



Instituto Tecnológico de Costa Rica

Administración de Tecnologías de
Información

Algoritmos y Estructura de Datos

TI-2402

Tarea Programada 2

Integrantes:

Melvin Brenes Gómez - 2013104822

Max Rodríguez Flores - 2014086106

Kevin Matamoros Serrano - 2014086158

Jueves 9 de Octubre de 2014

Tabla de Contenidos

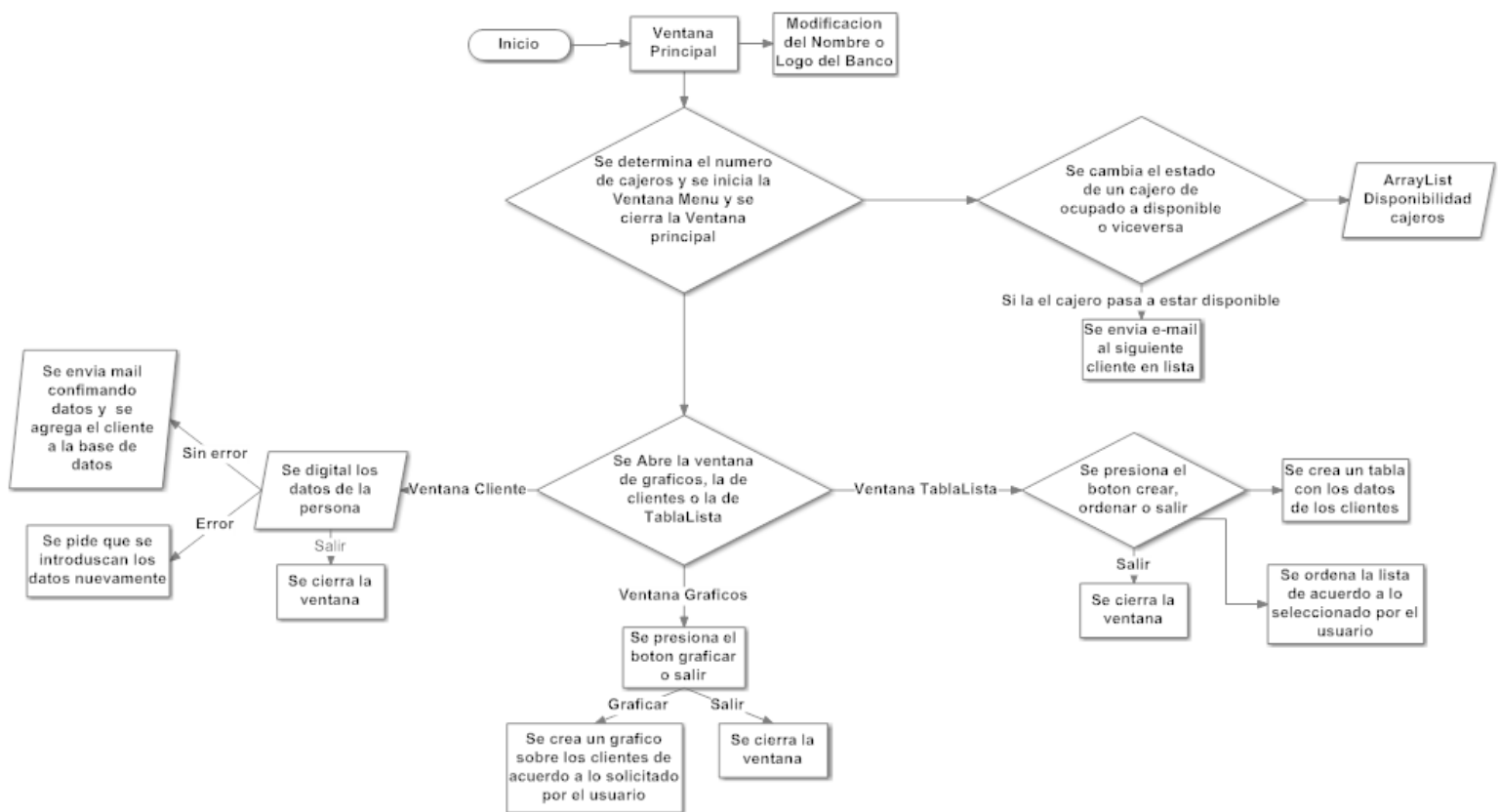
• Descripción del problema	3
• Diseño del programa: decisiones de diseño, algoritmos usados, diagramas lógicos	4
• Librerías usadas: especificación de las librerías usadas, y descripción de uso	4
• Análisis de resultados: objetivos alcanzados, objetivos no alcanzados.....	5
• Manual de usuario: instrucciones uso, de ejecución y de compilación	5
• Conclusión	12

