

UNIVERSIDADE FEDERAL DE PELOTAS
Faculdade de Educação
Programa de Pós-Graduação em Ensino de Ciências e Matemática



Produto Educacional

**TECHMATH: Ferramenta Interativa para a Aprendizagem de
Matemática Discreta**

Vanessa Mattoso Cardoso

Pelotas, 2024

Vanessa Mattoso Cardoso

**TECHMATH: Ferramenta Interativa para a Aprendizagem de
Matemática Discreta**

Produto educacional apresentado ao Programa de Pós-Graduação em Ensino de Ciências e Matemática (PPGECM) da Faculdade de Educação (FAE) da Universidade Federal de Pelotas, como requisito parcial para a obtenção do título de Mestre em Ensino de Ciências e Matemática.

Orientador: Prof. Dr. André Luis Andrejew Ferreira

Pelotas, 2024

INTRODUÇÃO

No Mestrado Profissional em Ensino, além da dissertação, para obtenção do título, o mestrando precisa desenvolver um processo ou produto educacional (PE) que seja aplicado em condições reais de sala de aula ou em outros ambientes de ensino.

Para a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), o PE é um recurso que apresenta estratégias educacionais para favorecer a prática pedagógica e tornar pública a pesquisa realizada durante o mestrado profissional. Podendo consistir em uma sequência didática, um aplicativo, um jogo, um vídeo, um conjunto de vídeo-aulas, um equipamento, um curso, entre outras possibilidades (CAPES, 2022).

Nesse sentido, este documento tem por finalidade apresentar o PE, na categoria aplicativo, que foi idealizado e implementado com base na dissertação intitulada “Explorando estratégias do Pensamento Computacional na Resolução de Problemas em Matemática Discreta: em busca da Aprendizagem Significativa”.

A pesquisa foi motivada pela necessidade de desenvolver estratégias que promovessem a aprendizagem dos conteúdos de Matemática Discreta (MD) com os estudantes do curso Tecnólogo em Análise e Desenvolvimento de Sistemas (TADS) do Instituto Federal Sul-rio-grandense do Câmpus Santana do Livramento (IFSul/SL).

Tal demanda justifica-se pelos altos índices de reprovação e evasão na disciplina (superiores a 50%) que é considerada, por autores como Parmar (2019) e Oliveira (2022) a espinha dorsal do curso.

Dentre as dificuldades apresentadas pelos estudantes, destaca-se a relacionada à resolução de problemas (seja em MD ou em disciplinas da área técnica como Lógica de Programação).

Na busca por pressupostos para desenvolver estratégias para a disciplina, com foco na resolução de problemas, as habilidades propostas pelo Pensamento Computacional (PC) apresentaram-se promissoras.

O PC, definido por Wing (2006) como um conjunto de estratégias da Ciência da Computação que auxilia na resolução de problemas em diversas

áreas, vem sendo apontado como uma das habilidades fundamentais para o pleno exercício da cidadania no século XXI (OECD, 2022).

Além disso, pesquisas como a realizada por Canal (2021) sugerem sua contribuição com a resolução de problemas matemáticos.

Ademais da resolução de problemas, no contexto da investigação que resultou no PE, ele surge como uma ferramenta com potencial de aproximar a MD da área técnica do curso, auxiliando na construção de uma proposta que atenda os pressupostos da Andragogia, Knowles (1968) e da Aprendizagem Significativa, Ausubel (1968) que foram balizadores das intervenções.

O PRODUTO EDUCACIONAL

Nessa seção será descrito o produto educacional desenvolvido, desde sua idealização até sua implementação.

Conceitos e elaboração

O PE foi pensado com o olhar voltado para o estudante, adulto, de um curso da área tecnológica. Desta forma, além de buscar promover a aprendizagem significativa de conceitos interdisciplinares entre MD e programação, visa fomentar a autonomia e o engajamento dos estudantes por meio de uma ferramenta interativa.

A escolha da ferramenta (aplicativo) reflete um esforço em alinhar a prática pedagógica às demandas contemporâneas, ao mesmo tempo em que responde às especificidades do curso e à carência de recursos didáticos adequados.

Podemos definir uma ferramenta interativa como um recurso tecnológico criado para mediar e facilitar a comunicação ativa entre o usuário e o conteúdo com o qual ele interage. Assim, destaca-se como mediadora de práticas pedagógicas, capazes de engajar e conectar os aprendizes em um cenário permeado por tecnologias digitais, mesmo que o acesso e o grau de interação possam variar entre diferentes contextos sociais (GARCIA et al., 2011).

Para elaboração do PE, a pesquisadora analisou, qualitativamente, durante as intervenções, o impacto de questões que relacionam os conteúdos da disciplina de MD com a área técnica do curso na construção da aprendizagem dos estudantes.

Dessa forma, foram selecionados os textos bases e as questões (anexo) que serviram como ponto de partida para o desenvolvimento do aplicativo.

Valorizando o protagonismo dos estudantes, e as experiências vivenciadas durante a pesquisa, optou-se por desenvolver o produto por meio da programação em blocos (através da ferramenta App Inventor). A ferramenta desenvolvida pelo Instituto de Tecnologia de Massachusetts (MIT) é um ambiente virtual de programação intuitiva, e gratuita, criada para iniciantes na área.

O App Inventor foi apresentado e explorado pelos estudantes, durante as aulas da disciplina de MD, para o desenvolvimento de aplicativos propostos e projetados por eles.

Sendo assim, a turma familiarizou-se com as possibilidades de implementação e o ambiente de programação e, através de uma análise coletiva das questões, pode propor os suportes a serem ofertados para fomentar a construção da aprendizagem.

A implementação da ferramenta interativa foi realizada por um dos sujeitos da pesquisa, sob a orientação da pesquisadora.

Na sequência, o aplicativo desenvolvido como PE resultante da dissertação de mestrado será apresentado.

Identificação

Título: TECHMATH: Ferramenta Interativa para a Aprendizagem de Matemática Discreta

Área de Conhecimento: Ensino

Modalidade: Tecnológico

Público-alvo: Estudantes de cursos na área tecnológica (como TADS)

Palavras-chave: Matemática Discreta, Pensamento Computacional, Programação, Recurso Educacional Aberto (REA)

Objetivos

- Promover a aprendizagem significativa de conceitos de Matemática Discreta através de uma proposta interdisciplinar;
- Desenvolver as habilidades do Pensamento Computacional para resolução de problemas;
- Oferecer suporte pedagógico por meio de dicas e inteligência artificial;
- Fomentar a autonomia e o engajamento dos estudantes.

