



CENTRO UNIVERSITARIO ANHANGUERA PITAGORAS UNOPAR DE CAMPO GRANDE

CURSO: ENGENHARIA DE SOFTWARE

DISCIPLINA: ALGORITMOS E PROGRAMAÇÃO ESTRUTURADA

ATIVIDADE PRÁTICA – UNIDADE 2, AULA 2

ESTRUTURAS DE CONTROLE E REPETIÇÃO

Aluno: Edmar Radanovis.

RA: 2025223493

Polo: Itapira / SP-UN944038

Ano 2025 / 2º Semestre

ANHANGUERA EDUCACIONAL

ENGENHARIA DE SOFTWARE

DISCIPLINA: Algoritmos e Programação Estruturada

PROFESSOR: Dr. Gilberto Fernandes Junior.

ALUNO: Edmar Radanovis

RA: 2025223493

TÍTULO: Relatório da Aula Prática – Estruturas de Controle e Repetição – Unidade 2 –
Aula 2

POLO: Itapira / SP-UN944038

Monte Sião, 22 de Novembro de 2025.

RESULTADOS DA ATIVIDADE PRÁTICA

Proposta:



Estruturas de Controle e Repetição - Estruturas de Repetição.

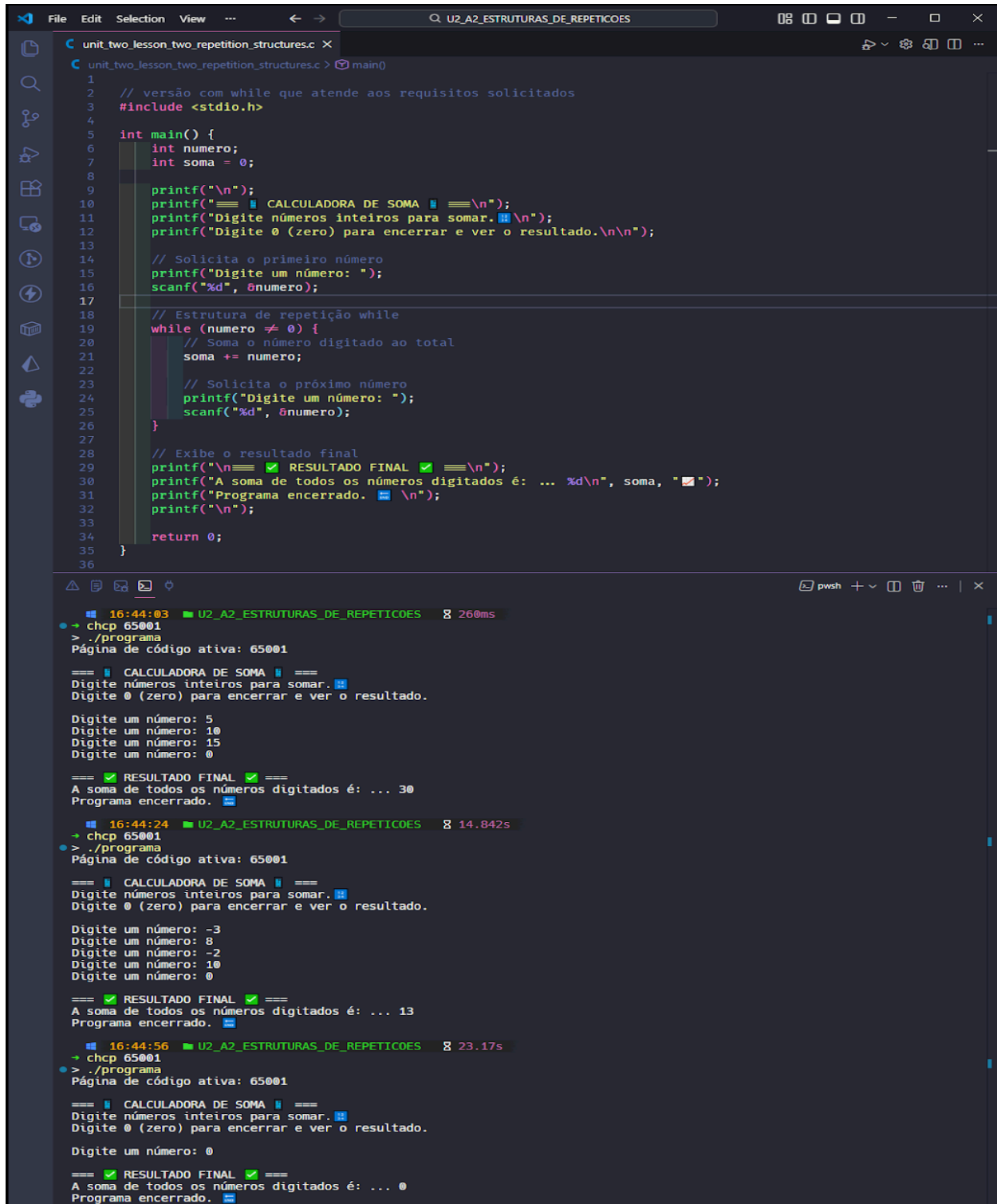
- ❖ Desenvolver um programa em **C** que utilize a estrutura de repetição `while` para calcular a soma de números inteiros até que o usuário digite zero, conforme roteiro da disciplina.