Descripción del problema

Lo solicitado es un programa para un banco, que permita añadir a las personas, que entren al banco a realizar transacciones, a una cola. También se solicita que exista prioridad para cierto tipo de personas, de forma que se atienda primero a las personas discapacitadas, luego los adultos mayores, mujeres embarazadas, clientes corporativos y clientes regulares en ese orden. El programa debe mandar un correo electrónico a cada cliente cuando llega al banco para confirmar sus datos y otro correo cuando es su turno, donde se indica a cuál caja debe pasar. También se debe poder visualizar toda la información de los clientes en una tabla de información y gráficos pastel y de barras sobre cantidad de clientes por los tipos de clientes y por las horas y fechas de en que fueron atendidos los clientes. Debe ser posible modificar la cantidad de cajas que posee el banco. También se deben guardar los datos en una base de datos para poder guardarlos y que queden almacenados incluso después de cerrar el programa.

Diseño del programa

En cuanto a decisiones del diseño del programa se usó el color rojo basándonos en BAC San José, se crearon ventanas pequeñas e intuitivas para que sea fácil su manejo, se usaron algoritmos simples para reducir los tiempos de espera, el programa tiene una función que actualiza los estados de las cajas continuamente para evitar problemas a la hora de asignar los clientes a las cajas. Las cajas se presentan en un Combo box para que sea más fácil su manejo. Para facilitar el diseño gráfico decidimos la utilización del programa Netbeans que nos da opciones de agregar el objeto y luego programarlo.



Librerías usadas

- **Email:** usada para mandar e-mails
- **Jcommon:** Usada por Jfreechart para cosas como mostrar textos, mostrar imágenes y manejo de código
- **Jfreechart:** Librería usada para crear los gráficos

Análisis de Resultados

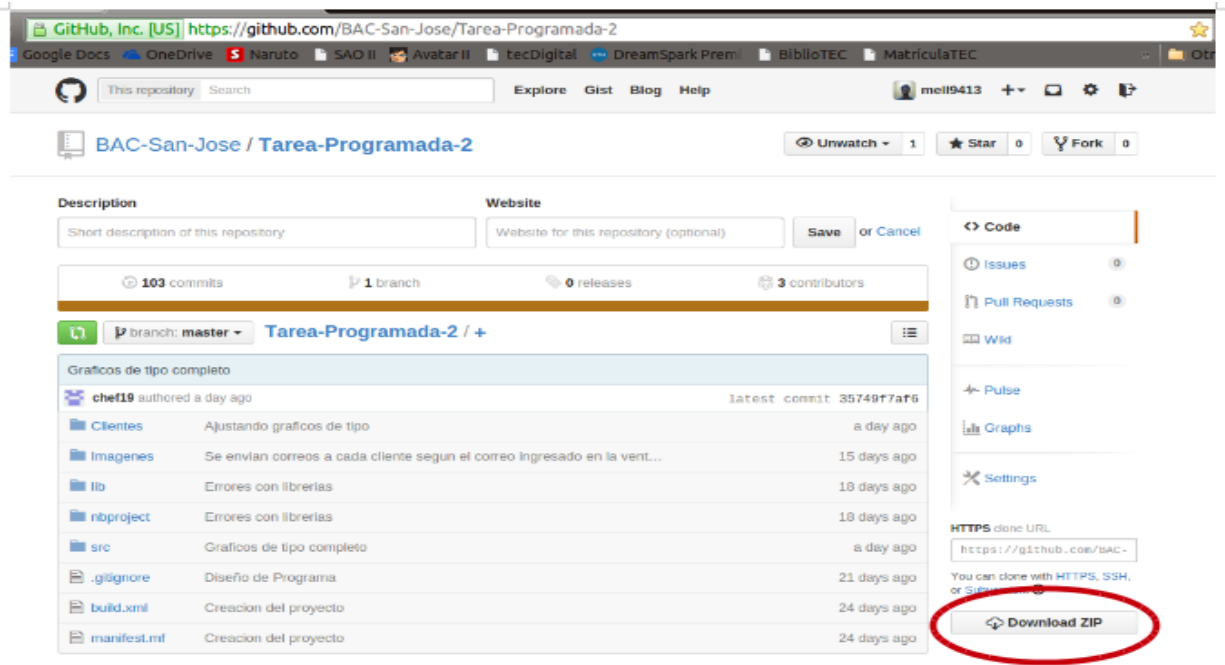
No se pudieron cumplir con todas las funciones solicitadas, la única función que no se pudo hacer fue la de ordenamiento, esta fue una de las más difíciles funcionalidades que tenía esta tarea programada y no la empezamos a hacer con suficiente tiempo por lo tanto no la pudimos hacer, aparte de esto todo lo demás fue logrado, el sistema permite agregar clientes, ordenarlos por su prioridad, enviar un email de confirmación y cuando sea su turno, mostrar los datos ordenados en una tabla y mostrar gráficos de cantidad de clientes por tipo de clientes, por hora y fecha en que fueron atendidos. El orden por prioridad se logró gracias a que se usaron distintas colas y estas se revisaban por orden de prioridad y luego estos se agregaban a base de datos ya ordenados, de esta forma se puede acceder a cada cliente fácilmente, también se crearon distintas listas para cada dato del cliente de tal forma en que los el cliente que esta en la posición 4 tiene sus datos en cada lista en la posición 4, así se pueden extraer sin dificultades.

