

1. Aplique as 4 Habilidades do Pensamento Computacional e descreva sua solução quanto:

A. Decomposição:

O divide o problema em várias etapas:

- Solicitar o nome do usuário e exibir mensagem de boas-vindas. Receber o número de matérias.
- Para cada matéria, solicitar as notas dos quatro bimestres.
- Calcular a média das matérias.
- Exibir opções para o usuário (ver matérias, ver a média geral ou sair).
- Verificar a situação do aluno com base na escolha e nas médias calculadas.
- Encerrar o programa quando a opção de saída for selecionada.

B. Reconhecimento de Padrões:

- Padrão de entrada de dados: Repetição na leitura do nome das matérias e das notas de cada bimestre.
- Cálculo da média: Processo repetido para cada matéria, somando as quatro notas e dividindo por quatro.
- Verificação da situação: Um padrão lógico simples compara a média com valores específicos para determinar se o aluno está aprovado ou reprovado.
- Menu de opções: O fluxograma repete o processo de escolha até que o usuário finalize o programa.

A. Abstração:

- A lógica do algoritmo é baseada em decisões condicionais simples, como comparação de médias para determinar a situação do aluno.
- Foco em entradas simples (nome do usuário, matérias e notas) e cálculos básicos (média aritmética).
- O menu é apresentado de forma abstrata e direta, sem preocupação com tratamento de erros ou validação de dados avançada.

B. Algoritmos (Textual/Narrativo):

1. Solicitar o nome do usuário e exibir uma mensagem de boas-vindas.
2. Perguntar quantas matérias o usuário deseja adicionar.
3. Para cada matéria:
 - Solicitar o nome da matéria e as notas dos quatro bimestres.
 - Calcular a média das notas.
4. Exibir um menu para que o usuário escolha entre:
 - Ver a situação de uma matéria.
 - Ver a média do semestre.
 - Encerrar o programa.
5. Continuar mostrando o menu até que o usuário escolha encerrar.

2. Considerando as diferentes fases de um algoritmos descreva sua solução quanto:

A. Entrada:

- Nome do usuário.
- Quantidade de matérias.
- Nome de cada matéria.
- Notas dos quatro bimestres de cada matéria.
- Escolha do usuário (ver média ou encerrar).

A. Processamento:

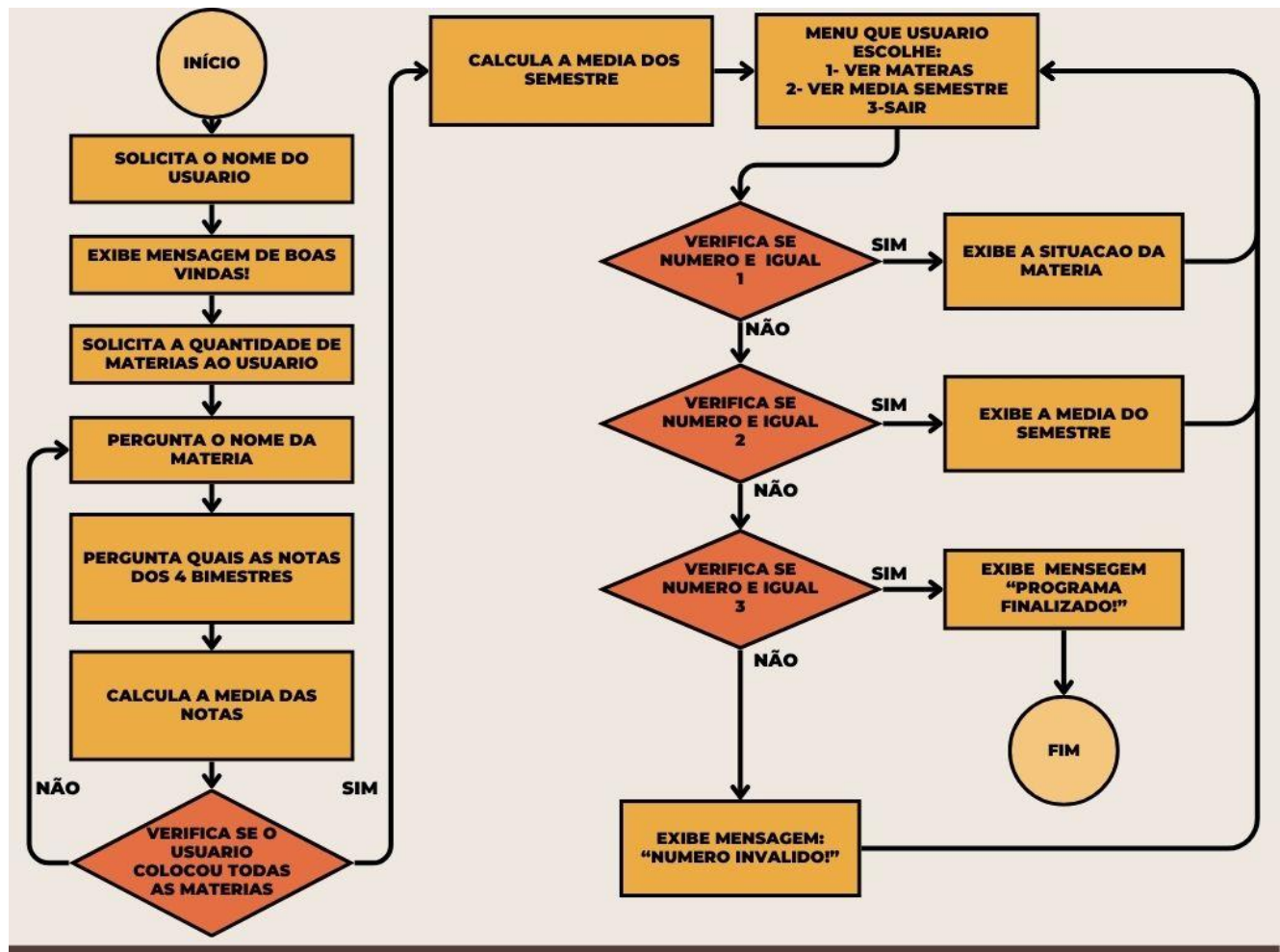
- Calcular a média de cada matéria com base nas notas fornecidas.
- Comparar as médias calculadas com limites fixos para determinar a situação do aluno.

