

JACKSON W. PLUSKOTA

JOÃO PAULO AIRES

MARIA JOÃO VARANDA PEREIRA

Metodologias ativas



Estratégias de ensino em
Programação

Metodologias ativas

Estratégias de ensino em
programação

Jackson William Pluskota

João Paulo Aires

Maria João Varanda Pereira

Metodologias ativas



Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

Importante: Muito cuidado e técnica foram empregados na edição deste livro. No entanto, não estamos livres de pequenos erros de digitação, problemas na impressão ou de alguma dúvida conceitual. Avise-nos por e-mail:

jackson.pluskota@gmail.com

Jackson William Pluskota

João Paulo Aires

Maria João Varanda Pereira

Metodologias ativas

Estratégias de ensino em
programação

Ponta Grossa
2023

**Aprender exige curiosidade. Aprender
verdadeiramente é uma aventura
emocionante e desafiadora**

Freire (2001).

SUMÁRIO

APRESENTAÇÃO DO PRODUTO	9
INTRODUÇÃO	11
METODOLOGIAS ATIVAS	15
ESTRUTURA DE DECISÃO SIMPLES E COMPOSTA: ESTRATÉGIAS E APLICAÇÃO NA SALA DE AULA.....	19
Estrutura de Decisão Simples e Composta: Estratégias e Aplicação na Sala de Aula Invertida.....	21
Estrutura de Decisão Simples e Composta: Estratégias e Aplicação na PBL	26
Estrutura de decisão simples e composta: Estratégias e aplicação na gamificação.....	39
Estrutura de Decisão Simples e Composta: Estratégias e Aplicação na TBL	47
MODULARIZAÇÃO: ESTRATÉGIAS E APLICAÇÃO NA SALA DE AULA INVERTIDA	54
Modularização: Estratégias e Aplicação na Sala de Aula Invertida	56
Modularização: Estratégias e Aplicação na PBL	62
Modularização: Estratégias e Aplicação de na Gamificação.....	67
Modularização: Estratégias e Aplicação de na TBL	75
ESTRUTURAS DE REPETIÇÃO: ESTRATÉGIAS E APLICAÇÃO NA SALA DE AULA INVERTIDA	87
Estruturas de Repetição: Estratégias e Aplicação na Sala de Aula Invertida	89
Estruturas de Repetição: Estratégias e Aplicação do PBL	94
Estruturas de Repetição: Estratégias e Aplicação na Gamificação.....	102
Estruturas de repetição: Estratégias e Aplicação na Estratégia TBL	111
VETORES: ESTRATÉGIAS E APLICAÇÃO COM METODOLOGIAS ATIVAS	116

Vetores: Estratégias e Aplicação na Sala de Aula Invertida	118
Vetores: Estratégias e Aplicação na Aprendizagem Baseada em Problemas	122
Vetores: Estratégias e Aplicação na Gamificação	129
Vetores: estratégias e Aplicação na Aprendizagem Baseada em Times .	136
MATRIZES: ESTRATÉGIAS E APLICAÇÃO COM METODOLOGIAS ATIVAS	141
Matrizes: Estratégias e Aplicação de na Sala de Aula Invertida	143
Matrizes: Estratégias e Aplicação na aprendizagem baseada em problemas	153
Matrizes: Estratégias e Aplicação na Gamificação	161
Matrizes: Estratégias e Aplicação de na TBL.....	169
CONSIDERAÇÕES.....	174
REFERÊNCIAS.....	176

AGRADECIMENTOS

Agradeço a minha família por acreditarem em mim e em meus irmãos. Principalmente a meus pais Jacksane e Claudinei, por terem se esforçado para sempre me manter na escola e sempre me ensinar que a educação é o melhor caminho para alcançar nossos objetivos;

Agradeço meus irmãos Jean e Jennifer, pelas conversas e apoio.

Agradeço a minha esposa Thais, por todos os conselhos conversas e apoio nos momentos de dificuldades.

E em memória a minha vó, por tudo que ela me ensinou, que vou levar para minha vida.

Agradeço a meu professor orientador pela oportunidade e, por ter acreditado em nossos projetos e a meus professores que ajudaram durante as dúvidas que tive ao longo do mestrado.

APRESENTAÇÃO DO PRODUTO

Esta produção é derivada da dissertação intitulada “ESTRATÉGIAS DE ENSINO COM METODOLOGIAS ATIVAS EM CONTEÚDOS DE PROGRAMAÇÃO PARA ALUNOS INGRESSANTES”, apresentada ao Mestrado em Ensino de Ciência e Tecnologia, da Universidade Tecnológica Federal do Paraná - Campus Ponta Grossa, desenvolvida pelo estudante de mestrado Jackson William Pluskota, sob orientação do Professor Dr. João Paulo Aires, e coorientação da Professora Dra. Maria João Varanda Pereira.

Desta forma, é com satisfação que apresentamos a vocês um documento abrangente contendo planos de aula detalhados com estratégias utilizando metodologias ativas para o ensino de programação. As estratégias, são destinadas ao uso em disciplinas ofertadas para alunos ingressantes, que terão seu primeiro contato com programação.

Neste material, exploramos cuidadosamente uma variedade de conceitos fundamentais, incluindo estruturas de decisão, estruturas de repetição, matrizes, vetores e modularização.

O objetivo principal desse documento é proporcionar que a disciplina ofertada promova um ambiente de aprendizado dinâmico e engajador, no qual os alunos possam adquirir conhecimentos em programação enquanto se envolvem ativamente no processo de aprendizagem. Para alcançar esse objetivo, são utilizadas metodologias ativas de ensino, como o Aprendizado Baseado em Problemas (PBL), a Aprendizagem Baseada em Tarefas (TBL), a Gamificação e a Sala de Aula Invertida.

O Aprendizado Baseado em Problemas (PBL) é uma abordagem na qual os alunos são desafiados a resolver problemas do mundo real, aplicando os conceitos e as técnicas de programação diante de um problema proposto pelo professor. Ao trabalharem em equipe, eles são incentivados a buscar soluções, aprofundar sua compreensão dos tópicos e desenvolver habilidades de pensamento crítico.

A Aprendizagem Baseada em Times (TBL) é uma metodologia que promove a aprendizagem ativa por meio de atividades práticas em duplas. Neste documento, propomos uma série de tarefas que os alunos devem realizar, com

o objetivo de aprimorar suas habilidades de programação e consolidar os conceitos aprendidos.

A Gamificação é uma estratégia que introduz elementos de jogos no processo de aprendizagem, tornando-o mais envolvente e motivador. Por meio de desafios, recompensas e competições amigáveis, os alunos são incentivados a se dedicarem e a explorarem a programação de forma divertida e interativa.

A Sala de Aula Invertida é uma abordagem em que os alunos adquirem conhecimento prévio por meio de recursos online, como vídeos e materiais de leitura, antes das aulas presenciais. Durante as aulas, o tempo é dedicado a atividades práticas, discussões e esclarecimentos de dúvidas. Essa metodologia permite uma melhor utilização do tempo em sala de aula e incentiva a participação ativa dos alunos.

É importante ressaltar que o material desenvolvido possui uma estrutura lógica e progressiva, facilitando a assimilação dos conceitos por parte dos alunos. São apresentadas 25 estratégias de ensino, sendo quatro estratégias diferentes para cada um dos conteúdos abordados (decisão, repetição, modularização, vetores e matrizes). Além disso, foi elaborado com base em práticas de ensino de programação em consonância com as diretrizes curriculares vigentes.

Acreditamos que esse documento servirá como um guia para os professores, para aplicação de metodologias de ensino (e atividades) diferenciadas em sala de aula. Tais atividades, proporcionarão uma base adequada para o ensino de programação aos alunos ingressantes, preparando-os para os desafios futuros e despertando seu interesse por essa área tão relevante. A flexibilidade e adaptabilidade do material permitem que ele seja ajustado de acordo com as necessidades e particularidades de cada disciplina, garantindo uma experiência de aprendizado personalizada e eficaz.

Estamos ansiosos para receber seus comentários e sugestões sobre o material apresentado, a fim de aprimorá-lo e torná-lo ainda mais adequado às demandas dos alunos e das instituições de ensino.

Atenciosamente

Jackson William Pluskota

João Paulo Aires

Maria João Varanda Pereira

INTRODUÇÃO

Metodologias ativas são abordagens de ensino que colocam o estudante no centro do processo de aprendizagem, incentivando a participação ativa e a construção do conhecimento de forma colaborativa. Essas metodologias buscam estimular o pensamento crítico, a resolução de problemas e a criatividade, ao mesmo tempo em que promovem a autonomia do estudante e a sua capacidade de tomar decisões e buscar informações (MAZUIM; GOMES, 2016; SILVA, 2017; RIBEIRO, 2019; RABELO, 2019).

Algumas das metodologias ativas mais conhecidas incluem: aprendizagem baseada em problemas, sala de aula invertida, aprendizagem cooperativa, gamificação, entre outras (SILVA, 2017). Cada uma delas possui características próprias e pode ser aplicada em diferentes contextos educacionais, desde o ensino básico até o ensino superior. O objetivo geral de sua utilização, é proporcionar uma aprendizagem mais significativa e engajadora para os estudantes.

Desta forma, as metodologias ativas, perpassam diversos passos para serem aplicadas, desde o exame de sado, para que o professor possa aproximar-se da realidade dos estudantes, até a passos como a identificação dos problemas a serem trabalhados em sala de aula, brainstorming, como no caso da ABP. Para que a partir destas informações, o professor, possa delimitar quais possíveis metodologias ativas, melhor de adequarão as demandas dos estudantes.

Como no caso, exemplificado, do problema de reprovações e evasões de estudantes dos cursos de programação. Aos quais, remetem a uma taxa de 44% a 51% de desistência nos cursos de Licenciatura, bacharelado e tecnologia da computação. Certamente, que, estas desistências perpassam diversas razões, tais como dificuldade de familiarização com a disciplina e questões de aprendizagem anteriores ao curso de graduação (HOED, 2016). No entanto, estas compreensões somente se darão, pelo estudo prévio do professor em relação a disciplina, ao curso e aos estudantes.

Por questões como estas, que se aponta as potencialidades das metodologias ativas, as quais em geral, demandarão do professor e dos

estudantes a compreensão de que, esta metodologia parte da aproximação da realidade, análise das demandas e do desenvolvimento das aulas, baseadas nelas.

Neste sentido, seguindo as etapas de cada metodologia ativa que melhor se adequa das demandas dos estudantes, é possível fomentar o ensino de programação, pois essa área do conhecimento exige muito mais do que a simples memorização de conceitos e comandos. É necessário que os estudantes desenvolvam habilidades práticas, tais como a resolução de problemas, a criatividade, a capacidade de trabalhar em equipe e a persistência na busca de soluções, favorecendo a tomada de decisão do futuro profissional.

Algumas das metodologias ativas que podem ser aplicadas no ensino de programação inclui, a Aprendizagem Baseada em Problemas (ABP), no qual os estudantes podem ser desafiados a criar um projeto prático utilizando os conceitos e habilidades aprendidas em aula. Esse projeto pode envolver a criação de um aplicativo, um jogo ou até mesmo um site. Dessa forma, os estudantes podem aplicar o conhecimento, adquirido em sala de aula, de forma prática e contextualizada.

Além da ABP, tem-se a Sala de Aula Invertida. Nesta abordagem, o professor disponibiliza materiais (notas de aula, videoaulas, infográficos, apresentações, entre outros) e atividades online para que os estudantes possam estudar em casa. Em sala de aula, o professor tira dúvidas sobre o conteúdo previamente estudado, por meio de perguntas e, os estudantes, podem trabalhar em projetos e exercícios práticos, sob a supervisão do professor.

Adicionalmente, tem-se a metodologia da Gamificação, que pode ser utilizada para tornar o aprendizado (em especial, o de programação) mais divertido e envolvente. Os estudantes podem ser desafiados a completar missões e a acumular pontos, à medida em que desenvolvem habilidades e conhecimentos acerca do conteúdo trabalhado. Destaca-se que, a Gamificação, usa estratégias de jogos (fases, missões, pontos, evolução do personagem), em uma atividade que não é um jogo.

Por esta razão, considerando que as metodologias ativas podem ajudar os estudantes a desenvolverem habilidades práticas e críticas durante o aprendizado, este livro foi organizado.

Os conteúdos trabalhados no *e-book*, foram retirados de disciplinas iniciais de programação (em virtude da similaridade dos conteúdos) ofertadas nos cursos de Engenharia (disciplinas Computação 1 e Fundamentos de Programação) e Ciência da Computação (disciplina de Algoritmos), tendo como assuntos:

- Estrutura de decisão simples e composta
- Modularização
- Estrutura de Repetição
- Vetores e Matrizes

O conteúdo deste e-book, foi pensado buscando apresentar estratégias de ensino de programação baseadas em quatro diferentes metodologias ativas: Sala de Aula Invertida, a Aprendizagem Baseada em Problemas, a Gamificação e a Aprendizagem Baseada em Times. Para que o professor possa aplicar em sua aula, as estratégias de aprendizagem foram organizadas em um plano de aula detalhado, indicando: os objetivos da aula, as habilidades a serem desenvolvidas, o conteúdo a ser trabalhado, as plataformas que podem ser usadas, entre outras informações relevantes para que a metodologia seja empregada.

Cabe destacar que, as estratégias apresentadas são ideias do que pode ser desenvolvido em sala, para gerar maior engajamento no ensino de programação. Entretanto, estas são flexíveis e podem ser efetuados os ajustes que o professor julgar necessário, tais como a plataforma que tenha maior afinidade, o exercício proposto, a quantidade de atividades a serem desenvolvidas.

Importa frisar que estas estratégias, são resultados da pesquisa realizada no Mestrado em Ensino de Ciência e Tecnologia (PPGECT) da Universidade Tecnológica Federal do Paraná - Campus Ponta Grossa, sob a orientação do Prof. Dr. João Paulo Aires.

Deste modo, este ebook encontra-se dividido em seis capítulos, dos quais, cada capítulo, representa quatro planos de aula, baseados em quatro tipos diferentes de metodologias ativas. Desta forma, o capítulo primeiro deste e-book trata sobre os conceitos de metodologias ativas e suas potencialidades na educação. Por seguinte, o segundo capítulo, apresenta quatro estratégias de ensino da disciplina de Estrutura de Decisão Simples e Composta.

