



Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica

DOCUMENTACIÓN

Escuela de Computación

Administración De Tecnologías De Información

TI-2402 Algoritmos y Estructuras de Datos

Profesor: Andrei Fuentes Leiva

Tarea Programada #2

Título: "Sistema Citas Bancario"

Lenguaje de Programación: Java

Integrantes:

Guillermo Ávila

Marvin Fernández Coto

Carlos Ramírez Cerdas

Tabla de Contenido

• Descripción del problema.....	3
• Solución al problema:.....	4
• Diseño del programa:.....	4
• Decisiones de diseño:.....	4
• Diagrama lógico.	5
• Objetivos alcanzados:	6
• Objetivos no alcanzados:	6
• Librerías usadas	6
• Manual de usuario	7
• Conclusión personal.	22

Descripción del Problema:

Se creará un programa para Sucursales Bancarias en este caso se estará trabajando con el BAC San José. Se usaran las estructuras de datos vistas en clase para implementar las funciones pedidas, además se usaran algoritmos de ordenamiento para facilitar el eso de los reportes en una tabla dinámica. El programa también chonta con dos tipos de gráficos los cuales permiten observar diversas características entre las cuales destacan los tipos de clientes, cantidad de clientes atentados por hora y por fecha.

El programa consiste en un sistema que va a ser utilizado por el oficial de seguridad de la sucursal bancaria, el cual les asignara una ficha electrónica a los clientes para evitar que estos tengan que realizar filas en el lugar. Una vez este desocupado algún cajero el programa le asignara un cliente al cajero de acuerdo a ciertos factores, el más importante es un orden de prioridad que va de la siguiente manera:

Persona con Discapacidad

Adulto Mayor

Mujer Embarazada

Cliente Corporativo

Cliente Regular

Todas las advertencias se enviaran por medio de un correo electrónico el cual será agregado a la hora de solicitar la cita. El lenguaje de programación empleado será Java y se ejecutara en el sistema operativo Linux.

Solución al problema.

Se irá detallando como se le va a dar solución al problema, donde indicaremos el diseño del programa, las decisiones que se tomaron, los algoritmos que se usaron entre otros aspectos.

Diseño del Programa

El programa contara con varias pantallas para la facilitación, se iniciara con una pantalla principal la cual tendrá un mensaje de bienvenida en la parte superior, además permitirá agregar el nombre del banco, el numero de cajeros disponibles y agregarle un logo al banco, después se mostrara otra pantalla, donde se observa el logo del banco, y los cajeros disponibles, aunque la función principal de esta pantalla será agregar clientes al sistema bancario, la cual hay que suministrar dos datos el nombre y el correo del cliente, además permite el manejo de los cajeros por medio de dos botones uno para ponerlo disponible y otro para ocuparlo, en tal caso que el funcionario de este cubículo tenga algún imprevisto, otra función importante de esta pantalla es el botón de reportes el cual nos abrirá una pantalla nueva la cual permitirá ver los datos en una manera más organizada, y también mostrar la información más clara y precisa para el entendimiento de todas las personas.

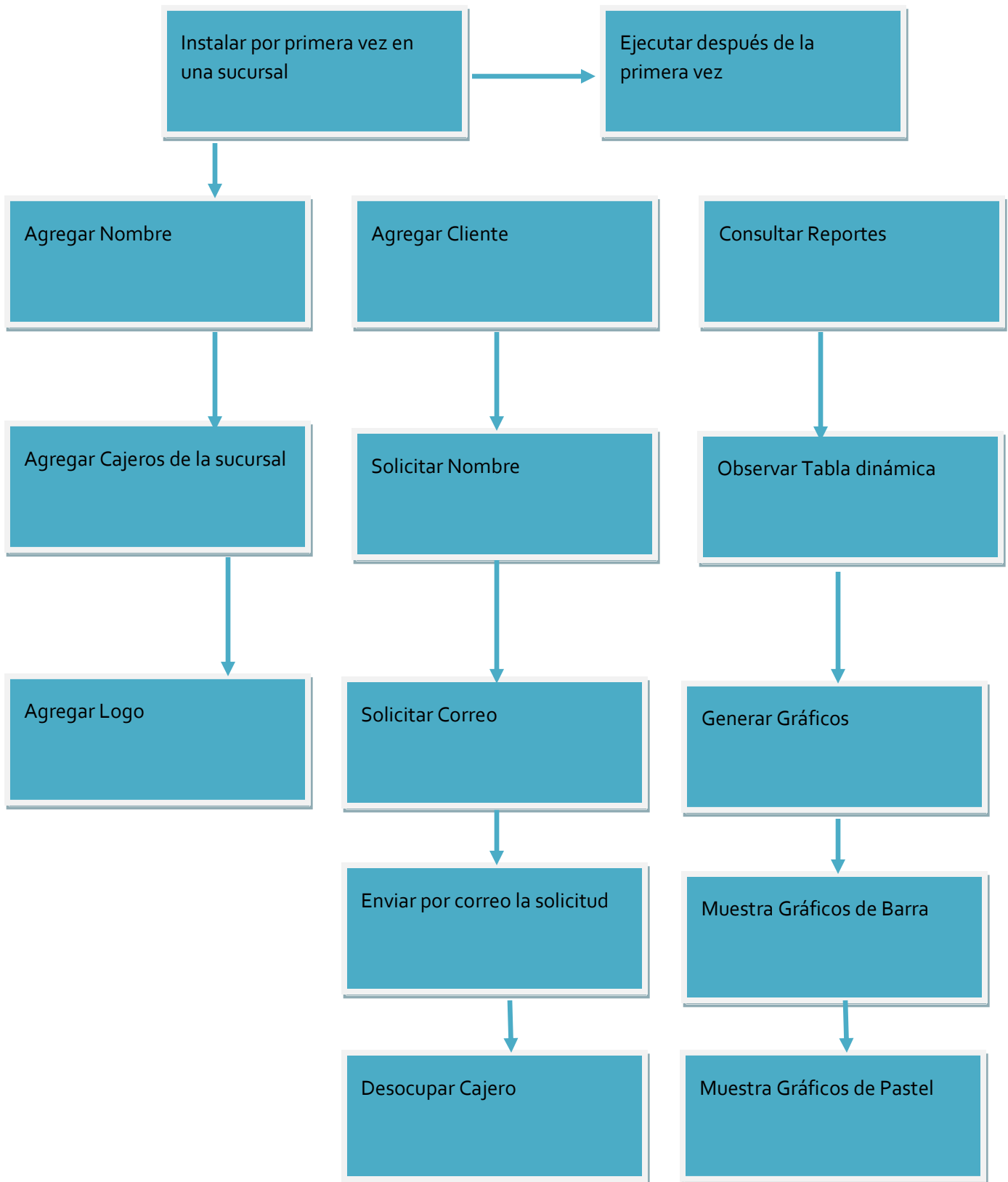
Decisiones de Diseño.

