Joaquim Rafael Mariano Prieto Pereira - 10408805
Henrique Arabe Neres de Farias - 10410152

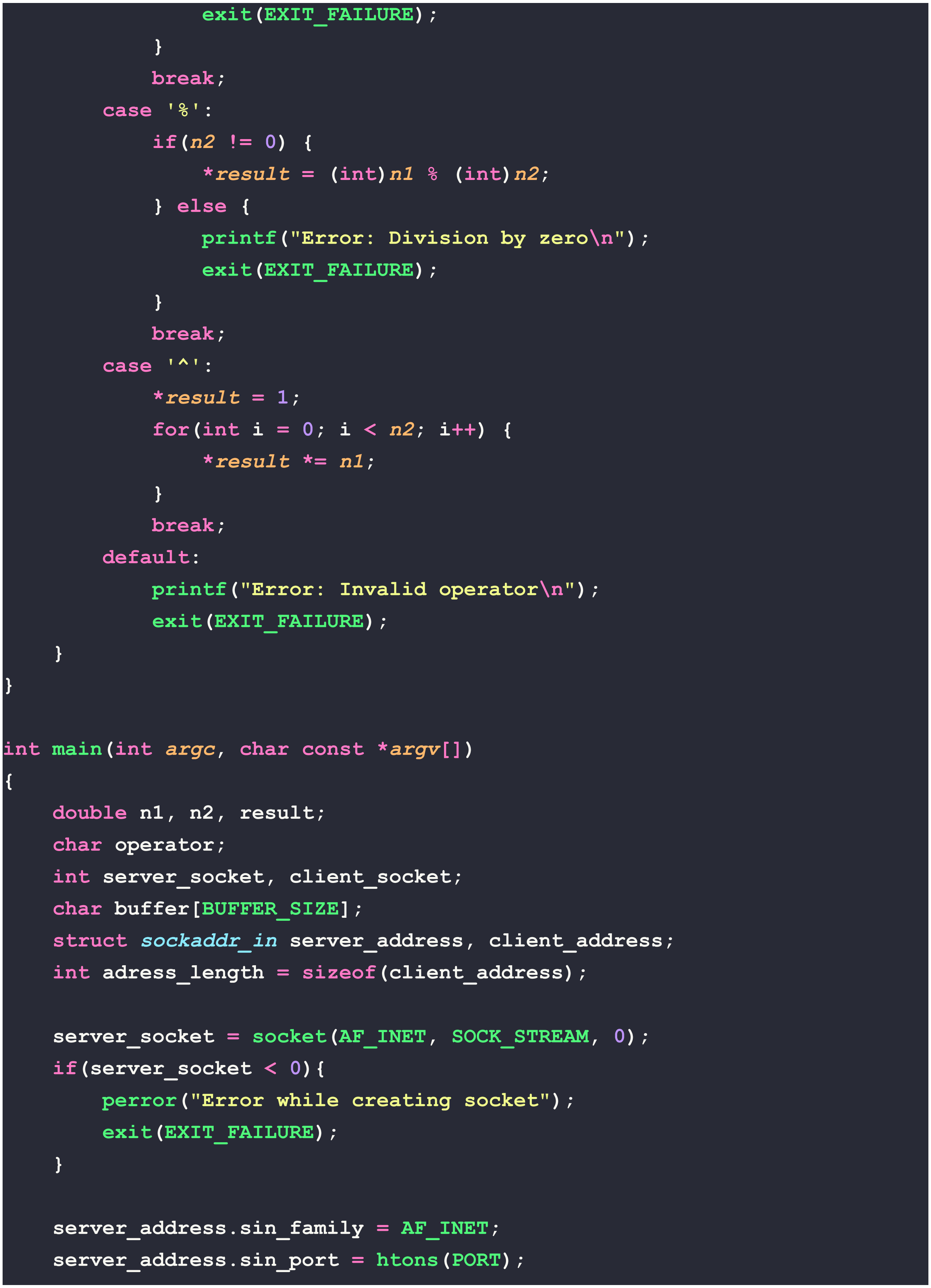
- **Repositório**

https://github.com/joaquimrafael/DistributedSystems

- **Código**

1. **Servidor.c**

```c
/*Laboratorio de Computacao Distribuida
Joaquim Rafael Mariano Prieto Pereira - 10408805
Henrique Arabe Neres de Farias - 10410152
Exercicio: Atv. Socket Calc */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <string.h>
#include <netinet/in.h>

#define PORT 8080
#define BUFFER_SIZE 1024

void calculator(double n1, double n2, char operator, double *result) {
    switch(operator) {
        case '+':
            *result = n1 + n2;
            break;
        case '-':
            *result = n1 - n2;
            break;
        case '*':
            *result = n1 * n2;
            break;
        case '/':
            if(n2 != 0) {
                *result = n1 / n2;
            } else {
                printf("Error: Division by zero\n");
```

