**UNIVERSIDADE DE FRANCA – UNIFRAN**

FABIO TOLEDO BONEMER DE SALVI

**ROTINA DE ATIVIDADE FÍSICA**

Experiência Prática III

ARARAQUARA – SÃO PAULO

2025

FABIO TOLEDO BONEMER DE SALVI

**ROTINA DE ATIVIDADE FÍSICA**

Experiência Prática III

Trabalho apresentado ao curso de Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade de Franca – UNIFRAN, como requisito para obtenção de nota de Experiência Prática III na disciplina Algoritmos e Pensamento Computacional.

ARARAQUARA – SÃO PAULO

2025

# Introdução

A proposta visa exercitar a lógica e o pensamento computacional por meio da resolução de problemas reais através do desenvolvimento de um programa funcional em linguagem C, utilizando vetores e matrizes n armazenamento e manipulação de grandes volumes de dados organizados, explorando a criatividade e a interação com o usuário.

A lógica para a solução do problema foi implementado em linguagem de programação C utilizando o ambiente de desenvolvimento *Visual Studio Code* ([MICROSOFT CORPORATION, 2025](https://code.visualstudio.com/)) em conjunto com as ferramentas de desenvolvimento e bibliotecas fornecidas pelo projeto *Minimalist GNU for Windows* ([MINGW-W64 PROJECT, 2025](https://www.mingw-w64.org/)), que contém ferramentas e bibliotecas que possibilitaram o desenvolvimento da aplicação de forma nativa em ambiente *Windows*.

A instalação e configuração do ambiente de desenvolvimento foi realizada seguindo os passos descritos em [MICROSOFT (2025](https://code.visualstudio.com/docs/cpp/config-mingw)) onde esta descrito como configurar o compilador GCC C++ (g++) e o depurador GDB.

O diretório contendo o repositório do projeto desta atividade está disponível em [Salvi (2025)](https://github.com/engfabiodesalvi/rain-by-zone.git).

# Descrição do Problema

O objetivo foi desenvolver um programa de computador em linguagem C que armazenasse e manipulasse grandes volumes de dados organizados na medição dos dados pluviométricos de uma determinada cidade, armazenando os registros na forma de uma matriz de duas dimensões, sendo uma dimensão responsável pelas informações do intervalo temporal das medições, definido em **dias,** e a outra dimensão responsável pelas informações da intensidade pluviométrica[[1]](#footnote-1) da chuva, comumente definida em **mm/h**.

A visualização dos dados é feita através de um gráfico de duas dimensões ilustrando as intensidades de cores, em num estilo denominado *Heatmap*[[2]](#footnote-2).

A solução proposta permite ao usuário visualizar as informações pluviométricas diárias registradas e registrar um novo conjunto de informação pluviométricas diárias.

## Requisitos que foram atendidos

O programa desenvolvido atende aos seguintes itens:

### Utiliza matrizes para armazenar os dados de intensidade pluviométrica;

### Período de análise defino em 15 dias e a 10 bairros para organizar os dados na matriz;

### Uso dos laços de repetição (*for*, *while*, *etc*) na leitura dos dados, no processamento dos cálculos, e na exibição dos resultados, evitando repetição na inclusão dos dados.

### Possui estrutura de repetição (laço) *while*, *for* e *do-while*;

### Contém estrutura de decisão (condicional) *if/else/else if* ou *switch*;

### Utiliza variáveis com nomes legíveis e significativos, que facilitem a compreensão do código;

### As mensagens e saídas para o usuário são claras e objetivas;

### O código está bem indentado e comentado adequadamente, explicando trechos importantes e a lógica usada;

### Permite que o usuário realize múltiplas operações até optar por sair do programa (*loop* principal).

### Utiliza funções e comandos básicos como *printf*, *scanf*, *if/else*, *switch*, *while*, *for*, entre outros.

### O programa foi testado em diferentes cenários e entradas de dados, funcionando corretamente.

## Organização do Código

### Apresentação do sistema: uma mensagem inicial explica o que o programa faz;

### Menu de opções: foi implementado dentro de um *loop while*, permitindo a repetição das ações;

### Tratamento das opções: uso de *if*/*else* ou *switch* no controle das funcionalidades conforme a escolha do usuário;

### Validação das entradas: garante que o usuário informe valores válidos e trata entradas incorretas com mensagens amigáveis;

### Mensagens de encerramento: ao sair do programa, apresenta uma mensagem de agradecimento;

# Código do programa em linguagem C

#include "ANSI-color-codes.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <Windows.h> // SetConsoleOutputCP()

#include <math.h> // Required for round() and pow()

#include <time.h> // For time() function

#include <limits.h>

#include <float.h> // FLT\_MAX

#include <locale.h>

#include <wchar.h>

// Three-Color Scale

// Define Start, Midpoint and End Colors

// Definindo a estrutura com as três cores bases do Heatmap

struct CoresHeatmap {

    int rgbBaixo[3]; // Cor do valor baixo

    int rgbMedio[3]; // Cor do valor médio

    int rgbAlto[3];  // Cor do valor alto

};

// Define estrutura com valores máximo e mínimo das medidades de intensidade de chuva

struct Limites {

    float valorMin;

    float valorMax;

};

// Define estrutura com os dados a serem impressos e com as configurações do gráfico

struct ConjuntoDados {

    // Valores máximo e mínimo das intensidades de chuva inseridas pelo usuario

    struct Limites limites;

    // Valores máximo e mínimo das intensidades de chuva dos valores aleatórios

    struct Limites limitesAleatorio;

    // Three-Color Scale

    // Escala de três cores

    // Variável com as três cores do Heatmap

    struct CoresHeatmap coresHeatmap;

    // Matriz nome dos bairros

    //char bairros[10][21];

    wchar\_t bairros[10][21];

    // Matriz nome dos bairros padrão

    //char bairrosPadrao[10][21];

    wchar\_t bairrosPadrao[10][21];

    // Matriz dados de intensidade de chuva por dia e bairro

    float medidaPluviometricaDiaBairro[15][10];

    // Matriz dados aleatórios de intensidade de chuva por dia e bairro

    float medidaPluviometricaDiaBairroAleatorio[15][10];

};

// Define estrutura com as opções de manipulacao do menu

struct Menu {

    // Definindo as variáveis de manipulação dos menus

    int menu;

    int returnMenu;

    int sairMenu;

    // Definindo as variáveis de manipulação das opções do menu

    int opcao;

    int returnOpcao;

    // Faixa de opções do menu

    int opcaoMinMax[2];

    // Opção selecinada pelo usuário

    // Dados aleatorios - 1 / Dados inseridos pelo usuario - 2

    int tipoDado;

    // Bairro selecionado

    int bairro;

    // novo bairro

    //char novoBairro[21];

    wchar\_t novoBairro[21];

    // Array para armazenar o retorno da função fgets()

    // [Erro -> NULL | Success -> aponta para a variável utilizada como buffer]

    char \*returnNovoCaracteres;

    // Indice para determinar o dia durante entrada de dados de intensidade pluviométrica

    int indiceDia;

    // novo valor com máximo de 6 dígitos - o útimo caracter é o '\n'

    char novoValor[7];

    // numero de valores inseridos na variavel novoValor

    int returnNovoValor;

    // nova Intensidade pluviométrica

    float novaIntensidadePluviometrica;

};

// Protótipo das funções

void apresentacaoSistema();

void menuInicio();

void menuPrincipal();

void menuVisualizarIntensidadePluviometrica(struct Menu \*menu, struct ConjuntoDados \*conjuntoDados);

void menuRegistrarIntensidadePluviometrica(struct Menu \*menu, struct ConjuntoDados \*conjuntoDados);

void menuFim();

void defineCoresHeatmap(struct CoresHeatmap \*coresHeatmap);

void inicializaMatriz(float medidaPluviometricaDiaBairro[15][10], struct Limites \*limites);

void defineLimites(float medidaPluviometricaDiaBairro[15][10], struct Limites \*limites);

//void imprimirHeatmap(char bairros[10][21], float medidaPluviometricaDiaBairro[15][10], struct Limites \*limites, struct CoresHeatmap \*coresHeatmap);

void imprimirHeatmap(wchar\_t bairros[10][21], float medidaPluviometricaDiaBairro[15][10], struct Limites \*limites, struct CoresHeatmap \*coresHeatmap);

void mensagemEncerramento();

int validaOpcaoMenu(struct Menu \*menu);

int validaEntradaNumerica(struct Menu \*menu);

int validaEntradaCaracters(struct Menu \*menu);

// Número máximo de dias

const int MAX\_DIAS = 15;

// Número máximo de bairros

const int MAX\_BAIRROS = 10;

// Número máximo de caracteres no nome do bairro

const int MAX\_BAIRRO\_CARACTERES = 21;

// Heatmap de intencidade pluviométrica

int main() {

    //  Mostrar caracteres acentuados

    //SetConsoleOutputCP( CP\_UTF8 );

    // habilita UTF-8 no terminal

    setlocale(LC\_ALL, "Portuguese\_Brazil.UTF-8");

    // Set the locale for numeric formatting

    // setlocale(LC\_NUMERIC, "C");

    // Declarando a estrutura de manipulação do menu

    struct Menu menu;

    // Definindo as variáveis de manipulação dos menus

    menu.menu = 0;

    menu.returnMenu = 0;

    menu.sairMenu = 0;

    // Definindo as variáveis de manipulação das opções do menu

    menu.opcao = 0;

    menu.returnOpcao = 0;

    // Faixa de opções do menu

    menu.opcaoMinMax[0] = 0;

    menu.opcaoMinMax[1] = 0;

    // Declarando a estrutura do conjunto de dados

    // Inicializando os dados com valores nulos

    struct ConjuntoDados conjuntoDados = {0};

    // Definindo as cores de transição do Heatmap

    defineCoresHeatmap(&conjuntoDados.coresHeatmap);

    // Matriz dias e bairros

    // Matriz medidaPluviometricaBairro aleatoria

    // Definindo valores nulos

    // i - dias / j - bairros

    for (int i; i < MAX\_DIAS; i++) {

        for (int j; j < MAX\_BAIRROS; j++){

            conjuntoDados.medidaPluviometricaDiaBairro[i][j] = 0.0;

            conjuntoDados.medidaPluviometricaDiaBairroAleatorio[i][j] = 0.0;

        }

    }

    // Valores máximo e mínimo das intensidades de chuva inseridas pelo susuário

    conjuntoDados.limites.valorMin = 0;

    conjuntoDados.limites.valorMax = 0;