Algoritmos Utilizados.

Se utilizaron algoritmos de búsqueda secuencia l en el sistema como por ejemplo cuando hay que consultar la cantidad de personas atendidas en una hora determinada, o si deseamos saber cuántas personas de determinado tiempo se han atendido, sin embargo se intentara mejorar este aspecto como reto personal, además en las pilas se utiliza el Bubble sort el cual es muy importante dentro del organigrama del programa, ya que es que ordena la lista cada vez que se agrega un cliente, además se utiliza en la tabla dinámica, la cual tiene varias opciones de ordenar .

Todos los datos se guardaran en una cola, la cual a su vez contiene listas doblemente enlazadas, lo cual permite tener más acceso a los datos de la aplicación.

Diagrama Lógico



Librerías Usadas:

Java mail 1_4_7: Permite enviar y recibir correos desde java

Jcommon- 1.0.17 Librería con funciones para crear gráficos y figuras geométricas, además de líneas.

Jfreechart 1.0.14 Librería que permite crear graficas tipo pastel

Análisis de Resultados:

Objetivos Alcanzados:

Se logró realizar todos los objetivos descritos en las instrucciones de la tarea programada #2 entre los principales están:

- Agregar Cliente a la cola de Clientes
- Asignar Fichas a clientes Vía correo
- Agregar Imagen al programa
- Establecer Cantidad de Cajeros
- Búsquedas Avanzadas
- Diseñar Interfaz Grafica

Objectives No Alcanzados:

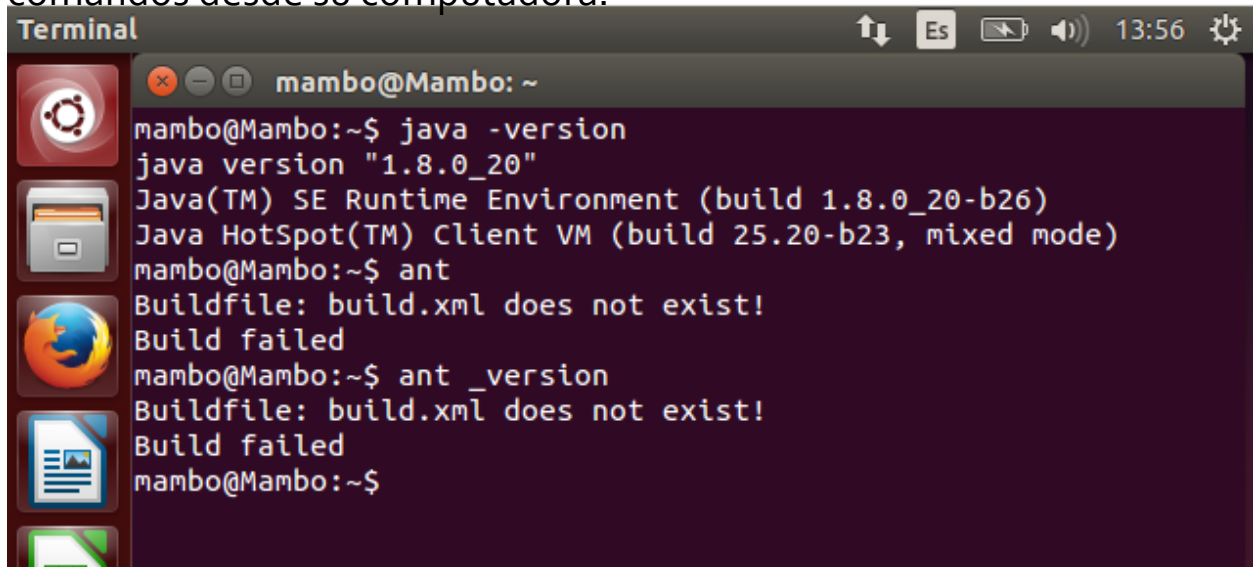
En este proyecto se lograron alcanzar todos los objetivos del proyecto, entre las funciones más complejas fue desarrollar los gráficos y la tabla dinámica, sin embargo cumplen con los requisitos, ahora por iniciativa del grupo se tratara de mejorar las funciones y agregar otras funciones nuevas, las cuales harán que la aplicación sea más completa.

Manual de Usuario

Requisitos del Programa

- JDK 1.8
- JAVA
- ANT 1.9.3
- Linux

Para verificar su versión de Java y Ant puede ejecutar los siguientes comandos desde su computadora.

A screenshot of a Linux terminal window titled "Terminal" with a dark purple background. The window shows the user "mambo@Mambo: ~" at the prompt. The user enters the command "java -version", and the output shows "java version "1.8.0_20"" followed by details about the Java(TM) SE Runtime Environment and Java HotSpot(TM) Client VM. Then, the user enters "ant", and the output shows "Buildfile: build.xml does not exist!" and "Build failed". Finally, the user enters "ant _version", and the output shows "Buildfile: build.xml does not exist!" and "Build failed". The terminal window has a sidebar on the left with icons for various applications like a file manager, web browser, and terminal. The top of the window shows system status icons like network, battery, and volume, along with the time "13:56".