- Requisitos:

1. Utilizar estrutura de repetição while com teste no início do loop
2. Ler números inteiros do usuário continuamente
3. Calcular a soma de todos os números inseridos
4. Utilizar o número zero como valor sentinela para encerramento
5. Exibir o resultado final da soma quando o loop for encerrado

link do repositório no GitHub:

https://github.com/ed-radanovis/Eng_Software_A-P-E_U2-A2_11-2025.git



The image shows a code editor window with a C program and a terminal window below it. The C program is a sum calculator using a while loop. The terminal shows the execution of the program with three different inputs: 5, 10, 15; -3, 8, -2, 10; and 0.

```
1 // versão com while que atende aos requisitos solicitados
2 #include <stdio.h>
3
4 int main() {
5     int numero;
6     int soma = 0;
7
8     printf("\n");
9     printf("=== CALCULADORA DE SOMA ===\n");
10    printf("Digite números inteiros para somar.\n");
11    printf("Digite 0 (zero) para encerrar e ver o resultado.\n\n");
12
13    // Solicita o primeiro número
14    printf("Digite um número: ");
15    scanf("%d", &numero);
16
17    // Estrutura de repetição while
18    while (numero != 0) {
19        // Soma o número digitado ao total
20        soma += numero;
21
22        // Solicita o próximo número
23        printf("Digite um número: ");
24        scanf("%d", &numero);
25    }
26
27    // Exibe o resultado final
28    printf("\n=== RESULTADO FINAL ===\n");
29    printf("A soma de todos os números digitados é: ... %d\n", soma, " ");
30    printf("Programa encerrado.\n");
31    printf("\n");
32    return 0;
33 }
```