Funcionalidades

- Questões Interdisciplinares: problemas que integram conceitos de matemática discreta e lógica de programação.
- Dicas de Pensamento Computacional: orientações para auxiliar na resolução de problemas.
- Suporte Matemático: explicações teóricas e práticas sobre os conceitos envolvidos.
- Auxílio com Inteligência Artificial: Ferramenta que fornece feedback em tempo real sobre as respostas.
- Roteiro para Desenvolvimento de Programas: orientações para criar soluções computacionais baseadas nos problemas apresentados.
- Feedback instantâneo, sinalizando se a resposta está correta ou incorreta.

Fundamentação Teórica

O aplicativo foi fundamentado nos princípios da aprendizagem significativa de Ausubel (1968), que valoriza a construção do conhecimento a partir da integração de novos conceitos aos saberes pré-existentes, e na andragogia de Knowles (1968) que reconhece as especificidades da aprendizagem em adultos.

O pensamento computacional, Wing (2006) foi incorporado como uma competência essencial para resolver problemas de forma sistemática e criativa, constituindo-se o principal articulador entre os conceitos matemáticos e a área do curso.

Desenvolvimento

O "TECHMATH" foi desenvolvido em parceria entre a pesquisadora e um estudante utilizando o App Inventor. As etapas incluíram:

- Coleta de feedback dos estudantes durante as aulas.
- Prototipação do design do aplicativo.
- Implementação e testes de usabilidade.

Instruções de Uso

- Acesso: O aplicativo estará (após validação da banca) disponível em repositório educacional aberto (REA) e o link para realizar o download pelo link será disponibilizado neste documento. Enquanto isso, pode ser acessado a partir do qrcod a seguir.



- Instalação: Baixe o arquivo APK e instale em dispositivos Android.
- Utilização: Navegue pelas questões e utilize os recursos de dicas, suporte matemático e inteligência artificial para aprimorar seu aprendizado.

Interfaces

Na sequência será apresentada as interfaces de algumas das telas disponíveis no aplicativo. Elas seguem um padrão, e as questões abordadas nas demais podem ser acessadas pelo qrcode ou acompanhadas no texto em anexo.

Na figura 1 está representada a tela de acesso (início do app).

Figura 1 – Tela de acesso do TECHMATH



Fonte: Elaborada pela pesquisadora (2024).

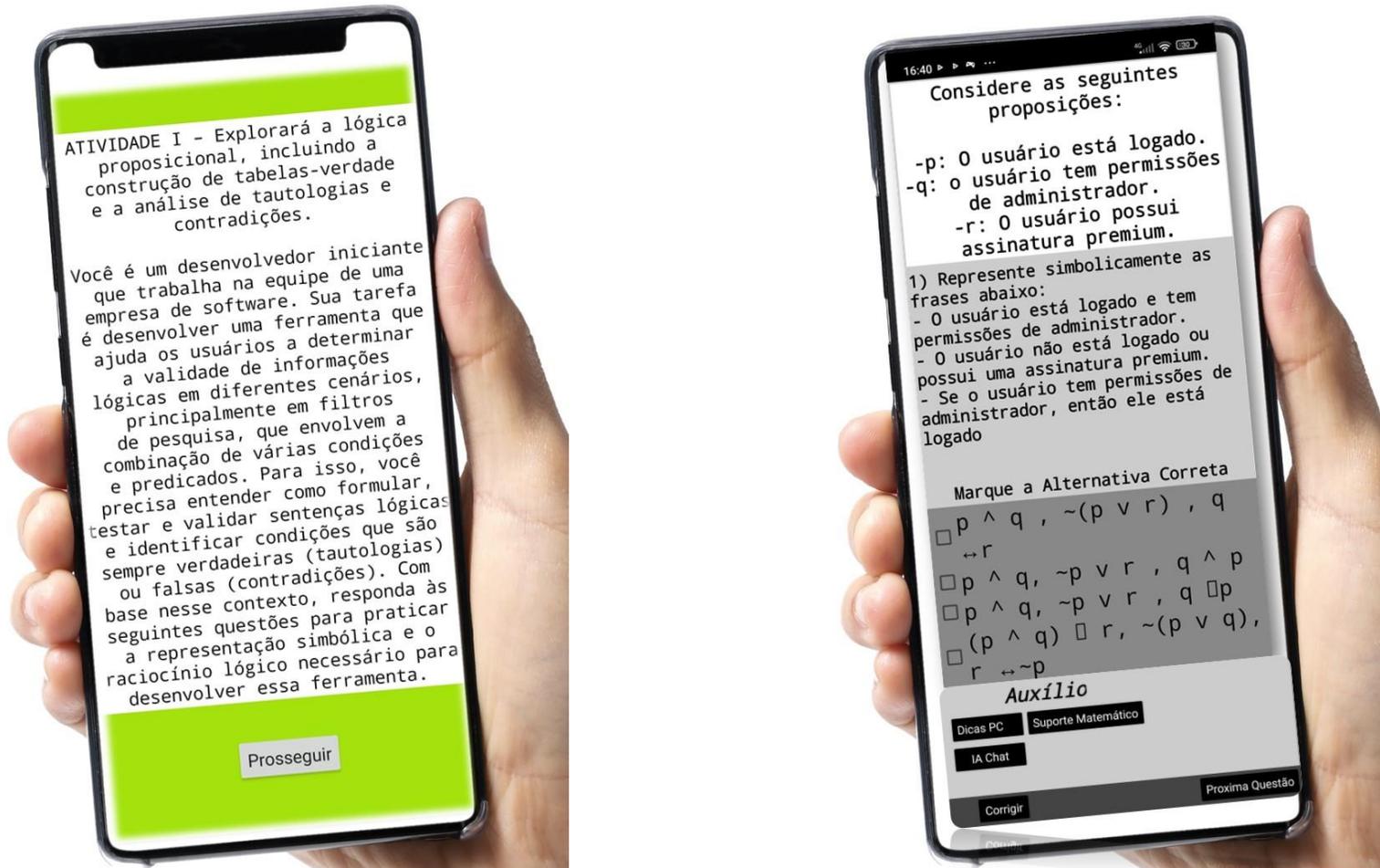
Ao clicar em “Iniciar” o usuário começa a navegar pelas questões que vêm antecedidas por uma contextualização e sucedidas por auxílios para a construção do aprendizado.