    // Valores máximo e mínimo das intensidades de chuva dos valores aleatórios

    conjuntoDados.limitesAleatorio.valorMin = 0;

    conjuntoDados.limitesAleatorio.valorMax = 0;

    // Matriz nome dos bairros

    //char bairros[10][21] = {

    //     "Bairro/Região  1", "Bairro/Região  2", "Bairro/Regiao  3", "Bairro/Região  4", "Bairro/Região  5",

    //     "Bairro/Região  6", "Bairro/Região  7", "Bairro/Região  8", "Bairro/Região  9", "Bairro/Região 10"

    // };

    // Largura lógica de caracteres

    wchar\_t bairros[10][21] = {

        L"Bairro/Região  1", L"Bairro/Região  2", L"Bairro/Regiao  3", L"Bairro/Região  4", L"Bairro/Região  5",

        L"Bairro/Região  6", L"Bairro/Região  7", L"Bairro/Região  8", L"Bairro/Região  9", L"Bairro/Região 10"

    };

    // Definindo os nome iniciais para os bairros

    for (int i; i < MAX\_BAIRROS; i++) {

        //strcpy(conjuntoDados.bairros[i], bairros[i]);

        //strcpy(conjuntoDados.bairrosPadrao[i], bairros[i]);

        wcscpy(conjuntoDados.bairros[i], bairros[i]);

        wcscpy(conjuntoDados.bairrosPadrao[i], bairros[i]);

    }

    // Mensagem inicial

    apresentacaoSistema();

    // Menu 0 - Menu principal

    menu.menu = 0;

    menu.sairMenu = 0;

    // Loop dos menus

    do {

        if (menu.menu == 0) {

            // Menu para a opção 0

            // Primeiro menu:

            // Menu principal - "Menu de serviços"

            // Loop que garente uma entrada numérica válida

            do {

                // Menu principal -> "Menu de serviços"

                menuInicio();

                menuPrincipal();

                // Opção selecionada pelo usuario

                printf("Digite a opção desejada: ");

                menu.returnOpcao = scanf("%d", &menu.opcao);

                menuFim();

                // Opções do menu

                menu.opcaoMinMax[0] = 1;

                menu.opcaoMinMax[1] = 3;

                // Limpando o buffer de entrada!!!!

                while(getchar() != '\n');

            } while (!validaOpcaoMenu(&menu));

            // Analisa a opção selecionada

            if (menu.opcao >= 1 && menu.opcao <= 2) {

                menu.menu = menu.opcao;

            } else {

                // Sair do programa

                menu.sairMenu = 1;

            }

        } else if (menu.menu == 1) {

            // Menu 1 - Selecionar entre visualizar dados aleatórios ou visualizar dados registrados pelo usuário

            // Menu para a opção 1

            // Passos para chegar a este menu:

            // "Serviços disponíveis" -> "1 - Visualizar os registros da intensidade de chuva"

            // Loop que garante uma entrada numérica válida

            do {

                // Menu "Visualizar os registros da intensidade de chuva"

                // O usuário deve escolher entre visualizar os dados leatórios ou visualizar dados registrados pelo usuário

                menuInicio();

                menu.tipoDado = 0;

                menuVisualizarIntensidadePluviometrica(&menu, &conjuntoDados);

                // Opcao selecionada pelo usuario

                printf("Entre com a opção: ");

                menu.returnOpcao = scanf("%d", &menu.opcao);

                menuFim();

                // Opções do menu

                menu.opcaoMinMax[0] = 1;

                menu.opcaoMinMax[1] = 3;

                // Limpando o buffer de entrada!!!!

                while(getchar() != '\n');

            // Valida a entrada numerica antes de prosseguir

            } while (!validaOpcaoMenu(&menu));

            // Verifica qual opção de conjunto de dados foi escolhida

            if (menu.opcao >= 1 && menu.opcao <= 2) {

                    // Registros aleatórios - 1 / Registros inseridos pelo usuário - 2

                    menu.tipoDado = menu.opcao;

                    // Primeiro dígito (1) - "Visualizar os registros da intensidade de chuva"

                    // Segundo dígito (opcao) - Identifica se os dados são aleatórios ou são dados inseridos pelo usuário

                    menu.menu = menu.opcao + 10;

            } else {

                    // Opcao 3 - Voltar

                    // Volta para o menu principal

                    menu.menu = 0;

            }

        } else if (menu.menu >= 11 && menu.menu <= 12) {

            // Visualizar dados aleatórios ou visualizar dados inseridos pelo usuário

            // Menu 11 e 12 para as opções 1 e 2

            // Passos para chegar a este menu:

            // "Serviços disponíveis" -> "1 - Visualizar os registros da intensidade de chuva"

            // "Selecione um conjunto de registros para visualizar:" -> "1 - Registros aleatórios

            // ou

            // "Selecione um conjunto de registros para visualizar:" -> "2 - Registros inseridos pelo usuário

            // Loop que garante uma entrada numérica válida

            do {

                // Menu 11 "Utilizando dados aleatórios"

                //ou

                // Menu 12 "Utilizando dados inseridos pelo usuário"

                // O usuário deve escolher entre visualizar os dados leatórios ou visualizar dados registrados pelo usuário

                menuInicio();

                menuVisualizarIntensidadePluviometrica(&menu, &conjuntoDados);

                // Opcao selecionada pelo usuario

                printf("Entre com a opção: ");

                menu.returnOpcao = scanf("%d", &menu.opcao);

                menuFim();

                // Opções do menu

                menu.opcaoMinMax[0] = 1;

                menu.opcaoMinMax[1] = 1;

                // Limpando o buffer de entrada!!!!

                while(getchar() != '\n');

            // Valida a entrada numerica antes de prosseguir

            } while (!validaOpcaoMenu(&menu));

            // Verifica qual opção de conjunto de dados foi escolhida

            if (menu.opcao == 1) {

                    // Opcao 1 - Voltar

                    // Volta para o menu anterior

                    menu.menu = 1;

            }

        } else if (menu.menu == 2) {

            // Menu 2 - Selecionar uma categoria para editar o nome e inserir as

            // informações de intensidade de chuva

            // Menu para a opção 2

            // Passos para chegar a este menu:

            // "Serviços disponíveis" -> "2 - Registrar a intensidade pluviométrica"

            // "Selecione um bairro para inserir os dados:" -> "1 - Bairro 1" ao "10 - Bairro 10"

            // Loop que garante uma entrada numérica válida

            do {

                // Menu "Registrar a intensidade pluviométrica"

                // O usuário deve escolher um bairro para adicinar os dados de intensidade pluviométrica

                menuInicio();

                menu.bairro = 0;

                menuRegistrarIntensidadePluviometrica(&menu, &conjuntoDados);

                // Opção selecionada pelo usuario

                printf("Entre com a opção: ");

                menu.returnOpcao = scanf("%d", &menu.opcao);

                menuFim();

                // Opções do menu

                menu.opcaoMinMax[0] = 1;

                menu.opcaoMinMax[1] = MAX\_BAIRROS + 1;

                // Limpando o buffer de entrada!!!!

                while(getchar() != '\n');

            // Valida a entrada numerica antes de prosseguir

            } while (!validaOpcaoMenu(&menu));

            // Verifica qual opção de conjunto de dados foi escolhida

            if (menu.opcao >= 1 && menu.opcao <= MAX\_BAIRROS) {

                    // Bairro 1 - [1] / Bairro MAX\_BAIRROS - [MAX\_BAIRROS]

                    menu.bairro = menu.opcao;

                    // Primeiro dígito (2) - "Registrar a intensidade pluviométrica"

                    // Segundo dígito (opcao) - Número do bairro selecionado

                    menu.menu = menu.opcao + 200;

            } else {

                    // Opcao 11 - Voltar

                    // Volta para o menu principal

                    menu.menu = 0;

            }

        } else if (menu.menu >= 201 && menu.menu <=210){

            // Menu para editar o nome da categoria

            // Menus 201 a 210

            // Opções 1 a 10

            // Menu para as opções 1 a 10

            // Passos para chegar a este menu:

            // "Serviços disponíveis" -> "2 - Registrar a intensidade pluviométrica"

            // "Selecione um bairro para inserir os dados:" -> "1 - Bairro 1" ao "10 - Bairro 10"

            // Menu "Editar nome do bairro"

            menuInicio();

            menuRegistrarIntensidadePluviometrica(&menu, & conjuntoDados);

            printf("Digite um novo nome para o bairro/região ou tecle \"Enter\": ");

            // Nome do bairro digitado

            // A função fgets foi utilizada para obter strings com espaços.

            //menu.returnNovoCaracteres = fgets(menu.novoBairro, 21, stdin);

            // Verifica se fgets() leu todos os caracteres da entrada

            // Se retornar NULL significa que ainda restam caracteres a serem lidos

            if (fgetws(menu.novoBairro, 21, stdin) != NULL) {

                // Verifica se o carater de fim de linha foi lido

                //  retorno NULL da função strchr() significa que o caracter não esta presente no vetor

                if (wcschr(menu.novoBairro, '\n') == NULL ) {

                    // Termina de ler os caracters não lidos!

                    // Um valor igual a EOF significa que o fim do buffer foi alcançado.

                    char c;

                    while ((c = getchar()) != '\n' && c != EOF);

                }

            }

            menuFim();

            // Eliminando '\n' da string

            //menu.novoBairro[strcspn(menu.novoBairro, "\n")] = 0;

            size\_t len = wcslen(menu.novoBairro);

            if (len > 0 && menu.novoBairro[len - 1] == L'\n') menu.novoBairro[len - 1] = L'\0';

            // Verifica se a entrada de caracteres é válida

            // Verifica se foi digitado o nome de um novo bairro

            if (wcslen(menu.novoBairro) > 0) {

                // Renomeia o bairro

                wcscpy(conjuntoDados.bairros[menu.bairro-1], menu.novoBairro);

                // Imprime uma mensagem confirmando os dados modificados

                printf("\n -> Bairro/região renomeado(a)!\n");

                wprintf(L" [Novo nome para o bairro/região: %ls]\n", conjuntoDados.bairros[menu.bairro - 1]);

            }

            // Ir para o menu para inserir os dados de intensidade pluviométrica

            // Inicia a captura de dados pelo dia 1

            menu.indiceDia = 1;

            menu.menu = menu.menu \* 100 + menu.indiceDia;

        } else if (menu.menu >= 20101 && menu.menu <= 21016) {

            // Menu para inserir as informações de intensidade de chuva medidas em mm/h

            // Menus 201.01 a 210.16

            // Opções 1 a 15

            // Menu para os dias 1 a 15

            // Menu para a opção 16 -> menu com a opção voltar

            // Passos para chegar a este menu:

            // "Serviços disponíveis" -> "2 - Registrar a intensidade pluviométrica"

            // "Selecione um bairro para inserir os dados:" -> "1 - Bairro 1" ao "10 - Bairro 10"

            // "Inserir dados diários de intensidade de chuva" -> "Dia 1" a "Dia 15"

            do {

                // Menu "Inserir dados de intensidade pluviometrica"

                menuInicio();

                menuRegistrarIntensidadePluviometrica(&menu, & conjuntoDados);

                if (menu.indiceDia <= MAX\_DIAS) {

                    printf("Digite um novo valor em [mm/h] ou tecle \"Enter\" : ");

                    // Limpa a variável novoValor

                    strcpy(menu.novoValor, "");

                    // Define como valor padrão o valor atual

                    menu.novaIntensidadePluviometrica = conjuntoDados.medidaPluviometricaDiaBairro[menu.indiceDia-1][menu.bairro - 1];

                    // A função fgets foi utilizada para obter strings com espaços.

