Universdad Nacional de Colombia

Práctica 2
Informe de elaboración

Sistemas Operativos

- Jose Alberto Cordoba Higuera
- Joel Esteban Diaz Arévalo
- Karen Medina Valderrama

---

1. Introducción

Para el desarrollo de esta práctica se requiere la creación de un programa en el que, utilizando los conceptos de comunicación entre procesos, se ejecute una conexión cliente-servidor que gestione la información de los tiempos de viaje entre zonas de Bogotá suministrada por Uber-Movement en un archivo CSV. El programa manejará los siguientes datos con sus respectivos nombres de variables:

- ID del origen: sourceid
- ID del destino: dstid
- Hora del día: hod
- Media del tiempo de viaje: mean_travel_time
- Desviación estándar del tiempo de viaje: standard_deviation_travel_time
- Media geométrica del tiempo de viaje: geometric_mean_travel_time
- Desviación geométrica estándar del tiempo de viaje: geometric_standard_deviation_travel_time

Para esto, se manejará la lógica del programa principal en dos programas:

1.1. Servidor:

El programa servidor deberá gestionar el archivo suministrado por Uber, que almacena los tiempos de viaje. A su vez, el servidor podrá recibir peticiones de clientes a través de la red, para realizar cada una de las solicitudes especificadas previamente: Ingresar origen, Ingresar destino, Ingresar hora, Buscar tiempo de viaje medio. Al entrar en ejecución, el servidor no mostrará ningún menú ni mensaje alguno. El servidor se limitará a gestionar las operaciones con los clientes y un archivo log donde se registrarán las operaciones que se estén realizando. El formato log de cada operación es el siguiente:

[Fecha YYYYMMDDTHHMMSS] Cliente [IP] [búsqueda - origen - destino]

1.2. Cliente:

El programa cliente deberá mostrar el siguiente menú:

Bienvenid@:

1. **Ingresar origen** (Recibe un número entre 1 y 1160)
2. **Ingresar destino** (Recibe un número entre 1 y 1160)
3. **Ingresar hora** (Recibe un número entre 0 y 23)
4. **Buscar tiempo de viaje medio** (Valida e imprime el resultado de la búsqueda)
5. **Salir** (Finaliza el programa del cliente)

Cuando se digite una opción, esta deberá ser enviada al servidor a fin de iniciar su gestión.

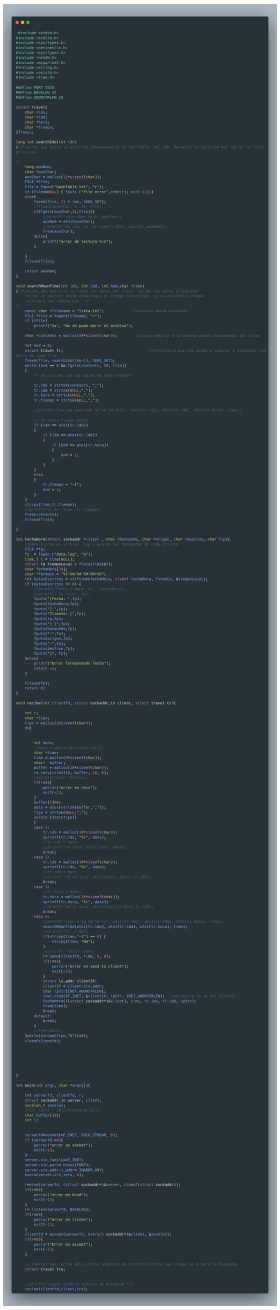
2. Desarrollo

2.1. Código

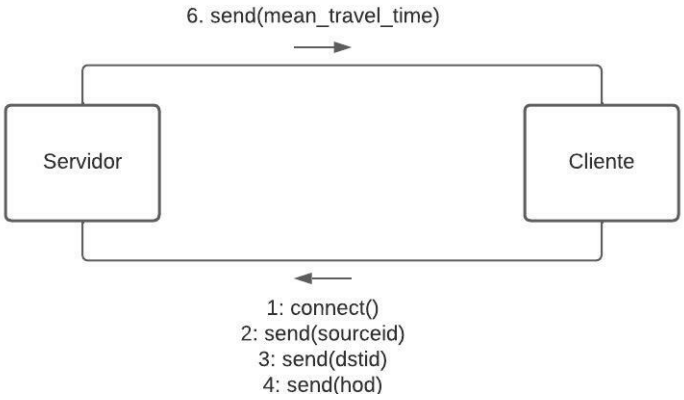
El programa está compuesto por dos archivos .c principales:
p2-client.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <time.h>
#include <string.h>
#include <termios.h>