O estudante tem acesso à questões que poderão ser respondidas e corrigidas (feedback) diretamente, ou solicitar dicas (opcional) e, quando desejar, avançar para a próxima questão.

Desta forma, a interação se torna personalizada, dependendo das escolhas e necessidades individuais de cada aluno.

A Figura 2 apresenta a tela de contextualização e o texto para a questão 1, respectivamente.

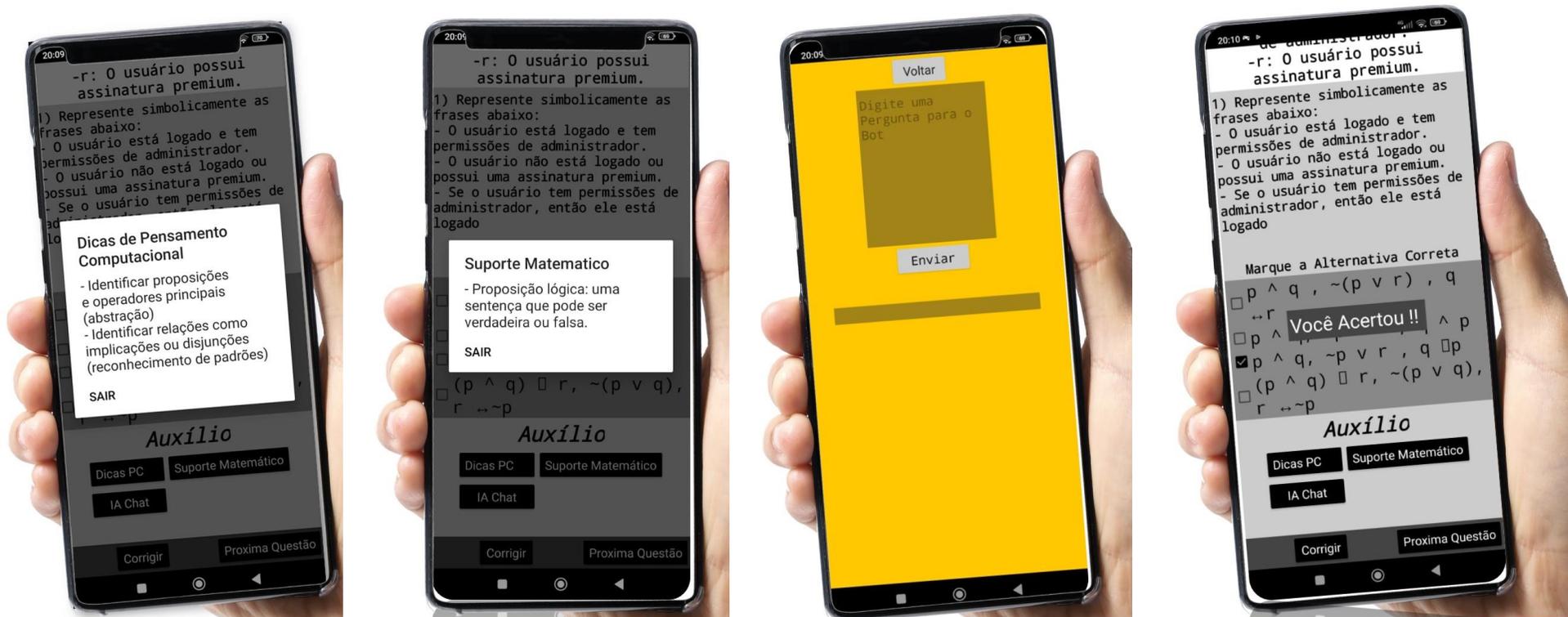
Figura 2 – Telas informativas sobre a questão 01



Fonte: Elaborada pela pesquisadora (2024).

Após ter acesso à questão, clicando em Dicas PC, Suporte Matemático, IA Chat e Corrigir, o estudante tem acesso, respectivamente, as telas representadas na Figura 3.

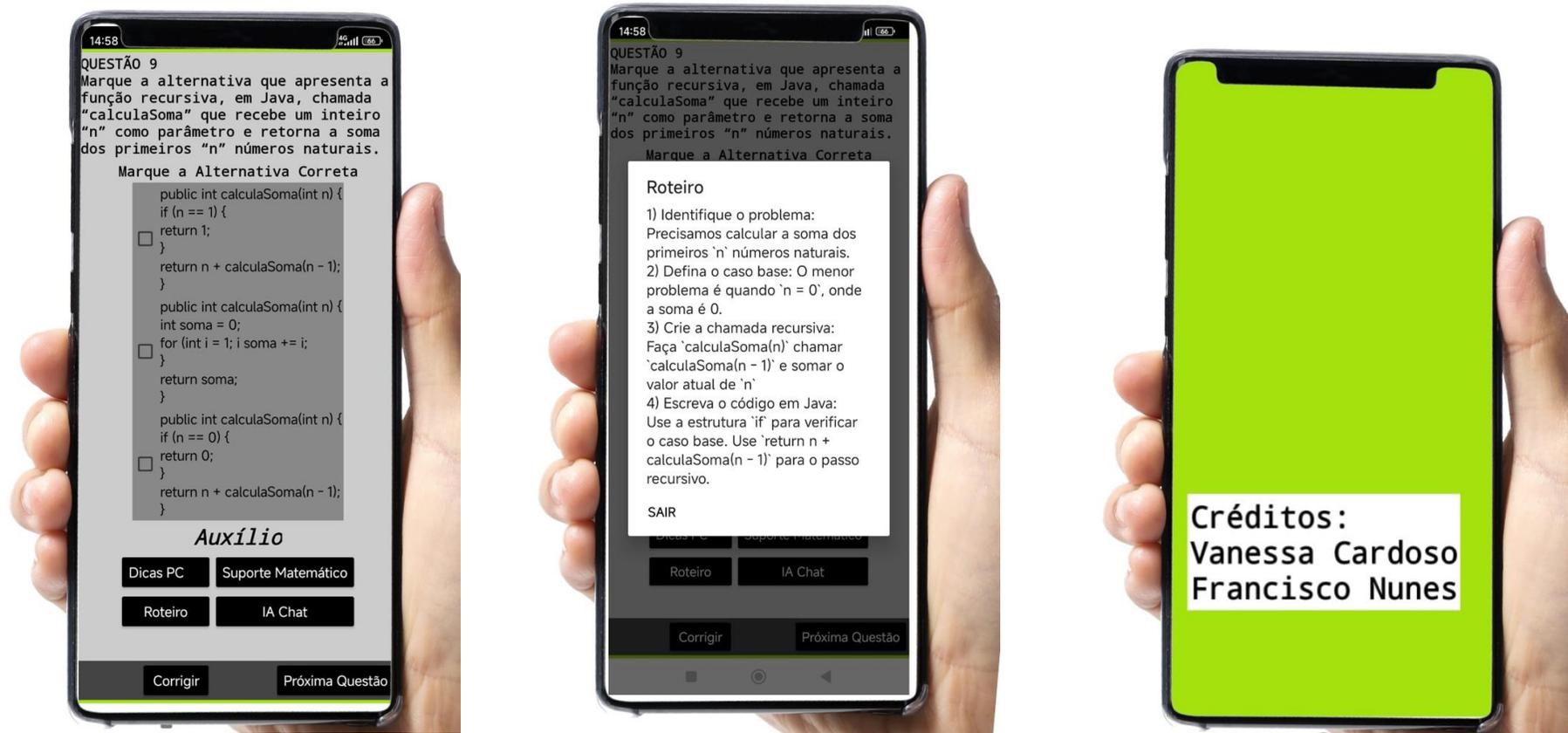
Figura 3 – Telas referentes ao suporte à questão 01