                    // Verifica se fgets() leu todos os caracteres da entrada

                    // Se retornar NULL significa que ainda restam caracteres a serem lidos

                    if (fgets(menu.novoValor, 7, stdin) != NULL) {

                        // Verifica se o carater de fim de linha foi lido

                        //  retorno NULL da função strchr() significa que o caracter não esta presente no vetor

                        if (strchr(menu.novoValor, '\n') == NULL ) {

                            // Termina de ler os caracters não lidos!

                            // Um valor igual a EOF significa que o fim do buffer foi alcançado.

                            char c;

                            while ((c = getchar()) != '\n' && c != EOF);

                        }

                    }

                } else {

                    // Opção selecionada pelo usuario

                    printf("Entre com a opção: ");

                    menu.returnOpcao = scanf("%d", &menu.opcao);

                    // Opções do menu

                    menu.opcaoMinMax[0] = 1;

                    menu.opcaoMinMax[1] = 1;

                }

                menuFim();

            // Valida a opcao do menu apenas se os valores de intensidade de chuva de todos os dias tiverem sido atualizados

            } while (

                (!validaOpcaoMenu(&menu) && (menu.indiceDia > MAX\_DIAS)) ||

                (!validaEntradaNumerica(&menu) && (menu.indiceDia <= MAX\_DIAS))

            );

            // if (menu.indiceDia <= MAX\_DIAS) {

            // }

            // Verifica o indice do dia

            if (menu.indiceDia >= 1 && menu.indiceDia <= MAX\_DIAS) {

                // // Define como valor padrão o valor atual

                // menu.novaIntensidadePluviometrica = conjuntoDados.medidaPluviometricaDiaBairro[menu.indiceDia-1][menu.bairro - 1];

                if (menu.returnNovoValor)

                    conjuntoDados.medidaPluviometricaDiaBairro[menu.indiceDia-1][menu.bairro - 1] = menu.novaIntensidadePluviometrica;

                // Aponta para o menu do próximo dia

                menu.indiceDia++;

                menu.menu = (menu.bairro + 200) \* 100 + menu.indiceDia;

            } else {

                // Opcao voltar

                // Menu de seleção de bairros

                menu.menu = 2;

            }

        } else {

            // Menu principal

            menu.menu = 0;

        }

    // Sair - sairMenu igual a 1 encerra o sistema!

    } while (!menu.sairMenu);

    mensagemEncerramento();

    return 0;

}

void apresentacaoSistema() {

    // Apresnetação do sistema

    printf("\n");

    printf("####################################################################\n");

    printf("###### Sistema de Registro de Intensidade Pluviométrica v1.0 #######\n");

    printf("####################################################################\n");

    printf("\* Sobre:\n");

    printf("-> Desenvolvido por Fabio Toledo Bonemer De Salvi\n");

    printf("\n");

    printf("\* Visualize os registros de intensidade pluviométrica!\n");

    printf("-> Os registros são apresentados em um gráfico comparando os\n");

    printf("  bairros, ou regiões, ao longo dos dias!\n");

    printf("-> As intensidade pluviométricas são apresentadadas por meio de um\n");

    printf("  gráfico de intensidade de cores no estilo \"Heatmap\"!\n");

    printf("-> Os menores valores são representados pela cor azul, os valores\n");

    printf("  médios pela cor laranja e os valores maiores pela cor vermelha!\n");

    printf("\n");

    printf("\* Registre os dados da intensidade pluviométrica!\n");

    printf("-> Entre com o nome do bairro e com os valores da intensidade\n");

    printf("  pluviométrica!\n");

    printf("-> Limitado a 10 bairros e a 15 dias de dados por bairro!\n");

    printf("####################################################################\n");

}

void menuInicio() {

    printf("\n");

    printf("####################################################################\n");

    printf("###### Sistema de Registro de Intensidade Pluviométrica v1.0 #######\n");

    printf("####################################################################\n");

}

void menuPrincipal() {

    // Menu 0 - Menu principal

    // Menu inicial

    printf("\n");

    printf(">> Serviços disponíveis:\n");

    printf("1 - Visualizar os registros de intensidade pluviométrica\n");

    printf("2 - Registrar uma intensidade pluviométrica\n");

    printf("3 - Sair\n");

}

void menuVisualizarIntensidadePluviometrica(struct Menu \*menu, struct ConjuntoDados \*conjuntoDados) {

    switch(menu->menu) {

        case 11:

            // Menu 11 - Menu Visualizar os registros da intensidade pluviométrica com dados aleatórios

            printf("======= Visualizar os registros da intensidade pluviométrica =======\n\n");

            printf(">> Gráfico \"Heatmap\" para os dados aleatórios:\n\n");

            // Inicializar dados aleatórios de intensidade pluviométrica na matriz

            inicializaMatriz(

                conjuntoDados->medidaPluviometricaDiaBairroAleatorio,

                &conjuntoDados->limitesAleatorio);

            // Imprimir os valores aleatrios da matriz da intensidade pluviométrica no estilo Heatmap

            imprimirHeatmap(

                conjuntoDados->bairrosPadrao,

                conjuntoDados->medidaPluviometricaDiaBairroAleatorio,

                &conjuntoDados->limitesAleatorio,

                &conjuntoDados->coresHeatmap

            );

            printf("\n1 - Voltar\n");

            break;

        case 12:

            // Menu 12 - Menu Visualizar os registros da intensidade pluviométrica com dados inseridos pelo usuário

            printf("======= Visualizar os registros da intensidade pluviométrica =======\n\n");

            printf(">> Gráfico \"Heatmap\" para os dados inseridos pelo usuário:\n\n");

            // Imprimir os valores inseridos pelo usuário na matriz de intensidade pluviométrica no estilo Heatmap

            imprimirHeatmap(

                conjuntoDados->bairros,

                conjuntoDados->medidaPluviometricaDiaBairro,

                &conjuntoDados->limites,

                &conjuntoDados->coresHeatmap

            );

            printf("\n1 - Voltar\n");

            break;

        case 1:

        default:

            // Menu 1 - Menu Visualizar os registros da intensidade pluviométrica

            printf("======= Visualizar os registros da intensidade pluviométrica =======\n\n");

            printf(">> Selecione um conjunto de registros para visualizar:\n");

            printf("1 - Registros aleatórios\n");

            printf("2 - Registros inseridos pelo usuário\n");

            printf("3 - Voltar\n");

            break;

    }

}

void menuRegistrarIntensidadePluviometrica(struct Menu \*menu, struct ConjuntoDados \*conjuntoDados) {

    if(menu->menu >= 201 && menu->menu <= 210) {

        // Menu 201 a 210 - Menu

        printf("=============== Registrar a intensidade pluviométrica ==============\n\n");

        // Imprime o bairro selecinado

        wprintf(L">> Bairro/região selecionado(a): %ls\n", conjuntoDados->bairros[menu->bairro-1]);

    }else if(menu->menu >= 20101 && menu->menu <= 21016) {

        // Menu 20101 a 21015 - Menu

        printf("=============== Registrar a intensidade pluviométrica ==============\n\n");

        // Imprime o bairro selecinado

        wprintf(L">> Bairro/região selecionado(a): %ls\n", conjuntoDados->bairros[menu->bairro-1]);

        printf(">> Registros de intensidade pluviométrica:\n") ;

        // Imprime os dias

        printf("  %-5s", "Dia");

        for(int i = 0; i < MAX\_DIAS; i++)

            printf("| %4d |", i + 1);

        printf("\n");

        // Imprime os dados de intensidade pluviométrica

        printf("  %-5s", "mm/h");

        for(int i = 0; i < MAX\_DIAS; i++)

            printf("|%6.1f|", conjuntoDados->medidaPluviometricaDiaBairro[i][menu->bairro-1]);

        printf("\n\n");

        // Verifica se o indiceDia é menor que o numero total de dias

        if (menu->indiceDia <= MAX\_DIAS) {

            printf(">> Dia: %d de %d.\n>> Intensidade pluviométrica: %6.1f [mm/h]\n",

                menu->indiceDia,

                MAX\_DIAS,

                conjuntoDados->medidaPluviometricaDiaBairro[menu->indiceDia - 1][menu->bairro-1]

            );

        } else {

            printf(">> Selecione uma opção:\n");

            printf("1 - Voltar\n");

        }

    } else {

        // Menu 2 - Menu Registrar a intensidade pluviométrica

        printf("=============== Registrar a intensidade pluviométrica ==============\n\n");

        printf(">> Selecione um bairro/região para inserir os dados:\n");

        for (int indiceBairro = 0; indiceBairro < MAX\_BAIRROS; indiceBairro++) {

            wprintf(L"%2d - %ls\n", indiceBairro + 1, conjuntoDados->bairros[indiceBairro]);

        }

        printf("11 - Voltar\n");

    }

}

void menuFim() {

    printf("####################################################################\n");

}

void mensagemEncerramento() {

    // Mensagem de encerramento

    printf("\n");

    printf("####################################################################\n");

    printf("###### Sistema de Registro de Intensidade Pluviometrica  v1.0 ######\n");

    printf("####################################################################\n");

    printf("Obrigado por utilizar nosso sistema!\n\n");

    printf("Volte sempre!\n");

    printf("####################################################################\n");

}

// Define as cores de transição do Heatmap

void defineCoresHeatmap(struct CoresHeatmap \*coresHeatmap) {

    // Definindo as cores de transição do HeatMap

    // light blue

    coresHeatmap->rgbBaixo[0] = 173;

    coresHeatmap->rgbBaixo[1] = 216;

    coresHeatmap->rgbBaixo[2] = 230;