No terceiro capítulo, apresentam-se as estratégias de ensino da disciplina de modularização, seguida da disciplina de vetores, no quarto capítulo, matrizes no quinto capítulo e as estratégias de ensino de estruturas de repetição no sexto e último capítulo. Cada disciplina, desenvolve-se por quatro diferentes metodologias ativas, isto é, da sala de aula invertida, da aprendizagem baseada em problemas, da gamificação e da aprendizagem baseada em times. Seguida das considerações finais, este e-book também aponta sugestões de atividades avaliativas, bem como das referências utilizadas para a construção das estratégias.

METODOLOGIAS ATIVAS

De acordo com Chagas (2020, p. 39) as metodologias ativas podem ser consideradas como “tecnologias que proporcionam engajamento dos educandos no processo educacional e que favorecem o desenvolvimento de sua capacidade crítica e reflexiva em relação ao que estão fazendo”.

Ainda, segundo Silva (2017, p. 18) as metodologias ativas são “métodos de ensino cujo foco do aprendizado concentra-se nos alunos, ou seja, os próprios alunos assumem a responsabilidade pelo seu processo de aprendizagem tornando-se assim mais ativo”, também compreendida como aprendizagem ativa.

Isto é, aprendizagens que de acordo com Gomes (2021) não são metodologias recentes, pois partem do princípio do “aprender fazendo”, “aprender por experiência” ou “aprender na prática”, que permite o desenvolvimento do processo de aprendizagem por meio da aproximação do estudante com o problema ou realidade estudada.

Segundo Freire (1987) quando uma realidade estudada é problematizada, é disparado no estudante um gatilho no processo de ensino e aprendizagem utilizando processos de observação, inspeção, exploração e análise. Assim, a aprendizagem começa a fazer sentido e, com isso, o ensino torna-se significativo.

Neste sentido, as metodologias ativas são métodos que permitem aos estudantes e professores caminharem além do espaço das quatro paredes da sala de aula, adentrando na realidade de cada estudante e assim, com eles, reinventar e desvelar um mundo além da tradicionalidade do ensino bancário, quebrando paradigmas, e saindo da zona de conforto, para construir conjuntamente um processo de aprendizagem por meio da inovação, da ciência e da corresponsabilidade (MAZUIM; GOMES, 2019).

Processo este, exemplificado pela fase de “o colher é de todos, o capinar é sozinho; o colher é coletivo, o capinar é individual”. Em outras palavras, pode-se dizer que construir conhecimento, não é algo que se possa construir sozinho, nem somente por meio do professor, isso porque, cada um constrói sua estrada

de aprendizagens de maneira diferente, pois depende de estímulos diferentes para que o processo de aprendizado ocorra completamente.

Isto ocorre, porque segundo Gomes e Mazuim (2019), ninguém pode pensar e recriar pelos outros, no sentido de que os seres humanos são seres que precisam do coletivo, mas terem a sua individualidade de saberes e crenças que os tornam competentes para atuar na sociedade.

Desta forma, compreendendo que o aprender é uma atividade coletiva em alguns momentos, mas também pessoal em outros, com as metodologias ativas, este aprender se torna menos complexo. Isto porque, as metodologias ativas são possibilidades de “desenvolver o processo de aprender, utilizadas pelos professores na busca de conduzir a formação crítica de futuros profissionais nas mais diversas áreas” (MAZUIM, GOMES, 2019, p. 14).

Estas possibilidades são, então, “estratégias de aprendizagem que tem a finalidade de impulsionar o estudante a descobrir um fenômeno, compreender seus conceitos e saber relacionar suas descobertas com seus conhecimentos já existentes” (SILVA et al, p. 32).

Como apontado por William Glasser (LIMA; SANTOS, 2020), por meio da Pirâmide de Aprendizagem (Figura 1), o maior aprendizado ocorre quando ensinamos outras pessoas (90%) e pouco se aprende, ao apenas ouvir (20%).

Figura 1 - Pirâmide de Aprendizagem de William Glasser



Fonte: Lima; Santos (2020, p. 7)

Este processo de aprendizagem ativa permite então “tornar o aluno, discente, um indivíduo, que reconstrói seu conhecimento, tornando-se autor e

gestor de sua aprendizagem” (MAZUIM, GOMES, 2019, p. 21). Com isso, ele rompe a característica tradicional da passividade sendo protagonista do processo, no qual o foco da aula é a aprendizagem do estudante.

No entanto este processo constitui o desafio de construir um saber que surja do próprio sujeito, sem que o(a) mesmo(a) reproduza de forma mecânica e acrítica conhecimentos de outrem (SILVA et Al., 2017). Adicionalmente, por esta razão que, dentre os desafios do ensino juntamente as metodologias ativas, está em estabelecer a autonomia e a liberdade do estudante, de forma que o mesmo, seja capaz de participar ativamente do seu processo de ensino e aprendizagem. Não sendo apenas um expectador na sala de aula, e sim, um sujeito crítico, reflexivo e construtor do conhecimento.

Neste sentido, de acordo com Krassmann (2016), Mazuim e Gomes (2019), Ribeiro, 2019), Costa (2020) e Lima e Mesquita (2021) existem diversas metodologias que podem ser empregadas em sala de aula, enquanto conjunto de abordagens que visam envolver os estudantes no processo de ensino e aprendizagem por meio de experiências significativas, que os permitem compreender a relevância de seu aprendizado.

É por este pressuposto, que neste livro, são transcritos, quatro tipos específicos de metodologias ativas (Sala de aula invertida, Aprendizagem Baseada em Problemas, Gamificação, Aprendizagem Baseada em Times), em forma de estratégias de ensino, registrados em planos de aula detalhados, para aplicação das metodologias em sala de aula para o ensino de programação.

Neste sentido, vale destacar, que, a metodologia ativa da Sala de Aula Invertida (*Flipped Classroom*) se trata de uma abordagem em que a exposição primária do conteúdo é feita antecipadamente ao estudante de forma individual e em casa. Na sala de aula, é feito o esclarecimento do conteúdo pelo professor.

Por sua vez, a Aprendizagem Baseada em Problemas (*Problem-Based Learning* PBL) é uma estratégia em que o “problema” ou desafio é o elemento central do processo de ensino e aprendizagem, por meio de um ambiente colaborativo.

Seguidamente, a Gamificação é uma estratégia que prevê a utilização dos elementos de design de jogos em contextos não lúdicos, para criar um

ambiente semelhante ao dos jogos, para motivar a ação, promover a aprendizagem e resolver problemas, tais como narrativa, sistema de feedback, sistema de recompensas, a diversão e a interatividade (KRASSMANN, 2016).

Logo, são, com base nestas estratégias, que se seguem os próximos capítulos deste livro.

ESTRUTURA DE DECISÃO SIMPLES E COMPOSTA: ESTRATÉGIAS E APLICAÇÃO NA SALA DE AULA

As estruturas de decisão são utilizadas em programação para controlar o fluxo de execução do código com base em condições específicas. Existem dois tipos principais de estruturas de decisão: simples e composta.

Uma estrutura de decisão simples é composta por uma única condição e duas possibilidades de fluxo, dependendo se a condição é verdadeira ou falsa. Em outras palavras, é uma estrutura de decisão que apresenta apenas um bloco de código condicional.

Um exemplo comum de estrutura de decisão simples é o "if" em várias linguagens de programação. Observe o exemplo de um pseudocódigo para uma estrutura de decisão simples:

Figura 2 - Pseudocódigo - Estrutura de Decisão Simples

```
se condição for verdadeira então
    executar este bloco de código
fim se
```

Fonte: Organizado pelo autor (2023).

Por outro lado, uma estrutura de decisão composta é composta por múltiplas condições e várias possibilidades de fluxo, dependendo de diferentes combinações de condições.

Essas estruturas podem ter múltiplos blocos de código condicional, permitindo caminhos diferentes de execução com base nas diferentes combinações de condições.

Exemplos comuns de estruturas de decisão compostas são "if-else" e "switch-case" em várias linguagens de programação.

Na figura 3, podemos observar um exemplo em pseudocódigo para uma estrutura de decisão composta:

Figura 3 - Pseudocódigo para uma estrutura de decisão composta

```
se condição 1 for verdadeira então
    executar este bloco de código
senão se condição 2 for verdadeira então
    executar este bloco de código
senão se condição 3 for verdadeira então
    executar este bloco de código
...
senão
    executar este bloco de código (caso nenhuma das condições anteriores
seja verdadeira)
fim se
```

Fonte: Organizado pelo autor (2023)

Em resumo, a diferença entre as estruturas de decisão simples e compostas está na quantidade de condições e possibilidades de fluxo que elas permitem. A estrutura de decisão simples apresenta apenas uma condição e duas possibilidades de fluxo, enquanto a estrutura de decisão composta permite múltiplas condições e diferentes caminhos de execução com base nas condições.

Desta forma, com base neste conceito, este capítulo divide-se em quatro diferentes metodologias ativas (Sala de aula Invertida, Aprendizagem baseada em Problemas, Gamificação e Aprendizagem baseada em Times), sendo uma metodologia para cada plano de aula, que tem como objetivo, apresentar os fundamentos da estrutura de decisão simples e composta, utilizando vídeo explicativo disponibilizado antecipadamente e aula expositiva, para explorar o conteúdo.

Boa leitura!

Estrutura de Decisão Simples e Composta: Estratégias e Aplicação na Sala de Aula Invertida

METODOLOGIA DE ENSINO: SALA DE AULA INVERTIDA

Conteúdo: Decisão simples e composta

Metodologia de ensino: Sala de Aula Invertida.

Descrição detalhada da estratégia: Resolução de questões, por meio da metodologia ativa da Sala de Aula Invertida.

PLANO DE AULA

Metodologia Ativa: Sala de Aula Invertida

Objetivo da aula: Apresentar os fundamentos da estrutura de decisão simples e composta, utilizando vídeo explicativo disponibilizado antecipadamente e aula expositiva, para explorar o conteúdo.

Fixar o conteúdo sobre decisão simples e composta, utilizando a metodologia ativa sala de aula invertida.

1) Metodologia a ser trabalhada

Procedimento:

A finalidade consiste em:

- Instigar o trabalho colaborativo entre os estudantes, para resolução das questões apresentadas na sala de aula. Para isso, antes da aula presencial, serão disponibilizados materiais de apoio, tais como o exemplo disponibilizado por Pluskota (2022) e listas e de exercícios para os estudantes buscarem realizar em casa, antes da aula, como no exemplo disponibilizado em Pereira (2022). Exemplos estes, relacionados a estrutura de decisão simples e composta. A

partir destes, os alunos deverão ser instigados a trazer para a sala de aula, suas dúvidas, quanto aos temas abordados nos materiais.

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos:

- **Momento 1:** Apresentação da ementa aos estudantes (anterior ao dia da aula);
- **Momento 2:** Apresentação de um vídeo explicativo, como no exemplo disponibilizado por Pluskota (2022) (<https://www.youtube.com/watch?v=vFEuFYUZTRw>) aos estudantes sobre o tema que será posteriormente, trabalhado em sala de aula.
- **Momento 3:** Disponibilização dos conteúdos sobre estrutura de decisão simples e composta, utilizando a plataforma Google Classroom, e-mail e/ou WhatsApp (anteriormente ao dia da aula, com no mínimo uma semana de antecedência);
- **Momento 4:** Apresentação do conteúdo sintetizado em sala de aula (15 minutos).
- **Momento 5:** Discussão sobre o conteúdo disponibilizado no ambiente virtual de aprendizagem (10 minutos).
- **Momento 6:** Apresentação da dinâmica da aula, destacando que ela, será desenvolvida com base nas atividades disponibilizadas na aula anterior, como por exemplo, as atividades de fixação e o vídeo explicativo (5 minutos);
- **Momento 7:** Separação dos estudantes em grupos (de dois a três componentes) para resolução das questões - ao menos, duas questões da lista de exercícios em anexo (15 minutos).
- **Momento 8:** promover discussão entre os estudantes, para compreender como os grupos solucionaram as questões. Posteriormente, as soluções deverão ser entregues ao professor(a).
- **Momento 9:** Discussão e feedback com os estudantes (5 minutos).

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente a estrutura de decisão simples e composta.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de decisão simples e composta.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de decisão simples e composta;
Fomentar o protagonismo do estudante em seu aprendizado, por meio da leitura antecipada do conteúdo; incentivar o trabalho em grupo; instigar a liderança em times.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Plataforma Google Classroom; WhatsApp e/ou e-mail.
- Moodle, para criação de salas de reunião (se ensino remoto);
- Vídeos explicativos e textos sobre a temática.
- Vídeo exemplo, disponível em:
<https://www.youtube.com/watch?v=vFEuFYUZTRw>
- Apresentação de slides;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em time;
Questões disponibilizadas, enviadas separadamente ao professor. No total de 12 questões. Todas deverão estar solucionadas.

7) Descrição detalhada da estratégia da aula

Aula de decisão simples e composta.:

- A proposta desta aula é, utilizando a estratégia da sala de aula invertida, instigar que os estudantes acessem o conteúdo a ser trabalhado em aula, antes que este seja apresentado pelo professor. Durante a aula, depois do acesso antecipado ao material, suas dúvidas possam ser solucionadas.
- Para esta finalidade seja alcançada, serão disponibilizados vídeos e textos explicativos sobre o tema que será abordado em sala de aula.
- Para fixação do tema, será disponibilizado aos estudantes, uma lista com 12 exercícios sobre o conteúdo.
- Para solucioná-los, os estudantes, deverão separar-se em grupo para desenvolver duas questões sorteadas pelo professor, para o referido time.
- À medida em que os times solucionarem seus exercícios, estes, deverão trocar de mesa para explicar ao grupo seguinte, como resolveu sua questão, e assim, sucessivamente, até todos os estudantes, passarem por todos os times.
- Ao final, o grupo, deverá indicar um líder, para que este, apresente as questões

que seu time resolveu e explicar como realizou o mesmo.

- Por seguinte o professor, deverá apresentar um feedback aos estudantes.
- A avaliação desta aula, ocorrerá, pela entrega individual das 12 atividades solucionadas ao professor.

8) Observações adicionais

É importante que:

- Oportunizar que os estudantes apresentem suas dúvidas em relação ao conteúdo estudado.
- Que todos os estudantes passem pelos outros grupos;
- Que seja entregue a (ao) professor (a) de forma individual, as doze questões solucionadas.

Desafios:

Professor: Produzir um vídeo explicativo, anteriormente ao dia da aula e disponibilizar em plataforma online; instigar o preparo dos estudantes antes da aula, bem como a comunicação e o trabalho em time.