Fonte: Elaborada pela pesquisadora (2024)

A Figura 4 apresenta a tela de uma questão que envolve programação, o comando adicional para esse tipo de questão (roteiro para implementação) e a tela final do aplicativo, respectivamente.

Figura 4 – Telas do aplicativo questão 9 e créditos



Fonte: Elaborada pela pesquisadora (2024)

Ao todo o aplicativo conta com 14 questões, que abordam os conceitos de: Lógica Proposicional, Tabela Verdade, Teoria dos Conjuntos, Funções em linguagem de programação, Recursão e Análise Combinatória.

CONSIDERAÇÕES FINAIS

O produto educacional TECHMATH, desenvolvido como requisito parcial para obtenção do título de mestre no Programa de Pós-Graduação em Ensino de Ciências e Matemática da Universidade Federal de Pelotas, apresenta-se como uma ferramenta promissora para o ensino de Matemática Discreta.

Durante a fase de intervenções, observou-se que a contextualização dos conceitos matemáticos com aplicações práticas da área tecnológica contribuiu significativamente para o engajamento e a compreensão dos estudantes.

A integração entre Matemática Discreta e Pensamento Computacional revelou-se uma estratégia eficaz para o desenvolvimento de competências interdisciplinares, especialmente na resolução de problemas complexos.

O TECHMATH não apenas facilita a aprendizagem significativa por meio de questões interativas e contextualizadas, mas também fomenta a autonomia dos estudantes, permitindo que explorem conceitos teóricos e práticos em um ambiente tecnológico intuitivo e acessível.

A aplicação de recursos como dicas, suporte matemático e inteligência artificial reforça o caráter inovador da ferramenta.

O TECHMATH foi projetado para atender às demandas contemporâneas de ensino, alinhando-se às diretrizes metodológicas e teóricas da linha de pesquisa em estratégias educacionais do PPGECM.

Por ser disponibilizado como um Recurso Educacional Aberto, ele democratiza o acesso ao conhecimento, e visa promover impacto positivo tanto na formação dos estudantes quanto na prática pedagógica dos educadores.

Acredita-se que a ferramenta apresenta potencial para transformar a abordagem do ensino de Matemática Discreta, e pode se tornar uma contribuição relevante para a área de ensino de matemática, mas especificamente, da Discreta.

REFERÊNCIAS

AUSUBEL, David P. **Educational psychology: a cognitive view**. Nova York: Holt, Rinehart and Winston, 1968.

CANAL, Ana Paula. **Pensamento Computacional articulado à Resolução de Problemas no ensino para formação inicial de professores de matemática: uma abordagem a partir da Teoria de Robbie Case**. 2021. 319f. Tese (Programa de Pós-Graduação em Ensino de Ciências e Matemática) - Universidade Franciscana, Santa Maria - RS.

CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). **Ensino: orientações para elaboração de propostas de cursos novos (APCN)**. <https://www.gov.br/capes/pt-br/centrais-de-conte/documentos/avali/E.pd>. Acesso em: 8 dez. 2024.

KNOWLES, M. S. **Andragogy, not pedagogy**. *Adult Leadership*, v. 16, n. 10, p. 350-352, 386, 1968.

GARCIA, Roberta V.; CORTEZ, Anne C. C. R. **Utilização de atividades de Pensamento Computacional aplicado à resolução de problemas nas aulas de Matemática**. Revista Interdisciplinar de Tecnologias na Educação [RINTE], v. 9, n. 1, Ed. Especial: IX SEC - Simpósio de Ensino de Ciências, 2023, p. 01-18.

OECD (Organização para a Cooperação e Desenvolvimento Econômico). **PISA 2022 - Matemática**. Disponível em: <https://pisa2022-maths.oecd.org/pt/index.html>. Acesso em: 14 dez. 2023.

OLIVEIRA, Carla. **Mas afinal, o que é essa tal de Matemática Discreta?** Artigo publicado em 26 de junho de 2022. Disponível em: <https://pt.linkedin.com/pulse/mas-afinal-o-que-%C3%A9-essa-tal-de-matem%C3%A1tica-discreta-carla-oliveira>. Acesso em: 13 jan. 2023.

PARMAR, M. **Matemáticas discretas: la columna vertebral de la informática**. Revista Electrónica, v. 4, n. 5, p. 52, 2019. SSRN.

WING, Jeannette M. **Computational thinking**. Communications of the ACM, [S.l.], v. 49, n. 3, p. 33–35, 2006. Disponível em: <http://portal.acm.org/citation.cfm?doid=1118178.1118215>. Acesso em: 18 mar. 2019.

