos mayores, mujer embarazada, cliente corporativo y por último cliente regular.

Un reto significativo e importante para la ejecución de nuestra tarea consistía en la forma de enviar correos electrónicos, ya que para ello se requería configurar el correo emisor, y por ende se debía utilizar un servidor SMTP y POP; para ser un poco más claro, al momento de enviar un correo electrónico estamos utilizando como medio un servidor SMTP y POP que es el encargado de hacer llegar el correo a su destino, lo podemos comparar con el servicio postal, para hacer entrega del correo necesitamos de cuatro datos importantes el origen, el destino, el medio que es el servidor, y el puerto que corresponde al que indicamos según las opciones a la hora de configurar nuestro correo.

# Diseño del programa

## Diseño de interfaz:

Estructura de ventanas de la interfaz

Ingresar número de cajeros

Ventana principal

Gráfico de barras

Gráfico Pastel

Tabla Historial

Gráfico de barras

En total son 5 las ventanas utilizadas, ya que la ventana ingresar numero de cajeros es parte de la ventana principal. Cada ventana tiene su propio archivo .java. La ventana inicial permite al usuario indicar la cantidad de cajeros que se van a establecer para la sucursal en la que se implemente. Posteriormente una segunda ventana que va a contener la cantidad de cajeros disponibles, la asignación de turnos, el tipo de cliente y la liberación de cajas.

Tanto la lista total de cajas y la lista de cajas ocupadas se muestra en un Jlist cada una. En el lado superior izquierdo se encuentran los datos que se deben ingresar para asignar turnos a los clientes. En un JcomboBox se encuentran las categorías de los clientes: discapacitado, adulto mayor, mujer embarazada, cliente corporativo y cliente regular. Cuando se presiona una caja ocupada, esta muestra los datos del cliente que está siendo atendido en los campos de texto que se encuentran en la parte inferior del Jlist.

En esta misma ventana se encuentra el botón de liberar caja, el cual desocupa alguna caja seleccionada y asigna a un nuevo cliente, según las prioridades de colas establecidas. También se encuentran los botones Historial y Estadísticas, el primero abre una ventana con un historial de los clientes atendidos, esta ventana se detalla más adelante. El botón estadísticas abre una ventana con el historial de clientes atendidos por medio de gráficos pastel y barras.

La ventana con la tabla del historial presenta los datos: tipo de cliente, nombre, correo electrónico, fecha y hora de atención. Dichos datos son almacenados en un archivo de texto de forma cronológica, tomando la fecha actual del sistema.

La tabla muestra los datos de forma cronológica predeterminadamente, pero un JcomboBox permite cambiar el orden. Se puede ordenar por nombre, donde serán ordenados alfabéticamente y ordenar por correo electrónico, que funciona igual a ordenar por nombre. Cuando se ordena por tipo de cliente el sistema los ordena de mayor a menor prioridad, en el orden mencionado anteriormente.

Cabe recalcar que los datos siempre quedan ordenados de forma cronológica aún cuando se ha visualizado otros órdenes.

Luego existe una ventana con un gráfico pastel que muestra los porcentajes del tipo cliente. Otro de los tres gráficos muestra la misma información, pero en gráfico de barras. El otro gráfico permite ver los clientes del banco, mediante métodos.

## Ingresar Datos del Cliente

En el momento que los clientes van ingresando al sistema de cajeros del banco, el guarda de seguridad debe indicar para cada uno su nombre, el correo electrónico y el tipo de usuario.

1. Cuando el guarda de seguridad ingrese a la ventana, primeramente deberá ingresar el nombre completo del cliente.
2. Posteriormente se indicará el correo electrónico, es importante que este sea registrado de manera correcta, con el fin de enviar al cliente su información y cajero correspondiente.

1. Por último, el guarda de seguridad va indicar el tipo de cliente, este con el fin de establecer la prioridad en la que va ser atendido.
2. Luego de presionar el botón agregar, estos datos son almacenados en una lista correspondiente al tipo de cliente.
3. Posteriormente los datos son almacenados en el archivo de texto.
4. Por último se envía un correo electrónico al cliente confirmando los datos ingresados.