B. Saída:

- Exibição da média de cada matéria.
- Exibição da situação do aluno para cada matéria (aprovado ou reprovado).
- Exibição da média do semestre (se o usuário optar).
- Mensagem final quando o usuário encerrar o programa.

Considerando as diferentes formas de representar um algoritmo descreva sua solução quanto:

(a) Fluxograma/Diagrama:



(b)Pseudocódigo:

Para Abrir o Código: [Clique aqui](#)

3. Traduza o seu algoritmo(pseudocódigo) para a sintaxe da linguagem C:

Para Abrir o Código em C: [Clique aqui](#)

Alunos: Daniel Silva Guedes (202208699804), Dennis Viera(matrícula), Maria Eduarda(matrícula)

4. Plus do time:

Fizemos uma atualização no sistema que agora permite o registo do nome de disciplinas, juntamente com as notas correspondentes a cada uma. Com isso, o sistema é capaz de calcular a média das notas em cada disciplina e classificar a situação do aluno como: "Aprovado" (média igual ou superior a 7.0), "Recuperação" (média entre 5.0 e 6.9) ou "Reprovado" (média abaixo de 5.0). Além disso, introduzimos um menu que possibilita ao usuário selecionar a disciplina da qual deseja consultar a média e a situação. Esse menu estará acessível até que o usuário opte por encerrar o programa, por meio de um comando específico.

5. Conclusão:

1. Fases de um Algoritmo Aplicadas ao Código:

- O problema foi bem definido: o algoritmo deve calcular as médias das matérias e determinar a situação do aluno, além de permitir que o usuário escolha qual matéria visualizar ou encerrar o programa.
- Foi desenhado o fluxo de operações, como a coleta dos nomes das matérias, as notas dos quatro bimestres, o cálculo da média, a verificação da situação do aluno e a interação com o usuário por meio de um menu.
- O fluxograma fornecido é uma excelente forma de planejar visualmente o que o algoritmo fará, ajudando a identificar possíveis falhas no fluxo lógico.
- A fase de implementação envolve a transformação do algoritmo planejado (em fluxograma e pseudocódigo) em um código executável, nesse caso, escrito em C. O código foi estruturado para seguir os passos planejados, como a coleta de dados, o cálculo da média e a exibição dos resultados.
- Após a implementação, o código deve ser testado com diferentes entradas (como várias combinações de notas e quantidades de matérias) para garantir que todas as condições e situações sejam corretamente tratadas (aprovado, recuperação, reprovado).

- Um código bem documentado facilita a compreensão e a manutenção. Comentários no código podem ser utilizados para descrever o que cada parte do algoritmo faz, ajudando futuros desenvolvedores ou usuários a modificarem e manterem o programa.
- A manutenção pode incluir melhorias no algoritmo ou correções de erros após a fase de testes.

2. Vantagens de utilizar diferentes representações de algoritmos.

- Fluxogramas: Visualizam o fluxo de controle, sendo úteis para entender a lógica do algoritmo e identificar problemas antes da codificação.
- Pseudocódigo: Permite descrever o algoritmo em uma linguagem próxima da humana, facilitando o planejamento e a comunicação entre membros da equipe antes da implementação.
- Código Fonte: A representação final e executável do algoritmo, que pode ser testada e utilizada diretamente para resolver o problema.

Eficiência e Desempenho:

- A linguagem C é conhecida por sua velocidade de execução e eficiência. Como é uma linguagem de baixo nível (mais próxima da linguagem de máquina), o desempenho do código tende a ser mais rápido.

Controle sobre recursos:

- C permite um controle mais direto sobre os recursos do sistema, como alocação de memória, o que é crucial em ambientes que exigem alta performance ou onde os recursos são limitados.

Portabilidade:

- O código escrito em C pode ser facilmente portado para diferentes plataformas, já que a linguagem é amplamente suportada em diversos sistemas operacionais.

Simplicidade da estrutura:

- No paradigma de programação estruturada, C é uma das linguagens mais utilizadas devido à sua simplicidade na criação de loops, condições e funções, permitindo uma lógica de programação clara e estruturada.

Desvantagens:

1. Gestão manual de memória:
 - Em C, o gerenciamento de memória é feito manualmente, o que aumenta a complexidade e a chance de erros como vazamentos de memória e falhas de segmentação.
2. Pouca abstração:
 - Comparado a linguagens mais modernas, como Python ou Java, a linguagem C oferece pouca abstração, o que pode tornar o desenvolvimento mais verboso e suscetível a erros, especialmente em grandes sistemas.
3. Falta de suporte a paradigmas modernos:
 - C não oferece suporte nativo para conceitos de orientação a objetos, por exemplo, o que torna o desenvolvimento de sistemas complexos mais desafiador do que em linguagens que suportam esses paradigmas.