    // orange

    coresHeatmap->rgbMedio[0] = 255;

    coresHeatmap->rgbMedio[1] = 165;

    coresHeatmap->rgbMedio[2] = 0;

    // yellow

    // coresHeatmap->rgbMedio[0] = 255;

    // coresHeatmap->rgbMedio[1] = 255;

    // coresHeatmap->rgbMedio[2] = 0;

    // red

    coresHeatmap->rgbAlto[0] = 255;

    coresHeatmap->rgbAlto[1] = 0;

    coresHeatmap->rgbAlto[2] = 0;

}

// Inicializa a matriz com valores aleatório de intensidade de chuva

void inicializaMatriz(float medidaPluviometricaDiaBairro[15][10], struct Limites \*limites) {

    // Inicia matriz com valores randomicos

    srand(time(NULL));

    // Define a variável que receberá os valores aleatórios dentro do laço de repetição

    float random\_in\_range;

    // Define o valor máximo com um valor mínimo

    limites->valorMax = 0.0;

    // Define o valor mínimo com um valór máximo

    limites->valorMin = FLT\_MAX;

    //printf("{\n");

    for (int j = 0; j < MAX\_BAIRROS; j++) {

        //printf("{ ");

        for (int i = 0; i < MAX\_DIAS; i++){

            // Generate a random number between 1 and 100

            random\_in\_range =  (float) ((rand() % 1000) + 1) / 10.0;

            medidaPluviometricaDiaBairro[i][j] = random\_in\_range;

            if (random\_in\_range > limites->valorMax)

                limites->valorMax = random\_in\_range;

            if (random\_in\_range < limites->valorMin)

                limites->valorMin = random\_in\_range;

            //printf("%3d%s ", medidaPluviometricaDiaBairro[i][j], (i == MAX\_DIAS - 1) ? "" : ",");

        }

        //printf("}%s\n", (j == MAX\_BAIRROS - 1) ? "" : ",");

    }

    //printf("}\n");

}

void defineLimites(float medidaPluviometricaDiaBairro[15][10], struct Limites \*limites) {

    // Inicia matriz com valores randomicos

    srand(time(NULL));

    // Define a variável que receberá os valores aleatórios dentro do laço de repetição

    float medidaIntensidadePluviometrica;

    // Define o valor máximo com um valor mínimo

    limites->valorMax = 0.0;

    // Define o valor mínimo com um valór máximo

    limites->valorMin = FLT\_MAX;

    //printf("{\n");

    for (int j = 0; j < MAX\_BAIRROS; j++) {

        //printf("{ ");

        for (int i = 0; i < MAX\_DIAS; i++){

            medidaIntensidadePluviometrica = medidaPluviometricaDiaBairro[i][j];

            if (medidaIntensidadePluviometrica > limites->valorMax)

                limites->valorMax = medidaIntensidadePluviometrica;

            if (medidaIntensidadePluviometrica < limites->valorMin)

                limites->valorMin = medidaIntensidadePluviometrica;

            //printf("%3d%s ", medidaPluviometricaDiaBairro[i][j], (i == MAX\_DIAS - 1) ? "" : ",");

        }

        //printf("}%s\n", (j == MAX\_BAIRROS - 1) ? "" : ",");

    }

    //printf("}\n");

}

void imprimirHeatmap(wchar\_t bairros[10][21], float medidaPluviometricaDiaBairro[15][10], struct Limites \*limites, struct CoresHeatmap \*coresHeatmap) {

    // Cor final do Heatmap

    int rgbItem[3];

    int rgbTexto[3] = {0, 0, 0};

    // Para os dados normalizados calculados dentro do laço

    float norm\_val = 0.0;

    // Limite inferior normalizado

    float norm\_val\_lower;

    // Limite superior normalizado

    float norm\_val\_upper;

    // Determina os limites dos valores de intensidade de chuva

    defineLimites(medidaPluviometricaDiaBairro, limites);

    // Imprime os valores do gráfico Heatmap

    // Laço que percorre os bairros

    for (int i = 0; i < MAX\_BAIRROS; i++){

        // Imprime o nome do bairo na referida linha de dados

        wprintf(L"%-20ls-", bairros[i]);

        // Laço que percorre os dias

        for (int j = 0; j < MAX\_DIAS; j++) {

            // Imprime dados utilizando a escala de duas cores

            // rgbItem[0] = (int) coresHeatmap->rgbBaixo[0] + (coresHeatmap->rgbAlto[0] - coresHeatmap->rgbBaixo[0]) \* norm\_val;

            // rgbItem[1] = (int) coresHeatmap->rgbBaixo[1] + (coresHeatmap->rgbAlto[1] - coresHeatmap->rgbBaixo[1]) \* norm\_val;

            // rgbItem[2] = (int) coresHeatmap->rgbBaixo[2] + (coresHeatmap->rgbAlto[2] - coresHeatmap->rgbBaixo[2]) \* norm\_val;

            // Verifica a diferença entre o limite maximo e  limite mínimo

            if (limites->valorMax - limites->valorMin) {

                // Fórmula básica para normalizar os dados entre o menor e  maior

                norm\_val = (float) (medidaPluviometricaDiaBairro[j][i] - limites->valorMin) / (limites->valorMax - limites->valorMin);

                // Garante que os valores normalizados estejam entre 0.0 e 1.0

                if (norm\_val < 0)

                    norm\_val = 0;

                else if (norm\_val > 1)

                    norm\_val = 1;

            } else {

                norm\_val = 0.0;

            }

            // Imprime os dados considerando uma transição entre três cores

            // Seleciona entre a metade inferior ou supeior para realizar a transição de cores

            // A cor supeior equivale ao valor 1

            // A cor central equivale ao valor 0.5

            // A cor inferior equivale a valor 0

            if (norm\_val <= 0.5) {

                // Transição entre as cores rgbBaixo e rgbMedio

                // Renormaliza o valor para estar entre 0.0 e 1.0 com um décimo de presição

                norm\_val\_lower = round(norm\_val / 0.5 \* 10) / 10;

                rgbItem[0] = (int) coresHeatmap->rgbBaixo[0] + (coresHeatmap->rgbMedio[0] - coresHeatmap->rgbBaixo[0]) \* norm\_val\_lower;

                rgbItem[1] = (int) coresHeatmap->rgbBaixo[1] + (coresHeatmap->rgbMedio[1] - coresHeatmap->rgbBaixo[1]) \* norm\_val\_lower;

                rgbItem[2] = (int) coresHeatmap->rgbBaixo[2] + (coresHeatmap->rgbMedio[2] - coresHeatmap->rgbBaixo[2]) \* norm\_val\_lower;

                // Imprime os valores de intensidade de chuva que correspondem às cores da metade inferior

                printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm%6.1f \e[0m",

                    rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], medidaPluviometricaDiaBairro[j][i]);

                // Imprime os valores normalizados que correspondem às cores da metade inferior

                //  printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm%6.2f \e[0m",

                //     rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], norm\_val\_lower);

            } else {

                // Transição entre as cores rgbMedio e rgbAlta

                // Renormaliza o valor para estar entre 0.0 e 1.0 com um décimo de presição

                norm\_val\_upper = round((norm\_val - 0.5) \* 10) / (0.5 \* 10);

                rgbItem[0] = (int) coresHeatmap->rgbMedio[0] + (coresHeatmap->rgbAlto[0] - coresHeatmap->rgbMedio[0]) \* norm\_val\_upper;

                rgbItem[1] = (int) coresHeatmap->rgbMedio[1] + (coresHeatmap->rgbAlto[1] - coresHeatmap->rgbMedio[1]) \* norm\_val\_upper;

                rgbItem[2] = (int) coresHeatmap->rgbMedio[2] + (coresHeatmap->rgbAlto[2] - coresHeatmap->rgbMedio[2]) \* norm\_val\_upper;

                // Imprime os valores de intensidade de chuva que correspondem às cores da metade superior

                printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm%6.1f \e[0m",

                    rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], medidaPluviometricaDiaBairro[j][i]);

                // Imprime os valores normalizados que correspondem às cores da metade superior

                // printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm%6.2f \e[0m",

                //     rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], norm\_val\_upper);

            }

            //printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm (%d, %d, %d) \e[0m", rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2]);

            //printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm%4d\e[0m", rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], "");

        }

        printf("\n");

    }

    // Imprime a linha referente aos dias

    printf("%-20s-", "Dia      ");

    for(int i = 0; i < MAX\_DIAS; i++)

        printf("| %3d |", i + 1);

    printf("\n\n");

    // Imprime a barra de cores e os limites da escala

    // Imprime o valor inferior da escala de cores

    printf(" %.1f [mm/h] ", limites->valorMin);

    // printf(" %.1f [mm/m\u00B2] ", limites->valorMin);

    for (float norm\_val = 0.0; norm\_val < 1.0; norm\_val += 0.05) {

        // Imprime os dados para a transição entre duas cores

        // rgbItem[0] = (int) coresHeatmap->rgbBaixo[0] + (coresHeatmap->rgbAlto[0] - coresHeatmap->rgbBaixo[0]) \* norm\_val;

        // rgbItem[1] = (int) coresHeatmap->rgbBaixo[1] + (coresHeatmap->rgbAlto[1] - coresHeatmap->rgbBaixo[1]) \* norm\_val;

        // rgbItem[2] = (int) coresHeatmap->rgbBaixo[2] + (coresHeatmap->rgbAlto[2] - coresHeatmap->rgbBaixo[2]) \* norm\_val;

        // Imprime os dados considerando uma transição entre três cores

        // Seleciona entre a metade inferior ou supeior para realizar a transição de cores

        // A cor supeior equivale ao valor 1

        // A cor central equivale ao valor 0.5

        // A cor inferior equivale a valor 0

        if (norm\_val <= 0.5 || !(limites->valorMax - limites->valorMin)) {

            // Transição entre as cores rgbBaixo e rgbMedio

            // Verifica a deferença entre o limite máximo é mair do que limite mínimo

            if (limites->valorMax - limites->valorMin) {

                // Renormaliza o valor para estar entre 0.0 e 1.0 com um décimo de presição

                norm\_val\_lower = round(norm\_val \* 10) / (0.5 \* 10);

            } else {

                norm\_val\_lower = 0.0;