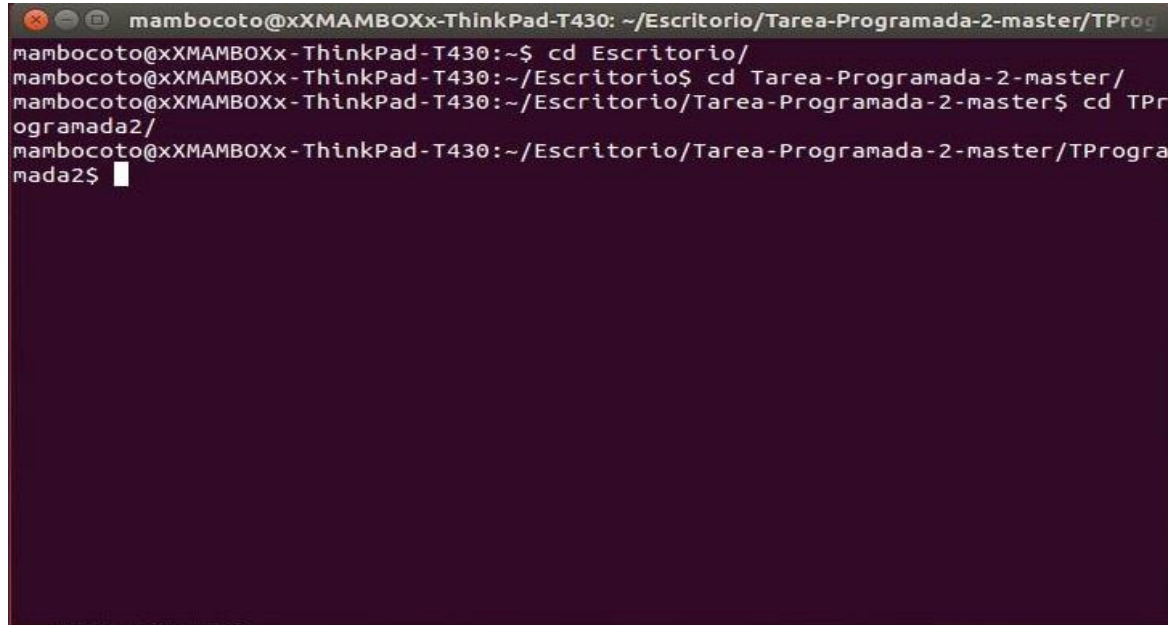
```
Terminal
mambo@Mambo: ~
mambo@Mambo:~$ java -version
java version "1.8.0_20"
Java(TM) SE Runtime Environment (build 1.8.0_20-b26)
Java HotSpot(TM) Client VM (build 25.20-b23, mixed mode)
mambo@Mambo:~$ ant
Buildfile: build.xml does not exist!
Build failed
mambo@Mambo:~$ ant _version
Buildfile: build.xml does not exist!
Build failed
mambo@Mambo:~$
```

En este caso vemos que la versión mostrada es 1.8.0_20

Link de descarga del zip: <https://github.com/xXMAMBOXx/Tarea-Programada-2>

NOTA: Para efectos de compilación con ANT se debe de extraer el .zip y luego se debe de entrar en la carpeta del proyecto llamada "Progra1" para así llamar al programa Ant desde la terminal y que compile y automáticamente cree la carpeta dist donde se encontrará el archivo Progra1.jar el cual puede ser ejecutado desde terminal o bien con doble click.

Ahora se debe de buscar la carpeta en la terminal de Ubuntu o Linux (Progra2)

A terminal window with a dark purple background. The title bar shows the user 'mambocoto' and the machine 'xXMAMBOXx-ThinkPad-T430'. The terminal text shows a series of 'cd' commands navigating through the file system: from the home directory to 'Escritorio/', then to 'Tarea-Programada-2-master/', and finally to 'TProgramada2/'. The prompt is currently at the 'TProgramada2\$' level with a cursor.

```
mambocoto@xXMAMBOXx-ThinkPad-T430: ~/Escritorio/Tarea-Programada-2-master/TProg  
mambocoto@xXMAMBOXx-ThinkPad-T430:~$ cd Escritorio/  
mambocoto@xXMAMBOXx-ThinkPad-T430:~/Escritorio$ cd Tarea-Programada-2-master/  
mambocoto@xXMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master$ cd TP  
rogramada2/  
mambocoto@xXMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master/TProgra  
mada2$
```

Ya por estar dentro de la carpeta ejecutamos la acción de ANT

Esto encontrara automáticamente la dirección del build.xml la cual será la encargada de guiar todo el proceso de compilación


```

mambocoto@xXMAMBOXx-ThinkPad-T430: ~/Escritorio/Tarea-Programada-2-master/TProg
[javadoc] Loading source file /home/mambocoto/Escritorio/Tarea-Programada-2-ma
ster/TProgramada2/src/Queue.java...
[javadoc] Loading source file /home/mambocoto/Escritorio/Tarea-Programada-2-ma
ster/TProgramada2/src/eMail.java...
[javadoc] Constructing Javadoc information...
[javadoc] Standard Doclet version 1.8.0_20
[javadoc] Building tree for all the packages and classes...
[javadoc] Building index for all the packages and classes...
[javadoc] Building index for all classes...

-javadoc-browse:

javadoc:

default:

BUILD SUCCESSFUL
Total time: 5 seconds
mambocoto@xXMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master/TProgra
mada2$ ls
build      Dias.obj  lib       manifest.mf  src
build.xml  dist     List.obj  nbproject    Tarea Programada 2.docx
mambocoto@xXMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master/TProgra
mada2$ cd dist