Terminal output:

```
16:44:03 U2_A2 ESTRUTURAS_DE_REPETICOES 8 260ms
+ chcp 65001
> ./programa
Página de código ativa: 65001

=== CALCULADORA DE SOMA ===
Digite números inteiros para somar.
Digite 0 (zero) para encerrar e ver o resultado.

Digite um número: 5
Digite um número: 10
Digite um número: 15
Digite um número: 0

=== RESULTADO FINAL ===
A soma de todos os números digitados é: ... 30
Programa encerrado.

16:44:24 U2_A2 ESTRUTURAS_DE_REPETICOES 8 14.842s
+ chcp 65001
> ./programa
Página de código ativa: 65001

=== CALCULADORA DE SOMA ===
Digite números inteiros para somar.
Digite 0 (zero) para encerrar e ver o resultado.

Digite um número: -3
Digite um número: 8
Digite um número: -2
Digite um número: 10
Digite um número: 0

=== RESULTADO FINAL ===
A soma de todos os números digitados é: ... 13
Programa encerrado.

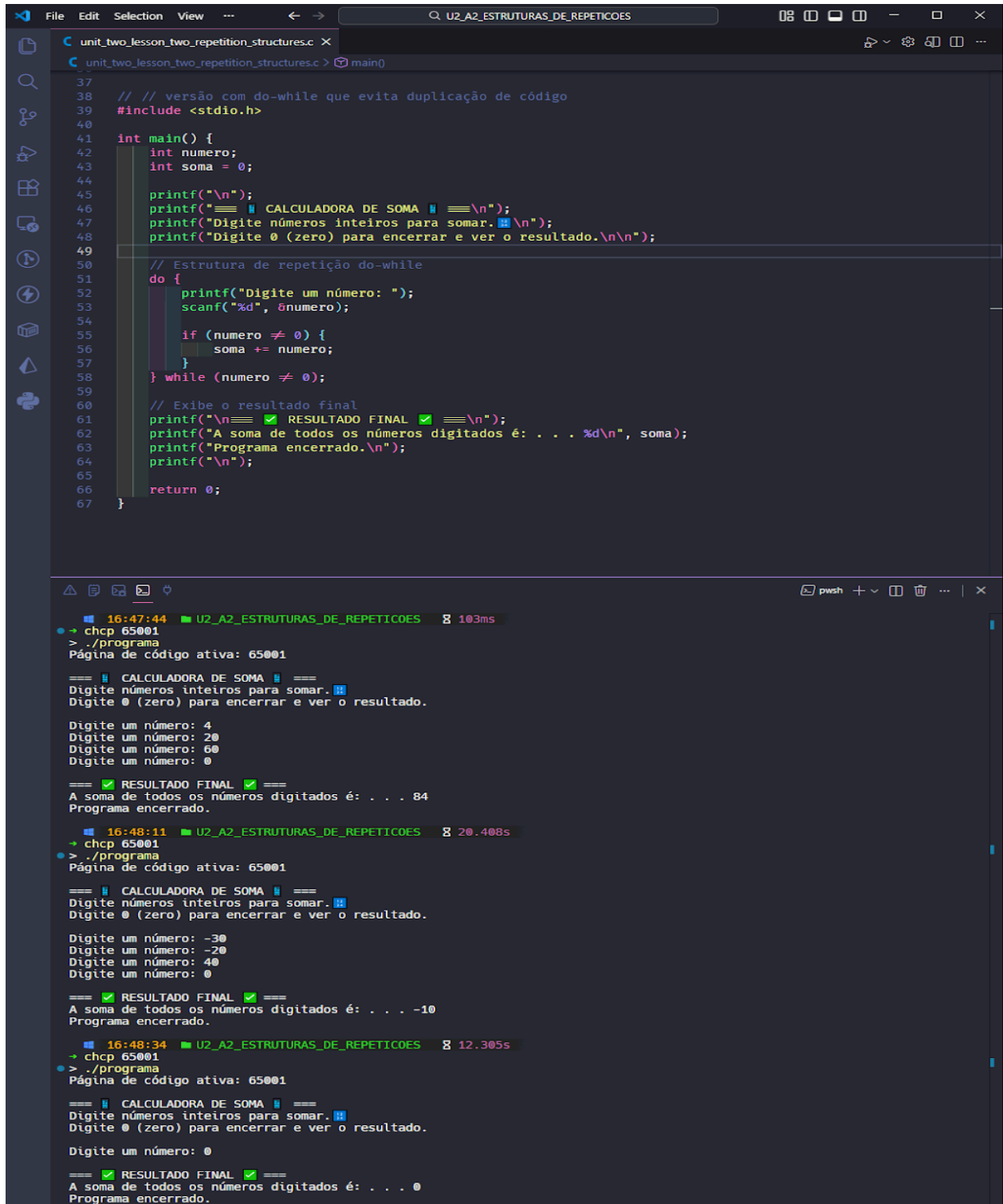
16:44:56 U2_A2 ESTRUTURAS_DE_REPETICOES 8 23.17s
+ chcp 65001
> ./programa
Página de código ativa: 65001

=== CALCULADORA DE SOMA ===
Digite números inteiros para somar.
Digite 0 (zero) para encerrar e ver o resultado.

Digite um número: 0

=== RESULTADO FINAL ===
A soma de todos os números digitados é: ... 0
Programa encerrado.
```

=> Figura 1: Print da tela com o código e os resultados da execução do script no terminal. (versão com while que atende aos requisitos solicitados).