Sugere-se: a criação de um fórum de discussão, tendo uma pergunta feita pelo professor, como condutora da discussão.

Estudantes: Realizar o máximo de atividades em casa e levar para o professor as dúvidas que surgiram a partir das mesmas. E assistir e anotar as observações do vídeo aula disponibilizado pelo professor e ou leituras que o mesmo disponibilizou.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre decisão simples e composta, tanto anteriormente, quanto no dia da aula;
- Instigar que os estudantes tragam de casa suas dúvidas (isso pode ser promovido, por meio de um fórum de discussão ou uma enquete no grupo no Whatsapp).
- Auxiliar na formação de equipes para construção da atividade avaliativa.
- Entregar feedback dos trabalhos e atividades

Ação dos estudantes:

- Solucionar a equação em grupo por meio do fórum e posteriormente de forma individual; indicar um líder para apresentar a solução para a turma e entregar ao professor a lista completa de exercícios solucionada.

Referência:

PEREIRA, Jonathan. **Lista de exercício de algoritmo**. S/A. Disponível em: <https://docente.ifrn.edu.br/jonathanpereira/disciplinas/algoritmos/lista-de-exercicios-2/view> Acesso em: 23 de dezembro de 2022.

PLUSKOTA, Jackson William. **Decisão Simples e Composta**. 2022. Vídeo Aula. Disponível em: <https://www.youtube.com/watch?v=vFEuFYUZTRw> Acesso em: 23 de dezembro de 2022.

Estrutura de Decisão Simples e Composta: Estratégias e Aplicação na PBL

METODOLOGIA DE ENSINO: APRENDIZAGEM BASEADA EM PROBLEMAS

Conteúdo: Estrutura de Decisão simples e composta

Metodologia de ensino: Aprendizagem baseada em problemas.

Descrição detalhada da estratégia: Resolução de questões utilizando a metodologia ativa Aprendizagem Baseada em Problemas.

PLANO DE AULA

Metodologia Ativa: Aprendizagem baseada em problemas.

Objetivo da aula:

- Apresentar uma situação-problema, envolvendo estrutura de decisão.
- Instigar que os/as estudantes discutam, com base nos seus conhecimentos prévios, sobre decisão simples e composta (via aula expositiva anterior ou sala de aula invertida) e criem hipóteses em um mapa mental sobre a solução do problema apresentado.

Fixar o conteúdo sobre decisão simples e composta via metodologia ativa da Aprendizagem baseada em problemas.

1) Metodologia a ser trabalhada

A finalidade consiste em:

- Instigar o trabalho colaborativo entre os estudantes, para resolução de uma situação-problema que contribua para o aprendizado do conteúdo sobre

Estrutura de Decisão Simples e Composta.

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos ao qual compreenderá em cinco etapas:

Etapa 1 - Estabelecendo relações com o problema: Dialogar com os estudantes sobre quais informações e conhecimentos prévios eles possuem a respeito dos conceitos sobre o uso da decisão no dia a dia.

Etapa 2 - Estabelecendo um plano de trabalho para a resolução do problema: A partir das discussões realizadas com auxílio do professor, os estudantes devem listar informações que forem obtidas por meio da apresentação do problema e discussão com os colegas, bem como dos seus conhecimentos prévios sobre o assunto.

Etapa 3 – Abordagem do problema: Construir um método para solucionar o problema apresentado, forma colaborativa.

Etapa 4 - Reequacionando o problema: após os trabalhos individual e coletivo dos estudantes, cada grupo deverá apresentar, à turma um relatório do trabalho desenvolvido o processo seguido pelos estudantes para o desenvolvimento do problema. A partir deste relatório, avaliar o desenvolvimento do problema e possíveis melhorias.

Etapa 5 - Elaborando e apresentando os produtos: nesta etapa, caberá ao professor fomentar um processo de avaliação, do qual, contabilize as percepções do professor, dos estudantes e dos grupos em si, da atividade total.

Exemplo de aplicação da estratégia:

Momento 1: Preparação Pré-aula:

- Apresentação do conteúdo aos estudantes (anterior ao dia da aula, preparando-os para a aula ativa);
- Disponibilização de um vídeo explicativo aos estudantes sobre o tema que será trabalhado em sala de aula, conforme o exemplo: <https://www.youtube.com/watch?v=vFEuFYUZTRw>
- Disponibilização dos conteúdos sobre Estrutura de decisão simples e composta utilizando a plataforma Google Classroom (como no exemplo

disponível na sala v54gkp7) disponível pelo link: <https://classroom.google.com/c/NTQxMjUxOTI4MDYy?cjc=v54gkp7> e-mail e ou WhatsApp (Anteriormente ao dia da aula, com no mínimo uma semana de antecedência);

Momento 2: Durante a aula:

- Apresentação do conteúdo sintetizado em sala de aula (15 minutos).

Momento 3: Apresentação da situação problema e da dinâmica da aula (10 minutos).

- A empresa Plusb Web está contratando uma equipe para compor seu novo projeto de desenvolvimento Web. Para isso, você terá que organizar uma equipe e resolver 50% das questões disponíveis em: <https://1drv.ms/w/s!AqgpbKeTHqetmpl8x0S8CT2plk7x7A> no menor tempo possível.

Momento 4: Separação dos estudantes em grupo para resolução das questões, ao menos, 50% questões da lista de exercícios em anexo (15 minutos).

Momento 8: Observar o desenvolvimento e organização dos estudantes quanto a escolha das questões para resolução, observando as seguintes notas:

Quadro 1 - Exemplo, organização dos estudantes

Ideias	Fatos	Questões de Aprendizagem	Plano de Ação
Como os grupos se dividiram	Discussão das oito questões apresentadas	Mediação dos conhecimentos que os estudantes possuem perante as questões	Observação do planejamento e ação dos estudantes

Nota: Organizado pelo autor.

Momento 9: Apresentação dos estudantes sobre seu plano de ação e questões que resolveram. Eles devem explicar o porquê escolheram as questões.

Momento 10: Discussão e feedback com os estudantes (5 minutos).

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente a estrutura de decisão simples e composta.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de decisão simples e composta.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de decisão simples e composta; fomentar o protagonismo do estudante em seu aprendizado, por meio da leitura antecipada do conteúdo; incentivar o trabalho em times; instigar a liderança em times.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Plataforma Google Classroom para disponibilização dos exercícios online e o vídeo explicativo: Sala – v54gkp7 em: <https://classroom.google.com/u/5/c/NTQxMjUxOTI4MDYy>
- Vídeos explicativos e textos sobre a temática: Vídeo exemplo, disponível em: <https://www.youtube.com/watch?v=vFEuFYUZTRw>
- Lista de exercícios, conforme o exemplo disponível em: <https://1drv.ms/w/s!AqqpbKeTHqetmpl8x0S8CT2plk7x7A>
- Feedback do professor para os estudantes;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades;
- Média da avaliação do professor e da autoavaliação dos estudantes.

7) Descrição detalhada da estratégia da aula

Aula de decisão simples e composta:

- A proposta desta aula é instigar que os estudantes, acessem o conteúdo/materiais antes que a aula ocorra. Durante a aula, as eventuais dúvidas devem ser solucionadas.
- Para esta finalidade, serão disponibilizados vídeos e textos explicativos sobre o tema que será abordado em sala de aula.
- Para fixação do tema. Será disponibilizado aos estudantes, uma lista com oito exercícios.
- Para solucioná-los, os estudantes, deverão separar-se em grupo para solucionar 50% questões no menor tempo possível.
- Ao longo do desenvolvimento dos exercícios, os grupos poderão consultar o professor para verificar se a questão está correta. Sugere-se uma premiação

à equipe vencedora (resolução de mais exercícios em menos tempo).

- Ao final, o grupo, deverá apresentar seu plano de ação para resolução das questões selecionadas pelo grupo.
- O professor, deverá apresentar um feedback aos estudantes.
- A avaliação desta aula, ocorrerá, por meio da observação do professor e de uma autoavaliação dos estudantes, no qual, permitirá calcular a média da pontuação obtida.

8) Observações adicionais

É importante que:

- Os estudantes estejam preparados para a dinâmica, antes da aula.

Desafios:

- Produzir/selecionar um vídeo explicativo, anteriormente ao dia da aula e disponibilizar em plataforma online;
- Instigar o preparo pré-aula, bem como a comunicação, o trabalho em grupo e o planejamento.
- Mediar o plano de ação dos estudantes.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre decisão simples e composta, tanto anteriormente, quanto no dia da aula;
- Instigar que os estudantes, tragam de casa suas dúvidas.
- Auxiliar na formação de equipes para construção da atividade avaliativa.
- Entregar feedback dos trabalhos e atividades

Ação dos estudantes:

- Construir um plano de ação para a resolução de 50% das questões disponibilizadas pelo professor e solucioná-las.

Observações:

- Ainda cabe como ações dos estudantes e professores de acordo com Camargo (2019, p. 117):

Quadro 2 - Indicações de ações nas Metodologias ativas

Estudantes	Professores
Reconhecer o sucesso no trabalho dos outros, além de perceber como colaboradores e fornecedores de recursos didáticos entre si (avaliação entre pares); Ser agente da própria aprendizagem, isto é, reconhecer o sucesso em seu próprio trabalho e se concentrar em como e o que estão aprendendo e onde ainda podem melhorar (autoavaliação).	Esclarecer e compartilhar as intenções de aprendizagem e critérios para obter sucesso: para que os alunos compreendam o que estão tentando aprender, por que e o que se espera deles; Organizar discussões e atividades efetivas em sala de aula de forma a evidenciar a aprendizagem: para criar um ambiente na sala de aula onde os alunos possam expor suas próprias ideias, pensar em voz alta e explorar a sua compreensão; Fornecer feedback que seja capaz de promover a aprendizagem: referenciando a qualidade de seu trabalho e o que os alunos podem fazer para torná-lo melhor;

Fonte: Camargo (2019).

REFERÊNCIAS

CARVALHO, Melisse. **Estratégias para avaliação na aprendizagem baseada em problemas.** In LOPES, Renato Matos. **APRENDIZAGEM BASEADA EM PROBLEMAS: Fundamentos para a aplicação no ensino médio e na formação de professores.** 2019. Disponível em: <https://educapes.capes.gov.br/bitstream/capes/432641/2/APRENDIZAGEM%20BASEADA%20EM%20PROBLEMAS%20-%20fundamentos%20para%20a%20sua%20aplica%C3%A7%C3%A3o%20no%20Ensino%20M%C3%A9dio%20e%20na%20Forma%C3%A7%C3%A3o%20de%20Professores.pdf> Acesso em: 26 de janeiro de 2023.

FERNANDES, Gustavo Q. **Exercícios sobre estruturas de decisão em algoritmos.** 2020. Disponível em: <https://mecatronicacemporcento.com.br/exercicios-sobre-estruturas-de-decisao-em-algoritmos/> Acesso em: 26 de janeiro de 2023.

ANEXO

SUGESTÃO DE LISTA DE EXERCÍCIOS

Estrutura de Decisão Simples e Composta

1 – Faça um algoritmo para ler um número real e informar ao usuário se ele é maior, menor ou igual a zero.

RESPOSTA

Algoritmo 1- Programa para comparar um número real com zero

```
1      inicio;
2      real: num;
3      escreva ("Digite um número real: ");
4      leia (num);
5      se (num > 0) entao
6          escreva ("O número é maior que zero!");
7      senao
8          se (num = 0) entao
9              escreva ("O número é igual a zero!");
10         senao
11             escreva ("O número é menor que zero!");
12         fimse;
13     fimse;
14     fim.
```

2 – Desenvolva um pseudocódigo para ler o nome e a idade de um indivíduo e imprimir “criança”, se a idade for menor ou igual a 13 anos, “adolescente”, no caso de idade estar entre 13 e 20 anos, e “adulto”, se for mais velho.

RESPOSTA

Programa para analisar a idade do usuário

```
1      inicio;
2      inteiro: idade;
3      escreva ("Digite a sua idade: ");
4      leia (idade);
5      se (idade <= 13) entao
6          escreva ("Criança!");
7      senao
8          se (idade <= 20) entao
9              escreva ("Adolescente!");
```

```

10      senao
11          escreva ("Adulto!");
12      fimse;
13      fimse;
14      fim.

```

3 – Elabore um algoritmo que leia 3 números reais, identifique e imprima o maior deles.

RESPOSTA

Programa para selecionar o maior número real

```

1      inicio;
2      real: A, B, C;
3      escreva ("Digite três números reais: ");
4      leia (A, B, C);
5      se (A >= B e A >= C) entao
6          escreva ("O maior número digitado é: ", A);
7      senao
8          se (B >= C) entao
9              escreva ("O maior número digitado é: ", B);
10     senao
11         escreva ("O maior número digitado é: ", C);
12     fimse;
13     fimse;
14     fim.

```

4 – Faça um pseudocódigo para ler um nome e imprimir o telefone correspondente de acordo com a lista mostrada abaixo. Se o nome não estiver na lista, o algoritmo deve imprimir a mensagem "Não consta na lista".

João: 222.22.22
Paulo: 333.33.33
Leonardo: 444.44.44
Márcio: 555.55.55

RESPOSTA

Programa para lista telefônica

```

1      inicio;
2      char: nome[1...10];
3      escreva ("Digite o nome de quem você quer o telefone: ");
4      leia (nome);

```

```

5      escolha (nome)
6      caso "João":
7          escreva ("O telefone do João é 222.22.22");
8      caso "Paulo":
9          escreva ("O telefone do Paulo é 333.33.33");
10     caso "Leonardo":
11         escreva ("O telefone do Leonardo é 444.44.44");
12     caso "Márcio":
13         escreva ("O telefone do Márcio é 555.55.55");
14     caso contrario:
15         escreva ("Esse nome não consta na lista!");
16     fimescolha;
17     fim.

```

5 – Desenvolva um algoritmo para calcular a área e o perímetro de uma circunferência (C), quadrado (Q), e retângulo (R). O usuário deverá escolher o polígono e informar os dados necessários para proceder os cálculos.