## Liberar Cajas Ocupadas

Para la liberación de cajas el guarda deberá seleccionar la caja que se desocupará, dicha caja se le asignará a un cliente en caso de que haya al menos uno en espera, en el caso de que haya más de uno, la caja que se desocupe, se le va a establecer al cliente según corresponda su prioridad. Dicha caja, una vez liberada desaparece del Jlist de cajas ocupadas, pero si existen clientes en espera, entonces la caja vuelve ha aparecer con los datos del nuevo cliente.

## Uso de estructuras de datos

En el caso de estructuras de datos implementadas, se utilizaron listas doblemente enlazadas por nodos. Estas listas son 2, una almacena la lista de clientes y la otra almacena la lista de cajas. En ambas listas se implementaron manualmente todos los atributos y métodos requeridos para que se acoplaran perfectamente a las funcionalidades requeridas.

Dichas listas funcionan como colas, es decir, el primer elemento ingresado es el primero en salir, pero esto solamente se cumple en la lista de los clientes. En el caso de la lista de cajas, no necesariamente la primera caja en ocuparse va a ser la primera en desocuparse, esto depende de la duración de los trámites que haga cada cliente.

## Algoritmos de ordenamiento

El único algoritmo de búsqueda implementado es el BubbleSort, el cual se utiliza cuando se necesita reordenar los datos de la tabla historial. Este algoritmo usa el método compareTo, para determinar qué valor es mayor, ya sea por nombre de cliente, tipo de cliente, correo electrónico. Solamente en el caso de fecha y hora es que no se utiliza el BubbleSort por el hecho de que los datos siempre permanecen ordenados cronológicamente. La tabla al ser una matriz va a ser una lista que contiene sublistas, por lo tanto cuando se necesita intercambiar datos, se intercambia toda la sublista a la cual pertenece el elemento que se necesita mover. Cuando se mueve una sublista “n” a la posición de una sublista “m” es necesario eliminar la sublista “n+1”, que es igual que sublista “m”. Esto es necesario, ya que la sublista “m” no es remplazada por la sublista “n”, solo se corre de posición.

# Librerías utilizadas

## Librería de interfaz gráfica java swing y awt

Son librerías que permiten crear objetos gráficos, agregarles o capturar eventos, ingresar, obtener y mostrar información. Estas librerías son propias de java, es decir, no se tiene que descargar por separado. Entre los objetos usados de estas librerías están:

1. JFrame: Permite crear ventanas. Se uso heredando la clase interfaz de la clase JFrame.
2. JButton: Permite crear botones. Se usaron métodos para especificar el tamaño de los botones.
3. JLabel: Sirve para crear etiquetas, ya sea con texto o imágenes. Se utilizaron métodos para cambiar el tamaño, color y tipo de letra.
4. JTextField: Funcionan para crear campos de texto. Se usaron para que el usuario digite información. Mediante métodos se extrajo información y también se mostró información.
5. JComboBox: sirve para crear campos de textos con opciones desplegables. Se hizo usó de dos JComboBox, uno para mostrar la cantidad de cajas disponibles que se van a utilizar, y otro con el fin de identificar los cajas que están siendo ocupadas.
6. JList: sirve para crear campos de lista. Se utilizaron métodos para extraer el dato seleccionado, el índice del dato seleccionado, agregar datos al final y eliminar datos.

## Librería de generar Gráficos JFreechart.

Podemos descargar gratuitamente esta librería de internet y nos permitirá realizar los gráficos en dos y tres dimensiones que más nos convengan para cada resultado de un informe, en este caso para obtener la cantidad de tipos de clientes que se han atendido.

Algunas de las características más importantes de esta librería son las siguientes:

* Gran variedad de tipos de gráficos. Entre los tipos de gráficos disponibles están los habituales de barras, líneas o tipo pastel, pero existen otros no tan habituales como los gráficos de Gantt, histogramas, de nubes de puntos, de áreas y muchos otros que amplían enormemente las posibilidades.
* Dispone de una buena documentación en forma de API.
* Está diseñada para que sea flexible con el fin de extenderla según más le convenga al usuario.
* Debido a su calidad de gratuita existe mucha información sobre ella en internet y eso facilita la labor de programar con ella y resolver dificultades.