ANEXO – SEQUÊNCIA DIDÁTICA IMPLEMENTADA NO TECHMATH

Tela 1 - anterior as questões 1, 2 e 3 (tipo uma apresentação)

ATIVIDADE I – Explorará a lógica proposicional, incluindo a construção de tabelas-verdade e a análise de tautologias e contradições.

Você é um desenvolvedor iniciante que trabalha na equipe de uma empresa de software. Sua tarefa é desenvolver uma ferramenta que ajuda os usuários a determinar a validade de informações lógicas em diferentes cenários, principalmente em filtros de pesquisa, que envolvem a combinação de várias condições e predicados. Para isso, você precisa entender como formular, testar e validar sentenças lógicas e identificar condições que são sempre verdadeiras (tautologias) ou falsas (contradições). Com base nesse contexto, responda às seguintes questões para praticar a representação simbólica e o raciocínio lógico necessário para desenvolver essa ferramenta.

Tela 2 – QUESTÃO 1

Considere as seguintes proposições:

- p: O usuário está logado.
- q: o usuário tem permissões de administrador.
- r: O usuário possui assinatura premium.

Represente simbolicamente as frases abaixo:

- i- O usuário está logado e tem permissões de administrador.
- ii- O usuário não está logado ou possui uma assinatura premium.
- iii- Se o usuário tem permissões de administrador, então ele está logado.

ALTERNATIVAS

- () $p \wedge q, \sim(p \vee r), q \leftrightarrow r$
- () $p \wedge q, \sim p \vee r, q \wedge p$
- () $p \wedge q, \sim p \vee r, q \rightarrow p$
- () $(p \wedge q) \rightarrow r, \sim(p \vee q), r \leftrightarrow \sim p$

SUPORTE

DICAS PC

- Identificar proposições e operadores principais (**abstração**)
- Identificar relações como implicações ou disjunções (**reconhecimento de padrões**)

MATEMÁTICA

- **Proposição lógica**: uma sentença que pode ser verdadeira ou falsa.

TELA 3 – QUESTÃO 2

Analise as proposições representadas acima e determine se alguma delas é uma tautologia ou uma contradição.

ALTERNATIVAS

() **Nenhuma proposição é tautologia ou contradição, pois as tabelas verdades mostram combinações verdadeiras e falsas.**

() Todas as proposições são contradições pois as tabelas verdades mostram combinações verdadeiras e falsas.

() Todas as proposições são tautologias já que as tabelas resultam sempre em verdadeiro.

() A proposição i é tautologia e as ii e iii são contradições, de acordo com as tabelas-verdades.

SUPORTE

DICAS PC

- Construção e interpretação de tabelas-verdade (**avaliação**)
- Implementação do código para teste (**pensamento algorítmico**)

MATEMÁTICA

- **Tabela-Verdade:** uma tabela que lista todas as combinações possíveis de valores de verdade para as proposições.
- **Tautologia:** uma proposição que é sempre verdadeira, independentemente dos valores das variáveis.
- **Contradição:** uma proposição que é sempre falsa, independentemente dos valores das variáveis.

ROTEIRO

- 1. Representar as proposições em variáveis booleanas (true/false).
- 2. Criar funções para avaliar conectivos lógicos: AND, OR, NOT, \rightarrow .
- 3. Implementar uma função para gerar tabelas-verdade iterando sobre combinações de valores.
- 4. Avaliar proposições e determinar se são tautologias ou contradições.

EXEMPLO DE CÓDIGO

Algoritmo (Portugol):

Variáveis: p, q, r (booleanas)

Função avaliar Tautologia (proposição):

Entrada: proposição lógica.

Saída: True se tautologia; False caso contrário.

Processo: Gerar tabelas-verdade e verificar se todas as saídas são verdadeiras.

TELA 4 – QUESTÃO 3

Analise as proposições representadas acima e determine se alguma delas é uma tautologia ou uma contradição, justificando sua resposta.

Imagine agora que, além de permissões específicas, há uma condição para todos os usuários que têm acesso a um recurso específico da aplicação. Considere o predicado $A(x)$: x possui acesso ao recurso específico. Utilizando quantificadores, represente as afirmações a seguir:

- Todos os usuários logados possuem acesso ao recurso específico.
- Existe pelo menos um usuário que não possui acesso ao recurso específico.

ALTERNATIVAS

- () $\forall x (p(x) \rightarrow A(x))$ e $\exists x (\sim A(x))$
- () $\forall x (p(x) \wedge A(x))$ e $\forall x (\sim A(x))$
- () $\exists x (p(x) \rightarrow A(x))$ e $\forall x (\sim A(x))$
- () $\forall x (p(x) \leftrightarrow A(x))$ e $\exists x (\sim A(x))$

SUPORTE

DICAS PC

- 1) Identificar proposições, operadores principais e quantificadores (**abstração**)
- 2) Identificar relações como implicações ou disjunções (**reconhecimento de padrões**)

MATEMÁTICA

Quantificadores: ' \forall ' significa 'para todos' e ' \exists ' significa 'existe pelo menos um'

TELA 5 (anterior as questões 4 e 5)

ATIVIDADE II – Abordará os conceitos de Fundamentos da Lógica e Teoria dos Conjuntos.

Você está desenvolvendo um programa em Portugol para uma aplicação que ajuda a organizar dados de clientes de uma empresa de tecnologia. A empresa possui diferentes tipos de clientes, categorizados conforme os produtos e serviços contratados. Para facilitar o processo de consulta e atualização de registros, é necessário compreender as operações de união, interseção, diferença e complemento entre conjuntos. Sua tarefa consiste em representar e

manipular os dados dos clientes utilizando conceitos de teoria de conjuntos, que servirão de base para o desenvolvimento do algoritmo em Portugol.

Considerando as seguintes informações, responda as questões 4 e 5.

- A: Clientes que possuem um contrato de manutenção.
- B: Clientes que possuem um contrato de consultoria.
- C: Clientes que possuem um contrato de software personalizado.

Tela 6 - QUESTÃO 4

De acordo com as informações anteriores, represente simbolicamente as operações abaixo e interprete o que cada conjunto representa no contexto da aplicação:

- Clientes que possuem um contrato de manutenção ou consultoria.
- Clientes que possuem contratos de manutenção e software personalizado.
- Clientes que possuem contrato de manutenção, mas não de consultoria.
- Clientes que não possuem contrato de consultoria.