The image shows a code editor window with a C program and a terminal window below it. The C program is a sum calculator using a do-while loop. The terminal shows three runs of the program with different inputs.

```
unit_two_lesson_two_repetition_structures.c X
unit_two_lesson_two_repetition_structures.c > main()
37
38 // // versão com do-while que evita duplicação de código
39 #include <stdio.h>
40
41 int main() {
42     int numero;
43     int soma = 0;
44
45     printf("\n");
46     printf("=== | CALCULADORA DE SOMA | ===\n");
47     printf("Digite números inteiros para somar. | |\n");
48     printf("Digite 0 (zero) para encerrar e ver o resultado.\n\n");
49
50     // Estrutura de repetição do-while
51     do {
52         printf("Digite um número: ");
53         scanf("%d", &numero);
54
55         if (numero != 0) {
56             soma += numero;
57         }
58     } while (numero != 0);
59
60     // Exibe o resultado final
61     printf("\n=== | RESULTADO FINAL | ===\n");
62     printf("A soma de todos os números digitados é: . . . %d\n", soma);
63     printf("Programa encerrado.\n");
64     printf("\n");
65
66     return 0;
67 }
```

Terminal output:

```
16:47:44 U2_A2 ESTRUTURAS DE REPETICOES 8 103ms
→ chcp 65001
> ./programa
Página de código ativa: 65001

=== | CALCULADORA DE SOMA | ===
Digite números inteiros para somar. | |
Digite 0 (zero) para encerrar e ver o resultado.

Digite um número: 4
Digite um número: 20
Digite um número: 60
Digite um número: 0

=== | RESULTADO FINAL | ===
A soma de todos os números digitados é: . . . 84
Programa encerrado.

16:48:11 U2_A2 ESTRUTURAS DE REPETICOES 8 20.408s
→ chcp 65001
> ./programa
Página de código ativa: 65001

=== | CALCULADORA DE SOMA | ===
Digite números inteiros para somar. | |
Digite 0 (zero) para encerrar e ver o resultado.

Digite um número: -30
Digite um número: -20
Digite um número: 40
Digite um número: 0

=== | RESULTADO FINAL | ===
A soma de todos os números digitados é: . . . -10
Programa encerrado.

16:48:34 U2_A2 ESTRUTURAS DE REPETICOES 8 12.305s
→ chcp 65001
> ./programa
Página de código ativa: 65001

=== | CALCULADORA DE SOMA | ===
Digite números inteiros para somar. | |
Digite 0 (zero) para encerrar e ver o resultado.

Digite um número: 0

=== | RESULTADO FINAL | ===
A soma de todos os números digitados é: . . . 0
Programa encerrado.
```

=> Figura 2: Print da tela com o código e os resultados da execução do script no terminal. (versão com do-while que evita duplicação de código).


Explicação da lógica utilizada:

A lógica utilizada para realizar a atividade baseia-se na aplicação prática das estruturas de repetição na linguagem **C**, conforme solicitado no roteiro, com foco principal na estrutura

while e apresentação alternativa do **do-while**.

- **Versão Principal com While:** Utiliza o laço **while** com a condição **numero != 0**, que é verificada no início de cada iteração. Esta abordagem requer a solicitação do primeiro número antes de entrar no loop, garantindo que a condição possa ser avaliada corretamente. A estrutura **while** é ideal quando é possível que o loop não execute nenhuma vez, como no caso de digitar zero imediatamente.
- **Versão Alternativa com Do-While:** Apresenta o laço **do-while** como alternativa, onde o bloco de código é executado pelo menos uma vez antes da verificação da condição. Esta versão elimina a duplicação da solicitação de entrada, porém requer uma verificação condicional interna **if** para evitar somar o zero final.
- **Acumulação de Valores:** Ambas as versões utilizam a variável **soma** para acumular o total dos números digitados, incrementando seu valor a cada iteração com o operador **+=**.
- **Valor Sentinela:** O número **zero** funciona como valor de parada, encerrando o loop quando detectado e exibindo o resultado final calculado.
- **Interface do Usuário:** O programa fornece instruções claras e feedback contínuo, solicitando números até que o usuário decida encerrar com zero.

Ressalvas sobre a versão do-while: Embora funcional e mais compacta, a versão com **do-while** requer uma verificação condicional interna (**if**) para evitar somar o zero, enquanto a versão com **while** segue mais fielmente a estrutura solicitada no roteiro com teste no início do loop.

Essa estrutura permite compreender na o funcionamento das estruturas de repetição na linguagem  e suas aplicações em cenários de entrada contínua de dados, atendendo aos objetivos de aprendizagem da unidade.