## Librería de utilidad JCommon.

Jcommon, es una colección de clases utilizadas por JFreeChart, y en general para cualquier tipo de aplicación Java. Esta librería incorpora una gran cantidad de clases de utilidad, que nos pueden ser muy útiles en nuestros desarrollos.

Entre estas utilidades, por ejemplo está la del centrado automático de ventanas en la pantalla, y una gran cantidad de utilidades más.

Librería de envió de correos electrónicos Java Mail

Esta librería nos permite enviar y recibir correos electrónicos por medio de un servidor SMTP, sin embargo en el presente trabajo nada mas haremos uso de la disponibilidad de enviar correos electrónicos.

En nuestra implementación utilizaremos la cuenta de de Windows live, así que los datos de configuración que se muestren en el código es para dicha cuenta. Si se tiene otro servidor de correo, se tendrá que mirar cómo es la configuración con tu servidor y cambiar los parámetros adecuados en la configuración de JavaMail.

La configuración de la cuenta para el envío de correos corresponde:

Requiere usuario y una contraseña. El usuario es la cuenta de correo, indicaremos por ejemplo, ejemplo@live.com

Protocolo smtp

TLS si está disponible

Puerto 587.

Clases y configuración de JavaMail para el envío de mensajes, para nuestro trabajo, hay tres clases de JavaMail implicadas:

Session: Representa la conexión con el servidor de correo. Hay que obtenerla pasándole los parámetros de configuración para el servidor de correo que estamos utilizando en nuestro caso live.

Transport: Clase para el envío de mensajes.

MimeMessage: El mensaje.

Multipart: ####

BodyPart:####

# Análisis de Resultados

## Objetivos Alcanzados

1. Cumplimiento de todas las funcionalidades del sistema de cajeros, establecidas por las especificaciones generales de la tarea.
2. Registro de información de los clientes que ingresan al sistema de cajeros del Bac San José, tal como su nombre, correo electrónico y tipo de cliente.
3. Disponibilidad de insertar la cantidad de cajeros disponibles para utilizar.
4. Asignación de cajeros a los clientes, con la funcionalidad de prioridad a la hora de asignar el cajero que se identifique como libre.
5. Liberación de cajeros, al seleccionar el cajero que se desea asignar como disponible, el sistema de forma automática establecerá el cajero al cliente que se encuentre en la cola, según su prioridad como tipo de cliente.
6. La implementación del historial en formato tabla para la presentación de datos de los clientes atendidos junto con el orden en que se desean mostrar.
7. Representación por medio de gráficos estadísticos, con el fin de demostrar la cantidad de clientes atendidos según su tipo (discapacitado, adultos mayores, mujer embarazada, cliente corporativo o cliente regular).
8. La implementación de la lista por medio de colas fue un gran logro, ya que se presentaron muchas dificultades en el proceso. Esto se debió a que la mayoría de las funcionalidades dependían fuertemente de la buena implementación de la lista y sus métodos para establecer los métodos de prioridad.
9. Un objetivo que se alcanza con cada tarea realizada es la adquisición de nuevo conocimiento o fortalecimiento del que ya se tiene. En este trabajo, dicho objetivo se cumplió a grandes rasgos porque se obtuvo conocimiento en el uso de nuevas herramientas, fortalecimiento sobre el conocimiento del lenguaje de programación java, estructuras de datos y la programación orientada a objetos.

## Objetivos no alcanzados

1. No implementar un método más eficiente a la hora de actualizar las modificaciones hechas a la tabla y a los gráficos en tiempo de ejecucción. El método utilizado para regrescar los datos de ambos elementos fue ocultar y desocultar la ventana, con setVisible(false) y setVisible(true).