            }

            rgbItem[0] = (int) coresHeatmap->rgbBaixo[0] + (coresHeatmap->rgbMedio[0] - coresHeatmap->rgbBaixo[0]) \* norm\_val\_lower;

            rgbItem[1] = (int) coresHeatmap->rgbBaixo[1] + (coresHeatmap->rgbMedio[1] - coresHeatmap->rgbBaixo[1]) \* norm\_val\_lower;

            rgbItem[2] = (int) coresHeatmap->rgbBaixo[2] + (coresHeatmap->rgbMedio[2] - coresHeatmap->rgbBaixo[2]) \* norm\_val\_lower;

            // Imprime os valores normalizados que correspondem às cores da metade inferior

            //printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm% 4.2f \e[0m", rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], norm\_val\_lower);

        } else {

            // Transição entre as cores rgbMedio e rgbAlta

            // Verifica a deferença entre o limite máximo é mair do que limite mínimo

            if (limites->valorMax - limites->valorMin) {

                // Renormaliza o valor para estar entre 0.0 e 1.0 com um décimo de presição

                norm\_val\_upper = round((norm\_val - 0.5)\*10) / (0.5 \* 10);

            } else {

                norm\_val\_lower = 0.0;

            }

            rgbItem[0] = (int) coresHeatmap->rgbMedio[0] + (coresHeatmap->rgbAlto[0] - coresHeatmap->rgbMedio[0]) \* norm\_val\_upper;

            rgbItem[1] = (int) coresHeatmap->rgbMedio[1] + (coresHeatmap->rgbAlto[1] - coresHeatmap->rgbMedio[1]) \* norm\_val\_upper;

            rgbItem[2] = (int) coresHeatmap->rgbMedio[2] + (coresHeatmap->rgbAlto[2] - coresHeatmap->rgbMedio[2]) \* norm\_val\_upper;

            // Imprime os valores normalizados que correspondem às cores da metade superior

            //printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm% 4.2f \e[0m", rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], norm\_val\_upper);

        }

        // Imprime a cor RGB

        //printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm (%d, %d, %d) \e[0m", rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], rgbItem[0], rgbItem[1], rgbItem[2]);

        // Imprime espaços em branco com cor de fundo

        printf("\e[0m\e[38;2;%d;%d;%dm\e[48;2;%d;%d;%dm%4s\e[0m", rgbTexto[0], rgbTexto[1], rgbTexto[2], rgbItem[0], rgbItem[1], rgbItem[2], "");

    }

    // Imprime  valor superior da escala de cores

    printf(" %.1f [mm/h]\n", limites->valorMax);

    // printf(" %.1f [mm/m\u00B2]\n", limites->valorMax);

}

int validaOpcaoMenu(struct Menu \*menu) {

    // Verifica se apenas um valor inteiro foi digitado

    if (menu->returnOpcao != 1) {

        printf("\nValor inválido!\n");

        return 0; // Entrada inválida

    }

    // Verifica se a opção digitada esta disponível no menu

    if (menu->opcao < menu->opcaoMinMax[0] || menu->opcao > menu->opcaoMinMax[1]) {

        printf("\nOpção inválida!\n");

        return 0; // Opção inválida

    }

    return 1; // Entrada numérica Ok

}

int validaEntradaNumerica(struct Menu \*menu) {

    if (menu->menu >= 20101 && menu->menu <= 21015) {

        // Extrai o valor de intensidade pluviometrica da variável novoValor

        // e armazena na variável novaIntensidadePluviometrica

        menu->returnNovoValor = sscanf(

            menu->novoValor,

            "%f", &menu->novaIntensidadePluviometrica

        );

        if (menu->returnNovoValor) {

            if (menu->novaIntensidadePluviometrica < 0.0 || menu->novaIntensidadePluviometrica > 10000) {

                printf("\nValor %6.1f inválido! Digite um valor entre 0,0 e 9999,9!\n", menu->novaIntensidadePluviometrica);

                return 0; // Entrada inválida

            }

        } else {

            printf("\nUtilizar valor padrão!\n");

            return 2; // Utilizar Valor Padrão

        }

    }  else {

        // Verifica se apenas um valor inteiro foi digitado

        if (menu->returnNovoValor != 1) {

            printf("\nValor inválido!\n");

            return 0; // Entrada inválida

        }

    }

    return 1; // Entrada numérica Ok

}

int validaEntradaCaracters(struct Menu \*menu) {

    // Verifica se huve problemas ao ler a entrada de caracteres

    if (menu->returnNovoCaracteres == NULL) {

        printf("\nEntrada inválida!\n");

        return 0; // Entrada inválida

    }

    return 1; // Entrada caracteres Ok

}

// run into prompt:

// $ prompt $G

// $ exemplo{number}.exe

// $ echo %errorlevel%

# Teste de Execução

O software desenvolvido foi compilado e executado em prompt de comandos utilizando o software *VS Code* ([MICROSOFT CORPORATION, 2025](https://code.visualstudio.com/)).

A execução do software, com as respectivas entradas de dados e mensagens mostradas ao usuário, estão dispostas nos tópicos subsequentes.

As opções do menu foram escolhidas digitando o número da opção desejada.

As informações da intensidade pluviométrica foram adicionadas acessando as respectivas opções para o registro de uma nova informação de intensidade pluviométrica.

Por definição a matriz de intensidade pluviométrica é pré-definida com nomes de bairros e valores de intensidades pluviométricas nulos.

A seleção da opção de registro de intensidade pluviométrica aponta para um menu de opções para a escolha do bairro onde serão inseridas as informações de intensidade pluviométrica. O bairro poderá ser renomeado somente nesta etapa.

A conclusão da definição do bairro aponta para a inserção dos dados de intensidade pluviométrica, que é realizado de forma sequencial. Após a inclusão do último valor é apresentado um menu com a opção de retorno ao menu de seleção dos bairros.

É possível editar as informações acessando novamente o bairro onde serão realizadas. O processo de inclusão de dados permite manter o valor anterior caso a entrada atual seja nula ou seja um valor em branco. Este recurso permite editar um determinado campo de forma eficiente.

Após o registro das informações pluviométricas é possível realizar sua visualização em gráfico de intensidade de cores, Heatmap, voltando ao menu principal e acessando a opção de visualizar as informações, seguido da seleção da opção de visualizar dados registrados. Um gráfico de intensidade de cores será apresentado com as informações registradas e indicando os valores máximo e mínimo em uma escala de cores na parte inferior do gráfico.

É possível visualizar dados simulados. Esta opção não interfere nos dados registrados anteriormente.

Os próximos tópicos ilustram estas etapas.

## Apresentação do sistema

Ao iniciar a execução do software a mensagem inicial da Figura 1 é apresentada no prompt de comandos.

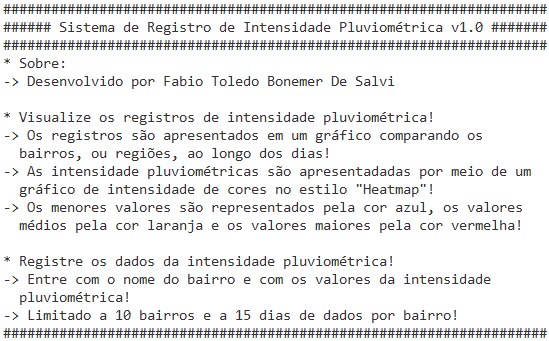


Figura - Tela de apresentação do software.

## Menu dos serviços disponíveis

O menu principal com os serviços disponíveis foi apresentado após a impressão da mensagem inicial de apresentação do sistema, mostrando as opções referentes a “**Visualizar os registros de intensidade pluviométrica**”, “**Registrar uma intensidade pluviométrica**” e a “**Sair**” do programa.

Para continuar foi preciso digitar o valor numérico referente à opção desejada.

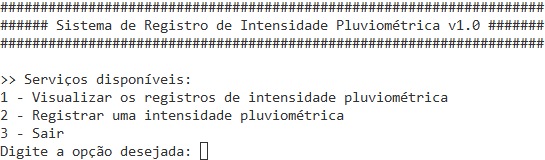


Figura - Menu dos serviços disponíveis.

## Visualização dos registros de intensidade pluviométrica

O acesso inicial foi realizado escolhendo a opção referente a visualizar os registros de intensidade pluviométrica com o objetivo de verificar o valor nulo desses registros nesta fase inicial de execução do software e para visualizar um conjunto de dados aleatórios de intensidade pluviométrica.

Para essa finalidade foi escolhida a opção “**1 – Visualizar os registros de intensidade pluviométrica**”, no menu de “**Serviços disponíveis**” (Figura 2), dando acesso ao menu “**Selecione um conjunto de registros para visualizar:**” (Figura 3) para a escolha entre as opções **“1 - Registros aleatórios”** ou **“2 - Registros inseridos pelo usuário”**:

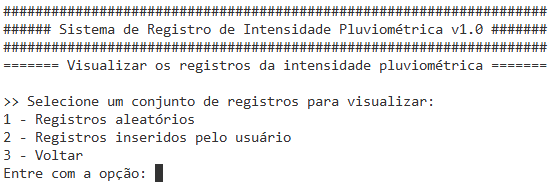


Figura - Menu de escolha do conjunto de registros.

A escolha da opção “**1 – Registros aleatórios**” resultou na apresentação de um gráfico de intensidade de cores, no estilo *Heatmap*, com dados aleatórios da intensidade pluviométrica, servindo apenas como exemplo (Figura 4).

Após a visualização do gráfico de registros aleatórios a opção “**1 – Voltar**” foi escolhida para retornar ao menu “**Selecione um conjunto de registros para visualizar:**” (Figura 3), onde foi escolhida a opção “**2 – Registros inseridos pelo usuário**” que resultou na apresentação do um novo gráfico de intensidades de cores, no estilo “*Heatmap*”, contendo os dados de intensidade pluviométrica inseridos pelo usuário, que nesta etapa da execução foram dados nulos (Figura 5).

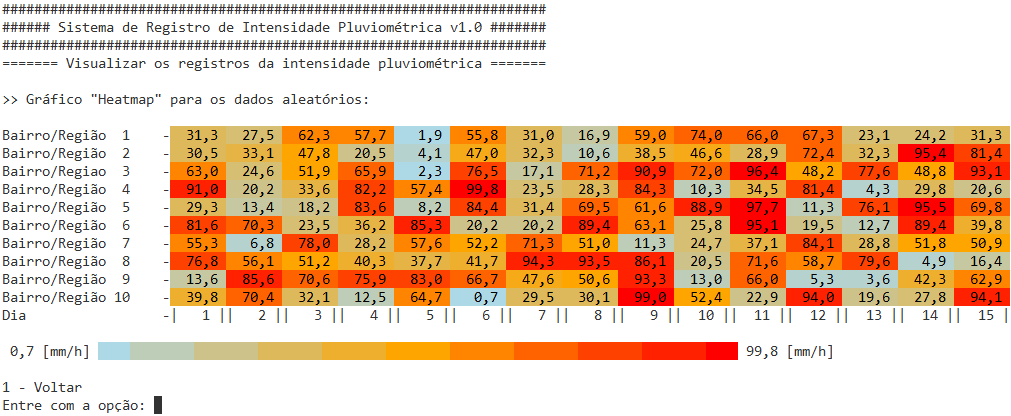


Figura – Gráfico de intensidade de cores, no estilo Heatmap, para um conjunto de dados aleatórios. Destaque para os valores máximo e mínimo que determinam o início e o fim da transição das cores.

