

ATIVIDADES PRÁTICAS SUPERVISIONADAS

Ciência da Computação

2ª Série

Programação Estruturada I

A Atividade Prática Supervisionada (ATPS) é um procedimento metodológico de ensino-aprendizagem desenvolvido por meio de etapas, acompanhadas pelo professor, e que tem por objetivos:

- ✓ Favorecer a autoaprendizagem do aluno.
- ✓ Estimular a corresponsabilidade do aluno pelo seu aprendizado.
- ✓ Promover o estudo, a convivência e o trabalho em grupo.
- ✓ Auxiliar no desenvolvimento das competências requeridas para o exercício profissional.
- ✓ Promover a aplicação da teoria na solução de situações que simulam a realidade.
- ✓ Oferecer diferenciados ambientes de aprendizagem

Para atingir estes objetivos, a ATPS propõe um desafio e indica os passos a serem percorridos ao longo do semestre para a sua solução.

Aproveite esta oportunidade de estudar e aprender com desafios da vida profissional.



AUTORIA:

Ivair Teixeira

Faculdade Anhanguera de Valinhos

Rodrigo Amaral Rocha

Faculdade Anhanguera de Valinhos

COMPETÊNCIAS E HABILIDADES

Ao concluir as etapas propostas neste desafio, você terá desenvolvido as competências e habilidades que constam, nas Diretrizes Curriculares Nacionais, descritas a seguir.

- ✓ Conhecer os aspectos teóricos, científicos e tecnológicos relacionados à computação.
- ✓ Operar equipamentos computacionais e sistemas de *software*.
- ✓ Analisar, documentar e solucionar problemas e necessidades passíveis de solução via computação.
- ✓ Desenvolver pesquisa científica e tecnológica.

Produção Acadêmica

- Relatório 1 - Planejamento Inicial. Contendo um esboço do programa em português estruturado, com uma descrição das ações e variáveis necessárias ao aplicativo básico.
- Relatório 2 - Comando `system()`. Listagem com as opções da função `system()` que podem ser utilizadas no aplicativo de forma criar uma interface mais agradável.
- Relatório 3 - Aplicativo básico. Contendo o código fonte e telas do aplicativo básico, que utiliza os comandos de atribuição, leitura, escrita e a estrutura de seleção, para realizar as funcionalidades propostas nas etapas 1 e 2.
- Relatório 4 - Algoritmos Avançados. Contendo a descrição das técnicas utilizadas para realizar as opções: Fatorial, Eponenciação e Raiz quadrada.
- Relatório 5 Aplicativo Final. Contendo o código fonte e telas do aplicativo avançado, que utiliza as estruturas de iteração e as estruturas de dados compostas para complementar o aplicativo com as funcionalidades propostas nas etapas 3 e 4.

Participação

Para a elaboração desta atividade, os alunos deverão previamente organizar-se em equipes de 4 a 5 participantes e entregar seus nomes, RAs e e-mails ao professor da disciplina. Essas equipes serão mantidas durante todas as etapas.

DESAFIO

A matemática e a informática são áreas que estão fortemente relacionadas e apresenta constante evolução, muitas vezes, decorrente da influência que existe entre elas. De um lado, o avanço da computação foi alavancado pela crescente necessidade de programas voltados a resolução de problemas matemáticos. De outro, a alta capacidade de cálculo dos computadores permite explorar aplicações da matemática que seriam inviáveis se realizadas por operações puramente manuais.

O desafio proposto consiste na elaboração de um aplicativo, em linguagem de programação C, que apresenta uma calculadora que ofereça operações básicas nas etapas 1 e 2 e mais avançadas nas etapas 3 e 4 da ATPS. A elaboração do aplicativo pode ser segmentada para ser planejada, codificada e testada por etapas, de forma incremental e evolutiva. A tela inicial do aplicativo deve exibir ao usuário um *menu* com as operações e receber do usuário a opção desejada. Na sequência são solicitados os parâmetros para aquela operação, por exemplo, se o usuário escolheu a opção de soma, deverá haver a entrada dos dois números, para concluir o cálculo e apresentar o resultado obtido. Após isso o programa fica em pausa com a apresentação da mensagem: "Pressione qualquer tecla para continuar". No final de cada operação o aplicativo deve limpar a tela e retornar ao *menu* principal.