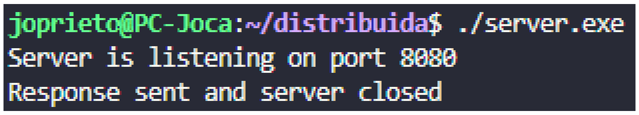


```c
                exit(EXIT_FAILURE);
            }
            break;
        case '%':
            if(n2 != 0) {
                *result = (int)n1 % (int)n2;
            } else {
                printf("Error: Division by zero\n");
                exit(EXIT_FAILURE);
            }
            break;
        case '^':
            *result = 1;
            for(int i = 0; i < n2; i++) {
                *result *= n1;
            }
            break;
        default:
            printf("Error: Invalid operator\n");
            exit(EXIT_FAILURE);
    }
}

int main(int argc, char const *argv[])
{
    double n1, n2, result;
    char operator;
    int server_socket, client_socket;
    char buffer[BUFFER_SIZE];
    struct sockaddr_in server_address, client_address;
    int adress_length = sizeof(client_address);

    server_socket = socket(AF_INET, SOCK_STREAM, 0);
    if(server_socket < 0){
        perror("Error while creating socket");
        exit(EXIT_FAILURE);
    }

    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(PORT);
    server_address.sin_addr.s_addr = INADDR_ANY;

    if (bind(server_socket, (struct sockaddr *) &server_address, sizeof(server_address)) < 0) {
        perror("Erro no bind");
        exit(EXIT_FAILURE);
    }

    if(listen(server_socket, 1) < 0) {
        perror("Error while listening");
        exit(EXIT_FAILURE);
    }
    printf("Server is listening on port %d\n", PORT);

    client_socket = accept(server_socket, (struct sockaddr *) &client_address, (socklen_t*)&adress_length);
    if(client_socket < 0) {
        perror("Error while accepting connection");
        exit(EXIT_FAILURE);
    }

    int read_size = read(client_socket, buffer, BUFFER_SIZE - 1);
    if(read_size < 0) {
        perror("Error while reading from socket");
        close(client_socket);
        close(server_socket);
        exit(EXIT_FAILURE);
    }
    buffer[read_size] = '\0';

    if(sscanf(buffer, "%lf %c %lf", &n1, &operator, &n2) != 3) {
        char *msg = "Invalid input format. Expected: <number> <operator> <number>";
        send(client_socket, msg, strlen(msg), 0);
    }else { calculator(n1, n2, operator, &result);
        char result_str[100];
        sprintf(result_str, "Result: %.2lf", result);
        send(client_socket, result_str, strlen(result_str), 0);
    }
    printf("Response sent and server closed\n");
    close(client_socket);
    close(server_socket);
    return 0;
}
```

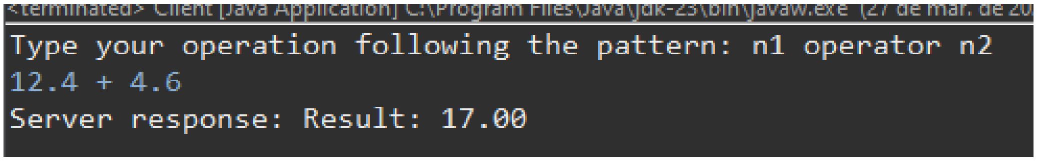
2. **Client.java**

```
/*Laboratorio de Computacao Distribuida
Joaquim Rafael Mariano Prieto Pereira - 10408805
Henrique Arabe Neres de Farias - 10410152
Exercicio: Atv. Socket Calc */
import java.io.*;
import java.net.*;
public class Client {
        public static void main(String[] args) {
                String serverAddres = "127.0.0.1"; // loopback
                int port = 8080;

                try(Socket socket = new Socket(serverAddres, port);
                        PrintWriter out = new PrintWriter(socket.getOutputStream(),
true);
              BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
              BufferedReader stdIn = new BufferedReader(new
InputStreamReader(System.in))) {

                                System.out.println("Type your operation following the
pattern: n1 operator n2");
                                String outputUser = stdIn.readLine();


                                out.println(outputUser);

                                String response = in.readLine();
                                System.out.println("Server response: "+ response);
                }catch (UnknownHostException  e) {
                        System.err.println("Unknown Host: "+ e.getMessage());
                }catch (IOException e) {
                        System.err.println("IO error: "+ e.getMessage());
                }
        }
}
```