ALTERNATIVAS:

a) $A \cup B$ (Clientes que possuem um contrato de manutenção ou consultoria)

$A \cap C$ (Clientes que possuem contratos de manutenção e software personalizado)

$A - B$ (Clientes que possuem contrato de manutenção, mas não de consultoria)

B' (Clientes que não possuem contrato de consultoria)

b) $A \cap B$ (Clientes que possuem um contrato de manutenção ou consultoria)

$A \cup C$ (Clientes que possuem contratos de manutenção e software personalizado)

$A - B$ (Clientes que possuem contrato de manutenção, mas não de consultoria)

C' (Clientes que não possuem contrato de consultoria)

c) $A \cup B$ (Clientes que possuem contrato de manutenção ou software personalizado)

$A \cap B$ (Clientes que possuem contratos de manutenção e consultoria)

$A - B$ (Clientes que possuem contrato de manutenção, mas não de software personalizado)

B' (Clientes que não possuem contrato de manutenção).

SUPORTE

DICAS DE PC

- 1) Substitua as informações referentes a cada conjunto pela letra que o representa (**abstração**)
- 2) Compare os operadores união, intersecção, diferença e complementar com sua linguagem usual - e, ou, mas não/menos, não – (**reconhecimento de padrões**)

MATEMÁTICA

União entre dois conjuntos (U) - incluiu todos os elementos que pertencem ao primeiro **ou** ao segundo conjunto.

Intersecção entre dois conjuntos (\cap) – inclui todos os elementos que pertencem ao primeiro **e** também ao segundo conjunto.

Diferença entre dois conjuntos (-) – Inclui todos os elementos que pertencem ao primeiro conjunto **e não** pertencem ao segundo.

Complementar (`) – Inclui todos os elementos que **não** pertencem ao conjunto.

Tela 7 - QUESTÃO 5

Escreva um programa em Portugol que receba informações sobre quais contratos um cliente possui e, com base nessas informações, exiba uma mensagem sobre os serviços que ele pode acessar. O programa deve perguntar ao usuário se ele possui cada tipo de contrato e, depois, exibir a mensagem correspondente.

Requisitos:

1. Pergunte ao usuário se ele possui um contrato de manutenção (A), consultoria (B) e/ou software personalizado (C).
2. Com base nas respostas do usuário, exiba as mensagens:
 - Se o cliente possui apenas o contrato de manutenção, exiba: "Você tem acesso ao serviço de manutenção."
 - Se o cliente possui apenas o contrato de consultoria, exiba: "Você tem acesso ao serviço de consultoria."
 - Se o cliente possui apenas o contrato de software personalizado, exiba: "Você tem acesso ao serviço de software personalizado."
 - Se o cliente possui mais de um contrato, exiba: "Você tem acesso a múltiplos serviços."

Marque a alternativa que apresenta uma possibilidade correta de código

3. Obtenha os dados do usuário: Use comandos de entrada para capturar as respostas (S/N) sobre os contratos.
4. Desenvolva as condições: Comece com as condições mais específicas (apenas um contrato ativo). Depois, implemente a condição para múltiplos contratos.
5. Teste o algoritmo: Verifique todas as combinações possíveis para garantir que o algoritmo funciona corretamente.

Tela 8 (anterior as questões 6,7 e 8)

ATIVIDADE III – Teoria dos conjuntos

Texto para as questões 6,7 e 8: Imagine que você está desenvolvendo um sistema de recomendação para uma plataforma de streaming. Os conjuntos a seguir representam categorias de filmes vistos por dois usuários diferentes em um mês:

$A = \{\text{Ação, Comédia, Drama, Suspense}\}$

$B = \{\text{Aventura, Comédia, Ficção Científica, Drama}\}$

Tela 9 (QUESTÃO 6)

A partir das informações, fornecidas na tela anterior, marque a alternativa que apresenta todos os elementos de $A \cup B$ e a interpretação da união de conjuntos no contexto de recomendação de filmes.

ALTERNATIVAS

() $A \cup B = \{\text{Comédia, Drama}\}$ – A união representa todas as categorias de interesse dos dois usuários, ajudando o sistema a determinar quais são as preferências gerais.

() **$A \cup B = \{\text{Ação, Aventura, Comédia, Drama, Suspense, Ficção Científica}\}$ - A união representa todas as categorias de filmes assistidas por pelo menos um dos dois usuários, ajudando o sistema a recomendar uma maior diversidade de opções.**

() $A \cup B = \{\text{Ação, Aventura, Comédia, Comédia, Drama, Drama Suspense, Ficção Científica}\}$ - A união representa todas as categorias de filmes assistidas pelos dois usuários, ajudando o sistema a recomendar uma maior diversidade de opções, com destaque para as maiores preferências.

SUPORTE

DICAS PC

Reconhecimento de Padrões: Entenda que $A \cup B$ inclui todos os elementos de ambos os conjuntos sem duplicatas.

Abstração: Concentre-se apenas na operação e elimine informações irrelevantes.

Pensamento Algorítmico: Construa métodos para listas os elementos.

MATEMÁTICA

União ($A \cup B$) – Todos os elementos pertencentes a pelo menos um dos conjuntos.

Tela 10 (QUESTÃO 7)

No mesmo sentido da questão 6, marque a alternativa que apresenta todos os elementos de $A \cap B$ e a explicação de como a interseção de conjuntos pode ajudar na recomendação de filmes específicos para ambos os usuários.

ALTERNATIVAS

() $A \cap B = \{\text{Ação, Aventura, Comédia, Suspense, Ficção Científica}\}$ – A interseção mostra as categorias menos assistidas, o que pode ajudar o sistema a sugerir filmes que provavelmente agradarão aos dois.

() $A \cap B = \{\text{Ação, Aventura, Comédia, Drama, Suspense, Ficção Científica}\}$ – A interseção mostra as os gostos dos usuários, o que pode ajudar o sistema a sugerir filmes que provavelmente agradarão aos dois.

() $A \cap B = \{\text{Comédia, Drama}\}$ – A interseção mostra as categorias em comum, o que pode ajudar o sistema a sugerir filmes que provavelmente agradarão aos dois.

SUPORTE

DICAS PC

Reconhecimento de Padrões: Observe que $A \cap B$ inclui apenas elementos presentes em ambos.

Abstração: Concentre-se apenas na operação e elimine informações irrelevantes.

Pensamento Algorítmico: Construa métodos para listas os elementos.