```

Aquí se ve cómo se va dando el proceso de compilación a través de la terminal de Ant.

```

mambocoto@xXMAMBOXx-ThinkPad-T430: ~/Escritorio/Tarea-Programada-2-master/TProg
[javadoc] Loading source file /home/mambocoto/Escritorio/Tarea-Programada-2-ma
ster/TProgramada2/src/Lista.java...
[javadoc] Loading source file /home/mambocoto/Escritorio/Tarea-Programada-2-ma
ster/TProgramada2/src/Program.java...
[javadoc] Loading source file /home/mambocoto/Escritorio/Tarea-Programada-2-ma
ster/TProgramada2/src/Queue.java...
[javadoc] Loading source file /home/mambocoto/Escritorio/Tarea-Programada-2-ma
ster/TProgramada2/src/eMail.java...
[javadoc] Constructing Javadoc information...
[javadoc] Standard Doclet version 1.8.0_20
[javadoc] Building tree for all the packages and classes...
[javadoc] Building index for all the packages and classes...
[javadoc] Building index for all classes...

-javadoc-browse:

javadoc:

default:

BUILD SUCCESSFUL
Total time: 5 seconds
mambocoto@xXMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master/TProgra
mada2$

```

Y al finalizar este el siguiente comando adentro de la carpeta generada automáticamente por Ant “dist”, se mostrara la pantalla principal del programa Sistema Citas Bancario

```
mambocoto@xMAMBOXx-ThinkPad-T430: ~/Escritorio/Tarea-Programada-2-master/TProg
[javadoc] Loading source file /home/mambocoto/Escritorio/Tarea-Programada-2-ma
ster/TProgramada2/src/eMail.java...
[javadoc] Constructing Javadoc information...
[javadoc] Standard Doclet version 1.8.0_20
[javadoc] Building tree for all the packages and classes...
[javadoc] Building index for all the packages and classes...
[javadoc] Building index for all classes...

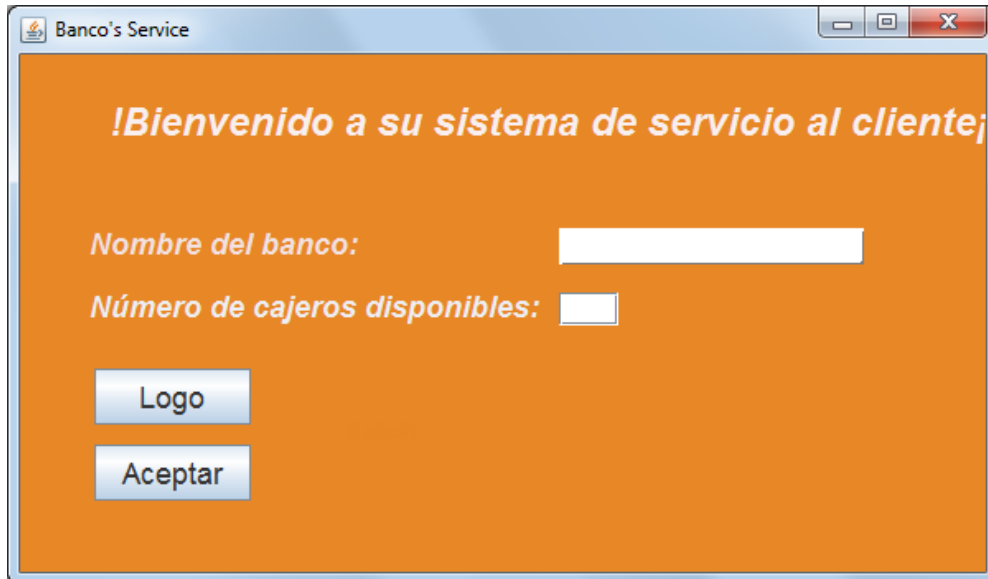
-javadoc-browse:

javadoc:

default:

BUILD SUCCESSFUL
Total time: 5 seconds
mambocoto@xMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master/TProgra
mada2$ ls
build      Dias.obj  lib       manifest.mf  src
build.xml  dist      List.obj  nbproject    Tarea Programada 2.docx
mambocoto@xMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master/TProgra
mada2$ cd dist
mambocoto@xMAMBOXx-ThinkPad-T430:~/Escritorio/Tarea-Programada-2-master/TProgra
mada2/dist$ java -jar TProgramada2.jar
```

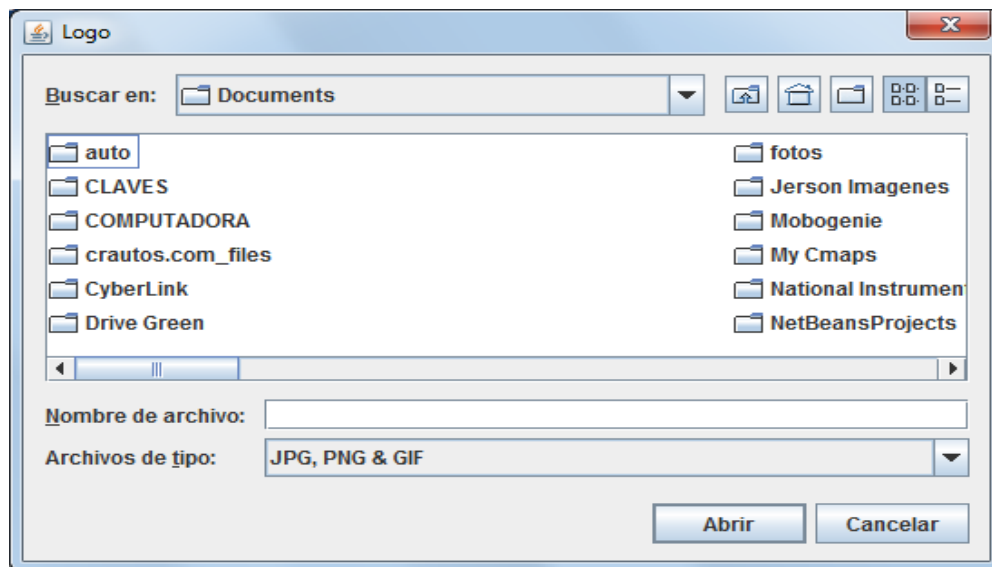
¿Cómo usar el Programa?