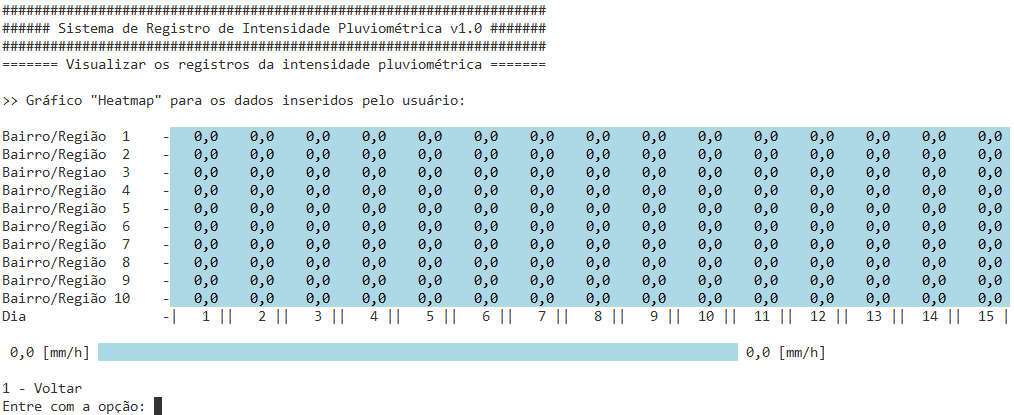


Figura - Gráfico de intensidade de cores, no estilo Heatmap, para um conjunto de dados, inicialmente com valores nulos, inserido pelo usuário. Destaque para os valores máximo e mínimo que determinam o início e o fim da transição das cores.

O retorno ao menu principal contendo os “**Serviços disponíveis**” foi realizado digitando os valores correspondentes às opções “**1 – Voltar**” neste menu e “**3 – Voltar”** no menu subsequente.

## Registro de intensidades pluviométricas

A inclusão das intensidades pluviométricas foi realizada selecionando a opção “**2 – Registrar uma intensidade pluviométrica**”, no menu principal dos “**Serviços disponíveis”** (Figura 6).

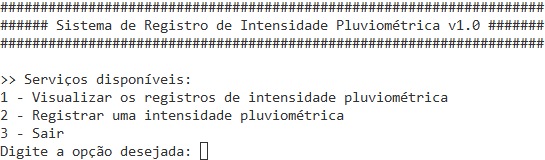


Figura - Menu dos serviços disponíveis.

O primeiro registro de intensidade pluviométrica foi realizado selecionando a opção “**1 – Bairro/ Região 1”** no menu “**Selecione um bairro/região para inserir os dados:**” (Figura 7).

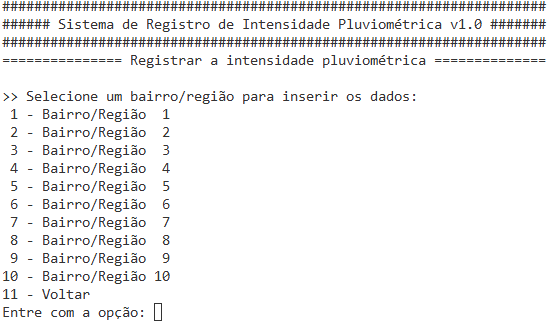


Figura 7 - Menu de escolha do bairro, ou região, para a inserção dos dados de intensidade pluviométricas.

Na sequência foi apresentado o menu “**Bairro selecionado: Bairro/Região 1**” onde foi digitado “**Centro**” (Figura 8), seguindo a ordem dos dados presentes na Tabela 1.

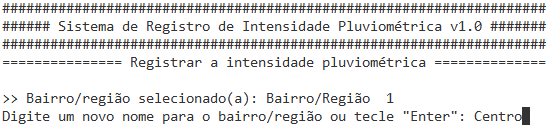


Figura - Menu para renomear a categoria, ou bairro. Tecle “Enter” para manter o nome original.

Uma mensagem foi impressa no “*prompt”*, parte superior da Figura 9, confirmando que o Bairro/região foi renomeado(a).

E na sequência tem-se o menu atualizado “**Bairro/região selecionado(a): Centro”** com os dados iniciais nulos de intensidade pluviométrica (Figura 9).

O registro das intensidades pluviométricas foi realizado digitando os valores numéricos presentes na Tabela 1, selecionando o bairro/região e iniciando a inserção sequencial dos dados pelo “**Dia 1**” e finalizando no “**Dia 15**”. Nesta parte foi possível manter um valor atual da intensidade pluviométrica apenas teclando “Enter”. Os valores decimais foram inseridos utilizando “**,**” no lugar de “**.**” devido a configuração “setlocale(LC\_ALL, "Portuguese\_Brazil.UTF-8")” utilizada para modificar as configurações de caracteres e números para os padrões brasileiros, imprimindo caracteres acentuados e utilizando os padrões de separação decimal com vírgula “**,**”. Para o “**Dia 1”** foi inserido o valor **12 mm/h** (Figura 9).

Tabela 1 - Valores de intensidade pluviométrica utilizados durante a execução do programa.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bairro/Região** | **Dias** (valores de intensidade pluviométrica em mm/h) | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| Centro | 12,0 | 0,0 | 28,5 | 4,0 | 0,0 | 10,0 | 45,0 | 18,0 | 2,5 | 0,0 | 22,0 | 11,0 | 0,8 | 20,0 | 6,0 |
| Jardim das Flores | 5,0 | 0,0 | 18,0 | 2,5 | 0,0 | 6,0 | 38,0 | 12,0 | 1,0 | 0,0 | 15,0 | 7,5 | 0,0 | 14,0 | 3,0 |
| Universitário | 15,0 | 1,5 | 30,2 | 6,0 | 0,0 | 12,0 | 50,0 | 20,0 | 3,0 | 0,0 | 25,0 | 13,0 | 1,0 | 22,0 | 8,0 |
| Industrial | 8,0 | 0,0 | 22,0 | 3,0 | 0,0 | 8,5 | 42,0 | 14,0 | 0,5 | 0,0 | 18,0 | 9,0 | 0,0 | 16,0 | 4,0 |
| Vila Nova | 2,0 | 0,0 | 10,5 | 1,0 | 0,0 | 4,0 | 30,0 | 8,0 | 0,0 | 0,0 | 9,0 | 4,0 | 0,0 | 7,0 | 1,0 |
| Marítimo | 0,0 | 0,0 | 5,0 | 0,0 | 0,0 | 1,0 | 25,0 | 6,0 | 0,0 | 0,0 | 7,0 | 2,0 | 0,0 | 5,0 | 0,0 |
| Boa Vista | 7,0 | 0,0 | 15,0 | 2,0 | 0,0 | 5,5 | 34,0 | 10,0 | 0,5 | 0,0 | 13,0 | 6,0 | 0,0 | 12,0 | 2,0 |
| Jardim Sul | 3,0 | 0,0 | 9,5 | 0,5 | 0,0 | 3,0 | 28,0 | 7,0 | 0,0 | 0,0 | 10,0 | 4,0 | 0,0 | 8,0 | 1,0 |
| Parque Oeste | 0,0 | 0,0 | 7,0 | 0,0 | 0,0 | 2,0 | 20,0 | 5,0 | 0,0 | 0,0 | 6,0 | 1,5 | 0,0 | 4,0 | 0,0 |
| Lagoa | 4,0 | 0,0 | 12,0 | 1,0 | 0,0 | 4,5 | 32,0 | 9,0 | 0,0 | 0,0 | 11,0 | 3,5 | 0,0 | 9,0 | 2,0 |

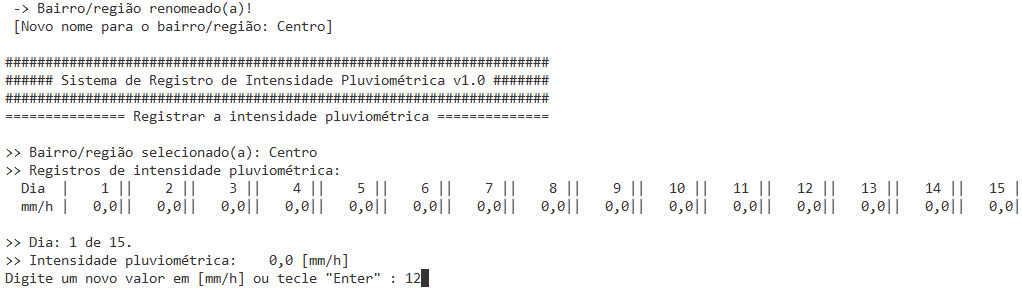


Figura - Menu para incluir a intensidade pluviométrica para o bairro/região Centro. Os dados anteriores são apresentados e atualizados durante a inserção dos dados. O dado 12 mm/h foi inserido ao dia 1.

O menu de inserção dos dados de intensidade pluviométrica foi atualizado após a inserção da intensidade pluviométrica do “**Dia 1**”, mostrando o novo registro impresso no “*prompt*” e solicitando a inserção da intensidade pluviométrica para o “**Dia 2**”, onde foi teclado “Enter” (Figura 10).

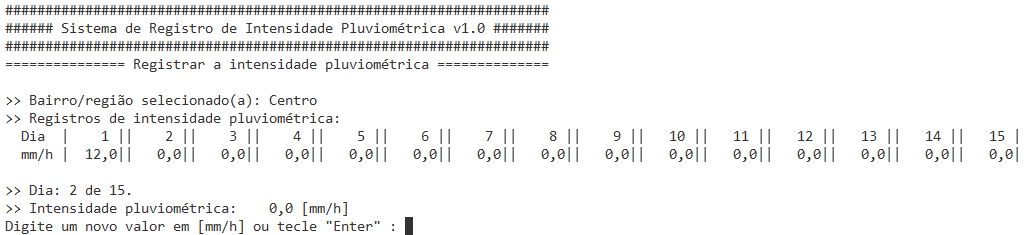


Figura – Menu atualizado com a intensidade pluviométrica do dia 1 e solicitação de inserção da nova intensidade pluviométrica para o dia 2. O dado 0 mm/h foi inserido ao dia 2 teclando “Enter”.