#define PORT 3535
#define BUFFERL 10

int mygetch ( void )
{
//Función  importante para el "Press any key to continue"
    int ch;
    struct termios oldt, newt;

    tcgetattr ( STDIN_FILENO, &oldt );
    newt = oldt;
    newt.c_lflag &= ~( ICANON | ECHO );
    tcsetattr ( STDIN_FILENO, TCSANOW, &newt );
    ch = getchar();
    tcsetattr ( STDIN_FILENO, TCSANOW, &oldt );
    return ch;
}

struct datosBusqueda{
    int origen;
    int destino;
    int hora;
};

int revisarDatos(struct datosBusqueda data, int aux){
//esta función no sirve que se de la orden de inicio de búsqueda si no se han modificado los datos primero
if(data.origen==-1){
    printf("No ha ingresado el ID del origen \n");
    aux = -1;
}
if(data.destino==-1){
    printf("No ha ingresado el ID del destino \n");
    aux = -1;
}
if(data.hora==-1){
    printf("No ha ingresado la hora del viaje \n ");
    aux = -1;
}
return aux;
}
int menu(){



 return 0;
}

int main(int argc, char *argv[]){

    time_t tiempo_inicio, tiempo_final;
    double segundos;
    int clientfd,r;
    int select = 0;
    int aux = 0;
    struct sockaddr_in client;
    struct hostent *he;
    struct datosBusqueda d1 = {-1,-1,-1};
    char buffer[BUFFERL];
    clientfd = socket(AF_INET, SOCK_STREAM, 0);

    if(clientfd<0){
        perror("error en socket");
        exit(-1);
    }
    client.sin_family = AF_INET;
    client.sin_port = htons(PORT);
    inet_aton("127.0.0.1", &client.sin_addr);

    r= connect(clientfd, (struct sockaddr*)&client, (socklen_t)sizeof(struct sockaddr));
    if(r<0){
        perror("error en connect");
        exit(-1);
    }
    startover:
        do {
        system("clear");
        printf("Bienvenide al menu principal\n");
        printf("1. Ingresar origen \n");
        printf("2. Ingresar destino \n");
        printf("3. Ingresar hora\n");
        printf("4. Buscar tiempo de viaje medio \n");
        printf("5. Salir \n");
        printf("Seleccione una opción (1-5) \n");
        scanf("%d", &select);
        }
        while((select < 1) && (select > 4));

        switch(select)
        {
            case 1:
                printf("Origen\n");
                scanf("%d", &d1.origen);
                sprintf(buffer,"%d", d1.origen);
                strcat(buffer , ",1" );
                r=send(clientfd, buffer, BUFFERL, 0);
                if(r<0){
            perror("error en send");
            exit(-1);
            }
            system("clear");
            printf("Has ingresado el dato: %d en origen. \n", d1.origen);
            printf("%c Press any key to continue.\n", getc(stdin));
                mygetch();
                goto startover;
            case 2:
                printf("Destino \n");
                scanf("%d",&d1.destino);
                sprint(buffer,"%d", d1.destino);
                strcat(buffer , ",2" );
                r=send(clientfd, buffer, BUFFERL, 0);
                if(r<0){
            perror("error en send");
            exit(-1);
            }
            system("clear");
            printf("Has ingresado el dato: %d en destino. \n", d1.destino);
            printf("%c Press any key to continue.\n", getc(stdin));
                mygetch();

                goto startover;
            case 3:
                printf("Hora\n");
                scanf("%d", &d1.hora);
                sprintf(buffer,"%d", d1.hora);
                strcat(buffer , ",3 " );
                r=send(clientfd, buffer, BUFFERL, 0);
                if(r<0){
            perror("error en send");
            exit(-1);
            }
            system("clear");
            printf("Has ingresado el dato: %d hora. \n", d1.hora);
            printf("%c Press any key to continue.\n", getc(stdin));
                mygetch();
                goto startover;
            case 4:
                system("clear");
                printf("Buscando...\n");
                strcpy(buffer, "0,4");
                if(revisarDatos(d1, aux)==-1){
                    printf("Ingrese los datos completos \n");
                    goto startover;
                } else {
                    printf ("Origen = %d, Destino = %d, Hora = %d \n", d1.origen, d1.destino, d1.hora);
                    r=send(clientfd, buffer, BUFFERL, 0);
                    if(r<0){
                        perror("error en send");
                        exit(-1);
                    }
                //printf("%c Press any key to continue.\n", getc(stdin));
                //mygetch();
                r= recv(clientfd, buffer, 8, 0);
                if(r<0){
                    perror("error en recv");
                    exit(-1);
                }
                buffer[r]=0;
                printf("\n Tiempo de viaje medio: %s", buffer);}
                printf("%c Press any key to continue.\n", getc(stdin));
                mygetch();
                goto startover;
            case 5:
                system("clear");
                printf("Saliendo...\n");
                printf("%c Press any key to continue.\n", getc(stdin));
                mygetch();
                printf("Bye.\n");
                strcpy(buffer, ",5");
                r=send(clientfd, buffer, BUFFERL, 0);
                if(r<0){
            perror("error en send");
            exit(-1);
            }
            close(clientfd);
                exit(0);
                //return 0;
            default:
                printf("Por favor selecione una opción válida\n");
                goto startover;
        }

