Programación Concurrente Cliente Servidor

Marcela Cascante Quirós

Diego Delgado

Alejandro Loaiza

Jose David Mora Loría

**Trabajo Escrito**

Jean Carlos Rojas

**“Sistema Automatizado de Control Bancario”**

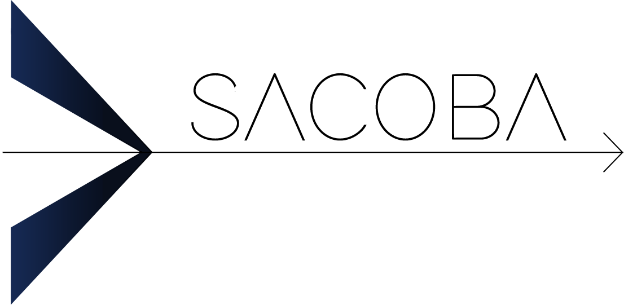
****

Tabla de contenido

[Objetivos 3](#_Toc459873797)

[Objetivo General 3](#_Toc459873798)

[Objetivos específicos 3](#_Toc459873799)

[Introducción 4](#_Toc459873800)

[Desarrollo 5](#_Toc459873801)

[SACOBA 5](#_Toc459873802)

[Herencia 6](#_Toc459873803)

[Hilos 7](#_Toc459873804)

[TCP 7](#_Toc459873805)

[Listas o Arreglos 8](#_Toc459873806)

[Listas Enlazadas 8](#_Toc459873807)

[Árboles 8](#_Toc459873808)

[Colas 8](#_Toc459873809)

[Excepciones 9](#_Toc459873810)

[Conclusiones 10](#_Toc459873811)

[Referencias 11](#_Toc459873812)

# Objetivos

## Objetivo General

Crear un sistema automatizado de control bancario para agilizar todos los trámites posibles dentro del mismo, brindando una mejor calidad de servicio a los usuarios.

## Objetivos específicos

* Crear un servidor que se encargue de manipular la información adecuadamente
* Mantener el orden a la hora de atender a los clientes.
* Crear diferentes tipos de clientes con funciones específicas.

# Introducción

El trabajo que se presenta a continuación pretende encontrar la mejor forma de solucionar los problemas a la hora de organizar las diferentes diligencias que se pueden realizar en una entidad bancaria.

Todo esto por medio de una serie de clientes conectados a un servidor, los cuales desempeñan diferentes funciones para agilizar la atención al usuario. Los anteriores se encuentran conectados por medio de TCP y logran intercambiar la información necesaria para que la aplicación funcione correctamente.

EL proyecto se presentó como una respuesta al problemas de las largas filas bancarias las cuales consumen horas del usuario innecesariamente, esto sucede en especial en las fechas de pago, a pesar de que ahora muchos de los tramites se manejan a través de internet, por lo que se decidió crear una herramienta que se encargue de organizar, de la manera más eficiente, la forma en la que se atienden las necesidades de cada persona.

# Desarrollo

Una de las principales características que tenemos en el lenguaje JAVA es la de ser un lenguaje compilado e interpretado. Todo programa en Java es de compilarse y el código que se genera “bytecodes” es interpretado por una máquina virtual. El código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma.

*“Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzará a ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto.” (Prof. Jean Carlos R.)*

## SACOBA

El Sistema de Control Bancario (SACOBA) brindará de una manera rápida y accesible, una gran ayuda para eliminar las filas en los bancos de forma precisa y ordenada. El usuario al ingresar al banco se registrará en una de las puertas con su cédula o invitado en caso de ser nuevo cliente y proveerá al computador cuál es el motivo de su visita seleccionando una de las opciones disponibles, sea algún trámite, transferencia, problema con las cuentas, entre otras situaciones.

Este proyecto ayudará a los bancos a mantener un orden de una forma más precisa que con los métodos actuales, asignándoles a los usuarios una prioridad dependiendo de la importancia de la solicitud que desean realizar. Los usuarios podrán ingresar desde una interfaz gráfica y sencilla a realizar los procesos necesarios y mantenerse al tanto de su posición en la fila. Al mismo tiempo los agentes del banco podrán estar al tanto de qué clientes tienen que atender, tendrán una breve idea del trámite que los usuarios necesitan y podrán prepararse de antemano a las necesidades del usuario, agilizando así el proceso que estos requieren.

De esta forma vamos a implementar los distintos temas vistos en clase, *Programación Concurrente y Cliente Servidor*, los cuales vamos a ir desarrollándolos conforme vamos estructurando el programa del Sistema de Control Bancario (SACOBA).  
En el proyecto vamos a desarrollar y estructurar muy bien los siguientes temas: Herencia, Interfaces Gráficas, Hilos, TCP, Listas o Arreglos, Listas Enlazadas, Arboles, Colas y Excepciones.

## Herencia

La herencia induce una jerarquía en forma de árbol sobre las clases con raíz en la clase Object.   
Una clase se dice que hereda o extiende a otra clase antecesora. La palabra reservada “extends” sirve para indicar que una clase extiende a otra. La clase que extiende a otra hereda todos los atributos y métodos de la clase antecesora que no sean privados (private). La clase antecesora puede extender a su vez otra clase.

Una de las ventajas a la hora de implementar este tema en nuestro proyecto fue que usamos las variables y métodos de una clase existente en otra clase, facilitándonos el compartimiento de los diversos métodos.