Novamente o menu foi atualizado após a inclusão da intensidade pluviométrica do “**Dia 2**”, mostrando o novo registro impresso em “*prompt*” e solicitando a inserção da intensidade pluviométrica para o “**Dia 3**”, onde foi inserido o valor **28,5 mm/h** (Figura 11).

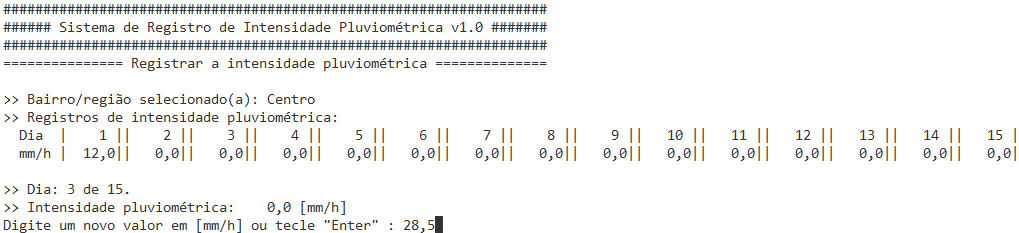


Figura - Menu atualizado com a intensidade pluviométrica do dia 2 e solicitação de inserção da nova intensidade pluviométrica para o dia 3. O dado 28,5 mm/h foi inserido ao dia 3.

O menu atualizado da Figura 12 foi apresentado após a inserção de todos os valores referentes ao bairro/região Centro.

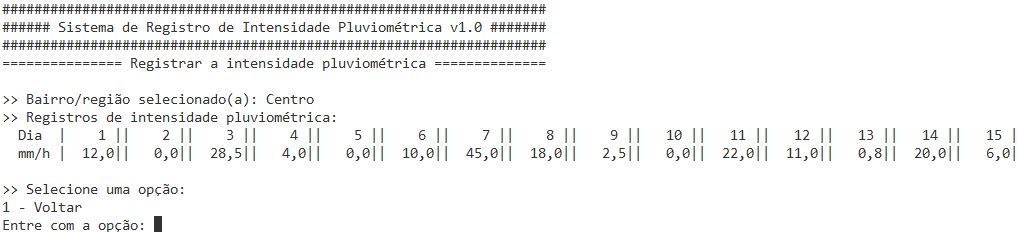


Figura - Bairro/região Centro atualizado com os valores de intensidade pluviométrica.

Uma visualização dos dados inseridos foi realizada voltando ao menu de “**Serviços disponíveis**”, através das respectivas opções “**Voltar**”, com posterior seleção das respectivas opções “**1 – Visualizar os registros de intensidade pluviométrica**” e “**2 – Registros inseridos pelo usuário**” (Figura 13).

Foi realizada a volta ao menu de “**Serviços disponíveis**” através das respectivas opções “**Voltar**” após visualizar o gráfico com os dados inseridos.

No menu de “**Serviços disponíveis**”foi selecionada a opção “**2 - Registrar uma intensidade pluviométrica**”, sendo impresso o menu atualizado “**Selecione um bairro/região para inserir os dados**” (Figura 14).

Os dados restantes da Tabela 1 foram inseridos seguindo os passos descritos anteriormente.

Os valores de intensidade pluviométrica inseridos devem estar na faixa de valores que inicia em **0.0 mm/h** e vai até **9999.9 mm/h.** Para manter o valor anterior é necessário deixar o campo vazio e teclar “Enter”.

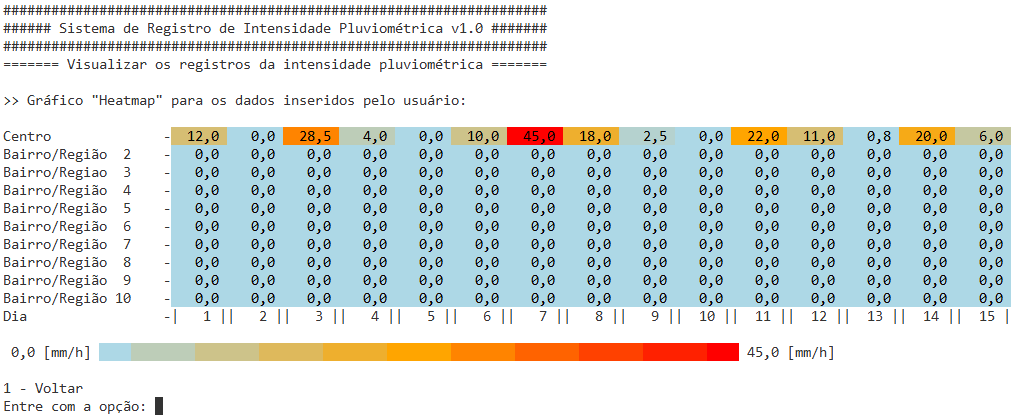


Figura - Gráfico de intensidade de cores, no estilo Heatmap, para um conjunto de dados atualizados. Destaque para os valores máximo e mínimo que determinam o início e o fim da transição das cores.

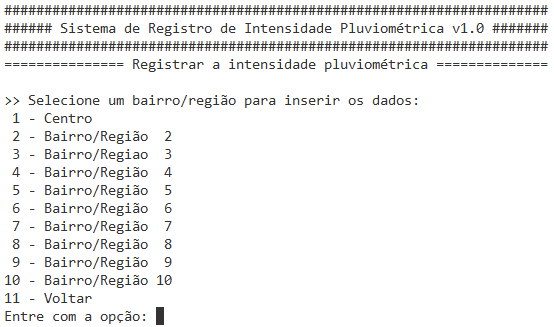


Figura - Acima a mensagem confirmando a inclusão da atividade física diária e abaixo o retorno ao menu inicial.

Com essa orientação o restante dos dados fora inserido:

* **A Figura 15** apresenta o resultado da inserção dos dados para o bairro/região **Jardim das Flores**.

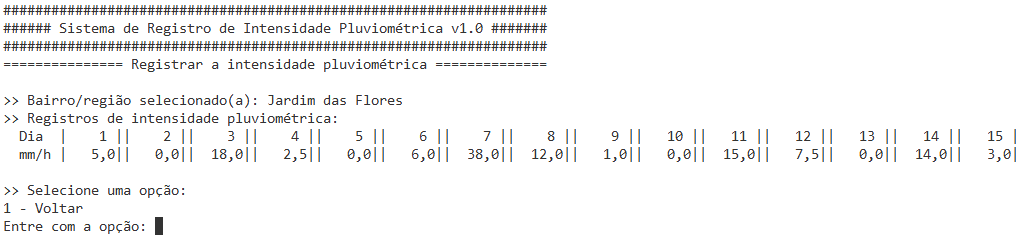


Figura - Bairro/região Jardim das Flores atualizado com os valores de intensidade pluviométrica.

* **A Figura 16** apresenta o resultado da inserção dos dados para o bairro/região **Universitário**.

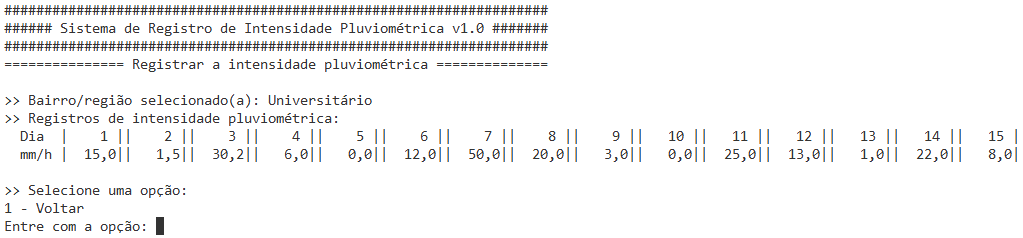


Figura - Bairro/região Universitário atualizado com os valores de intensidade pluviométrica.

* **A Figura 17** apresenta o resultado da inserção dos dados para o bairro/região **Industrial**.

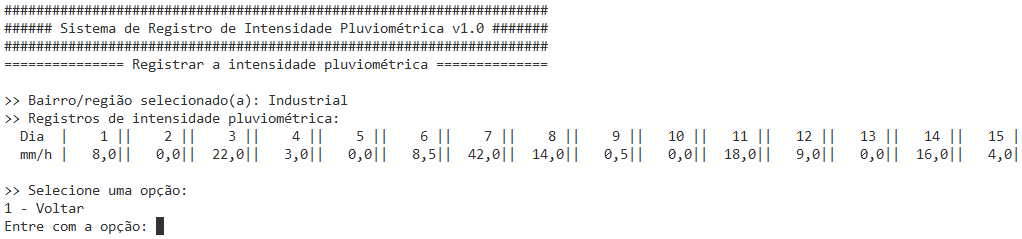


Figura - Bairro/região Industrial atualizado com os valores de intensidade pluviométrica.

* **A Figura 18** apresenta o resultado da inserção dos dados para o bairro/região **Vila Nova**.

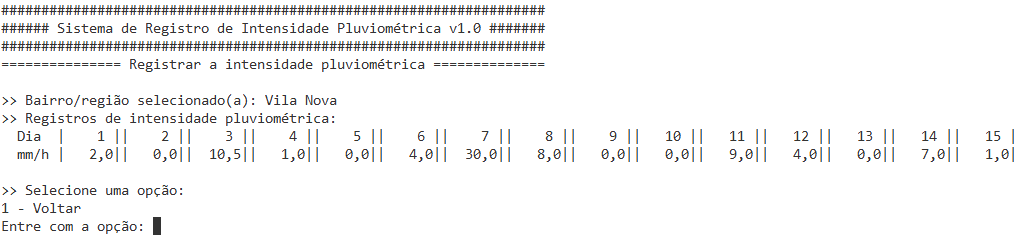


Figura - Bairro/região Vila Nova atualizado com os valores de intensidade pluviométrica.

* **A Figura 19** apresenta o resultado da inserção dos dados para o bairro/região **Marítimo**.