Para manejar las colas se implementó una clase cola con las funcionalidades básicas de una cola, además se implementó una clase arraylist para manejar los datos de los clientes y facilitar su uso.

Se usaron ventanas para simplificar la estructura del programa, así cada función solicitada fue acomodada en una ventana distinta, así el código es más simple y además es más fácil de usar para el usuario

Manual de Usuario

En el siguiente enlace, se encuentra el repositorio en donde esta almacenada la programación, <https://github.com/BAC-San-Jose/Tarea-Programada-2>



El botón que se encuentra encerrado en un círculo rojo generará un clon del repositorio en el escritorio y creará un archivo de formato zip, este archivo se descomprime para al final tener una carpeta en el escritorio llamada “Tarea-Programada-2-master”. Una vez realizado este paso, se procederá, desde la línea de comandos, a acceder a esta carpeta hasta donde se encuentra el archivo “build.xml”, el cual compila el ANT.

```
mell@Mell-xD-09: ~/Escritorio/Tarea-Programada-2-master
mell@Mell-xD-09:~$ cd Escritorio/Tarea-Programada-2-master/
mell@Mell-xD-09:~/Escritorio/Tarea-Programada-2-master$ ls
build.xml  Clientes  Imagenes  lib  manifest.mf  nbproject  src
mell@Mell-xD-09:~/Escritorio/Tarea-Programada-2-master$
```

Después se procede a realizar la compilación con ant, así que en esa misma dirección se procede a escribir “ant” y luego teclear “enter”. Al momento se podrá observar en la consola lo siguiente:

```
mell@mell-xD-09: ~/Escritorio/Tarea-Programada-2-master
src/Queue.java...
[javadoc] Loading source file /home/mell/Escritorio/Tarea-Programada-2-master/
src/Reloj.java...
[javadoc] Loading source file /home/mell/Escritorio/Tarea-Programada-2-master/
src/VentanaCliente.java...
[javadoc] Loading source file /home/mell/Escritorio/Tarea-Programada-2-master/
src/VentanaMenu.java...
[javadoc] Loading source file /home/mell/Escritorio/Tarea-Programada-2-master/
src/VentanaPrincipal.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: 8 seconds
mell@mell-xD-09:~/Escritorio/Tarea-Programada-2-master$
```

Cuando se realiza este paso, en la carpeta donde se encuentra el “build.xml” se generan dos carpetas creadas por “ant”, la carpeta “build” y “dist”. En esta última encontramos el archivo Tarea-Programada2.jar, este archivo es la programación. El mismo puede llamarse desde la consola, ingresando en la dirección en que se encuentra y luego correrlo con el siguiente comando: “java -jar Tarea-Programada-2.jar”. Pero también, se puede correr el programa al dar doble click al archivo Tarea-Programada-2.jar.

Una vez que se inicia el programa se mostrará la ventana de inicio donde se puede modificar el logo del banco(1), el nombre(2) y el número de cajas(3) o se puede presionar el botón “Predeterminado”(4) para devolver el nombre y el logo a sus datos predeterminados.