RESPOSTA

Programa para calcular área e perímetro

```

1      inicio;
2      char: geometria;
3      real: lado1, lado2, raio, perimetro, area;
4      escreva ("Digite a geometria desejada (C, Q, R): ");
5      leia (geometria);
6      escolha (geometria)
7      caso "C":
8          escreva ("Digite o raio da circunferência: ");
9          leia (raio);
10         perimetro ← 2 * 3.14 * raio;
11         area ← 3.14 * pot(raio, 2);
12         escreva ("O perímetro da circunferência é: ", perimetro);
13         escreva ("A área da circunferência é: ", area);
14     caso "Q":
15         escreva ("Digite o comprimento de um dos lados do quadrado: ");
16         leia (lado1);
17         perimetro ← 4 * lado1;
18         area ← pot(lado1, 2);
19         escreva ("O perímetro do quadrado é: ", perimetro);
20         escreva ("A área do quadrado é: ", area);
21     caso "R":

```

```

22     escreva ("Digite o comprimento dos dois lados do retângulo: ");
23     leia (lado1, lado2);
24     perimetro ← 2 * (lado1 + lado2);
25     area ← lado1 * lado2;
26     escreva ("O perímetro do retângulo é: ", perimetro);
27     escreva ("A área do retângulo é: ", area);
28     caso contrario:
29         escreva ("Não existe essa opção de figura geométrica!");
30     fimescolha;
31     fim.

```

6 – Escreva um pseudocódigo que leia o código de um determinado produto e apresente a sua classificação de acordo com a tabela abaixo:

Quadro 3 - Lista de códigos e suas respectivas classificações para o exercício 6

Lista de códigos e suas respectivas classificações para o exercício 6	
Código	Classificação
1	Alimento não perecível
2, 3 ou 4	Alimento perecível
5 ou 6	Vestuário
7 até 12	Limpeza
Qualquer outro código	Opção inválida

Nota: Organizado pelo autor (2023)

RESPOSTA

Programa para imprimir a classificação de um produto

```

1     inicio;
2     inteiro: codigo;
3     escreva ("Digite o código da classificação: ");
4     leia (codigo);
5     se (codigo = 1) entao
6         escreva ("Alimento não perecível!");
7     senao
8         se (codigo = 2 ou codigo = 3 ou codigo = 4) entao
9             escreva ("Alimento perecível!");

```

```

10     senao
11         se (codigo = 5 ou codigo = 6) entao
12             escreva ("Vestuário!");
13     senao
14         se (codigo >= 7 e codigo <= 12) entao
15             escreva ("Limpeza!");
16     senao
17         escreva ("Opção Inválida!");
18     fimse;
19     fimse;
20     fimse;
21     fimse;
22     fim.

```

7 – Elabore um algoritmo que leia dois números reais, a operação aritmética e, então, calcule e imprima o resultado da operação desejada. Utilize os símbolos (+, -, *, /) para identificar a operação aritmética. Caso a operação pretendida seja diferente das implementadas, o pseudocódigo deverá exibir a mensagem “Operação Inválida” para o usuário.

RESPOSTA

Programa para fazer uma operação aritmética entre dois números

```

1     inicio;
2     char: operacao;
3     real: num1, num2, resultado;
4     escreva ("Digite dois números reais e a operação desejada: ");
5     leia (num1, num2, operacao);
6     escolha (operacao)
7         caso "+":
8             resultado ← num1 + num2;
9             escreva ("O resultado da soma é: ", resultado);
10        caso "-":
11            resultado ← num1 – num2;
12            escreva ("O resultado da subtração é: ", resultado);;
13        caso "*":
14            resultado ← num1 * num2;
15            escreva ("O resultado da multiplicação é: ", resultado);
16        caso "/":
17            resultado ← num1 / num2;
18            escreva ("O resultado da divisão é: ", resultado);
19        caso contrario:

```

```

20     escreva ("Operação inválida!");
21     fimescolha;
22     fim.

```

8 – Desenvolva um algoritmo que leia o valor de uma determinada compra e o código da forma de pagamento. Com base neste código, o pseudocódigo deverá calcular e imprimir o valor a pagar, a forma de pagamento e o valor de cada parcela, se for o caso.

Quadro 4 - Lista de códigos e suas respectivas classificações para o exercício 8

Código	Classificação
1	À vista, com 8% de desconto
2	À vista no cartão, 4% de desconto
3	Em 2x, preço normal sem juros
4	Em 4x, preço acrescido de 8%
Qualquer outro código	Opção inválida

Fonte: Organizado pelo autor (2023)

RESPOSTA

Programa para calcular as formas de pagamento

```

1     inicio;
2     int: codigo;
3     real: valor, parcela;
4     escreva ("Digite o valor a ser pago e a forma de pagamento: ");
5     leia (valor, codigo);
6     escolha (codigo)
7     caso 1:
8         valor ← valor * 0.92;
9         escreva ("O valor a ser pago é de R$: ", valor);
10    caso 2:
11        valor ← valor * 0.96;
12        escreva ("O valor a ser pago é de R$: ", valor);
13    caso 3:
14        escreva ("O valor a ser pago é de R$: ", valor);
15        escreva ("Cada parcela será de R$: ", valor/2);
16    caso 4:
17        valor [Símbolo] valor * 1.08;

```

```
18     escreva ("O valor a ser pago é de R$: ", valor);
19     escreva ("Cada parcela será de R$: ", valor/2);
20     caso contrario:
21         escreva ("Opção inválida!");
22     fimescolha;
23     fim.
```

Estrutura de decisão simples e composta: Estratégias e aplicação na gamificação

METODOLOGIA DE ENSINO: GAMIFICAÇÃO

Conteúdo: Estrutura de Decisão Simples e Composta

Metodologia de ensino: Gamificação. Também pode ser utilizado juntamente a Sala de aula invertida, como uma preparação dos estudantes para discussões em sala de aula presencial.

Descrição da estratégia: Com base na Metodologia ativa da gamificação e utilizando a Plataforma *Google Classroom*, o professor deverá interagir com os estudantes durante o aprendizado do conteúdo sobre Estrutura de Decisão Simples e Composta. A interação se dará, via mural e lista de atividades, na qual, os estudantes verão em cada observação do professor e em sala de aula, durante o desenvolvimento da aula ativa presencial.

PLANO DE AULA

Metodologia Ativa: Gamificação

Objetivo da aula:

Apresentar, de forma lúdica, como funciona a estrutura de decisão simples e composta aos estudantes. Destacando que uma aula lúdica, é aquela aula que utiliza de elementos de jogos, diversão e interatividade para tornar o processo de aprendizado mais envolvente, motivador e efetivo. Ela busca engajar os alunos de forma ativa e proporcionar experiências práticas que facilitem a compreensão e a retenção do conhecimento (PAPERT, 2008).

1) Metodologia a ser trabalhada

Baseada na metodologia da gamificação, serão realizados os seguintes

procedimentos/etapas:

- Criação de uma sala de aula no Google Classroom, com a temática “caça ao tesouro”.
- Aplicação via Sala de aula Invertida ou em aula presencial dos desafios para realização das missões para conquista do tesouro para o título de “Rei dos Piratas”. É válido destacar que os desafios têm como finalidade, tornar a aula mais competitiva e divertida, instigando os estudantes relacionar o aprendizado a diversão e não unicamente a obrigação de realização das atividades de fixação.
- Realização e esclarecimento das missões, de acordo com a observação das dificuldades dos estudantes.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente a estrutura de decisão simples e composta.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Estrutura de Decisão Simples e Composta:

- Valores Lógicos.
- Operadores Relacionais.
- Operadores Lógicos.
- Comandos de Seleção Condicional.
- Estrutura condicional Simples e Composta em C.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Estrutura de Decisão Simples e Composta;
- Aprendizagem Ativa.
- Trabalho em grupo.
- Competitividade positiva.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Disponibilização de textos introdutórios, para apresentar os conceitos

Estrutura de Decisão Simples e Composta, a serem discutidos em sala de aula.

- Os materiais serão disponibilizados em uma sala de aula, na plataforma Google Classroom.

6) Forma de avaliação

Observação da participação dos alunos por meio da conquista das missões.

7) Descrição detalhada da estratégia da aula

- Antes da aula:
 - Preparação da disciplina durante as aulas anteriores, relacionada a Estrutura de Decisão Simples e Composta.
 - Apresentação aos estudantes, da metodologia que será utilizada ao longo das aulas;
 - Disponibilização de materiais de estudos em casa, utilizando a sala de aula na plataforma Google Classroom.

Em sala de aula presencial ou em meio remoto, sugere-se as seguintes etapas (utilizando uma sala de aula criada no Google Classroom ou similar):

Etapa 1: Cenário da aula gamificada:

☞ **1.1** Nesta etapa o professor deverá planejar o jogo que será realizado em sala de aula, desde as atividades de direcionamento, até os desafios gamificados.

☞ Caberá ainda ao professor, na etapa do planejamento, considerando a atividade do Classroom, personalizar o banner, para cativar a atenção dos estudantes. Para isso, é interessante criar uma arte, que esteja relacionada com o objetivo da aula.

☞ Por exemplo, Rei dos Piratas. Você está em *One Piece* e deverá encontrar o ouro o mais rápido possível e se tornar o Rei dos Piratas.

☞ Desta forma, a arte, terá que estar relacionada a piratas, tesouro e o conteúdo em si, neste caso, “estrutura de decisão simples e composta”.

☞ Esta arte, poderá ser construída, utilizando a ferramenta *Google Slides*, *Gimp* ou até mesmo no *Canva* na dimensão pixel de 1600X400. Esta imagem deverá ser baixada em PNG para o upload no banner do Google Classroom. Como por exemplo na Figura 4:

Figura 4 - Banner



Fonte: Organizado pelo autor (2023)

Etapa 2: Missões: Com a aula e jogo planejado. Adentrar em Google Formulários para desenvolvimento das aulas em formato Escape Room. Por se tratar da estratégia de gamificação, na qual usam-se estratégias de jogos nos processos de ensino, é necessário criar fases ou níveis e um cenário para o jogo. Este processo pode ser realizado utilizando o Google Slides, de forma que os estudantes possam visualizar o que devem fazer para conquistar o **primeiro lugar** e se tornar o “Rei dos Piratas”.

☞ É válido construir links que levem os estudantes a acessarem as etapas dos estudos e cumprir as missões.

☞ Começando do simples para o complexo, como níveis em si. Estas etapas podem ser realizadas utilizando perguntas no formato verdadeiro ou falso, por meio do *Google Forms*). Deve-se utilizar tópicos para destacar (dar ênfase) as fases, integrando outras ferramentas para realização das tarefas e missões, tais como slides e o Youtube.

☞ Por se tratar de uma estratégia de jogo, é necessário apresentar recompensas ao final das etapas, como no caso de emblemas, estrelas, distintivos e/ou certificados. Estes, são concedidos aos jogadores, por alcançarem níveis nas pontuações.

☞ Crie um Google Forms para cada nível de pontuação, no qual os estudantes preenchem as informações, gerando uma planilha para totalizar o ranking final de todas as pontuações.

Exemplo: No Google Formulários (para melhor compreensão, segue o exemplo no link:

<https://docs.google.com/forms/d/e/1FAIpQLSeyX4KNzggIqHxZUGM143dSSH5QOvBJKycbefKlIjev5Ape9w/viewform>)

Sessão 1: Explicar a missão e solicitar os dados dos estudantes

Título: Rei dos Piratas: Uma aventura pela estrutura de decisão simples e composta.

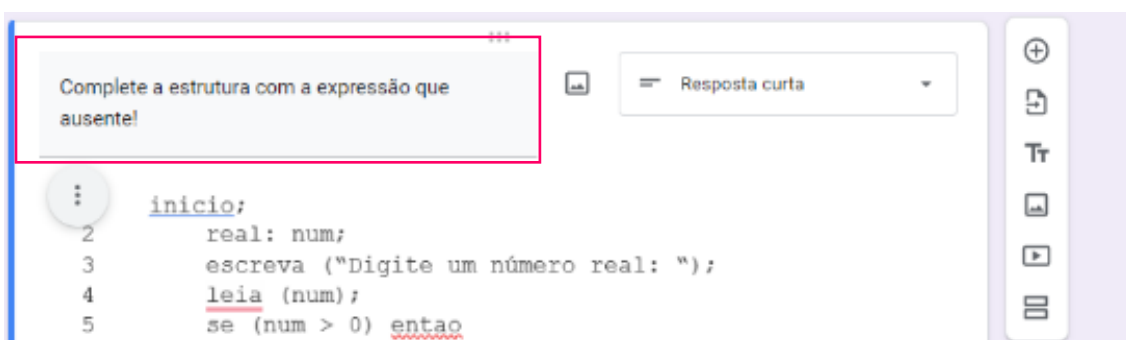
Explicação/Comentário: Complete as missões e conquiste o título do novo Rei do Piratas (e o prêmio tão desejado). Lembre-se: para cada uma das perguntas, há apenas uma resposta correta. Cabe a você completá-la e encontrar o erro.

Sessão 2: Construa a atividade/missão a partir da validação das atividades.

Para validação:

Clicar em “ADICIONAR PERGUNTA” após escrever a questão que pretende direcionar aos estudantes; Exemplo: *Complete a estrutura com a expressão que está ausente!*”

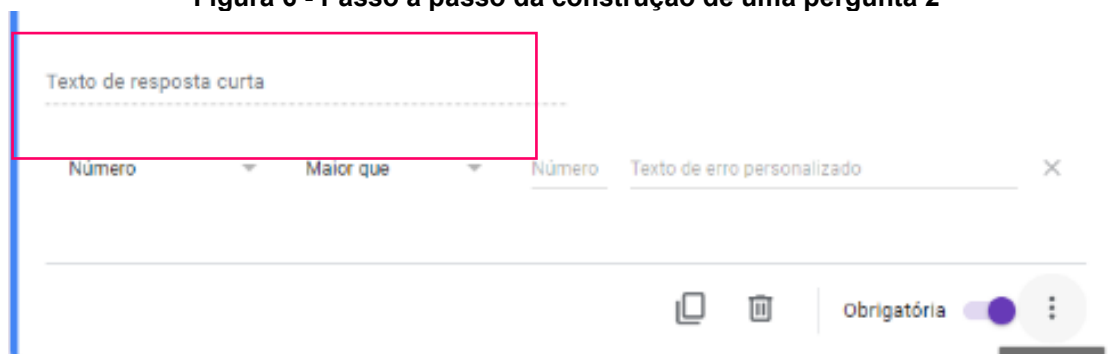
Figura 5 - Passo a passo da construção de uma pergunta 1



Fonte: Organizado pelo autor (2023)

Clicar em “RESPOSTA CURTA”.