MATEMÁTICA

Interseção ($A \cap B$): Apenas os elementos pertencentes a ambos os conjuntos.

Tela 11 (QUESTÃO 8)

Você precisa programar, em pseudocódigo, uma função que receba dois conjuntos de categorias e retorne à união e à interseção desses conjuntos. Marque a alternativa que apresenta uma solução possível para essa programação.

ALTERNATIVAS

()

```
função calcularUniaoIntersecao(conjuntoA, conjuntoB)
```

```
intersecao <- conjuntoA + conjuntoB
```

```
uniao <- vazio
```

```
retorne (uniao, intersecao)
```

```
fim função
```

() **correta**

```
função calcularUniaoIntersecao(conjuntoA, conjuntoB)
```

```
uniao <- conjuntoA + conjuntoB
```

```
intersecao <- vazio
```

```
para cada elemento em conjuntoA
```

```
  se elemento pertence a conjuntoB
```

```
    adicione elemento em intersecao
```

```
fim para
```

```
retorne (uniao, intersecao)
```

```
fim função
```

SUPORTE

DICAS PC

Abstração – Utilização da linguagem lógica de conjuntos

Pensamento Algorítmico – Estrutura da função

Avaliação – Testes em Portugol para verificar a resposta correta.

Aproveite, também, para propor soluções mais eficientes que levem ao mesmo resultado.

ROTEIRO

1. Definição de Função: Crie uma função que receba dois conjuntos como entrada.

2. Calcular a União: Utilize uma estrutura de dados que permita armazenar todos os elementos de ambos os conjuntos. Remova duplicatas (se necessário, usando listas ou sets).

3. Calcular a Interseção: Percorra os elementos de um conjunto e verifique se cada elemento também pertence ao outro. Adicione os elementos compartilhados em um novo conjunto.

4. Retornar Resultados: Retorne os resultados de ambas as operações em um formato acessível, como uma dupla.

5. Testes: Teste a função com conjuntos de exemplo para validar os cálculos.

Tela 12 (texto para as questões 9 e 10)

ATIVIDADE IV - Função e recursão

Você foi contratado por uma empresa para desenvolver um programa em Java que realiza uma análise de dados para uma aplicação financeira. Um dos requisitos é criar uma função que calcule a soma dos primeiros 'n' números naturais de forma recursiva.

Tela 13 (QUESTÃO 9)

Marque a alternativa que apresenta a função recursiva, em Java, chamada "calculaSoma" que recebe um inteiro "n" como parâmetro e retorna a soma dos primeiros "n" números naturais.

ALTERNATIVAS

()

```
public int calculaSoma(int n) {  
    if (n == 1) {  
        return 1;  
    }  
    return n + calculaSoma(n - 1);  
}
```

()

```
public int calculaSoma(int n) {  
    int soma = 0;  
    for (int i = 1; i <= n; i++) {  
        soma += i;  
    }  
    return soma;  
}
```

() **CORRETA**

```
public int calculaSoma(int n) {  
    if (n == 0) {  
        return 0;  
    }  
    return n + calculaSoma(n - 1);  
}
```

SUPORTE

DICAS PC

Abstração: Identifique o padrão repetitivo na soma dos números naturais e veja como o processo pode ser reduzido a chamadas recursivas.

Reconhecimento de padrões: Entenda a relação entre o número atual e a soma dos números anteriores.

Algoritmos: Estruture o algoritmo para definir claramente o caso base e o processo recursivo.

Avaliação: Realize testes para verificar a estrutura do programa. Aproveite para propor e testar, alternativas mais eficientes de solução.

MATEMÁTICA

A fórmula da soma dos primeiros n números naturais é $S = n(n+1)/2$, mas o objetivo aqui é implementar isso por meio de uma abordagem recursiva,

com a soma se reduzindo a subproblemas menores até alcançar o caso base (0). A fórmula pode ser utilizada para testar o programa.

ROTEIRO

1. Identifique o problema: Precisamos calcular a soma dos primeiros n números naturais.
2. Defina o caso base: O menor problema é quando $n = 0$, onde a soma é 0.
3. Crie a chamada recursiva: Faça `calculaSoma(n)` chamar `calculaSoma(n - 1)` e somar o valor atual de n .
4. Escreva o código em Java: Use a estrutura `if` para verificar o caso base. Use `return n + calculaSoma(n - 1)` para o passo recursivo.

Tela 14 (QUESTÃO 10)

Marque a alternativa que explica corretamente como a função recursiva funciona, detalhando o processo de chamadas e retornos de cada etapa até alcançar o caso base. Como suporte, use as dicas oferecidas na questão 9.

ALTERNATIVAS

- () A função `calculaSoma` começa somando o número n e entra em um loop infinito.
- () **A função verifica se n é igual a zero (caso base). Se for, retorna 0. Caso contrário, retorna $n + \text{calculaSoma}(n - 1)$, somando progressivamente até atingir o caso base.**
- () A função soma diretamente todos os números de 1 até n progressivamente.
- () A função apenas retorna o valor de n , somado ao caso base.

Tela 15 (texto para as questões 11 e 12)

ATIVIDADE V – Função, Recursão e Análise Combinatória

Em programação, a recursão é uma técnica essencial para resolver problemas em que uma função é chamada repetidamente dentro de si mesma. No contexto de análise de sistemas, a recursão pode ser usada para resolver problemas que possuem subproblemas menores de natureza semelhante. Essa questão explora o cálculo de fatorial de um número, uma função comumente usada na análise de algoritmos e em estruturas de programação.

Tela 16 (QUESTÃO 11)

Você precisa escrever um pseudocódigo da função recursiva para calcular o fatorial de um número inteiro não-negativo n ($n!$). A função deve receber n como entrada e retornar o valor de $n!$. Lembre-se de que, por definição, o fatorial de 0 ($0!$) é igual a 1.

Diante disso, marque a alternativa que apresenta um pseudocódigo que atende a solicitação:

ALTERNATIVAS

Correta

Função Fatorial(n):

Se $n == 0$:

Retorne 1

Senão:

Retorne $n * \text{Fatorial}(n - 1)$

Função Fatorial(n):

Se $n \leq 1$:

Retorne 1

Senão:

Retorne $\text{Fatorial}(n) * (n - 1)$

Função Fatorial(n):

Enquanto $n > 0$:

Retorne $n * \text{Fatorial}(n - 1)$

Retorne 1

SUPORTE

DICAS PC

Reconhecimento de Padrões: Identificar o padrão de multiplicação sucessiva.