A continuación se presenta la pantalla inicial del programa, la cual contiene un mensaje de bienvenida, además trae varias funciones, ya que el programa está diseñado para ser usado en cualquier sucursal, independientemente de las condiciones del personal entonces se deben llenar varios campos los cuales solo se llenan cuando se instala la aplicación en la sucursal, después el sistema ingresa directo como se mostrara más adelante.

Primero debe indicar el nombre del Banco, seguidamente el numero de cajeros de la sucursal, aquí se deben incluir todos tanto los disponibles como los que están inhabilitados. Más adelante el administrador del sistema los puede inhabilitar.

Seguidamente se debe ingresar el logo de la compañía, el cual abrirá otra pantalla donde se debe escoger la ruta de la imagen, una vez seleccionada la imagen se le dará abrir.



Una vez concluidos los pasos anteriores nos debe quedar una pantalla como la siguiente, donde tenemos el nombre, los números de cajeros y el logo, ahora podemos presionar aceptar.



Seguidamente mostrara esta pantalla la cual será la pantalla de bienvenida después de haberla instalado la primera vez.

Esta pantalla nos permitirá ir agregando los clientes conforme lleguen a la sucursal, se le deberá solicitar al cliente el nombre, y el correo, ya que será por este medio que se notificara de la cita, además se debe verificar el tipo de cliente ya que esto afecta en el orden de ser atendido.



BAC TEC

Nombre:

Correo:

Cliente Regular ▼

Añadir

Reportes

BAC

Cajeros:

- Cajero #0 Disponible
- Cajero #1 Disponible
- Cajero #2 Disponible

Disponible

Aquí se va agregando los clientes, y si hay un cajero disponible se le asignara automáticamente, si no quedara en cola hasta que se libere

Para liberar un cajero lo seleccionamos y se le da el botón liberar, y así cada vez que se termine de atender a cada cliente.

BAC TEC

Nombre:

Correo:

▼

BAC

Cajeros:

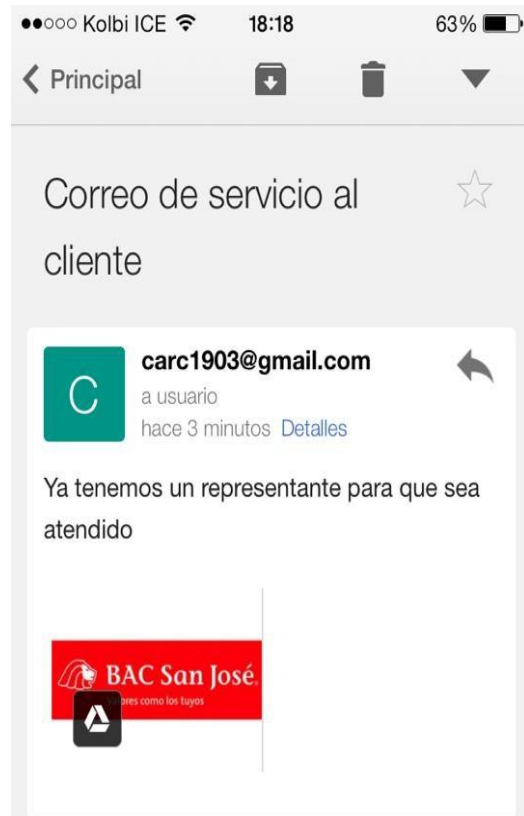
- Cajero #0 Ocupado
- Cajero #1 Ocupado
- Cajero #2 Ocupado

Cuando el sistema registra un cliente, enviara un correo a la persona actual indicándole que su solicitud se realizo con éxito, de igual manera cuando hay un cajero disponible y se le asignan un cliente inmediatamente se le enviara un correo a la persona asignada para que proceda a la debida ventanilla.

Aquí se ve un ejemplo de cuando un cliente es registrado en el sistema.



Aquí se ve un ejemplo de cuando el cliente es atendido por un cajero.



El sistema tiene varios reportes los cuales se pueden consultar ya sea al final del día, o en cualquier momento así sea su necesidad.

Seleccionamos el botón Reportes

BAC TEC

Lista de Clientes ordenada por:

☐ Nombre ☐ Correo ☒ Hora ☐ Fecha ☐ Prioridad

Nombre	Correo	Hora	Fecha	Tipo
carlos	carlos19ramirez...	23:14:41	2014-10-14	Regular
maria	maria@hotmail.c...	23:14:50	2014-10-14	Regular
alonso	alonso@hotmail.c...	23:17:03	2014-10-14	Corporativo
alonso	alonso@hotmail.c...	23:17:06	2014-10-14	Especial
alonso	alonso@hotmail.c...	23:17:07	2014-10-14	Especial
alonso	alonso@hotmail.c...	23:17:09	2014-10-14	Regular
andreina	andreina@hotmail...	23:17:20	2014-10-14	Regular
luisa	luisa@hotmail.com	15:21:08	2014-10-13	Embarazada
maritza	mcr@hotmail.com	15:23:42	2014-10-13	Corporativo

Fechas de atención al cliente:

2014-10-14

Cantidad de atendidos:

Especiales:

A.Mayor:

M.Embarazada:

C.Corporativo

C.Regulares:

Graficar

jButton1

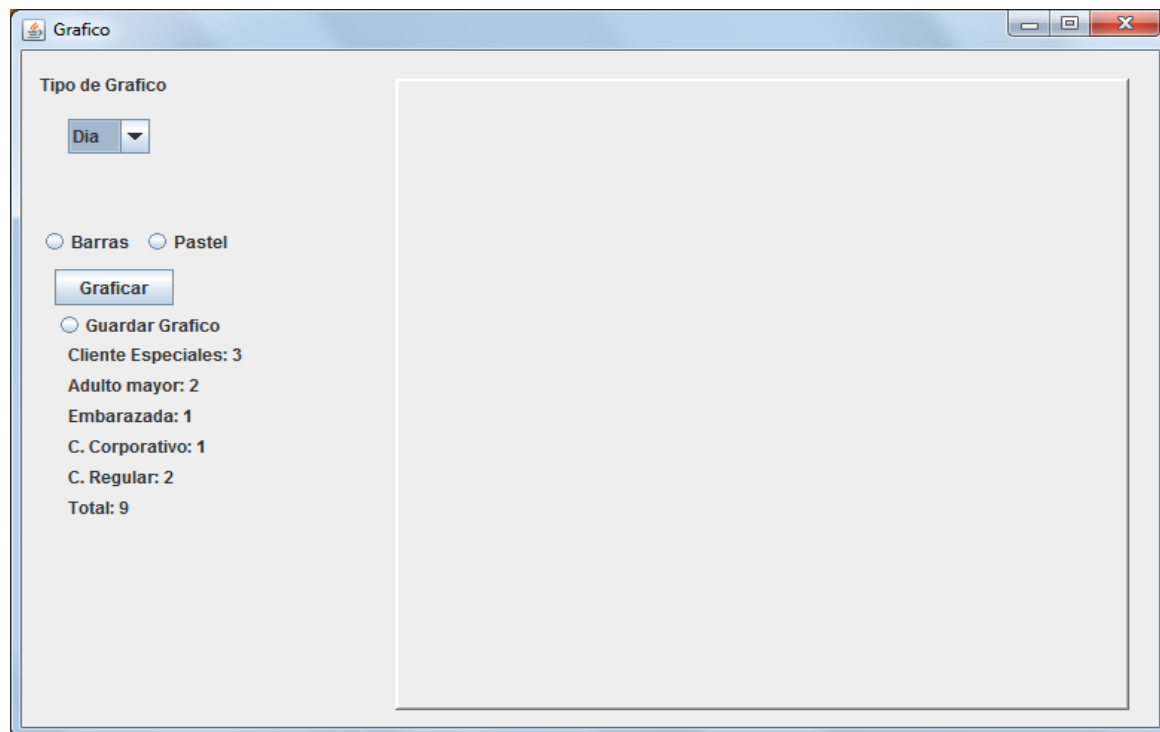
Aquí podemos observar una tabla dinámica la cual nos muestra la información ordenada ya sea por nombre, correo, hora, fecha, prioridad. Simplemente se selecciona la que le interese al usuario y se ordenara.

Además en el recuadro de la parte inferior se ve las fechas que se ha atendido a los clientes, las cuales muestran varias características como cantidad de atendidos, y de cada tipo nos muestra la cantidad de personas. En este caso se debe seleccionar la fecha para ver los reportes.

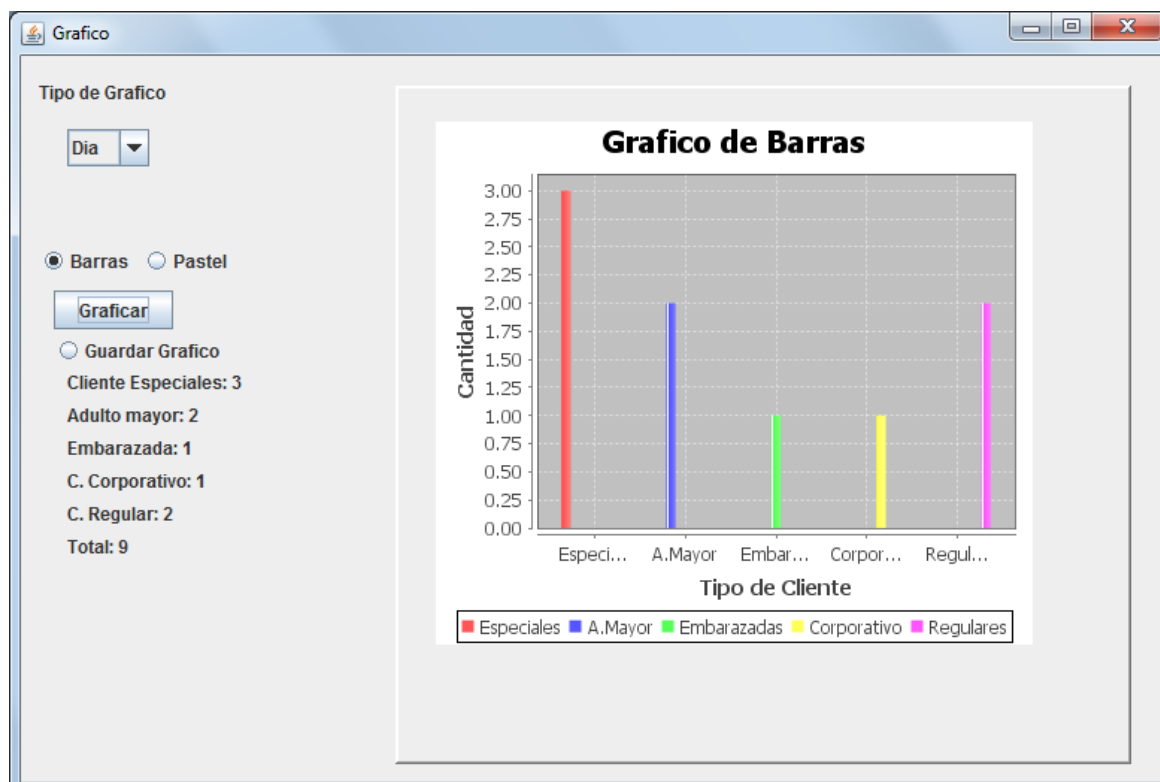


La otra opción de reportes son los gráficos, donde se deben llenar varios datos para poder obtener el grafico deseado, lo primero es seleccionar si lo desea por día o por hora seguidamente el tipo de grafico, las opciones disponibles es de barras o de pastel y por ultimo seleccionar si desea guarda el grafico.