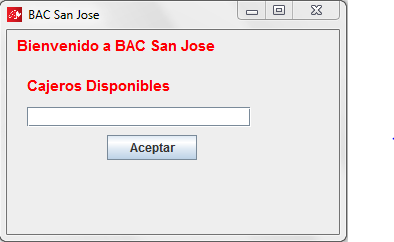
# Manual de usuario

## Compilación:

Para compilar está tarea programada solo es necesario ingresar a la carpeta del proyecto, desde la terminal de ubuntu o alguna otra distribución de Linux. Posterior mente se escribe el comando ant y el programa iniciará.

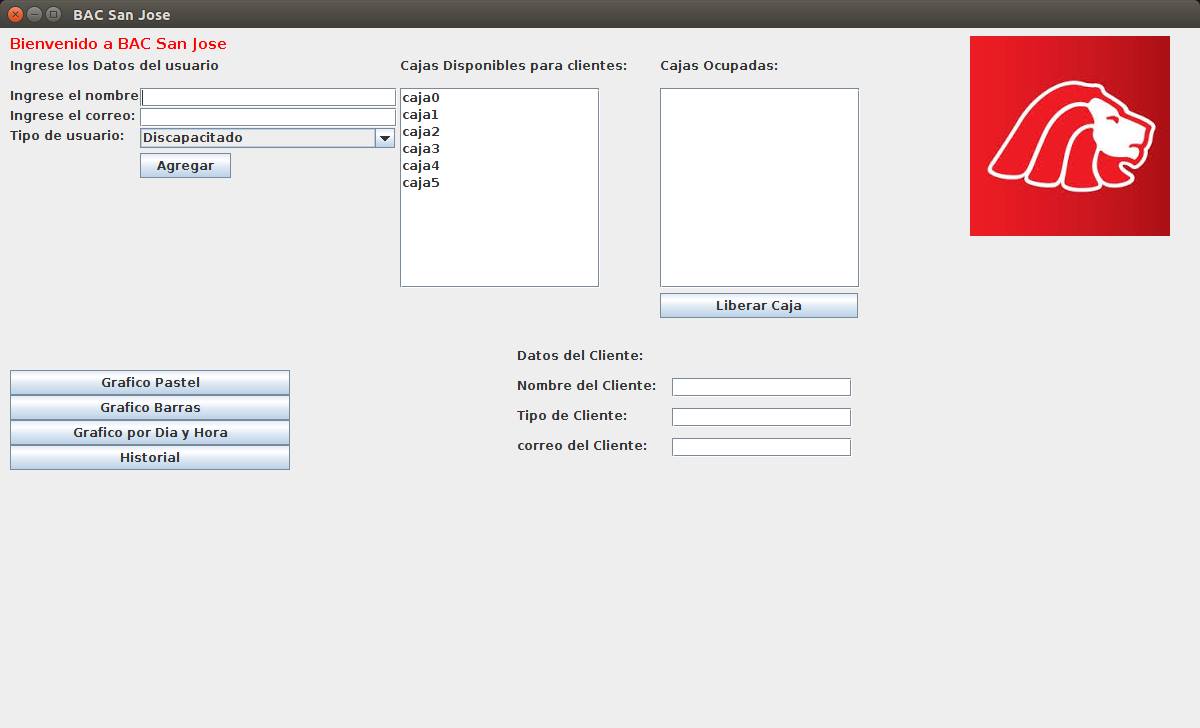
## Interfaz y funcionalidades

Cuando se ejecute el programa, se deberá indicar el número de cajas con disponibilidad para utilizar. Seguidamente se muestra la ventana inicial del sistema.



Aquí se ingresaran la cantidad de cajeros y presionamos aceptar.

