UNIVERSIDAD CATÓLICA

INGENIERÍA EN SISTEMAS

ESTRUCTURA DE DATOS

PROYECTO PROGRAMADO

LIC. ANDRÉS JIMÉNEZ LEANDRO

JORGE ESTEBAN ROJAS UGALDE

LUIS ALONSO CORELLA CHAVEZ

SAN CARLOS

DICIEMBRE, 2016

Tabla de contenido

[Resumen Ejecutivo 3](#_Toc468377177)

[Descripción 4](#_Toc468377178)

[Abstract 5](#_Toc468377179)

[Objetivos 6](#_Toc468377180)

[Objetivos Generales 6](#_Toc468377181)

[Objetivos Específicos 6](#_Toc468377182)

[Introducción 7](#_Toc468377183)

[Presentación y análisis del problema 8](#_Toc468377184)

[Diagrama de Clases 9](#_Toc468377185)

[Análisis del problema 10](#_Toc468377186)

[Análisis de la solución 10](#_Toc468377187)

[Discusión de resultados y Conclusiones 11](#_Toc468377188)

[Recomendaciones 12](#_Toc468377189)

[Cronograma de trabajo 13](#_Toc468377190)

[Anexos 15](#_Toc468377191)

# Resumen Ejecutivo

¿De qué trata el proyecto?

Este proyecto abarca una aplicación de escritorio creada en el lenguaje Java en el entorno NetBeans, que crea automáticamente colas de Banco mediante programación orientada a objetos, con el uso de listas, ciclos for, ciclos while-do, condiciones if-else, herencia, entre otros

# Descripción

Este documento abarca la documentación respectiva del proyecto programado de colas de Banco. Este documento incluye, aparte de las cosas obligatorias de un documento escrito, un resumen ejecutivo que resume el proyecto, una descripción que resume el documento escrito, un abstract en inglés, objetivos generales y específicos del proyecto, una introducción al proyecto que explica porque se hace el proyecto y que se incluye, marco teórico, una descripción del problema, el diagrama de clases, el análisis del problema donde se analiza el problema que se desea solucionar desde el punto de vista de las implicaciones y los restos de este proyecto.

También incluye un análisis de la solución donde se explica lo que se implementó para solucionar el problema, lo que faltó y lo que se podría mejorar en el proyecto. También se recomiendan cosas que, alguien que este haciendo un proyecto similar, pueda tomar en cuenta.

# Abstract

This project covers a desktop application created in the Java language in the NetBeans environment, which automatically creates Bank queues using object-oriented programming, using lists, for cycles, while-do cycles, if-else conditions, inheritance, among others.

This Java-based object-oriented programming project is based on a Banking queues program, which aims to streamline bank operations when organizing the queues and deliver a ticket to the user through previously established priorities, Which, in this case, includes 5 bank teller and a service platform, which only serves a specific letter. There is a single row that will have a maximum of twenty-five customers who will be vacated when there is a bank teller available and are attended or because they retired. It includes an impression of the clients served and the clients who retired before being attended. The clients served are printed for each bank teller and the average time in which the customers were served.

It implements a tolerance time of the clients waiting between three and forty-five minutes. Customers who left unattended should print on the final prints. Already organized in the bank teller, the clients have an average duration of procedures that go around between the five and thirty minutes.

With the realization of this project you can get more knowledge about cycles, lists, rows and queues, inheritance and many more things, better teamwork and not ideas of the working world in the career of engineering systems.

We solved all the problems that presented us with everything seen in class and also investigated what we did not know how the implementation of time.

# Objetivos

## Objetivos Generales

* Implementar una aplicación para el manejo de filas en un banco utilizando estructuras de datos dinámicas.

## Objetivos Específicos

* Implementación de filas y colas con prioridad.
* Agregar filas con un máximo de clientes.
* Implementación de un tiempo de espera y de tolerancia de los clientes.
* Distribución de los clientes en 5 cajas y una plataforma.