Interfaces Graficas

**En nuestro proyecto vamos a tener una interfaz gráfica de usuario,** interfaz de usuario o GUI (Graphical User Interface): es el entorno de objetos gráficos disponibles para un usuario en el marco de una aplicación o sistema operativo.   
También podemos decir, que una interfaz gráfica es un conjunto de componentes gráficos que posibilitan la interacción entre el usuario y la aplicación. Es decir ventanas, botones, combos, listas, cajas de diálogo, campos de texto, etc.

## Hilos

Podemos decir que un Hilo es un salida de control independiente dentro de un programa. Como tal tiene su propia pila de llamadas, pero todos los Hilos creados en el mismo programa comparten el mismo espacio de direcciones, lo que implica que comparten todas las instancias del programa excepto las locales.   
Nuestros programas pueden tener un gran número de Hilos ejecutándose al mismo tiempo y todos ellos comparten el uso de la CPU.

## TCP

Nuestro objetivo con este tema, es que, con el uso del protocolo TCP, las aplicaciones pueden comunicarse en forma segura (gracias al sistema de recibo del protocolo TCP. Esto significa que los routers (que funcionan en la capa de Internet) sólo tienen que enviar los datos en forma de datagramas, sin preocuparse con el monitoreo de datos porque esta función la cumple específicamente el protocolo TCP.

Algunas características importantes que debemos tener en cuenta a la hora de implementar el Protocolo de Control de Transmisión, son las siguientes:

* TCP permite colocar los datagramas nuevamente en orden cuando vienen del protocolo IP.
* Los puertos pueden ser entre 0 a 65.535, algunos sistemas operativos pueden utilizar los menores a 1024.
* TCP permite que el monitoreo del flujo de los datos y así evitar la saturación de la red.
* TCP permite comenzar y finalizar la comunicación amablemente

## Listas o Arreglos

Permiten almacenar un conjunto de datos del mismo tipo, organizados en una sola columna y uno o más renglones.

Nos basamos en algunas características importantes a la hora de implementar los arreglos y listas, son los siguientes:

* Ambos representan secuencias de elementos de un tipo.
* Los arreglos tienen longitud fija; las listas, no.
* Los elementos de un arreglo pueden accederse en forma independiente
* Los de la lista se acceden secuencialmente, empezando por la cabeza

## Listas Enlazadas

La lista enlazada básicamente es la lista enlazada simple, la cual tiene enlace por nodo. Este enlace apunta el siguiente nodo o indica que tiene la dirección en memoria del siguiente nodo en la lista, o al valor NULL o a la lista vacía, si es el último nodo.

## Árboles

Un árbol es una estructura de datos divididos (no lineal) que puede representarse como un conjunto de nodos enlazados entre sí por medio de ramas. La información contenida en un nodo puede ser de cualquier tipo simple o estructura de datos.

## Colas

Una cola es una [estructura de datos](http://es.wikipedia.org/wiki/Estructura_de_datos), caracterizada por ser una secuencia de elementos en la que la operación de introducción “push” se realiza por un extremo y la operación de extracción “pop” por el otro. También se le llama estructura “[FIFO](http://es.wikipedia.org/wiki/FIFO)” (del inglés First In First Out), debido a que el primer elemento en entrar será también el primero en salir.

En el proyecto vamos a tener que al ingresar estos dos datos (cedula y motivo de la visita), el servidor agregará al usuario a una de las tres colas disponibles: Trámites, Plataforma y Cuentas. Cada cola tiene su prioridad y llevará el conteo de los usuarios presentes en cada una y les asignará un número de secuencia según la cola a la que sean asignados y número de usuario ingresado a la misma.  
Al ingresar al banco luego de registrarse, los usuarios podrán ver en un monitor la cantidad de personas en cada cola, así como que usuario sigue en la fila y a que caja deberá proceder cuando la respectiva caja para esta cola esté libre.

## Excepciones

Una excepción es una situación extraña a la que llega la ejecución de un programa. Estas situaciones extrañas pueden ser desde el intento de abrir un fichero que no existe, a la división por cero.

# Conclusiones

Se ha logrado crear el programa con total funcionalidad y cumpliendo los objetivos propuestos al inicio de este trabajo, además esto se logró por medio de la implementación de cada uno de los temas vitos en clase.

La conexión cliente-servidor se realizó de una manera satisfactoria por medio de la implementación de TCP. El servidor funciona de manera correcta a la hora de manipular la información que recibe.

Se crearon diferentes tipos de clientes que cumplen funciones muy específicas lo cual brinda un mejor orden a la hora de orientar a los usuarios.

En general el programa cumple satisfactoriamente todas las expectativas, desde su código hasta su interfaz gráica.

# Referencias

Calzada, R. (2011). *Estructura de Datos.* Madrid: estructura-de-datos-garo.blogspot.com.

Fernandez, O. B. (2014). *Desarrollo de Proyectos Informáticos con Tecnologia Java.* España: Universidad Despacho.

Fernández, O. B. (2015). *Introducción al lenguaje de programación Java.* España: Universidad Despacho.

Khaled, O. (2016). *TCP/IP.* España, Madrid: Applied Science & Technology Source.

Lapuente, M. J. (2013). *Interfaz Grafica.* España: hipertexto.info.

Rancel, M. R. (2006). *Aprende a programar.* España: aprenderaprogramar.com.

Rojas, J. C. (2016). *TCP.* Costa Rica: PowerPoint Online.

Saturno, W. (2013). *Pilas, Listas y Colas Estructuras de Datos.* Republica Dominicana: LinkedIn Corporation.