Posteriormente que se seleccionan los cajeros disponibles, se nos muestra una ventana con todos los elementos necesarios para el registro de los clientes, asignación y liberación de cajeros, además con las opciones de mostrar el historial con los datos de clientes atendidos y los gráficos estadísticos para visualizar los porcentajes con respecto al tipo de cliente atendido. En la siguiente imagen se aprecia esta ventana:



**1**

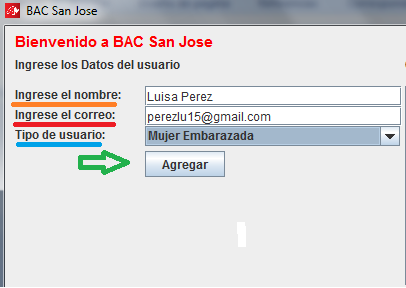
**3**

**2**

**4**

**5**

### Apartado 1:



Como se muestra en la figura, el guarda deberá ingresar los siguientes datos de cada cliente al entra al sistema de cajeros:

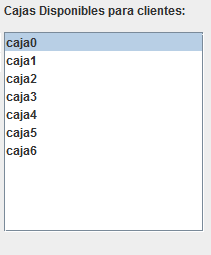
En este campo, se indicara el nombre y apellidos del cliente.

Aquí se va a establecer el correo electrónico del cliente, es importante dar el correo de forma adecuada, ya que por este medio se le proporcionara la información propia al cliente y su el numero de caja que debe ubicarse.

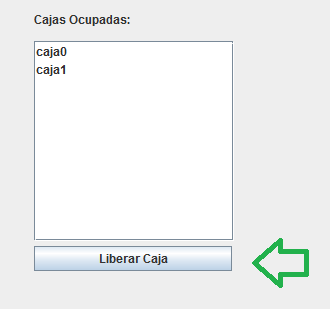
Seleccionar el tipo de cliente, preferiblemente para establecer la prioridad de atención.

### Apartado 2:

Como se puede apreciar en la figura de al lado, se muestra la cantidad de cajeros disponibles para asignar a los clientes del banco, si que el paso mencionado anteriormente se concreta, se seleccionara el cajero disponible al cliente y presionamos la opción de agregar.



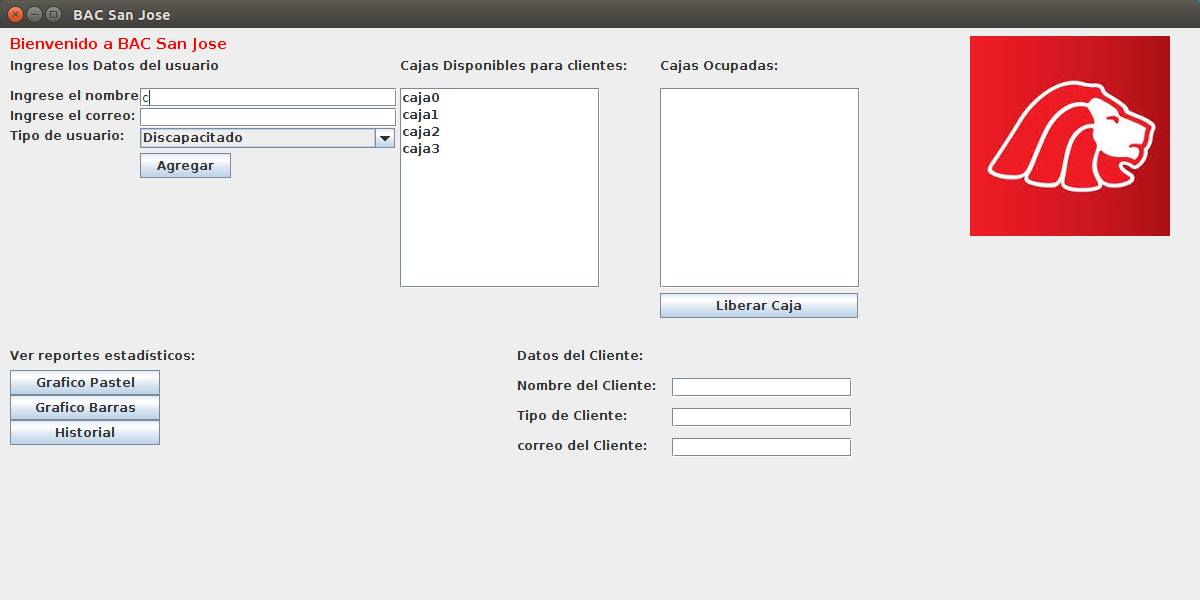
### Apartado 3:



Seguidamente se manifiestan los cajeros que se encuentran ocupados, para elegir el cajero a liberar, se selecciona como la flecha pequeña lo identifica, y posteriormente se presiona la opción “Liberar Caja”.