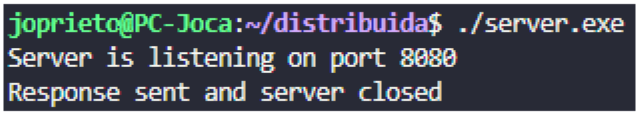
- **Testes**



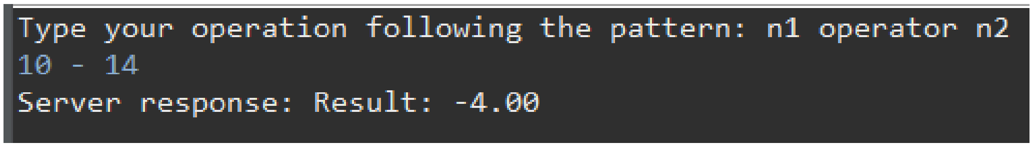
```
joprieto@PC-Joca:~/distribuida$ ./server.exe
Server is listening on port 8080
Response sent and server closed
```



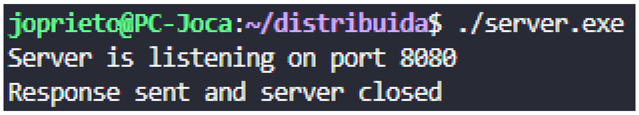
```
<terminated> Client [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (27 de mar. de 20
Type your operation following the pattern: n1 operator n2
12.4 + 4.6
Server response: Result: 17.00
```



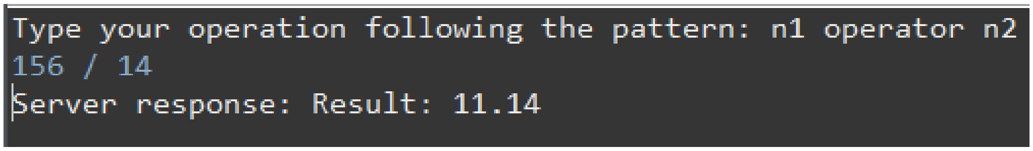
```
joprieto@PC-Joca:~/distribuida$ ./server.exe
Server is listening on port 8080
Response sent and server closed
```



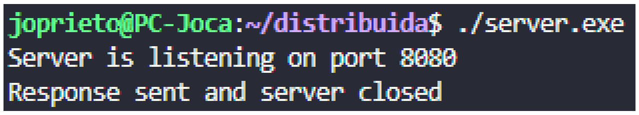
```
Type your operation following the pattern: n1 operator n2
10 - 14
Server response: Result: -4.00
```



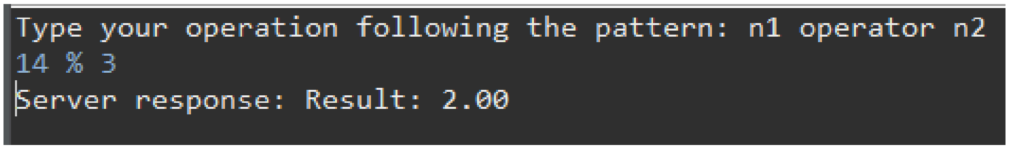
```
joprieto@PC-Joca:~/distribuida$ ./server.exe
Server is listening on port 8080
Response sent and server closed
```



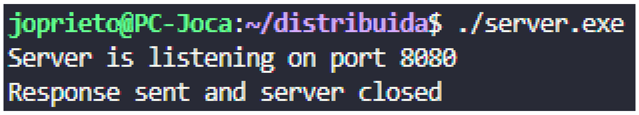
```
Type your operation following the pattern: n1 operator n2
156 / 14
Server response: Result: 11.14
```



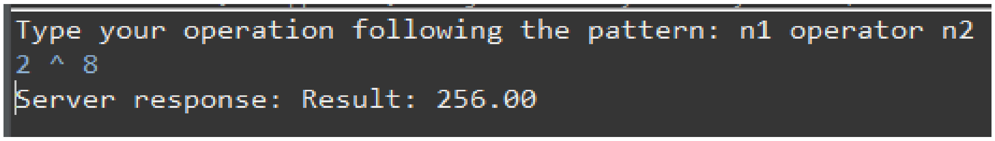
```
joprieto@PC-Joca:~/distribuida$ ./server.exe
Server is listening on port 8080
Response sent and server closed
```



```
Type your operation following the pattern: n1 operator n2
14 % 3
Server response: Result: 2.00
```



```
joprieto@PC-Joca:~/distribuida$ ./server.exe
Server is listening on port 8080
Response sent and server closed
```



```
Type your operation following the pattern: n1 operator n2
2 ^ 8
Server response: Result: 256.00
```