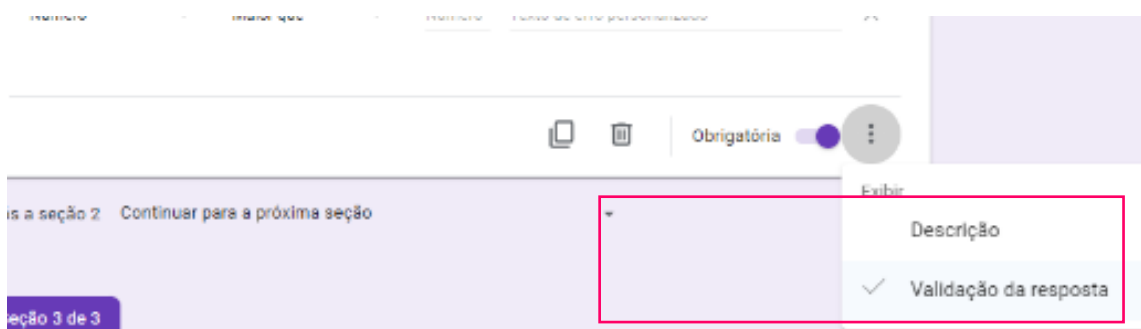
Figura 6 - Passo a passo da construção de uma pergunta 2



Fonte: Organizado pelo autor (2023)

Clicar em “MAIS OPÇÕES” e clicar em “VALIDAÇÃO DA RESPOSTA”.

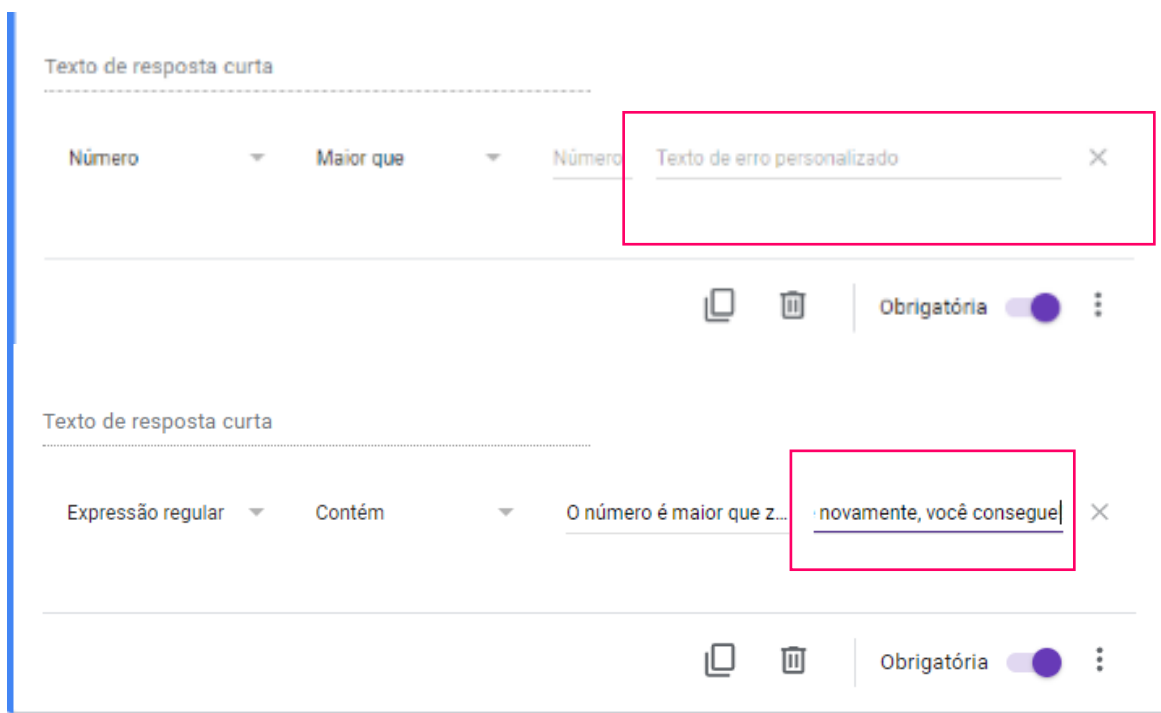
Figura 7 Passo a passo da construção de uma pergunta 3



Fonte: Organizado pelo autor (2023)

Clicar em “EXPRESSÃO REGULAR” clicar em “CONTÉM” e seguir para preencher em “PADRÃO” a resposta correta. Por seguinte, clicar em Criar o “TEXTO PERSONALIZADO”, com respostas como: “Ah Não! Tente novamente” ou “Parabéns!” se o estudante acertar. A questão automaticamente irá para a próxima sessão. Observe a Figura 8, que apresenta o passo a passo da construção de uma pergunta:

Figura 8 - Passo a passo da construção de uma pergunta 4



Fonte: Organizado pelo autor (2023)

A sequência apresentada na Figura 8 deverá ser seguida ao longo das questões/missões.

Após formular e finalizar a questão/missão, você deverá clicar em “NOVA SESSÃO” e adicionar/escrever “VOCÊ AVANÇOU” e em NOVA QUESTÃO

adicionar uma imagem com a frase “SUCESSO, ESSA MISSÃO FOI CONCLUÍDA!” Por exemplo na Figura 9:

Figura 9 – Exemplo de mensagem final em missão da Gamificação



Fonte: Organizado pelo autor (2023)

Etapa 5: Recompensa: Em cada um dos formulários, deixe uma mensagem em configuração (apresentação) parabenizando pela conquista, com o link em que o aluno vai acessar para gerar o seu certificado, no qual ela poderá personalizar com seu nome.

Feedback: Retornar aos estudantes, suas dúvidas e dificuldades, bem como diálogo sobre possíveis novos desafios e possibilidades de contribuição a sala de aula.

8) Observações adicionais

É importante, planejar a aula previamente, verificando e testando as missões, antes de levar a sala de aula.

Desafios:

Montagem da sala de aula gamificada.

Ação dos professores:

Planejamento, estruturação e avaliação da sala de aula.

Ação dos alunos:

Desenvolvimento das missões disponíveis na sala de aula e conquista do tesouro.

Feedback:

Entrega das avaliações e esclarecimento das missões que os estudantes possam não ter conseguido realizar.

REFERÊNCIAS

MECATRONICA. **Exercícios sobre estrutura de decisão em algoritmos.** 2023. Disponível em: <https://mecatronicacemporcento.com.br/exercicios-sobre-estruturas-de-decisao-em-algoritmos/> Acesso em: 18 de janeiro de 2023.

PAPERT, Seymour. **A máquina das crianças: repensando a escola na era da informática.** Porto Alegre: Artmed, 2008.

PLUSKOTA. Jackson W. **Modelo aula gamificada de Estrutura de Decisão Simples e Composta.** 2023. Disponível em: <https://forms.gle/kqdfa3C8mGRbTb2h6> Acesso em: 18 de janeiro de 2023.

RODRIGUES, Claudio C. **Linguagem C - Estrutura de Seleção Condicional.** UFU. 2021. Disponível em: <https://www.facom.ufu.br/~crlopes/IC/8-C-Condicionais.pdf> Acesso em: 11 de janeiro de 2023.

Estrutura de Decisão Simples e Composta: Estratégias e Aplicação na TBL

METODOLOGIA DE ENSINO: APRENDIZAGEM BASEADA EM TIMES

Conteúdo: Estrutura de Decisão Simples e Composta

Metodologia de ensino: Aprendizagem Baseada em Times

Descrição detalhada da estratégia: Trata-se de uma atividade em aula presencial no qual será realizada durante o ensino do conteúdo de Estrutura de Decisão Simples e Composta, considerando uma aula de 50 minutos.

PLANO DE AULA

Metodologia Ativa: Aprendizagem baseada em Times

Objetivo da aula:

Fixar o conteúdo de Estrutura de Decisão Simples e Composta utilizando a metodologia ativa *Team Based Learning* (TBL) ou Aprendizagem Baseada em Times, durante uma aula de 50 minutos, por meio da realização de uma atividade envolvendo o desenvolvimento de um site (*Google Sites*) e exercícios de fixação.

1) Metodologia a ser trabalhada

Procedimento:

- **Pré-aula**

Anteriormente ao dia da aplicação da TBL, será necessária uma breve explicação do conteúdo que será trabalhado na aula seguinte. Quer seja presencialmente, ou por meio de disponibilização de material didático, tais como artigos, vídeos.

- Para o desenvolvimento da aula serão necessários:
 - Computador ou *notebook* para acesso as atividades disponíveis no Google Site.

- Editor de textos *Word* (ou similar) ou editor de planilhas Excel (ou similar)
- Em caso de aula remota, pode ser adotada a plataforma *Google Meet* e *Moodle* (ou similar).

A atividade terá duração de 50 minutos.

- Para o desenvolvimento da aula deverão estar preparados alguns desafios, como os disponíveis no *link*: <https://forms.gle/c78x9xvmDZ9DxxjY9>
- A construção de um mapa mental sobre o conteúdo, para promover a discussão ao final da aula, gerando uma nova descrição/mapa coletivo dos temas tratados em aulas, pelos alunos.
- Preparação da aula no Google Sites, conforme disponível no *link*: <https://sites.google.com/view/estruturadedecisaosc/in%C3%ADcio>

Especificamente, durante o dia da aula.

- Indica-se dividir a turma em times, de forma aleatória, de modo, que não haja pessoas com vínculos comum, tais como colegas que geralmente fazem trabalhos juntos. Esta regra é importante, para o desenvolvimento do processo de organização, planejamento e liderança do time.
- Em sala de aula, apresentar atividade baseada na aprendizagem ativa, retomando a aula anterior de forma individual, e posteriormente, indicar os times de até quatro pessoas utilizando a metodologia TBL.

Metodologia de ensino: Aprendizagem Baseada em Times.

- A aplicação da metodologia TBL será realizada da seguinte forma:
 - **Fase da preparação:** Antes do dia da aula de TBL o professor deverá realizar um estudo prévio para preparar os estudantes para as atividades relacionadas a estrutura de decisão simples e composta. Esta preparação pode ser dar via materiais disponibilizados remotamente ou presencialmente, por meio de textos, vídeos e ou simulações.
 - **Teste de preparação individual:** em sala de aula, os estudantes poderão iniciar um teste individual para apreender o conteúdo que foi disponibilizado anteriormente a aula TBL.

- **Teste de preparação em equipe:** Divididos em equipes de forma aleatória pelo professor, os estudantes irão discutir e responder, o teste disponibilizado no *Google Site*.
- Neste site, haverá cinco questões:
 - Três questões que os estudantes verão automaticamente, se estão corretos ou não. Como no exemplo da imagem a seguir:

Figura 10 - Exemplos de questões sobre Estrutura de decisão simples e composta

Fonte: Organizado pelo autor (2023)

- E duas questões que deverão ser apresentadas ao professor e discutidas com ele, caso esteja ou não correta. De modo, que o professor retorne o *feedback* aos estudantes. Conforme o exemplo a seguir:

Figura 11 - Exemplo de pseudocódigo em Estrutura de decisão simples e composta

```

Complete o pseudocódigo para ler o nome e a idade de um indivíduo e imprimir
"criança", se a idade for menor ou igual a 13 anos, "adolescente", no caso de
idade estar entre 13 e 20 anos, e "adulto", se for mais velho.
1  inicio;
2
3  inteiro: idade;
4  escreva ("_____:");
5  leia (idade);
6  se (idade <= 13) entao
7    escreva ("Criança!");
8  senao
9    se (idade <= 20 0) entao
10     escreva ("Adolescente!");
11   senao
12     escreva ("_____!");
13   fimse;
14 fim.

 Adulto - Digite a sua idade
 Digite a sua idade - Adulto
 Adolescente - Digite a sua idade
 Digite a sua idade - Adolescente
  
```

Fonte: Organizado pelo autor (2023)

- **Discussão:** O professor finalizará a atividade realizando uma discussão com a turma sobre as principais dificuldades encontradas pelos estudantes. O Google Site, deverá ser composto por uma introdução que trate sobre os principais conceitos que precisarão ser apreendidos pelos estudantes. Considerando que em TBL os grupos discutem entre si sobre as respostas, para tentar convergir em uma resposta comum, havendo depois a intermediação do Professor. No caso, intermediação sobre os conceitos de Estrutura de Decisão Simples e Composta.

A atividade consiste em:

Discutir e aplicar os conceitos sobre Estrutura de Decisão Simples e Composta na Linguagem C, por meio da metodologia ativa TBL.

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos:

Momento 1: Preparação da aula, anteriormente ao dia da aula com TBL.

- Preparação dos materiais de apoio: Artigos, vídeos e ou simulação.
- Desenvolvimento do Google Sites disponível gratuitamente na Plataforma Google em: <https://sites.google.com/new>
- O mesmo pode seguir o exemplo disponível em: <https://sites.google.com/view/estruturadedecisaosc/in%C3%ADcio>

Momento 2: Aplicando o conhecimento aprendido.

- Neste momento caberá o professor retomar os temas tratados na aula anterior bem como discussões a partir dos materiais disponibilizados.
- Dividir de forma aleatória os estudantes em times com até quatro participantes para que realizam a atividades disponíveis no Google Sites.
- As atividades, deverão ser acompanhadas pelo professor por meio de mediação, caso os estudantes não consigam desenvolver.
- Ressalta-se que, três questões retornam feedback automático aos estudantes, de modo que podem compreender seus erros. Porém, em duas questões não retornará *feedback*, de modo que caberá ao professor corrigi-las e retornar as discussões com os estudantes

2) Quando aplicar este plano?

- Em conjunto, com o conteúdo referente a Estrutura de Decisão Simples e Composta.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Estrutura de Decisão:

- Operadores Relacionais.
- Expressões Lógicas.
- Estrutura de Decisão Composta.
- Estrutura de Decisão Simples.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Estrutura de Decisão;
- Incentivar o trabalho em time;
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Plataforma *Google Meet* (se ensino remoto);
- Moodle, para criação de salas de reunião (se ensino remoto);
- Editor de texto *word* ou Excel, para apresentação dos cálculos e times formados;
- *Feedback* do professor para os estudantes;

6) Forma de avaliação

- Resolução das atividades disponíveis *Google Sites* no, conforme exemplo disponível em:

<https://sites.google.com/view/estruturadedecisaosc/in%C3%ADcio>

7) Descrição detalhada da estratégia da aula

Aula Estrutura de Decisão Simples e Composta:

Momento 1

Caberá ao professor a preparação da aula baseada na aprendizagem ativa, anteriormente ao dia da aula com TBL. Ao qual, deverá ocorrer por meio

de materiais de apoio, tais como artigos, vídeos e ou simulações. Estes podem ser disponibilizados em sala de aula, ou por meio do Google Classroom.