}
```

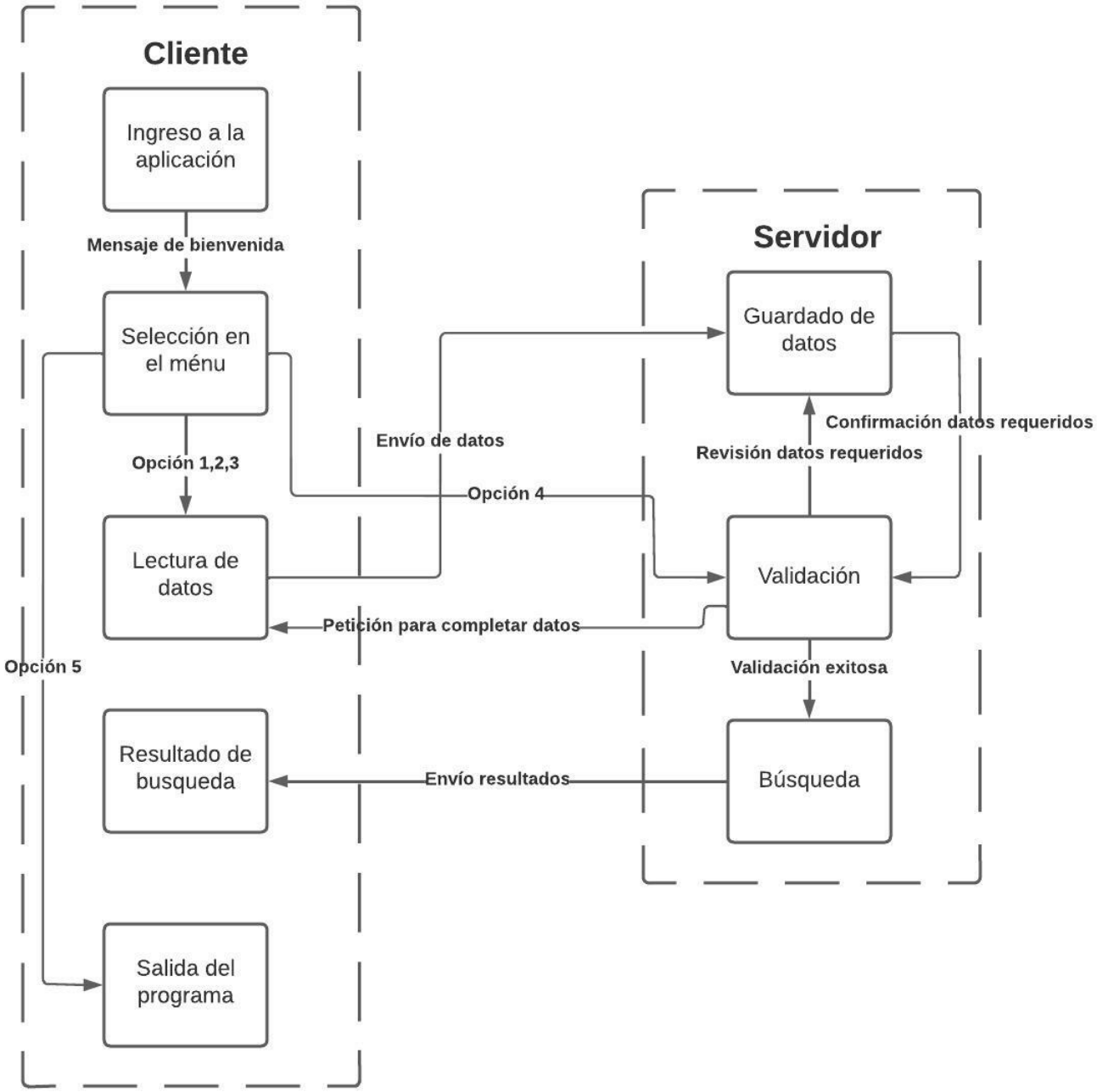
p2-server.c



2.2. Diagrama de comunicaciones



2.3. Diagrama de Bloques


Cliente
Ingreso a la aplicación
Mensaje de bienvenida
Selección en el ménu
Opción 1,2,3
Lectura de datos
Opción 5
Resultado de busqueda
Salida del programa
Servidor
Guardado de datos
Envío de datos
Opción 4
Confirmación datos requeridos
Revisión datos requeridos
Validación
Petición para completar datos
Validación exitosa
Búsqueda
Envío resultados