Abstração: Implementar a recursão como um conceito genérico para resolver problemas similares.

Pensamento Algorítmico: Criação de um passo a passo lógico para resolver o problema.

Avaliação: Avalie a solução proposta e tente desenvolver uma alternativa que possa ser mais eficaz para a situação acima.

MATEMÁTICA

O cálculo do fatorial é uma aplicação direta da multiplicação sucessiva de inteiros positivos até 1. Matematicamente, $n! = n \times (n-1) \times (n-2) \times \dots \times 1$, com $0! = 1$ como caso base.

ROTEIRO

Passo 1: Entendimento do Problema: O fatorial de um número n é definido como o produto de todos os números inteiros positivos menores ou iguais a n . Por convenção, o fatorial de 0 ($0!$) é igual a 1. Assim, o problema é calcular $n! = n \times (n-1) \times (n-2) \times \dots \times 1$.

Passo 2: Identificar o Caso Base: A recursão exige um caso base que interrompa a sequência de chamadas. Para o cálculo do fatorial: Caso base: Quando $n = 0$, o valor retornado deve ser 1.

Passo 3: Definir a Regra Recursiva O cálculo do fatorial para $n > 0$ pode ser reduzido ao cálculo do fatorial de $n-1$. Ou seja, $n! = n \times (n-1)!$.

Passo 4: Esboçar o Pseudocódigo Com base no caso base e na regra recursiva, o pseudocódigo deve ser estruturado.

Passo 5: Teste: Testar o pseudocódigo com diferentes valores de entrada para verificar sua funcionalidade.

Tela 17 (QUESTÃO 12)

Assinale a alternativa que descreve como a recursão funciona nesse caso, mencionando a chamada da função para subproblemas menores até chegar à condição de parada.

ALTERNATIVAS

- A função recursiva chama a si mesma com um valor de entrada reduzido, resolvendo subproblemas menores até atingir a condição de parada ($n == 0$), onde retorna o valor 1. O resultado final é obtido multiplicando os retornos das chamadas recursivas.**
- A função executa um loop até que o valor de n seja 0. Durante cada iteração, ela acumula o resultado multiplicando o valor atual de n , ignorando a chamada da função para valores menores.
- A função usa uma pilha de chamadas onde cada chamada é armazenada até atingir a condição de parada ($n == 0$). Depois, os valores retornados são multiplicados em ordem inversa.

() A função evita chamadas recursivas adicionais ao resolver diretamente o problema para qualquer valor de n, retornando 1 independentemente do valor de entrada.

SUPORTE

MATEMÁTICA

A recursão se baseia na definição matemática do fatorial: um problema é reduzido a si mesmo até atingir um caso base.

Tela 18 (texto base para as questões 13 e 14)

ATIVIDADE VI – Análise Combinatória

Uma empresa de tecnologia especializada em logística contratou você para desenvolver um sistema que analise a quantidade de maneiras possíveis de selecionar um grupo de representantes de um total de funcionários. O sistema precisa ser flexível para calcular quantas formas distintas é possível escolher diferentes quantidades de funcionários a partir de um total dado.

Tela 19 (QUESTÃO 13)

Você precisa Implementar uma função em Java chamada “calculaSelecao” que recebe dois inteiros, “totalFuncionarios” e “grupo”, e retorne o número de maneiras distintas de selecionar o grupo de funcionários, considerando o total fornecido. Marque a alternativa que atende a essa solicitação.

ALTERNATIVAS

() **correta**

```
public static long fatorial(int n) {  
    long resultado = 1;  
    for (int i = 1; i <= n; i++) {  
        resultado *= i;  
    }  
    return resultado;  
}  
  
public static long calculaSelecao(int totalFuncionarios, int grupo) {
```

```
        return fatorial(totalFuncionarios) / (fatorial(grupo) * fatorial(totalFuncionarios -
grupo));
    }
}
```

()

```
    public static long fatorial(int n) {
        long resultado = 1;
        for (int i = 1; i <= n; i++) {
            resultado *= i;
        }
        return resultado;
    }
}
public static long calculaSelecao(int totalFuncionarios, int grupo) {
    return totalFuncionarios * grupo;
}
}
```

()

```
    public static long fatorial(int n) {
        long resultado = 1;
        for (int i = 1; i <= n; i++) {
            resultado *= i;
        }
        return resultado;
    }
}
public static long calculaSelecao(int totalFuncionarios, int grupo) {
    return totalFuncionarios / 2;
}
}
```

SUPORTE

DICAS DE PC

Reconhecimento de Padrões: Identificar a relação entre os cálculos matemáticos e sua implementação em código.

Abstração: a partir da fórmula matemática traduzida para uma função Java.

Pensamento Algorítmico: projeção do algoritmo para calcular o fatorial e aplicar a fórmula de combinação.

Avaliação: Analise a solução proposta e estude a possibilidade de apresentar uma mais eficiente.

MATEMÁTICA

A fórmula de combinação é usada para calcular o número de maneiras de selecionar k elementos de um conjunto de n elementos sem considerar a ordem:

$$C(n, k) = n! / (k! \cdot (n-k)!).$$

Casos Especiais

- $C(n, 0) = 1$: Sempre há exatamente 1 maneira de escolher nenhum elemento.
- $C(n, n) = 1$: Sempre há exatamente 1 maneira de escolher todos os elementos.

ROTEIRO

1. Criação da Função Fatorial
2. Implementação da Função `calculaSelecao`
3. Teste do Código

Tela 20 (QUESTÃO 14)

Qual das alternativas descreve como o código resolve o problema de selecionar os funcionários, destacando o caso em que todos os funcionários são selecionados ou nenhum é escolhido. Use o Suporte da questão 13!

ALTERNATIVAS

- O código retorna 1 tanto para o caso em que todos os funcionários são selecionados ($C(n, n) = 1$) quanto para o caso em que nenhum funcionário é selecionado ($C(n, 0) = 1$).
- O código apresenta um erro ao tentar calcular $C(n, 0)$ ou $C(n, n)$, pois esses casos não estão definidos matematicamente.
- O código retorna $n!$ para o caso de $C(n, n)$ e 0 para $C(n, 0)$.