### Apartado 4:

En esta sección se muestran los datos del cliente que se encuentre en una de las cajas. Para visualizar estos datos solo es necesario seleccionar una de las cajas.

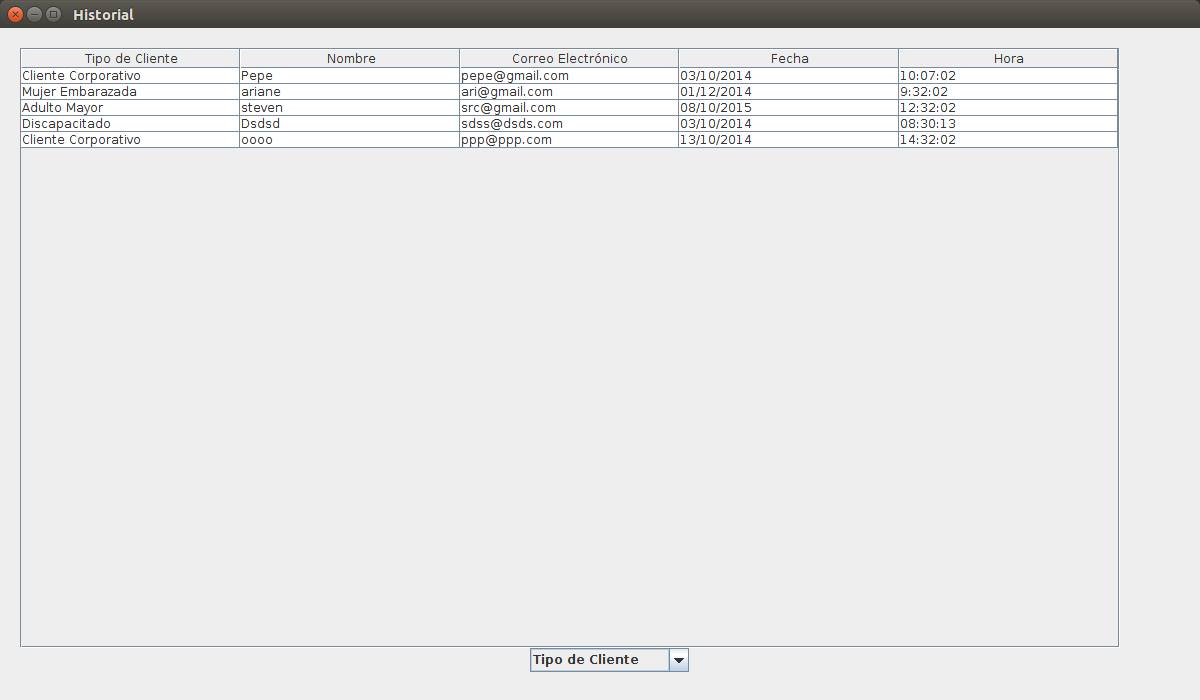


### Apartado 5:

En esta sección se encuentran los botones: gráfico de barras, gráfico pastel y el historial.

Cuando se presiona alguno de estos botones se nos abrirá una ventana que muestra el gráfico de barras, el gráfico pastel o la tabla de historial, según corresponda. Estas tres ventanas se muestran a continuación, según el botón presionado:

#### Botón Historial:



Como se observa en la imagen anterior, el botón Historial abre una ventana con una tabla. Dicha tabla contiene las columnas tipo de cliente, nombre de cliente, correo electrónico, fecha y hora. En la parte inferior de la tabla se puede observar un JcomboBox que permite ordenar los datos según la columna seleccionada. A continución se muestra como se ordena, según cada columna.