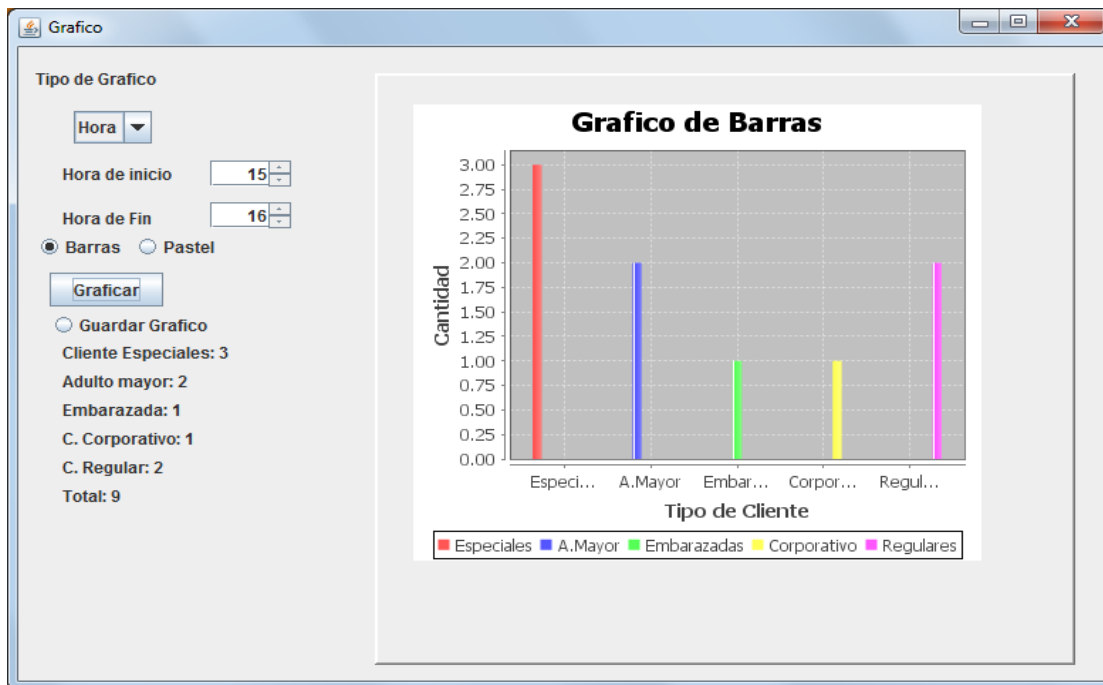
Una vez realizado los pasos anteriores se selecciona el botón de graficar.



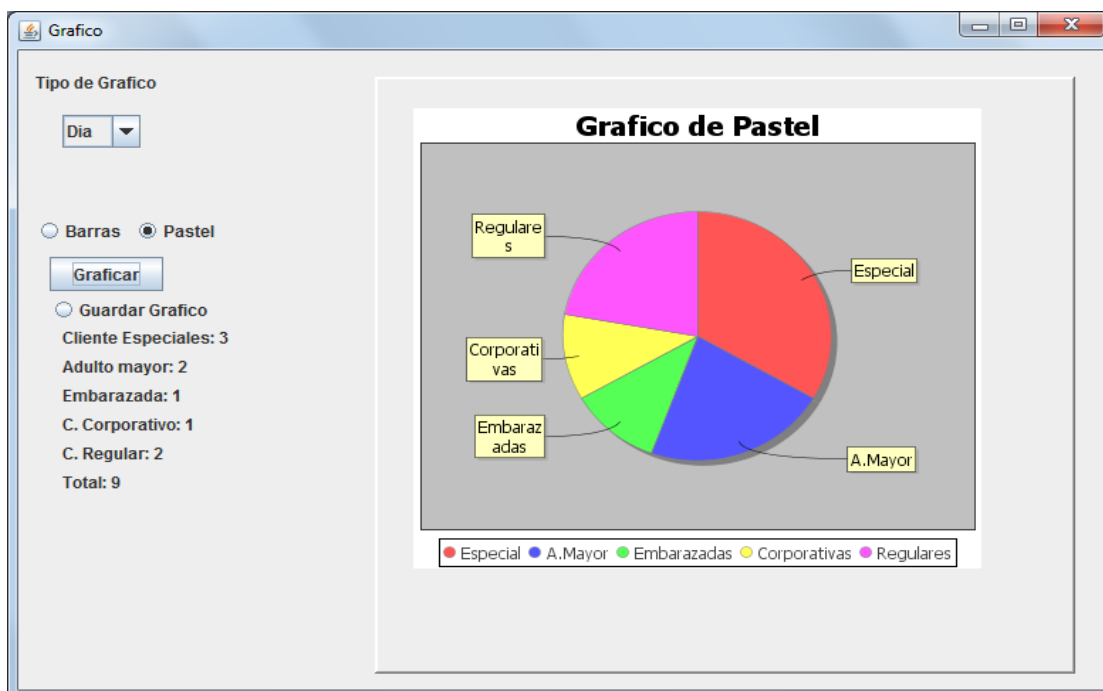
Aquí obtenemos el grafico de barras, donde se pueden observar la cantidad de personas atendidas en el día, la escala variará de acuerdo a los datos del sistema.



