UNIVERSIDAD TECNICA NACIONAL

INGENIERIA DEL SOFTWARE

PROYECTO I

RONALD ACUÑA  
DIEGO ALFARO

PROGRAMACION II

PROFESOR: ING.JOSE PABLO QUIROS.

OCTUBRE, 2013

Tabla de Contenido

[RESUMEN 3](#_Toc361254877)

[ABSTRACT 4](#_Toc361254878)

[INTRODUCCION 5](#_Toc361254879)

[PRESENTACION Y ANALISIS 6](#_Toc361254880)

[RESULTADOS 6](#_Toc361254881)

[ANALISIS DE LA SOLUCION 8](#_Toc361254882)

[MANUAL DE USUARIO 11](#_Toc361254884)

[RECOMENDACIONES 12](#_Toc361254884)

[CRONOGRAMA DE TRABAJO 13](#_Toc361254885)

[Bibliografía 13](#_Toc361254886)

## RESUMEN

La intención de la aplicación aquí presentada es emular una sucursal de un banco con cajeros y filas para usar métodos de ordenamientos por prioridades y nodos, ambos vistos en clase además de diseñar una interfaz gráfica para mostrar la simulación**.**

## ABSTRACT

# The intent of the application presented here is to emulate one bank query in which every client has his own random characteristics that will store and then show at the end of the simulation. We are going to use methods that we studied in class and others we found via researching, thus allowing the teacher to evaluate our development and progress during the actual course.

## 

## INTRODUCCION

En este primer proyecto el objetivo crear la simulación de una sucursal de banco y usar métodos de inserción y nodos (entre otras herramientas de programación) según sea necesario

Se deben crear cuatro cajeros que muestren una serie de datos así como una fila a la cual a cada cliente que ingrese se le deberán asignar datos ya establecidos

## PRESENTACION Y ANALISIS

## ¿Qué es lo que hay que resolver?

Se debe crear una fila de banco y se le debe asignar a cada cliente una prioridad que es basada en

• Si es una persona adulta mayor tiene prioridad alta, por lo que el tiquete vendría con el número, seguido por la letra A.

• Si es una mujer embarazada o con un niño en brazos, la prioridad es media, por lo que el tiquete vendría con el número, seguido por la letra B.

• En cualquier otro caso el tiquete vendría con el número, seguido por la letra

+el número de ficha y el tiempo que duró en ser atendido junto con el tiempo que duró atendiéndose (estos dos últimos no se muestran)

## Se deberá llevar un promedio del tiempo que tardó en ser atendidos, así como cuándo tardó cada cajero en promedio y cuántas personas atendió.

## Existirá una única fila, conforme se van desocupando los cajeros, se van atendiendo clientes.

## • La fila tendrá como máximo 10 clientes haciendo fila.

## • Existirán 4 cajas.

## • Cada cliente tendrá un tiempo de duración de trámite, el cuál será asignado de forma aleatoria de 2 a 10, y este corresponderá a minutos.

## • Cada caja será una lista, donde se irán enlazando los clientes que se van atendiendo

## RESULTADOS

## Se puede concluir que la programación con el lenguaje java en este caso específico usando las herramientas necesarias para emular la sucursal bancaria es bastante versátil en cuanto a lo que se puede hacer o crear, conociendo o investigando las múltiples funciones que tiene el lenguaje como los es “CompareTo” la cual fue de mucha ayuda al ubicar y asignar las prioridades.

## ANALISIS DE LA SOLUCION

Se hizo una pantalla de interfaz gráfica específicamente para la simulación total y los datos se pueden ver actualizados conforme se avanza en la simulación   
En cuanto a código

Se elaboró una clase Nodo con las variables

public cliente dato;

public Nodo siguiente;

public Nodo anterior;

y su método constructor

Una clase cjero con la variable

String nombre

Su método constructor y un toString

Esta clase sirve para establecer las cajas del banco las cuales serán instancias de lista

Una clase cliente que contiene

protected String nombre;

protected String cedula;

protected String tipo;

protected int tiempoAtendido;

protected String ficha;

Métodos constructor y set & get para las variables

Mediante los métodos (generarNumero **,** tipoDePersona**,** crearCedula y (generarNumeroNombre o generarNombreMujer))

Se define la prioridad “A” adulto mayor

“B” embarazada

“C” normal

generarNumero() = genera un entero aleatorio entre 1 y 3

tipoDePersona() = convierte ese entero en el estado “A” ”B” o “C” según sea

si ese estado es “A” o “C” generarNumeroNombre() crea un nombre aleatorio para esa persona y en caso de que sea “C” generarNombreMujer() genera el nombre

crearCedula() agrega la cedula y asignarTiempoAtendido() genera un entero aleatorio entre 2 y 10 para asignar el valor del tiempo de atención a este cliente

El método toString() se encarga de imprimir un mensaje dependiendo del tipo de persona que sean

Una clase listacajero con variables tipo

Nodo primero;

Nodo ultimo;

int tamaño;

Con método constructor y un método boolean vacía() que servirá para decir que la lista está vacía

Un método agregar que recibe por parámetro un dato que será el cliente de la lista de la fila y lo agregará a esta lista

Un método toString que imprime todos los clientes atendidos

Luego un método double promedioDeTiempoAtendiendo() para calcular la media del tiempo atendido de los clientes en total

Se elaboraron varios frames para la interfaz gráfica conectados entre todos e incluso animados por medio de música (el código fue sacado de la investigación que fue fuera del tema pero se incluyó por cuestiones estéticas personales a última hora

## Manual de usuario

## Se le presenta una pantalla con tres opciones

## Comenzar // para empezar la simulación

## Ver anteriores // para ver resultados anteriores (no disponible hasta haberse hecho una simulación

## Salir

## Presione comenzar

## Se le presentara una pantalla nueva que será la simulación en sí de la sucursal

## Presione generar lista para comenzar la simulación y mostrar la cola en el cuadro de texto del centro de la ventana la cual es la fila

## Presione atender cliente para que la simulación siga avanzando y entonces cuando se atienda un cliente automáticamente genere otro cliente y lo agregue a la fila borrando así el primero (que pasará a la cola de atendidos de la caja

## En cada caja se puede ver el promedio de tiempo conforme se avanza en la simulación y si se le da click al botón ver cola se puede ver la cola de atendidos hasta ese momento

## Cuando se le de click al botón salir al menú principal y luego click a ver anteriores entonces si mostrará los datos

## RECOMENDACIONES

En Java existen ciertas reglas de codiﬁcación que son comúnmente utilizadas por los programadores.

Conviene conocer y seguir estas reglas.

• Los nombres de las clases deben empezar por mayúscula.

• Los atributos y métodos de las clases deben empezar por minúsculas y si están formadas por varias palabras, se escriben sin espacios y la primera letra de cada palabra en mayúscula.

• Las instancias de las clases siguen la misma recomendación que los métodos y atributos.

• Las constantes se escriben en mayúsculas

## CRONOGRAMA DE TRABAJO

**El programa se comenzó el 15 de octubre y se trabajó por medio de skype y se trabajó además en días que se asistió a la universidad**

****

## Bibliografía

La materia y ejercicios vistos en clase

YouTube