BAC San José
Sistema de Atención a Clientes BAC

1 Cambiar Logo Hora: 20:22:42
Fecha: 3/10/2014

2 Cambiar Nombre

Cantidad de Cajas: 3 0

Aceptar 4 Predeterminado

Una vez que se establezca el número de cajas se puede acceder al menú principal

BAC San José
Sistema de Atención a Clientes BAC

Gráficos Hora: 20:36:45 Fecha: 3/10/2014

Cliente

Tabla Lista

Cajas

1

Disponibile Ocupada

Salir

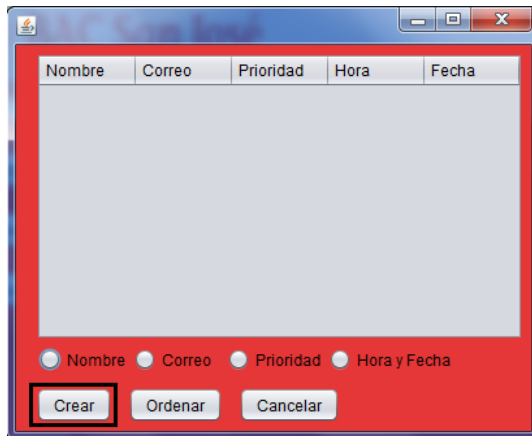
En el cuadro negro se puede ver una pequeña pestaña en la que podemos seleccionar alguna de las cajas, y en los círculos negros se aprecian las opciones “Disponible” y “Ocupada”, esto es para asignar la disponibilidad de las cajas y también comprobarla, el sistema se actualiza constantemente y si hay clientes en fila y cajas disponibles el sistema por sí solo asignará los clientes a las cajas. También se pueden consultar la hora y fecha en la esquina superior de la pantalla. Si se presiona el botón “Cliente”, se abrirá la siguiente ventana



En esta ventana se ingresan nuevos clientes al banco, el cuadro 1 es para ingresar el nombre del cliente, el 2 es para ingresar el correo electrónico del cliente y el 3 para asignar el tipo de cliente, pues esto definirá qué prioridad tendrá a la hora de ser atendido, el orden de prioridad es “Persona Discapacitada”, “Adulto Mayor”, “Mujer Embarazada”, “Cliente Corporativo” y “Cliente Regular”. una vez se ingresan los datos se puede presionar el botón “Aceptar”, de otra forma el sistema dará un error si hay un error en los datos o estan incompletos, si no hay ningún problema con los datos se verá este mensaje



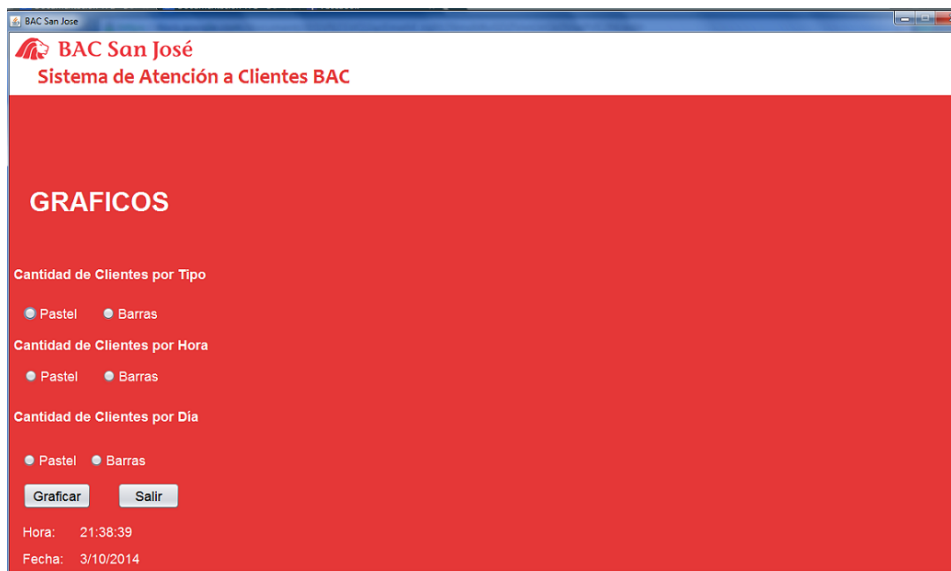
Una vez que registren todos los clientes deseados se puede volver al menú principal, si presionamos el botón “Tabla Lista” se abrirá la siguiente ventana.