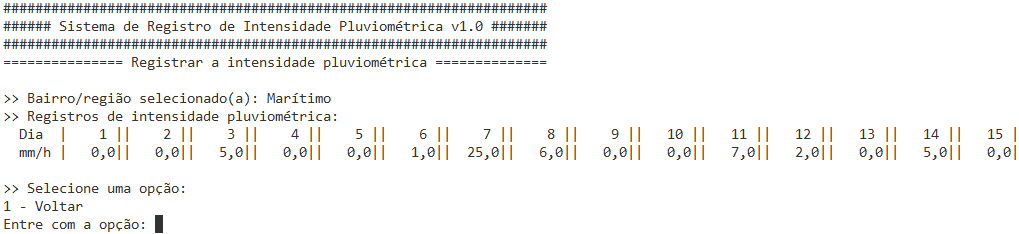


Figura - Bairro/região Marítimo atualizado com os valores de intensidade pluviométrica.

* **A Figura 20** apresenta o resultado da inserção dos dados para o bairro/região **Boa Vista**.

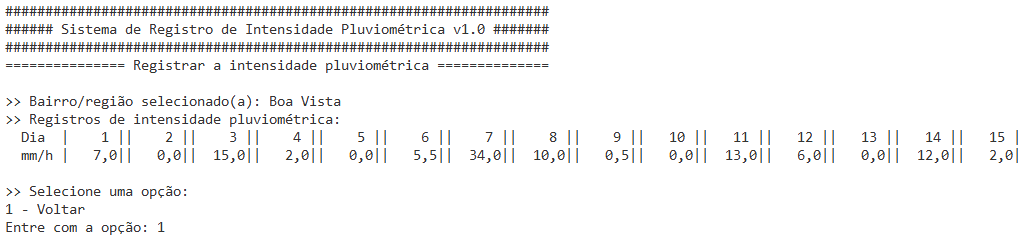


Figura - Bairro/região Boa Vista atualizado com os valores de intensidade pluviométrica.

* **A Figura 21** apresenta o resultado da inserção dos dados para o bairro/região **Jardim Sul**.

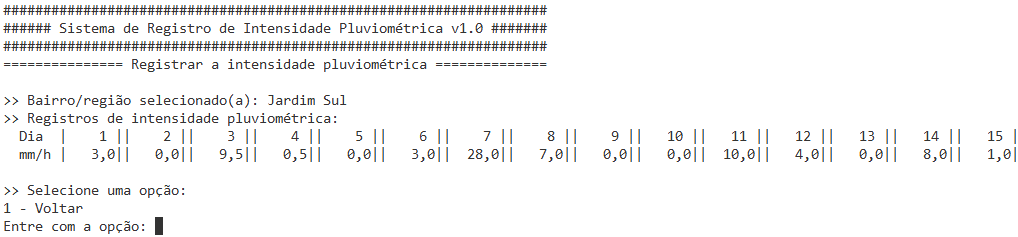


Figura - Bairro/região Jardim Sul atualizado com os valores de intensidade pluviométrica.

* **A Figura 22** apresenta o resultado da inserção dos dados para o bairro/região **Parque Oeste**.

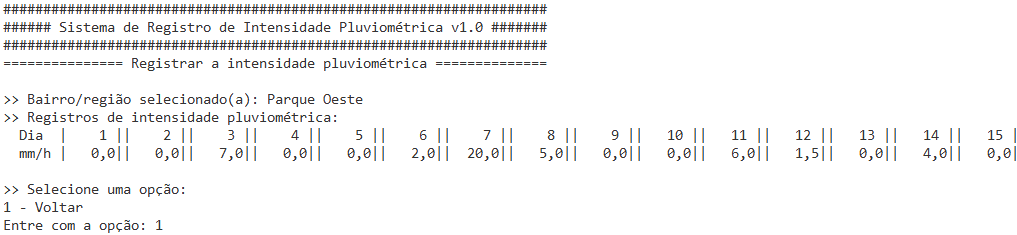


Figura - Bairro/região Parque Oeste atualizado com os valores de intensidade pluviométrica.

* **A Figura 23** apresenta o resultado da inserção dos dados para o bairro/região **Lagoa**.

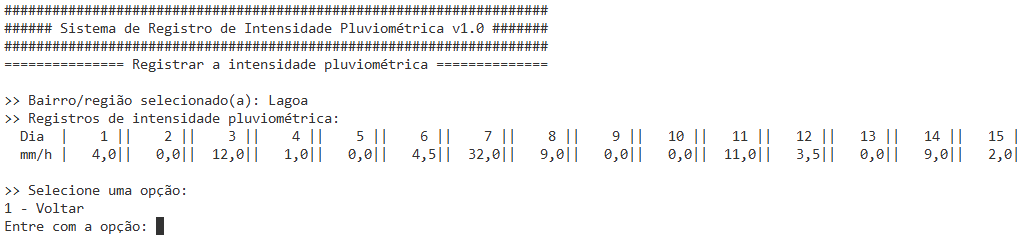


Figura – Bairro/região Lagoa atualizado com os valores de intensidade pluviométrica.

* Menu “**Selecione um bairro/região para inserir os dados**” atualizado com os novos bairros/regiões (Figura 24).

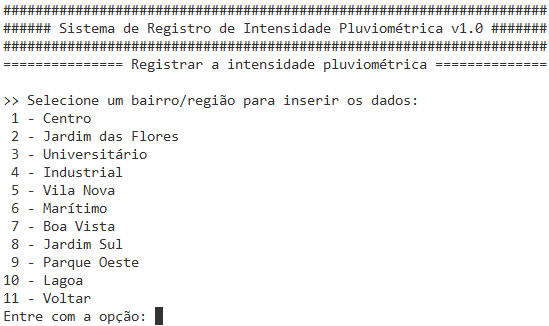


Figura - Menu atualizado com os bairros/regiões.

* Gráfico de intensidade de cores, *Heatmap*, com as informações atualizadas de intensidade pluviométricas (Figura 24).

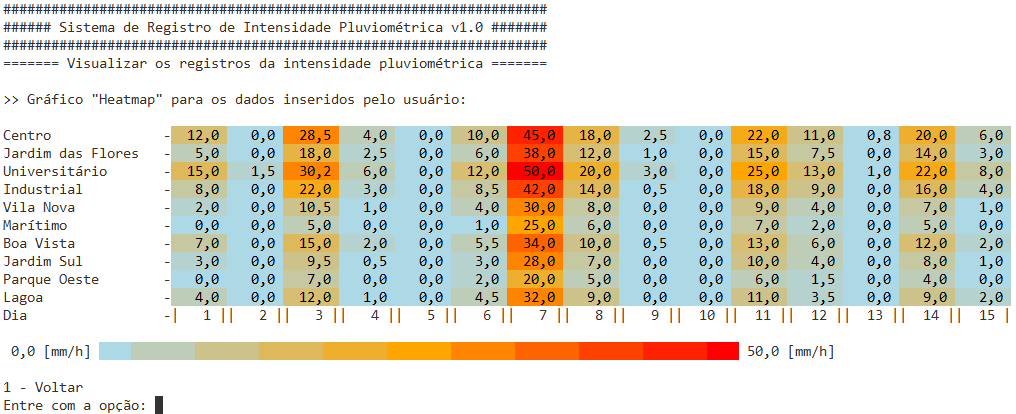


Figura - Gráfico de intensidade de cores, no estilo Heatmap, para um conjunto de dados atualizados. Destaque para os valores máximo e mínimo que determinam o início e o fim da transição das cores.

## Mensagem de encerramento do programa

Após realizar as operações descritas nos tópicos anteriores foi selecionado as opções referentes a “**Voltar”** em cada menu até o software retornar ao menu inicial com as opções referente aos “**Serviços disponíveis**” e foi selecionado a opção “**3 – Sair**” para encerrar a execução do software (Figura 26).

O software imprime no *prompt* uma mensagem de encerramento ao finalizar sua execução (Figura 27).

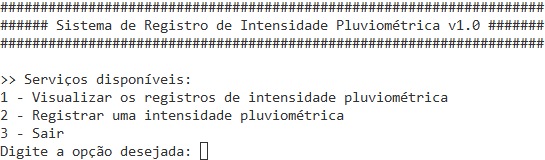


Figura - Menu dos serviços disponíveis.

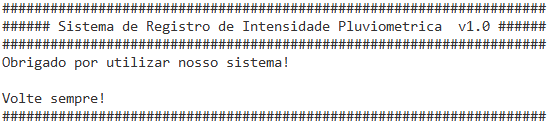


Figura – Mensagem de encerramento do programa.

# Conclusão

Objetivo desta experiência prática foi desenvolver um software em linguagem C para realizar o registro de um intervalo definido de dias de dados da intensidade pluviométrica em diferentes bairros/regiões com opção de visualizar essas informações pluviométricas adicionadas na forma de um gráfico de intensidades de cores, *Heatmap*.

O software desenvolvido foi compilado e executado, com o resultado de suas interações capturado em forma de imagens e listado nos tópicos anteriores.

As exigências de uso das estruturas de decisão, controle e repetição foram atendidas, indicando sucesso no desenvolvimento desta atividade.

O diretório contendo o repositório do projeto desta atividade está disponível em [Salvi (2025)](https://github.com/engfabiodesalvi/rain-by-zone.git).

# Referências

MICROSOFT. *Using GCC with MinGW*. Disponível em: [https://code.visualstudio.com/docs/cpp/config-mingw](https://code.visualstudio.com/docs/cpp/config-mingw?utm_source=chatgpt.com). Acesso em: 28 set. 2025.

MICROSOFT CORPORATION. *Visual Studio Code*. Versão 1.93. Redmond: Microsoft, 2025. Disponível em: <https://code.visualstudio.com/>. Acesso em: 28 set. 2025.

MINGW-W64 PROJECT. *MinGW-w64: Minimalist GNU for Windows*. [S.l.]: MinGW-w64, 2025. Disponível em: <https://www.mingw-w64.org/>. Acesso em: 28 set. 2025.

UNIVERSIDADE DE FRANCA (UNIFRAN). *Experiências Práticas – Fundamentos da Introdução à Programação e ao Pensamento Computacional*. Franca: UNIFRAN, [2025]. Apostila da disciplina Algoritmos e Pensamento Computacional, Curso EAD. Disponível em: <https://www.unifran.edu.br/>. Acesso em: 28 set. 2025.

SALVI, Fabio Toledo Bonemer De. *basic-food-basket-flowchart*. GitHub, 2025. Disponível em: <https://github.com/engfabiodesalvi/rain-by-zone.git>. Acesso em: 28 set. 2025.

1. A **intensidade pluviométrica** é definida como a quantidade de chuva que cai em uma determinada área durante um determinado período de tempo, sendo comumente expressa em **mm/h**. [↑](#footnote-ref-1)
2. ***Heatmap*** é definido como um mapa de intensidade de cores que representa as informações desejadas, podendo as informações serem numéricas ou textuais. [↑](#footnote-ref-2)