Avance Final

Curso: Programación Concurrente Cliente Servidor

Karl Aase Blanco

Pablo Bustos Cortés

Jeffry López Meneses

José Gabriel Rivas Zúñiga

Eduardo Leonel Castillo Rodas

Profesor: Fernando Salas

III Cuatrimestre 2019

Manual técnico del proyecto

Clases

Cliente

- frmMenu
- frmClient
- classLogin
- frmLogin
- MainClass
- frmNewClient

Servidor

- BusCall
- BusConfig
- Servidor
- ServerConfig
- Time

Cliente

MainClass

Funcionalidad:

Esta es considerada la clase main del código ya que inicializa el login para poder iniciar todo el programa.

frmMenu

Funcionalidad:

Esta es mostrada posterior al login y cuenta con 2 botones que permiten visualizar los buses pasando al Jframe “frmClient” o el botón salir el programa el cual envía un mensaje en el socket cómo Reply “False” para cerrar el servidor y posteriormente el cliente.

frmClient

Variables:

Socket sc;

DataInputStream in;

DataOutputStream out;

final String HOST = "127.0.0.1";

final int PUERTO = 5000;

boolean Bus1 = false;

boolean Bus2 = false;

boolean Bus3 = false;

boolean Bus4 = false;

boolean Bus5 = false;

Funcionalidad:

La clase se encarga de crear 5 buses los cuales consultarán y actualizarán la información en el Jframe correspondientemente al recibo de datos proveniente del servidor, estos buses se encuentran en un hilo sincronizado para recibir y actualizar en orden.

Métodos:

run () : Ejecuta el hilo.

ChangeColorLabels(String label) : Cambia los colores de los buses en la interfaz.

Filter (String msg, boolean Bus1, boolean Bus2, boolean Bus3, boolean Bus4, boolean Bus5): Actualiza la información de los buses en la interfaz.

SocketClient(): establece comunicación con el servidor.

frmLogin

Funcionalidad:

El jframe se encarga de capturar datos mediante un JTextField y un password field para poder verificar en la clase “classLogin”

classLogin

Variables:

ArrayList<String> users = new ArrayList<>();

ArrayList<String> pass = new ArrayList<>();

Funcionalidad:

Se encarga de verificar los usuarios y contraseñas introducidos en el Jframe “frmLogin” con respecto a los que se encuentres respectivamente en los Arraylist user y pass.

Métodos:

Verificar (String users, String pass): Verifica los datos introducidos con los que se encuentran en los Arraylist user y pass.

ArrayList<String> getUsers() : Retorna el usuario.

setUsers(ArrayList<String> users) : Establece usuario.

ArrayList<String> getPass(): Retorna contraseña.

setPass(ArrayList<String> pass) : Establece contraseña.

frmNewClient

Variables:

String name = txtName.getText();

String Pass = new String (txtPass.getPassword());

Funcionalidad:

Genera nuevos usuarios y los registra en el “classLogin” para luego ser verificados.

Servidor

BusCall

Variables:

boolean status = true;
BusConfig miHilo 1;
BusConfig niHilo 2;
BusConfig miHilo3;
BusConfig miHilo 4;
BusConfig miHilo 5;

Funcionalidad:

En esta clase se encarga de realizar el llamado y la creación adecuado de los buses, es decir, de los hilos, con su respectiva información y posición en la parada que se le asigne.

Métodos:

public BusCall() : Constructor de la clase, crea los hilos con sus información, esto gracias a la instancia de la clase BusConfig.

BusConfig

Variables:

Time t = new Time();
Thread Bus;
String p[];

```
int v = 0;
int count;
static String message = "";
static String busName = "";
static Server s = new Server ();
```

Funcionalidad:

Clase que se encarga de crear la configuración necesaria para el bus, es decir, desde su posición en las paradas, al igual que da el inicio al hilo del bus, lo cual lo pone en ejecución y realiza el recorrido según las paradas de la ruta.

Métodos:

BusConfig(String nombre, String parada[], int position): Constructor para controlar los buses y la info que contendrá cada uno de estos.
BusConfig crearYComenzar(String nombre, String parada[], int position): Inicia el hilo y retorna el estado.
Run (): ejecuta el hilo e imprime el estado de los buses y ejecuta el método rutas.
Rutas (): imprime las paradas de buses (actual y posterior) y el tiempo transcurrido.

Servidor

Variables:

```
public static Scanner v = new Scanner (System.in);

ServerSocket servidor = null;

 Socket sc = null;

DataInputStream in;

DataOutputStream out;

final int PUERTO = 5000;
```

String mensaje = "";

Thread server;

static int close = 0;

static boolean stop = true;

Funcionalidad:

Esta clase lleva el control adecuado de toda la funcionalidad del programa, desde el establecimiento del canal de comunicación para el cliente, como la recepción y el envío de la información al cliente que se encuentre realizando las consultas.

Métodos:

Run (): Ejecuta el hilo principal del programa, inicializando el servidor, aceptando la conexión con el cliente y administrador dicha conexión, es decir, realiza el manejo del socket en el cual dependiendo si el cliente desee cerrar conexión o bien que no se encuentre ningún bus en ruta, permitiendo al servidor poder concluir la comunicación de forma satisfactoria.

SendMessage (String Msg): Realiza el envío del mensaje de actualización de la ruta del bus.

Time

Variables:

String hora;

String minutos;

String segundos;

String ampm;

String time;

Calendar calendario;

Thread h1;

Funcionalidad:

Se encarga en la sincronización del tiempo de sleep y activación de los buses, con el fin de lograr una mejor funcionalidad de las diferentes partes.

Métodos:

Time (): Instancia del hilo Time y su ejecución.

Run (): Método de inicio del hilo, llama al método Capturador.

Capturador (): Captura el tiempo del sistema, asignando un valor a los atributos de hora, minutos y segundos.

TimeTransform(): Realiza la transformación de los minutos, segundos y horas en caso que sea necesario.