Objetivo do Desafio

Implementar uma calculadora via programação em Linguagem C. Deverá ser explorado o uso dos comandos e estruturas abordados na disciplina de programação estruturada na elaboração de um projeto prático que integre esses comandos de forma simples, porém funcional. Como resultado espera-se um conjunto de relatórios com o código fonte que apresente uma interface simples ao usuário com as funções de uma calculadora.

Livro Texto da Disciplina

A produção desta ATPS é fundamentada no livro-texto da disciplina, que deverá ser utilizado para solução do desafio:

MIZRAHI, Victorine Viviane. *Treinamento em Linguagem C: módulos 1 e 2*. PLT- 706. 1ª ed. São Paulo: Pearson - Prentice Hall, 2009.

ETAPA 1 (tempo para realização: 5 horas)

- ✓ Aula-tema: Introdução a Linguagem de programação C. Estrutura de um programa. Abordagem de compilação, linkedição e execução de um programa. Conceito e aplicações de variáveis. Exemplos e exercícios.

Esta atividade é importante para que você entenda o processo de declaração e definição de variáveis, os tipos de dados e seus relacionamentos.

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS

Passo 1 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Fazer a leitura do **Capítulo 1 - Introdução**, do livro texto MIZRAHI, Victorine Viviane. *Treinamento em Linguagem C*. PLT nº 706. 1ª ed. São Paulo: Pearson, 2007. p. 10 a 17, com foco nas variáveis e seus tipos.
2. Elaborar um esboço do programa, listando cada ação a ser executada e analisando criteriosamente quais os dados envolvidos, bem como os tipos de variáveis que melhor se adequam para armazená-los. As funcionalidades da 1ª etapa são:

- (+) Soma
- (-) Subtração
- (*) Multiplicação
- (/) Divisão
- (E) Exponenciação
- (R) Raiz Quadrada
- (P) Porcentagem

Na Tabela 1 são apresentados os itens que devem ser analisados referentes ao aplicativo básico.

Tabela 1 – Ações da primeira etapa.

Itens	
1 - Menu Principal.	5 - Operação de Divisão.
2 - Operação de Soma.	6 - Operação de Exponenciação.
3 - Operação de Subtração.	7 - Operação de Raiz Quadrada.
4 - Operação de Multiplicação.	8 - Operação de Porcentagem.

A descrição em português estruturado a seguir, apresenta o esboço do *menu* e da ação de soma.

Item 1: Menu principal.

Variável: **op** do tipo **caractere**.

Ações:

1. Apresentar na tela as opções do aplicativo básico. (Soma, Subtração, Multiplicação, Divisão, Exponenciação, Raiz Quadrada, Porcentagem e Sair).
2. Receber a opção e armazenar (**op**).

Item 2: Operação de soma.

Variáveis: **val1**, **val2** e **res** do tipo **inteiro**.

Ações:

1. Solicitar e armazenar o primeiro valor (**val1**).
2. Solicitar e armazenar o segundo valor (**val2**).
3. Realizar o cálculo e armazenar o resultado (**res**).
4. Apresentar o resultado na tela (**res**).

Passo 2 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Pesquisar no *site* sugerido a seguir, sobre a função **system()**. Este comando permite acessar funcionalidades oferecidas pelo MSDOS por *prompt* de comando e melhorar significativamente a operabilidade do aplicativo, torna-o mais amigável e atraente para o usuário.
2. Elaborar uma listagem com o nome e a descrição de quatro funcionalidades interessantes para serem utilizadas no programa. A função **system()** está descrita na biblioteca **stdio.h**, que deve ser incluída no início do programa. Um material de aula sobre esta função está disponível no site sugerido a seguir.

Site sugerido para pesquisa

- SYSTEM. Comando **System.doc**. Disponível em:
<https://docs.google.com/file/d/0B114oKTDsLjnQWQwYnhtWjFTLTA/edit?usp=sharing>. Acesso em: 04 abr. 2013.

Passo 3 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Redigir o **Relatório 1 - Planejamento Inicial**. Contendo um esboço do programa em português estruturado, com uma descrição das ações e variáveis necessárias ao aplicativo básico.

2. Redigir o **Relatório 2 - Comando `system()`**. Contendo a listagem com as opções da função `system()` que podem ser utilizadas no aplicativo de forma tornar a operação mais agradável.

ETAPA 2 (tempo para realização: 5 horas)

- ✓ **Aula-tema:** Apresentação de operadores aritméticos, lógicos e relacionais. Atribuição. Criação de expressões aritméticas, lógicas e relacionais. Tabela verdade. Exemplos e exercícios. Exercícios em laboratório. Apresentação teórica e prática da estrutura do comando de seleção `if`.