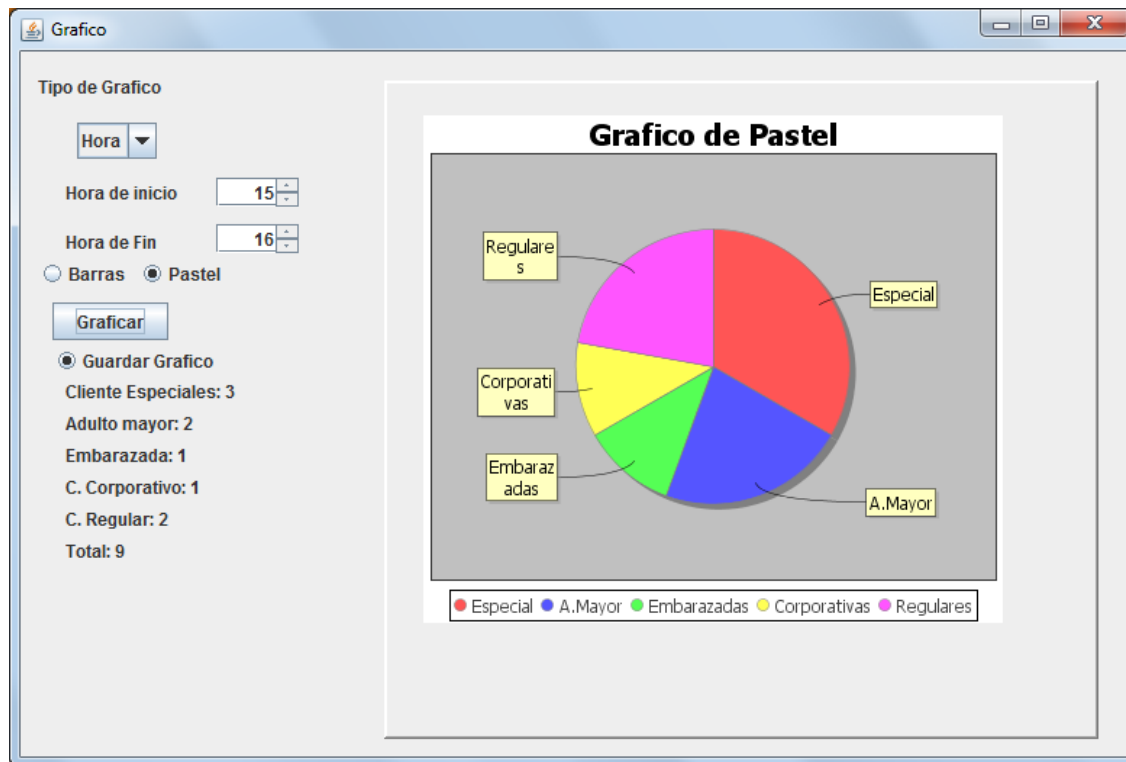
En el caso que se desee por hora se debe ingresar la hora inicial y la hora final, como se muestre en el siguiente ejemplo



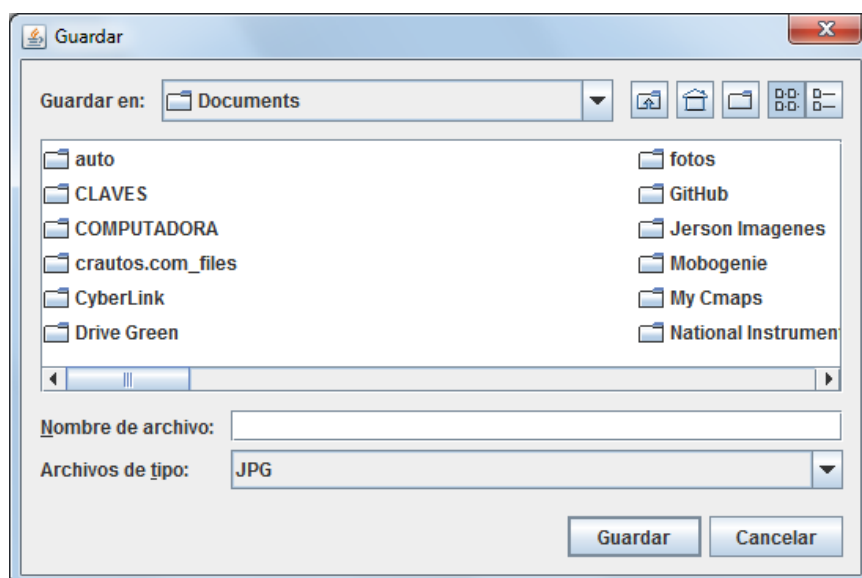
En el grafico de pastel se observan las diferentes particiones de acuerdo a los clientes atendidos, en tal caso que alguno de los tipos fuera cero, igual se mostrara el nombre y una pequeña línea.



Igualmente al anterior cuando es un grafico pastel por horas, debemos de seleccionar la hora inicial y la hora final.



En el caso anterior seleccionamos la opción de guardar por lo cual antes de hacer el grafico mostrara la siguiente pantalla. En la cual se debe especificar la ruta donde se desee guardar el grafico, y seleccionar el botón de guardar.



Para dudas o consultas no dude en comunicarse con nosotros por medio del correo electrónico

Banc_MMC @gmail.com

Conclusiones Personales:

Marvin Fernández Coto:

Realizar la segunda tarea programada manejando estructuras de datos como la cola fue fácil ya que se conocía estructuras de datos como una lista, la cual solamente se le agregaron métodos para poder moldear su estructura a una cola, también se incrementó el conocimiento del lenguaje de programación (Java), también se implementaron algoritmos de ordenamiento para dar prioridad al cliente y poder realizar la tabla dinámica.

Carlos Ramírez Cerdas

Este proyecto fomento el uso de colas para la solución de problemas cotidianos en la vida laboral, en este caso se hizo énfasis en el sistema bancario, pero si uno analiza hay muchas soluciones que se pueden mejorar, y otras más que se pueden implementar desde cero. El sistema de citas bancario nos da una idea del funcionamiento interno de muchos programas aunque hay mucho que mejorarle al programa.

Guillermo Ávila Chaves

En esta tarea aprendí mucho sobre gráficos en java, también aprendí el uso de las colas de prioridad y el ordenamiento de diferente tipos de datos, el sistema bancario nos dejó enseñanza de cómo funciona los dispositivos de los bancos los cuales son un con colas de prioridad, aprendí como usar diferentes tipos de ventanas a la vez sin caer en errores de programación como el WAW (write after write).