Como sugestão, o professor poderá desenvolver um *Google Sites*, para disponibilizar materiais. Sugere-se o Google Sites, para que o professor, possa testar outras TICs diferenciadas. Para isso, o professor, pode seguir o exemplo disponível em:

<https://sites.google.com/view/estruturadedecisaosc/in%C3%ADcio>

Momento 2: Aplicando o conhecimento aprendido.

Neste momento caberá o professor:

- Retomar os temas tratados na aula anterior bem como discussões a partir dos materiais disponibilizados.
- Dividir de forma aleatória os estudantes em times de até quatro pessoas para que juntos, realizem a atividades disponível no *Google Sites*.

As atividades, deverão ser acompanhadas pelo professor por meio de mediação, caso os estudantes não consigam desenvolver.

Ressalta-se que, três questões retornam *feedback* automático aos estudantes, de modo que podem compreender seus “erros”. Porém, duas questões não retornarão *feedback*, de modo que caberá ao professor corrigi-las e retornar as discussões com os estudantes.

Ao final, nas atividades acima, deverá ocorrer o feedback do professor e dos estudantes.

8) Observações adicionais

Os estudantes, em time, deverão resolver as atividades em sala de aula, durante o período de 50 minutos.

Desafios:

- Resolver em grupo a questão disponibilizada no período dos 50 minutos de aula.
- Seguir os quatro princípios essenciais da TBL

1ª Distribuição adequada dos grupos:

- Formar adequadamente os grupos, de modo que não formem times “panelinhas”, minimizando assim, possíveis barreiras durante a formação dos times.

2ª Tornar o estudante responsável por seu progresso:

- Contribuir para que os estudantes, tornem-se protagonistas de suas ações por meio de responsabilidades prévias coletiva e individuais (planejamento).

3ª Promoção da interação por meio das atividades:

- Fomentar o diálogo entre os times e possíveis planos de ação futuras com os resultados das atividades (artigo, site ou game por exemplo).

4ª Constante feedback:

- O papel do professor é mediar as interações dos estudantes, neste sentido, é importante dar retorno quanto as potencialidades e fragilidades dos times ao longo de suas atividades.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Estrutura de Decisão Simples e Composta, bem como sua aplicação;
- Auxiliar na formação dos times para construção da atividade avaliativa.
- Mediar as interações dos times.
- Gerar constante feedback.

Ação dos estudantes:

- Construir a equação individual e posteriormente em time, apresentar as conclusões atingidas.

Feedback:

Entregar *feedback* dos trabalhos e atividades.

MODULARIZAÇÃO: ESTRATÉGIAS E APLICAÇÃO NA SALA DE AULA INVERTIDA

A modularização é um conceito fundamental na programação e refere-se à divisão de um programa em módulos ou partes menores e independentes. Cada módulo contém um conjunto específico de funcionalidades ou tarefas relacionadas, facilitando o desenvolvimento, manutenção e compreensão do código.

A modularização tem como objetivo principal melhorar a organização, a legibilidade e a reutilização do código. Ao dividir um programa em módulos, cada parte pode ser desenvolvida separadamente, permitindo que diferentes desenvolvedores trabalhem em partes diferentes do projeto simultaneamente. Além disso, os módulos podem ser reutilizados em diferentes partes do programa ou em projetos futuros, economizando tempo e esforço de desenvolvimento.

Existem várias vantagens associadas à modularização:

- **Organização:** A divisão em módulos permite que o código seja estruturado de forma mais clara e organizada. Cada módulo tem uma responsabilidade específica, facilitando a localização e a compreensão de diferentes partes do programa.
- **Manutenção:** Com um programa modular, é mais fácil identificar e corrigir problemas específicos, pois cada módulo é independente. Isso torna as tarefas de manutenção, como correção de bugs ou implementação de melhorias, mais gerenciáveis e menos propensas a afetar outras partes do código.
- **Reutilização:** Os módulos podem ser reutilizados em diferentes projetos, o que economiza tempo e esforço de desenvolvimento. Uma vez que um módulo é desenvolvido e testado, ele pode ser incorporado em diferentes programas ou projetos, desde que cumpra os requisitos específicos.
- **Colaboração:** A modularização permite que diferentes desenvolvedores trabalhem em diferentes módulos simultaneamente, facilitando a colaboração e agilizando o processo de desenvolvimento.

- **Testabilidade:** Os módulos podem ser testados individualmente, o que facilita a identificação e solução de problemas. Isso contribui para a qualidade e confiabilidade do programa como um todo.

Em resumo, a modularização é uma abordagem de organização e estruturação do código em partes menores e independentes, chamadas de módulos. Essa prática traz benefícios como melhor organização, manutenção mais fácil, reutilização de código e colaboração eficiente entre desenvolvedores.

Desta forma, com base no conceito de modularização, este capítulo divide-se em quatro diferentes metodologias ativas (Sala de aula Invertida, Aprendizagem baseada em Problemas, Gamificação e Aprendizagem baseada em Times), sendo uma metodologia para cada plano de aula, que tem como objetivo, apresentar os fundamentos da estrutura de decisão simples e composta, utilizando vídeo explicativo disponibilizado antecipadamente e aula expositiva, para explorar o conteúdo.

Boa leitura!

Modularização: Estratégias e Aplicação na Sala de Aula Invertida

METODOLOGIA DE ENSINO: SALA DE AULA INVERTIDA

Conteúdo: Modularização

Metodologia de ensino: Sala de Aula Invertida

Descrição da estratégia: Construção de um site no aplicativo gratuito do *Google Sites* pelos estudantes, de modo que tragam de forma sintetizada, os conceitos de modularização.

PLANO DE AULA

Metodologia Ativa: Sala de Aula Invertida

Objetivo da aula:

Apresentar, de forma lúdica, como funciona a modularização de código (funções) aos estudantes.

1) Metodologia a ser trabalhada

Baseada na metodologia da Sala de Aula Invertida, terá os seguintes procedimentos:

- Disponibilização antecipada de materiais, por meio da plataforma *WebQuest* conforme disponível em: <https://sites.google.com/view/jacksonpluskota/página-inicial>, referentes a importância da programação e da resolução de problemas; caso o professor não tenha acesso, poderá utilizar o *Moodle*. Sugere-se a *WebQuest* para que o professor possa se aproximar de outras possibilidades de uso das tecnologias de informação e comunicação - TIC.
 - Conteúdo disponibilizado no Sites Google, ao qual contará com:
 - Textos em PDF.
 - Materiais visuais para os/as alunos acessarem remotamente,

sobre os conteúdos que serão trabalhados em sala de aula, tais como no exemplo disponível em:

<https://sites.google.com/view/jacksonpluskota/p%C3%A1gina-inicial>

Tais como Procedimentos e funções e passagem de parâmetros por valor e referência.

■ Atividade de fixação presentes no próprio site, a ser realizado em link disponibilizado do Google Forms conforme o exemplo:

<https://forms.gle/SfZhDoEjqrGDSifh8>

- Em sala de aula, a aula será dividida em quatro fases:
 - Um conjunto de perguntas e respostas, sobre o material anteriormente disponibilizado.
 - Retomando as leituras disponíveis no Sites Google.
 - Repasse do dever de casa da noite anterior:
 - Discutido com os estudantes sobre as suas dúvidas e dificuldades.
 - Preleção de novo conteúdo:
 - Aplicação de um Quiz para fixação dos conceitos, como disponibilizado por Lima (2021) em: <https://quizizz.com/admin/quiz/60ac0b3ceb63c5001bf986f8/aula12-funces-e-modularizacao>
 - Prática orientada e independente e/ou atividade de laboratório:
 - Construção de uma função para fixar o conceito de modularização.

Sala de Aula Invertida

Compreendendo que a Sala de aula Invertida é a inversão da abordagem educacional comum, em que o aluno pode acessar de sua casa a aula preparada pelo professor, devendo ser interativa e provocativa (ARAÚJO, 2019, p. 21).

A metodologia para a presente aula será a apresentação de um site disponível em <https://sites.google.com/view/jacksonpluskota/p%C3%A1gina-inicial>, no qual apresentará conteúdos relacionados à modularização, utilizando textos em PDF e vídeos explicativos, além de uma atividade de fixação e sugestão de leituras complementares.

Esta atividade consiste em instigar a compreensão dos conceitos de modularização, de forma autônoma e antes da aula com o professor. Tem-se como base, a *WebQuest*, isto é, um formato de aula orientada por meio do Google Sites, com recursos que podem ser utilizados para resolver tarefas provenientes da web (BACICH, 2020) ¹.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à modularização de programas.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Modularização:

- Procedimentos e funções.
- Passagem de parâmetros por valor e referência.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Modularização;
- Incentivar o trabalho em grupo;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Disponibilização de textos introdutórios, para apresentar os conceitos de Modularização a serem discutidos em sala de aula.
- Site disponível na plataforma do *Google Sites*.
- Quiz online registrado na plataforma *Quizzes*.

6) Forma de avaliação

- Observação da participação dos alunos por meio das atividades em grupo; Análise dos formulários respondidos, compondo a nota das atividades avaliativas

¹ BACICH, Lilian. Inovação na Educação. 2020. Disponível em: <https://lilianbacich.com/2020/03/> Acesso em: 30 de abril de 2022.

7) Descrição detalhada da estratégia da aula

- Antes da aula:
 - Preparação da disciplina durante as aulas anteriores, relacionada a funções;
 - Apresentação aos estudantes, da metodologia que será utilizada ao longo das aulas;
 - Desenvolvimento do Site no *Site Google*: Vale destacar que este, é um recurso do gratuito, disponível pela conta google. Neste, basta entrar em Google Sites com a conta pessoal ou institucional do Google e desenvolver o site explicativo, conforme o objetivo da aula.
 - Disponibilização de materiais de estudos em casa, utilizando o Sites *Google*. Conforme o exemplo disponível em: <https://sites.google.com/view/jacksonpluskota/página-inicial?authuser=0> contendo:
 - Início - Apresentação do conteúdo;
 - Introdução;
 - Conceituação;
 - Vídeos norteadores;
 - Atividade de fixação;
 - Referências;
- Em sala de aula, no dia seguinte:
 - Organização da turma em grupos aleatórios para discussão das atividades disponibilizadas (20 min).
Conforme o exemplo disponível em: <https://sites.google.com/view/jacksonpluskota/praticando?authuser=0>
Atenção: A atividade, deve estar previamente desenvolvida no Google Forms, para posteriormente ser alocada no Google Sites.
 - Discussão sobre a resolução do grupo, desta atividade em forma de apresentação, por meio da indicação de um líder (20 min).
A discussão, poderá partir da construção de um mapa mental, no qual os estudantes construirão com o professor as suas considerações.
 - Feedback do professor.

8) Observações adicionais

O Sites Google é gratuito. Por meio do link, é possível construir um novo site a partir do modelo. As Atividades do Google Forms, são sugestões baseadas nos textos disponibilizados nas referências utilizadas, podendo ser alterado, com maior ou menor número de questões.

Desta forma, os links para acesso as TCIs de apoio são:

- **WebQuest** – **Sites** **Google:**
<https://sites.google.com/view/jacksonpluskota/p%C3%A1gina-inicial>
- **Google Forms:** <https://forms.gle/Y83k3qZefJj36BfF7>

Desafios:

Professor: Disponibilizar os conteúdos sobre modularização de forma interativa e diferenciada aos estudantes por meio da construção do site de forma que se torne interativo e interessante aos estudantes (designer) para acessarem conteúdos sobre a modularização.

Ação dos professores:

Construir o site para compartilhar os conteúdos de forma interativa, de maneira que todo o conteúdo seja retomado durante a aula presencial, principalmente a atividade de fixação.

Ação dos alunos:

Leitura dos materiais disponibilizados pelo professor. Acompanhamento de, ao menos um dos vídeos. Resolução da atividade de fixação, para discussão em sala de aula. Trazer suas principais dúvidas, dificuldades durante a realização da atividade.

Feedback:

Esclarecimento das dúvidas e a junção de nova metodologia durante a aula para fixação dos conceitos, como a exemplo, do *Team Based Learning* para que, durante o trabalho em grupo, possam retomar as atividades e fixação do tema.

REFERÊNCIAS

LIMA, André. Aula 12: **Funções e Modularização**. 2021. Disponível em: <https://quizizz.com/admin/quiz/60ac0b3ceb63c5001bf986f8/aula12-funces-e-modularizacao> Acesso em: 26 de janeiro de 2023.

PLUSKOTA, Jackson W. **WebQuest – Sites Google**. 2023. Disponível em: <https://sites.google.com/view/jacksonpluskota/p%C3%A1gina-inicial> Desenvolvido em: 26 de janeiro de 2023

PLUSKOTA, Jackson W. **Google Forms. 2023. Disponível em:** <https://forms.gle/Y83k3qZefJj36Bf7> _ Desenvolvido em: 26 de janeiro de 2023.

Modularização: Estratégias e Aplicação na PBL

METODOLOGIA DE ENSINO: PROBLEM BASED LEARNING

Conteúdo: Modularização

Metodologia de ensino: Aprendizagem Baseada em Problemas

Descrição detalhada da estratégia: trata-se de uma atividade realizada em sala de aula, durante o ensino do conteúdo de modularização. Está sendo considerada uma aula de 50 minutos e terá como objetivo, fomentar a compreensão dos conceitos de Modularização, por meio da resolução de problemas usando a aprendizagem ativa.

PLANO DE AULA

Metodologia Ativa: Aprendizagem baseada em Problemas

Objetivo da aula:

Fixar o conteúdo de Modularização por meio de uma aula ativa de 50 minutos baseada na metodologia ativa Aprendizagem Baseada em Problemas.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - Disponibilização aos estudantes de:
 - um vídeo de apresentação da disciplina/aula/conteúdo;
 - material didático;
 - um vídeo de visualização obrigatória e um vídeo de visualização complementar.
 - Por meio de uma plataforma digital (*Google Classroom, Moodle*).
 - Formação de grupos, dos quais, cada grupo, receberá uma situação-problema formulada pelo professor.
 - Explicações sobre a estratégia PBL e o estudo dirigido.