Esta atividade é importante para praticar a utilização dos comandos de entrada e saída, ou seja, os comandos responsáveis pela elaboração da interface com o usuário. Estas funções estão descritas nas bibliotecas `stdio.h` e `stdlib.h`. As estruturas de seleção permitem realizar ações diferentes, em resposta a interação do usuário na interface.

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS

Passo 1 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Fazer a leitura do **Capítulo 1 - Introdução**, do livro texto MIZRAHI, Victorine Viviane. *Treinamento em Linguagem C*. PLT nº 706. 1ª ed. São Paulo: Pearson, 2007. p. 7 a 10, com foco na estrutura de controle Seleção.
2. Elaborar o código em Linguagem C, que apresente a tela inicial do aplicativo básico, conforme a Figura 1 apresentada a seguir. Capturar a opção com apenas uma, para isso ler sobre a função `getche()` nas páginas 30 a 32 do capítulo 2 - Operadores, do livro texto. Para testar, elaborar um código que, capture a tecla referente a opção, limpe a tela e apresente a tecla pressionada.

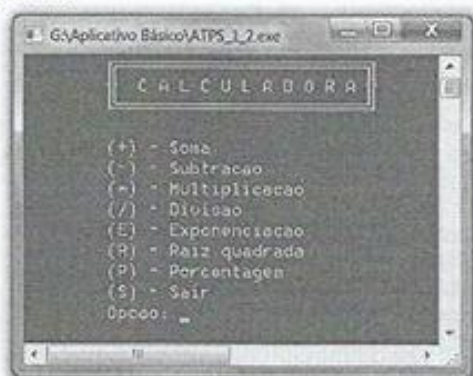


Figura 1 – Tela inicial do aplicativo básico.

O quadro ao redor da palavra calculadora pode ser obtido com a utilização de caracteres gráficos apresentados capítulo 1 - Introdução, do livro nas páginas 21 a 23. A formatação desta tela deve ser igual ao apresentado na Figura 1.

Passo 2 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Fazer a leitura do Capítulo 4 - Comando de decisões, do livro texto MIZRAHI, Victorine Viviane. *Treinamento em Linguagem C*. PLT nº 706. 1ª ed. São Paulo: Pearson, 2007. p. 93 a 97, com foco nas estruturas de seleção.
2. Desenvolver a estrutura de seleção que execute o algoritmo para resolver a operação desejada. Para obter os valores utilizados no cálculo leia o capítulo 2 do livro texto, sobre o comando `scanf()` nas páginas 28 a 30. Na Figura 2(A) e 2(B) e são apresentadas as telas de entrada de dados e na Figura 2(C) a tela de resultado da operação de multiplicação.



Figura 2 – Telas da operação de multiplicação

O código em Linguagem C, exibido na Figura 3, apresenta o algoritmo para executar a operação de soma e pode servir de modelo inicial para a elaboração e formatação dos algoritmos das demais operações.

```
switch(ch)
{
    /*operação soma, disparada pelo pressionamento do caractere
    '+'. Recebe os valores do usuário, realiza o cálculo e
    apresenta o resultado na tela.*/
    case '+': printf("\n\n\tS O M A\n");
              printf("\n\n\tPrimeiro valor: ");
              scanf("%d", &val1);
              printf("\n\tSegundo valor: ");
              scanf("%d", &val2);
              res = val1+val2;
              printf("\n\t%d + %d = %d", val1, val2, res);

    break;
}
```

Figura 3 – Código da Operação de Soma.

Um detalhe específico da operação de divisão é verificar se o segundo valor é zero, pois caso seja realizada divisão por zero ocorre um erro e o programa será interrompido. Esta tela é apresentada na Figura 4(A). Caso seja escolhida uma opção inválida deverá ser

impressa na tela a frase: "Opcao Invalida", conforme a Figura 4(B). Da mesma forma, caso seja escolhida a opção 's' será apresentada a frase: "programa finalizado" como pode ser visto na Figura 4(C).



Figura 4 – (A) Erro divisão por zero, (B) Opção Inválida e (C) Sair

Para as duas próximas opções, podem ser utilizadas as funções definidas na biblioteca `math.h`. As telas de entradas de dados e resultados para a operação de exponenciação é apresentada na Figura 5(A), Raiz Quadrada na Figura 5(B) e Porcentagem Figura 5(C).