# Introducción

Este proyecto realizado en Java mediante Programación Orientada a Objetos está basado en realizar un programa de colas de Bancos, el cual tiene como objetivo agilizar las operaciones de los bancos a la hora de organizar las filas y entregar un tiquete al usuario mediante prioridades previamente establecidas, que, en este caso, incluyen 5 cajas y una plataforma de servicios, que solo atiende a una letra en específico.

Existe una única fila que tendrá un máximo de 25 clientes los cuales la irán desocupando cuando haya una caja disponible y sean atendidos o bien porque se retiraron.

Se incluye una impresión de los clientes atendidos y los clientes que se retiraron antes de ser atendidos. Se imprimen por cada caja los clientes atendidos y el tiempo promedio en el que fueron atendidos los clientes.

# Presentación y análisis del problema

¿Qué es lo que hay que resolver?

Hay que hacer una fila de clientes con un máximo de 25 clientes los cuales serán acomodados en cada una de las cajas según sea su prioridad y también se enviarán a la plataforma a los clientes que así lo necesiten. Se implementa un tiempo de tolerancia de los clientes en espera de entre tres y cuarentaicinco minutos. Los clientes que se retiraron sin ser atendidos se tienen que imprimir en las impresiones finales.

Ya organizados en las cajas, los clientes tienen una duración promedio de trámites que rondan entre los 5 y 30 minutos.

# Diagrama de Clases



# Análisis del problema

Este proyecto es complicado ya que se tienen que implementar muchos ciclos y muchas listas en las cuales se realizarán operaciones random (aleatorias) las cuales nos devolverán el valor de las filas, clientes, operaciones de los clientes, tiempo de atención del cliente y los clientes que se retiran sin ser atendidos.

Con estos elementos vienen muchos retos ya que hay que implementar los diferentes valores en diferentes listas y métodos que tienen que ir sincronizados correctamente para que el programa corra correctamente.

# Análisis de la solución

Pudimos realizar la mayoría de implementaciones gracias a lo aprendido en clase en las últimas semanas como lo son las filas y colas con prioridad, los random que aprendimos a implementar en proyectos programados anteriores, así como la herencia y la programación de objetos en general.

Tambien realizamos los menus correspondientes de las opciones del programa con los switch-case aprendidos.

# Discusión de resultados y Conclusiones

Con la realización de este proyecto puede obtener más conocimientos acerca de los ciclos, los arreglos de listas, las filas y colas, la herencia y muchas cosas más, gracias a este proyecto vamos teniendo las bases para, a futuro, hay que nos hace trabajar mejor en equipo y nos da ideas del mundo laboral en la carrera de ingeniería en sistemas.

Resolvimos todos los problemas que se nos presentaron con todo lo visto en clase y también investigamos lo que no sabíamos cómo la implementación del tiempo.

# Recomendaciones

A la hora de realizar un proyecto de estas características, es muy importante primero analizar todo lo que se necesita, ya sean elementos que se tienen que implementar o pensar en el resultado en el cual se quiere terminar, ya que así nos damos una idea de todo lo que hay que hacer para realizarlo. También es muy importante investigar un poco más de lo que aprendemos en clase ya que en la universidad no vamos a aprender el 100% de lo que necesitamos y así vamos a tener más conocimientos de que implementar.

Asimismo, hay que validar siempre los ingresos de datos para que no se puedan ingresar datos no deseados. También hay que imprimirle al usuario todo lo que tiene que hacer para utilizar el programa correctamente y guiarlo en su uso.

# Cronograma de trabajo

Bitácora

|  |  |  |
| --- | --- | --- |
| Actividades | Descripción | Tiempo |
| Análisis y planteamientos del problema | Buscar que se tiene que hacer para realizar el proyecto | 2 horas |
| Investigación | Investigación de los pasos a seguir para realizar el programa | 24 horas |
| Estructuración | Planear la estructura del programa | 3 horas |
| Elaboración del programa | Programación del programa en NetBeans | 48 horas |
| Documento escrito | Realización de la documentación formal escrita | 15 horas |
| Pruebas y agregados al programa | Se hacen pruebas al programa de verificación | 2 horas |
| Total | Total de horas del proyecto | 94 horas |

# Anexos

 

 