Nombre o correo electrónico de Cliente

Tipo de Cliente

Más reciente

Más antigua

Fecha u hora

Z

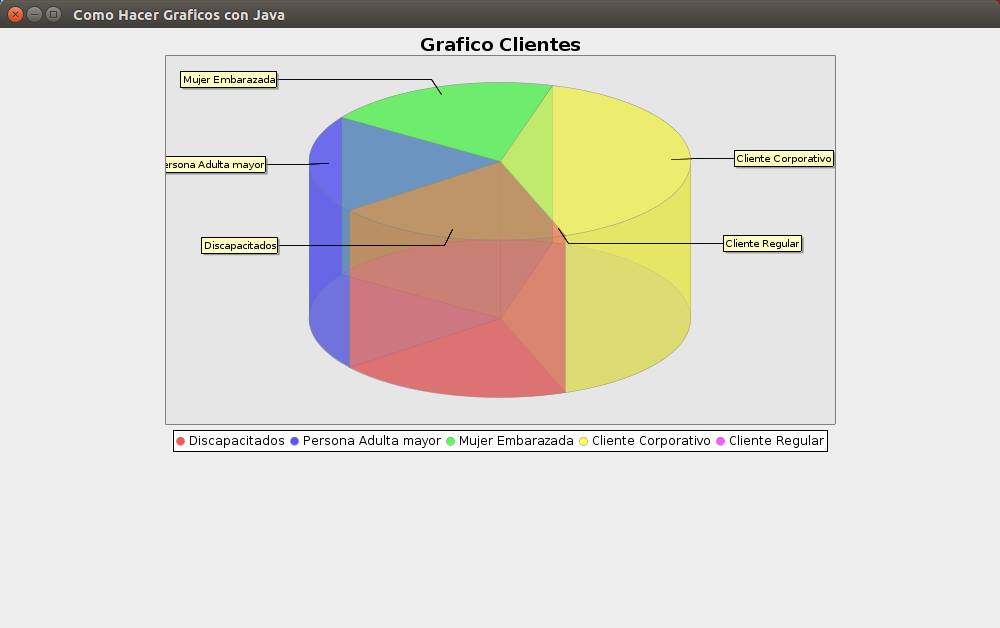
A

Menor prioridad

Mayor prioridad

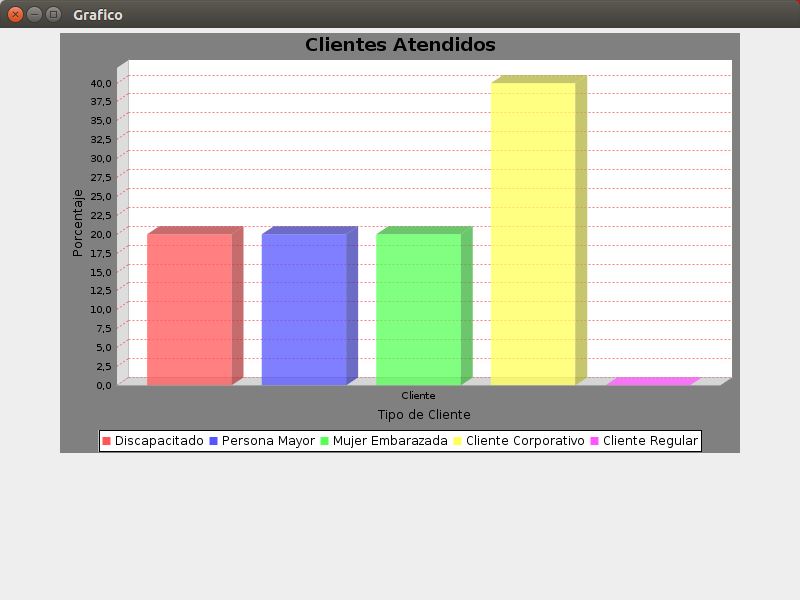
#### Botón Gráfico Pastel:

En esta ocasión se puede ver lo siguiente:



Botón Barras:

Muestra la siguiente información:



Botón Gráfico por fecha y hora:

Este gráfico permite visualizar la cantidad de clientes atendidos por hora. Este gráfico se presenta en la siguiente figuar.