Figura 5 – (A) Exponenciação, (B) Raiz Quadrada e (C) Porcentagem

Após cada sequência de execução o programa entra em pausa e deve ser finalizado ao pressionar qualquer tecla. Para reduzir a complexidade, não são necessários testes que realizem a validação dos valores digitados pelo usuário, por exemplo, digitação de letras quando é solicitado um valor numérico. No entanto a escolha da opção desejada no menu deve funcionar para caracteres maiúsculos ou minúsculos.

Passo 3 (Equipe)

Redigir o **Relatório 3 – Aplicativo Básico**. Contendo o código fonte e telas do aplicativo básico, que utiliza os comandos de atribuição, leitura, escrita e a estrutura de seleção, para realizar as funcionalidades propostas nas etapas 1 e 2. O código deve estar corretamente endentado e organizado, com comentários significativos que expliquem o funcionamento dos algoritmos.

ETAPA 3 (tempo para realização: 5 horas)

- ✓ **Aula-tema:** Apresentação do comando de repetição **while**. Apresentação do comando de repetição **for**. Apresentação do comando de repetição **do..while**.

Esta atividade é importante para entender os laços de repetição, seu modo de funcionamento, suas características e sua aplicabilidade. Operações repetitivas internas são altamente exploradas na programação e constituem a base para a resolução de uma infinidade de problemas

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS**Passo 1 (Equipe)**

Rever o material sobre as aplicações da função **system()**, disponibilizado no *site* a seguir, para ajustar a aparência e opções disponibilizadas na tela para a interface do aplicativo avançado. Alterar o título da janela, bem como as cores dos caracteres e fundo da tela.

Site sugerido para pesquisa

- **SYSTEM.** *Comando* *System.doc.* Disponível em:
<https://docs.google.com/file/d/0B114oKTDsLjnQWQwYnhtWjFTLTA/edit?usp=s_haring>. Acesso em: 04 abr. 2013.

Nessa etapa o programa deve ser acrescido das operações: (F) Fatorial, (B) Conversão Binário para Decimal e (D) Conversão Decimal para Binário. A nova aparência do aplicativo pode ser observada na Figura 6 apresentada a seguir.

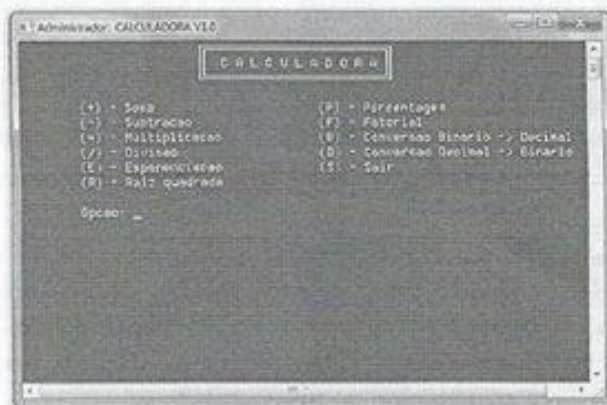


Figura 6 – Tela do aplicativo avançado

Passo 2 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Fazer a leitura do **Capítulo 3 – Laços**, do livro texto MIZRAHI, Victorine Viviane. *Treinamento em Linguagem C*. PLT nº 706. 1ª ed. São Paulo: Pearson, 2007. p. 69 a 71, com foco nas estruturas de iteração.

2. Elaborar o laço de repetição que limpa a tela e retorna a execução para o *menu* principal do programa, após cada sequência de execução, ou seja, o programa somente será finalizado com a escolha da opção: (S) – sair.

Passo 3 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Elaborar o código que resolva a opção Fatorial, com base nas estruturas de controle estudadas nas páginas 54 a 60 do capítulo 3 – Laços, do livro texto. Solicitar o valor, e imprimir o resultado no formato: "O fatorial de *val1* = *res*".
2. Atualizar os algoritmos das operações de Exponenciação e Raiz Quadrada, caso as soluções atuais tenha utilizado funções da biblioteca **math.h**. Nesta etapa não é permitido o uso de funções desta biblioteca, portanto os algoritmos devem ser substituídos por algoritmos equivalentes que utilizem as estruturas de controle para obter o resultado.

Passo 4 (Equipe)

Redigir o Relatório 4 – Algoritmos Avançados. Contendo a descrição das técnicas utilizadas para realizar as opções: Fatorial, Exponenciação e Raiz quadrada. O texto a seguir exemplifica o formato deste relatório no que diz respeito à operação de cálculo de fatorial.