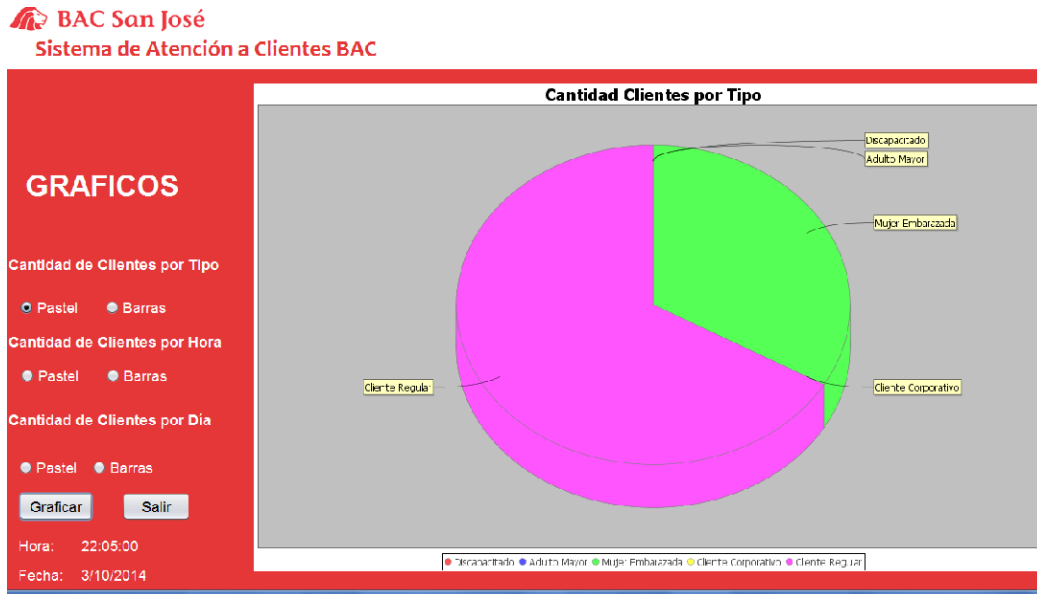
Si se presiona el botón “Crear” se creará una tabla con los datos de todos los clientes que se hayan introducido como se puede ver a continuación.

Nombre	Correo	Prioridad	Hora	Fecha
Max	max.andr...	Regulares	21:10:25	3/10/2014
Link	HeroeDel...	Regulares	21:23:41	3/10/2014
Nicole	nicole867...	Embarazadas	21:24:40	3/10/2014

Si en el menú principal se presiona el botón de “Gráficos” se abrirá la siguiente ventana



Los gráficos se dividen por tipo de dato y tipo de gráfico, los tipos de datos son “por tipo de cliente”, “por horas en que se atendieron” y “por fechas en que se atendieron”, y por tipo de gráfico se dividen en Pastel y de Barras. si se elige alguna de las 6 opciones y se presiona el botón “Graficar” se mostrará un gráfico del tipo elegido y con los datos elegidos, por ejemplo en esta imagen se escogió un gráfico pastel de tipo de clientes



Conclusión

Como conclusión, adquirimos conocimientos amplios en el lenguaje de programación Java. Se utilizó el IDE de Netbeans para facilitar la parte del diseño del programa. Comparada a la anterior Tarea Programada, los problemas que se presentaron se pudieron resolver más rápidamente pues ya tenemos más experiencia con Java y con Netbeans. El uso de Github fue mucho más intuitivo que la última vez, pues ya tenemos experiencia con este y ya no nos ocasiona problemas.

La experiencia de la anterior Tarea Programada nos ayudó mucho pues, desde un principio se resolvieron problemas como variables dudosas y funciones dudosas, usando nombre significativos y se intentó usar el menor número de métodos y variables para así evitar confusiones o perder el orden del código.

Al usar ventanas se simplificó el código porque así una ventana no queda sobrecargada con métodos sino que cada ventana se enfoca en un método, dándole así más eficiencia a los métodos y más funcionalidades.

La parte en la que mas usamos los conocimientos adquiridos en el curso fue en la parte de ingresar un cliente al sistema y cuando era el turno de un cliente de pasar a la caja, pues ambos implican el uso de colas el cual fue el tema que se vio en clase recientemente.

Esta vez pudimos realizar la mayoría de las funcionalidades solicitadas más rápido porque pudimos empezar con el trabajo más temprano, pues ya conocíamos bien Github y Netbeans, lo que nos permitió ponernos de acuerdo más rápido y empezar a trabajar más rápido, además de que ya no teníamos dudas sobre Github o Netbeans, esto nos ahorró mucho tiempo.

El trabajo no se sintió tan pesado como la vez pasada, esta vez fue mucho más fácil hacer cada método y clase. El código se hizo lo más eficiente posible, lo más simple posible y lo más manejable posible.