Item: fatorial.

Variável: **num**, **fat** do tipo caractere.

Descrição: A variável **fat** é inicializada como valor 1 e recebe-se do usuário o valor a ser calculado (**num**). Um laço **for()**, com decremento de 1, inicializa o seu contador (**i**) com o valor a ser calculado, que será decrementado a cada iteração. Na primeira iteração a variável **fat** acumula seu valor atual, que é 1 multiplicado pela valor da variável **i**. Nas demais iterações, **fat** acumula o produto do seu valor atual com o valor de **i** decrementado, até que **i** seja igual a 1. Na Figura 7 é apresentado um exemplo da simulação desta função.

```
Simulação:
fat = 1 e num=5.
1ª iteração: fat ← fat * 5 ou 1*5 = 5.
2ª iteração: fat ← fat * 4 ou 5 * 4 = 20.
3ª iteração: fat ← fat * 3 ou 20 * 3 = 60.
4ª iteração: fat ← fat * 2 ou 60 * 2 = 120
5ª iteração: fat ← fat * 1 ou 120 * 1 = 120.
Algoritmo em C.
printf("\n\n\tValor: ");
scanf("%d", &num);
for(i = num; i > 1; i--)
    fat *= i;
printf("\n\tO Fatorial de %d = %d", num, fat);
```

Figura 7 – Tela do aplicativo avançado

ETAPA 4 (tempo para realização: 5 horas)

- ✓ **Aula-tema:** Introdução a matrizes unidimensionais (vetores). Estudo de matrizes bidimensionais.

Esta atividade é importante para que possa praticar a utilização de estrutura de dados do tipo vetor. Os vetores unidimensionais ou multidimensionais são amplamente utilizados no armazenamento de dados e oferece uma estrutura organizada que pode ser explorada de forma rápida e eficiente por algoritmos de busca.

Para realizá-la, devem ser seguidos os passos descritos.

PASSOS

Passo 1 (Equipe)

Fazer as atividades apresentadas a seguir.

1. Fazer a leitura do **Capítulo 11 – Memória e Impressão de Caracteres**, do livro texto MIZRAHL, Victorine Viviane. *Treinamento em Linguagem C*. PLT nº 706. 1ª ed. São Paulo: Pearson, 2007. p. 129, com foco sistemas numéricos.
2. Fazer a leitura do material de aula sobre funções matemáticas disponibilizado no *site* a seguir.

Passo 2 (Equipe)

Desenvolver as funções de conversão de base de Binário para Decimal e Decimal para Binário. Na Figura 8(A) é apresentada a tela da conversão de Binário para Decimal e na Figura 8(B) a conversão de Decimal para Binário.

Site sugerido para pesquisa

- **FUNÇÕES.** *Funções matemáticas.* Disponível em: <https://docs.google.com/file/d/0B114oKTDsLjnTTFQY0c1SWJvSGc/edit?usp=sharing>. Acesso em: 04 abr. 2013.



Figura 8 – (A) Conversão Binário para Decimal e (B) Conversão Decimal para Binário

Passo 3 (Equipe)

Redigir o Relatório 5 - Aplicativo Final. Contendo o código fonte e telas do aplicativo avançado, que utiliza as estruturas de iteração e as estruturas de dados compostas para complementar o aplicativo com as funcionalidades propostas nas etapas 3 e 4.

Padronização

O material escrito solicitado nesta atividade deve ser produzido de acordo com as normas da ABNT, com o seguinte padrão (exceto para produções finais não textuais):

- em papel branco, formato A4;
- com margens esquerda e superior de 3cm, direita e inferior de 2cm;
- fonte *Times New Roman* tamanho 12, cor preta;
- espaçamento de 1,5 entre linhas;
- se houver citações com mais de três linhas, devem ser em fonte tamanho 10, com um recuo de 4cm da margem esquerda e espaçamento simples entre linhas;
- com capa, contendo:
 - nome de sua Unidade de Ensino, Curso e Disciplina;
 - nome e RA de cada participante;
 - título da atividade;
 - nome do professor da disciplina;
 - cidade e data da entrega, apresentação ou publicação.

Para consulta completa das normas ABNT, acesse a Normalização de Trabalhos Acadêmicos Anhanguera. Disponível em:
<http://issuu.com/normalizacao/docs/normalizacao_de_trabalhos_acad_m>. Acesso em: 13 maio 2014.