- Avaliação.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à modularização de programas.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Modularização:

- Procedimentos e funções.
- Passagem de parâmetros por valor e referência

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Modularização;
- Incentivar o trabalho em grupo;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Plataforma digital (Google *Classroom*, *Moodle*), para disponibilização de materiais pré-estabelecidos (textos e videoaulas);
- Apresentação no *Power Point* para o estudo dirigido, conforme exemplificado por Márquez (2013).
- Utilização do quadro para exemplificação dos estudos;
- *Feedback* do professor para os estudantes.

6) Forma de avaliação

- Resolução e envio das atividades via plataforma digital.

7) Descrição detalhada da estratégia da aula

Procedimento:

- Anteriormente ao dia aula, será apresentada a metodologia aos estudantes e informado sobre os materiais de estudos, disponíveis na plataforma digital.
- Na plataforma digital serão disponibilizados dois vídeos explicativos sobre modularização, conforme exemplos disponíveis em:

<https://sites.google.com/view/jacksonpluskota/videos?authuser=0>

Durante a aula:

- Divisão da turma em grupos com até quatro estudantes, para desenvolvimento das atividades
- Disponibilização aos grupos de uma situação problema, conforme os seguintes exemplos (SANTANA, 2013)²:
 1. Utilizando o conceito de função, faça um programa que receba os três lados de um triângulo e faça o seguinte: informar se os três lados realmente formam um triângulo e que tipo de triângulo as medidas formam.
 2. Faça um programa que receba dois números e descubra seu MMC. A identificação do MMC deve ser feita dentro de uma função que retorne o valor calculado. A leitura e a escrita dos dados, além da chamada do módulo MMC, devem ficar na função principal.
 3. Faça um programa que calcule a soma dos N primeiros números primos, sendo N definido pelo usuário na função main. O programa ainda deverá ter as funções **somaPrimos** e **primo**, sendo que a primeira função será responsável pela soma dos números que forem primos e a segunda, será responsável por verificar se o número em questão é primo ou não.
 4. Faça um programa que calcule a média aritmética de todas as turmas de uma escola. Considere como entrada os números de turmas e o número de alunos de cada turma. A média aritmética de cada turma deve ser apresentada, além de média geral, que será o resultado da média das turmas. O programa deverá ter as funções **mediaTurma** e **mediaEscola**. O usuário entrará, na função **main**, com a quantidade de turmas e, dentro da função **mediaEscola**, com o número de alunos de cada turma.
- Ao longo das discussões dos alunos, o professor deverá passar de grupo em grupo para verificar se ele não está disperso, bem como verificar as eventuais dúvidas.
- Cada grupo fará uma pergunta baseada no contexto do caso e procurará resposta dentro do material disponível para consulta, seguindo pelo método do estudo dirigido (baseado nos textos disponíveis no *Classroom*).

² Questões de exemplo, disponíveis em: <http://www.univasf.edu.br/~max.santana/material/introprog/listaExercicios04.pdf>

- Durante a aula, o professor poderá realizar algumas pausas para explicações sobre a estratégia PBL e o estudo dirigido.
- A avaliação se dará via conclusão das questões exemplificadas em Santana (2013) e entrega delas ao professor ao final da aula.

Estas questões, deverão ser retomadas na aula seguinte. Se o professor preferir, poderá solicitar que os estudantes, em grupo, terminem em casa e disponibilizem o código via *Classroom* para avaliação.

- A finalidade da estratégia, é:
 - Fomentar uma discussão nos grupos, da melhor forma de resolução do problema.
 - Resolver as questões disponibilizadas pelo professor.
 - Buscar diferentes soluções para um mesmo problema.
 - Instigar, diferentes formas de aprendizagem.

8) Observações adicionais

Na atividade os estudantes deverão explicar como chegaram ao resultado apresentado.

Desafios:

- Resolver em grupo a questão disponibilizada.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Modularização, bem como sua aplicação;
- Auxiliar na formação dos grupos para resolução da atividade disponibilizada.

Ação dos estudantes:

- Resolver a questão em grupo e apresentar o resultado.

Feedback:

- Entregar feedback dos trabalhos e atividades.

REFERÊNCIAS

MARQUEZ, Mercedes R. Gonzalez. **Modularização**. 2013. Disponível em: <http://www.comp.uems.br/~mercedes/disciplinas/2013/AEDI/AEDI-modularizacao.ppt> Acesso em: 27 de janeiro de 2023.

SANTANA, Max. **Exercícios de fixação sobre modularização**. 2013. Disponível em: <http://www.univasf.edu.br/~max.santana/material/introprog/listaExercicios04.pdf> Acesso em: 27 de janeiro de 2023.

Modularização: Estratégias e Aplicação de na Gamificação

METODOLOGIA DE ENSINO: GAMIFICAÇÃO

Conteúdo: Modularização

Metodologia de ensino: Gamificação

Descrição da estratégia:

Aula baseada em um Escape Room, sob a narrativa de uma equipe de trabalhadores que, atuam em uma empresa de programação e deverão desativar uma “casa autônoma/inteligente”. Esta missão, demandará dos estudantes, a resolução de pequenos quebra-cabeças (*Quizzes*) para chegar aos códigos de acesso à sala do servidor, para desativação do sistema da casa.

PLANO DE AULA

Metodologia Ativa: Gamificação.

Objetivo da aula:

Apresentar a modularização de funções, por meio de uma estratégia de ensino gamificada.

1) Metodologia a ser trabalhada

Baseada na metodologia da gamificação, serão utilizados os seguintes procedimentos:

- € Disponibilização de materiais antecipadamente, referente a relevância da programação e da resolução de problemas.
- € Disponibilização em ambiente virtual de aprendizagem de material em Power Point e/ou vídeos explicativos, sobre os conteúdos que serão trabalhados em sala de aula.
- € Criação de um Escape Room para aplicação prática da estratégia gamificada. Destacando que *Escape Room* (também conhecido como sala de

fuga, jogo de fuga ou sala de escape).

€ Aplicação de um *Quiz* para fixação dos conceitos.

Gamificação

- Para a presente aula, os estudantes terão acesso a uma narrativa referente a uma “casa inteligente”. O texto apresentando instiga a participação dos estudantes para a conclusão de missões, com a finalidade de buscar a compreensão dos conceitos de modularização.

- A atividade ocorrerá a partir de uma atividade coletiva, por meio da formação de grupos para realizar uma missão.
- Haverá um tempo cronometrado para realizar as missões;
- Haverá um prêmio para os vencedores.

A finalidade consiste em: Instigar a compreensão dos conceitos de modularização.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à modularização de programas.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Modularização:

- Procedimentos e funções. Passagem de parâmetros por valor e referência.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Modularização;
- Incentivar o trabalho em grupo;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Disponibilização de textos introdutórios para apresentar os conceitos a serem discutidos em sala de aula.
- Escape Room disponível na Plataforma do *Google Forms*.
- Quiz online disponível na plataforma *Quizzes*, disponível em: <https://quizizz.com/embed/quiz/6263146c4a71fc001d7c8efb>
- Feedback do professor para os alunos.

6) Forma de avaliação

- Observação da participação dos alunos por meio das atividades em grupo;
- Análise dos formulários respondidos (*quizz* e *Escape Room*).

7) Descrição detalhada da estratégia da aula

€ **Momento 1:**

Preparação das aulas

€ Primeiramente, será importante preparar as aulas que serão trabalhadas com os estudantes.

€ Caberá ao professor, construir a história/narrativa que será tratada em sala de aula e será a base para o *Escape Room*.

€ Como no seguinte exemplo:

A empresa Plus, localizada no vale do silício (EUA), criou uma IA para casa autônomas. Porém, um update no sistema com um bug, fez com que a casa não funcione direito, deixando todos os moradores trancados sem internet, alimentos ou água. A única maneira de abrir a casa e salvar a todos, é chegando até o servidor central da casa, onde a IA está instalada.

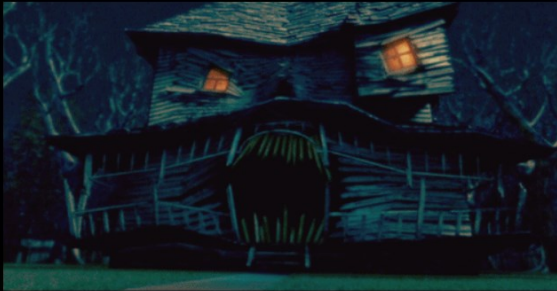
Precisamos de ajuda da melhor equipe desta empresa para resetarmos esta casa. Contudo, somente podem participar desse desafio, aqueles que estejam em processo de estudos sobre modularização. Como exemplo, indica-se acessar: [A casa inteligente.html](#)

- Estando com narrativa pronta, juntamente com as questões, que irá aplicar aos estudantes, caberá ao professor, acessar <https://twinery.org/> para acessar e se inscrever com suas contas institucionais ou do próprio Gmail no aplicativo da *Twinery* e no *Quizizz* disponível em: <https://quizizz.com/> ambos aplicativos, que possuem versões gratuitas.
- Por seguinte, caberá ao professor construir a narrativa, em sequência de esquema no *Twiner* e as questões no *quizzer*.
- Visualmente, o estudante terá a seguinte informação:

Figura 12 - Exemplo Design Twiner

A empresa Plus localizada no vale do silício (EUA) criou uma IA para casa autônomas. Porém, um update com bug fez com que a casa não funcione direito, deixando todos trancado dentro dela sem internet, alimentos ou água e, a única maneira de abrir a casa e salvar a todos, é chegando até o servidor central da casa onde a IA se encontra.

Precisamos de ajuda da melhor equipe desta empresa para resetarmos esta casa, contudo, somente podem participar a equipe que esteja em processo de estudos sobre modularização.

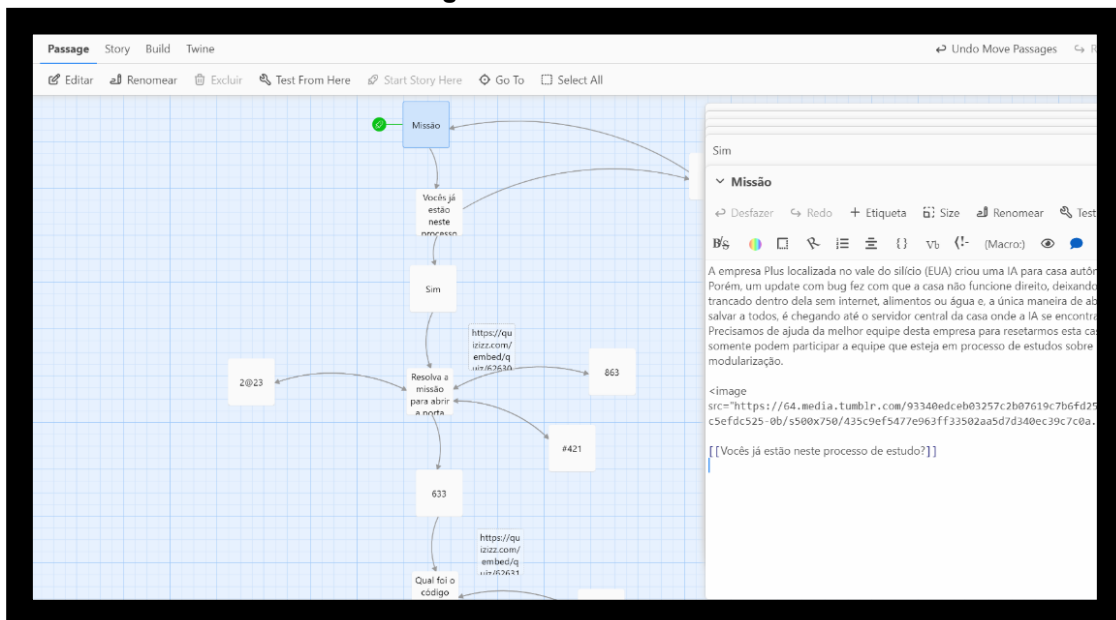


Vocês já estão neste processo de estudo?

Fonte: Organizado pelo autor (2023)

Por sua vez, a visão do professor, será a seguinte:

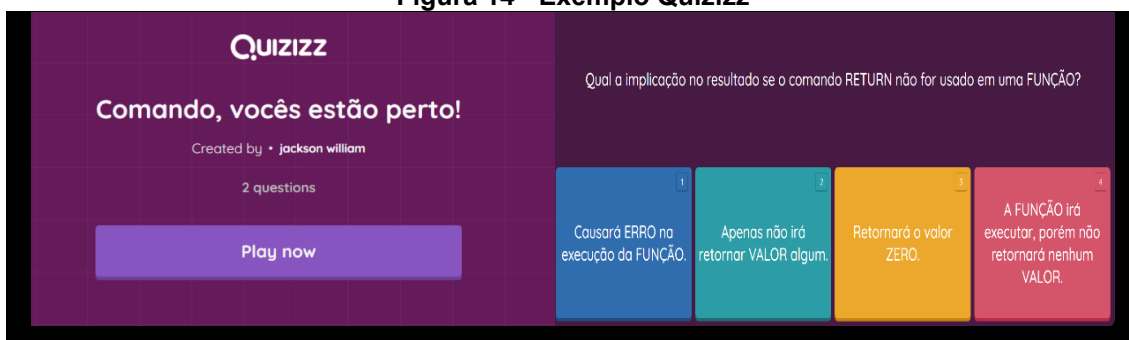
Figura 13 - Back Twiner



Fonte: Organizado pelo autor (2023)

Por este exemplo, é importante, estar com as questões disponíveis no quizzer, logo, a atividade, passa a ser construída ao mesmo tempo. Assim, no quizzer, os estudantes terão a seguinte visualização:

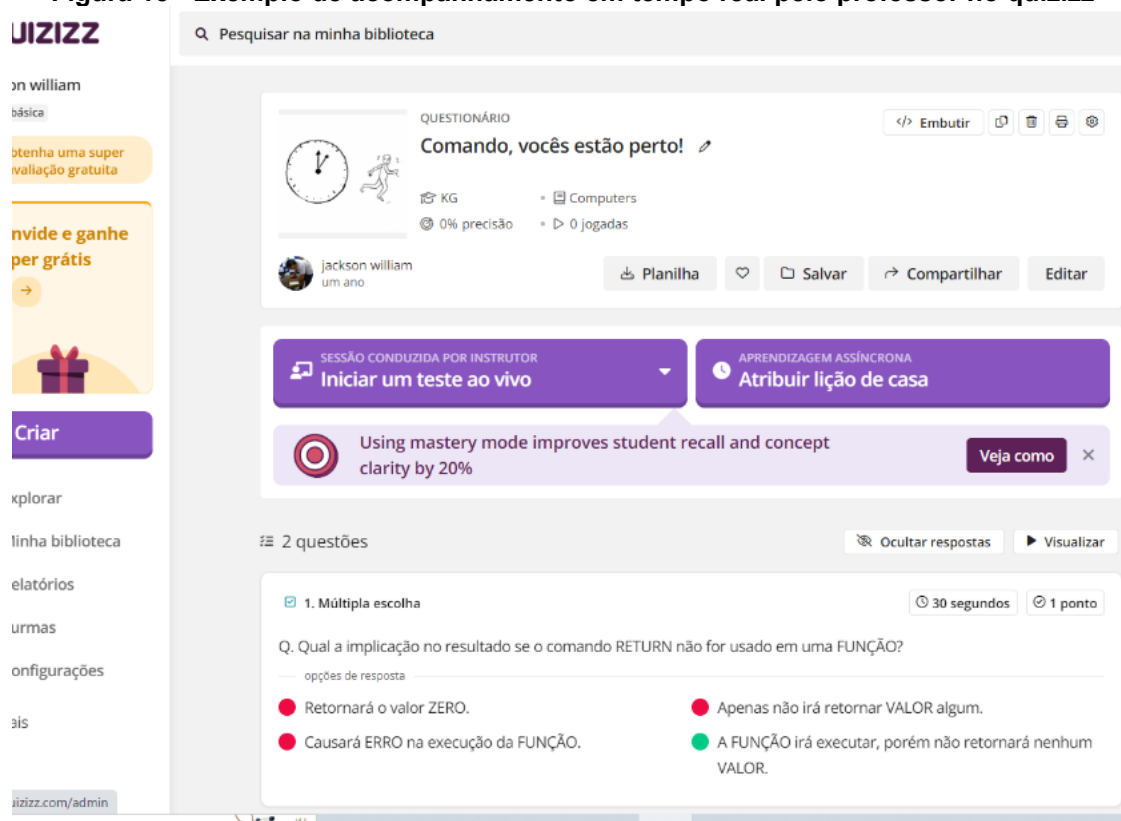
Figura 14 - Exemplo Quizizz



Fonte: Organizado pelo autor (2023)

É válido destacar que o professor, poderá acompanhar em tempo real, as informações respondidas ou não pelos estudantes, por meio de sua conta de administrador, como no exemplo a imagem a seguir:

Figura 15 - Exemplo de acompanhamento em tempo real pelo professor no quizizz



Fonte: Organizado pelo autor (2023)

☒ MOMENTO 2

Aplicando as aulas com a turma

- Na primeira aula, para introdução a modularização será apresentado aos estudantes a metodologia que será desenvolvida na sala de aula.
- Em seguida, será lembrado aos estudantes sobre os textos

disponibilizados anteriormente na sala de aula (5 min). Trata-se da fase de aquecimento da turma.

- Será disponibilizado aos estudantes, um *Escape Room* que ocorrerá da seguinte maneira:
 - Explicação da aula e da missão que ocorrerá durante a aula.
 - Disponibilização do link para acesso a *Twinery* ou pasta para que os estudantes o baixem no computador.
 - Leitura da narrativa disponível na plataforma *Twinery* (disponível em *Twine*). Vale ressaltar que para disponibilizar aos estudantes, sem precisar baixar no computador, é importante subir o arquivo HTML no servidor de hospedagem.
 - Acompanhamento da realização das missões.
 - Orientação aos estudantes que apresentem possíveis dificuldades na resolução dos exercícios da missão.
 - Entrega dos prêmios aos vencedores;
- O jogo será disponibilizado aos alunos para, em um período de 15 minutos, resolverem as missões disponibilizadas.

€ **Após a aplicação do jogo:** Dialogar com a turma, sobre suas dificuldades e facilidades com as atividades, buscando anotá-las para retomar na próxima aula.

8) Observações adicionais

Escape Room: Vale destacar que *Escape Room* ou *Fuga de sala*, é um tipo de atividade de entretenimento onde os participantes são trancados em uma sala tematicamente decorada e cheia de quebra-cabeças, enigmas e pistas. O objetivo é resolver esses desafios em equipe para encontrar uma saída da sala dentro de um limite de tempo específico, geralmente 60 minutos.

Os enigmas e quebra-cabeças podem variar amplamente em dificuldade e estilo, podendo incluir decifrar códigos, resolver quebra-cabeças físicos, manipular objetos, decodificar mensagens secretas, analisar pistas visuais e sonoras, entre outras atividades. A história ou tema da sala também desempenha um papel importante, adicionando imersão e contexto à experiência.

Escape rooms são frequentemente projetados para estimular a colaboração em equipe, o pensamento crítico, a criatividade e a comunicação entre os participantes. Eles têm ganhado popularidade como uma atividade de lazer, tanto para grupos de amigos quanto para atividades corporativas de construção de equipes.

Além das versões físicas, onde os participantes estão fisicamente presentes em uma sala real, também existem *escape rooms* virtuais que podem ser jogados online, muitas vezes envolvendo elementos de realidade virtual ou jogos digitais interativos.

Quanto aos Prêmios: O prêmio será direcionado pela participação dos alunos e pelo trabalho em equipe.

Links: Os links utilizados para acompanhamento do desenvolvimento dos alunos são:

- Q1: <https://quizizz.com/embed/quiz/62631962978c12001d879fe2>
- Q2: <https://quizizz.com/embed/quiz/6263146c4a71fc001d7c8efb>
- Q3: <https://quizizz.com/embed/quiz/6263047fd3b05c001d5ca7a9>
- Q4: <https://quizizz.com/embed/quiz/60c8ae7f532369001b7ca315>
- Q. **Final:**
https://docs.google.com/forms/d/1H65eIWAT0LPB0SHbyTaJNsluHwD6qJiv_ULj-YmPXLw/edit (edição) e <https://forms.gle/P91uLd2mK3GTBUQu7> (respostas)

Desafios:

- Professores: Planejar a aula e adicioná-las nos aplicativos.
- Estudantes: Concluir as missões do *escape room*.

Ação dos professores:

- Repassar o escape e fomentar a formação de equipes para discutir sobre a narrativa e a lógica presente nela.

Ação dos alunos:

- Seguir as indicações do escape, realizar as missões baseadas em leituras anteriores sobre a temática.

Feedback:

- Entregar um prêmio de participação para os alunos/as.

REFERÊNCIAS

Lima, André. **Aula Modularização de código no portugol studio**. Disponível em: <https://quizizz.com/join/quiz/61939d914b6e10001e46987f/start> Acesso em: 05 de maio de 05 de maio de 2022.

Lima, André. **Aula 12 de funções e modularização**. Disponível em: <https://quizizz.com/join/quiz/60ac0b3ceb63c5001bf986f8/start?from=admin> Acesso em: 05 de maio de 05 de maio de 2022.

PLUSKOTA, Jackson W. Texto APK Twine, 2022. Disponível em: [A casa inteligente.html](#) Desenvolvido em maio outubro de 2022.

PLUSKOTA, Jackson W. APK Twine, 2022. Disponível em: [Twine](#) Desenvolvido em maio outubro de 2022

Modularização: Estratégias e Aplicação de na TBL

METODOLOGIA DE ENSINO: APRENDIZAGEM BASEADA EM TIMES

Conteúdo: Modularização

Metodologia de ensino: Aprendizagem Baseada em Times

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de modularização, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de Modularização e a resolução de problemas, por meio do desenvolvimento de funções e a aprendizagem ativa. A atividade será dividida em dois momentos: 1) A apresentação de um “menu de equações”; e, 2) a resolução das funções que resolvem as equações, em um período de 50 minutos em sala de aula presencial.

PLANO DE AULA

Metodologia Ativa: Aprendizagem baseada em Times

Objetivo da aula:

- Desenvolver uma aula de 50 minutos utilizando metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação.

Fixar o conteúdo de Modularização utilizando a metodologia ativa *Team Based Learning* (TBL) ou Aprendizagem Baseada em Times.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - Editor de textos *Word* (ou similar) ou editor de planilhas *Excel* (ou similar)

- Em caso de aula remota, pode ser adotada a plataforma Google Meet e *Moodle* (ou similar).
- A atividade terá duração de 50 minutos.
- Para o desenvolvimento da aula deverão estar preparados alguns desafios, disponíveis em documentos do *Word* ou Excel:
 - Uma
 - listagem de seis cálculos simples, distribuídos aos alunos de acordo com a ordem alfabética do livro de chamadas. ex:

Tabela 1 - Listagem de seis cálculos simples- Lista de Chamada (sorteio)

Lista de chamada:	Lista de cálculo:	Distribuição das atividades:
estudante 1		estudante 1 -> Cálculo 1
estudante 2	Cálculo 1	estudante 2 -> Cálculo 2
estudante 3	cálculo 2	estudante 3 -> Cálculo 1
estudante 4		estudante 4 -> Cálculo 2
estudante 5		estudante 5 -> Cálculo 1
estudante 6		estudante 6 -> Cálculo 2

Fonte: Organizado pelo autor (2023)

- Adicionalmente, é criada uma listagem contendo os grupos que foram formados e o desafio que eles devem resolver., seguindo a distribuição dos cálculos simples a partir da lista de chamada. Ex:

Tabela 2 - Listagem contendo os grupos que foram formados pela tabela 1

Time A	Time B
estudante 1 -> Cálculo 1	estudante 2 -> Cálculo 2
estudante 3 -> Cálculo 1	estudante 4 -> Cálculo 2
estudante 5 -> Cálculo 1	estudante 6 -> Cálculo 2

Fonte: Organizado pelo autor (2023)

- Se a aula for remota, utilizar o Moodle e criar salas de reunião para cada time, para que os estudantes se comuniquem durante a atividade em time.

Especificamente, durante o dia da aula.

Caberá ao professor seguir os passos da metodologia de ensino, baseada na TBL. Tais como:

A aplicação da metodologia será realizada da seguinte forma:

- **Apresentação da atividade “menu de equações”:**
 - “Menu de equações”: será um código-fonte apresentado pelo professor no qual unirá todas as funções para serem testadas por ele e pelos estudantes.

Neste passo, o professor mostrará o “menu” com as opções, sendo o código que gera o menu, disponibilizado para os estudantes. Caberá as equipes, elaborarem as funções que serão chamadas pelo programa que contém o menu de operações.
- **Sorteio das equações:** será realizado utilizando o livro de chamada para que cada função seja distribuída pelo professor, para o estudante na ordem que estiver registrado na chamada (conforme exemplificado anteriormente). Caberá ao professor considerar as possibilidades de estudantes em sala de aula e o tempo de 50 minutos.
 - **Proposta para resolução das equações:** Após o sorteio das equações, será proposto ao estudante, de forma individual, descobrir como resolver essas equações utilizando a estratégia de função/modularização.
 - **Resolução das equações:** Em seguida, entre cinco a dez minutos de trabalho individual, os estudantes serão agrupados em times de estudantes, a partir do critério de que tenham equações iguais (sorteadas/distribuídas anteriormente para resolver individualmente) para discutir seus resultados e encontrar a melhor solução.
- **Implementação da equação:** Cada equipe deverá implementar uma função final para ser disponibilizada no “menu de equações”.
- **Teste do menu:** E por seguinte, poderão testar o “menu de equações” (cujo código será disponibilizado pelo professor), com as funções de cada time.

A finalidade consiste em:

- Fomentar o entendimento dos conceitos e benefícios do uso de módulos e entender como utilizar funções que já existem:
 - Definição;
 - Forma de utilização;
 - Vantagens de uso;
 - Parâmetros;
 - Identificador;
 - Valor de retorno;

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos:

- **Momento 1:** Apresentação do Problema: Como criar um programa “Menu de Equações?”
 - O desenvolvimento inicial ocorrerá de maneira individual, em torno de 20 minutos. Aleatoriamente, utilizando o livro de chamada, serão sorteadas para cada estudante, equações que devem ser resolvidas. Uma sugestão de cálculos simples, são:
 - 1-Cálculo do volume de um cubo;
 - 2-Cálculo da magnitude de um vetor;
 - 3-Cálculo da área de um quadrado;
 - 4-Cálculo de Bhaskara;
 - 5-Conversão monetária;
 - 6-Cálculo da área de um círculo, entre outros, a critério do professor.

Por seguinte, será indicado para o estudante, que individualmente, descubra uma forma de transcrever essas equações em uma função. Isto é, somente descrever em um documento word, sobre como será transcrita a equação em uma função e, como o estudante fará isso (lembrando que esse momento será em apenas 15 minutos).

Momento 2: Programação das equações.

- Neste momento serão desenvolvidas as funções em linguagem de programação, para serem adicionadas na ferramenta chamada de “menu de equações”. Este código-fonte será preparado (e disponibilizado) pelo professor

em que unirá todas as funções a serem criadas e testadas por ele e pelos estudantes.

- Este momento ocorrerá após o trabalho individual (nos 10 minutos seguintes).

- O professor irá apresentar para turma em um documento Excel ou Word os times formados a partir do sorteio individual em que, a partir do "momento 1" (após o sorteio de um dos seis problemas para cada estudante, logo, até quatro estudantes que fizeram a mesma equação formarão um time), construirão as funções a serem implementadas no código-fonte disponibilizado pelo professor, no sistema "menu de equações". Caso a atividade seja feita por meio remoto, será usado o *Moodle* e criado salas virtuais para cada time antes da aula, para os estudantes discutirem e para o professor observar e orientar cada time. A atividade baseada em problemas ocorrerá por meio de:

- Discussão coletiva: Uma discussão entre o time sobre a melhor forma de resolução desse problema, para a criação de uma função (de acordo com o resultado da atividade individual).

- Desenvolvimento de uma função final, da equação sorteada: esta função, irá agregar em um sistema de "menu de equações". Esta atividade deverá ocorrer durante 20 minutos.

- Teste: Cada time testará o "menu de equações" com as funções dos outros times, para ver o resultado do sistema "menu de equações".

Ao final do 'momento 2', nos últimos cinco minutos haverá um feedback do professor e dos estudantes.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à modularização de programas.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Modularização:

- Procedimentos e funções.
- Passagem de parâmetros por valor e referência.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Modularização;

- Incentivar o trabalho em time;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Plataforma Google Meet (se ensino remoto);
- Moodle, para criação de salas de reunião (se ensino remoto);
- Editor de texto word ou Excel, para apresentação dos cálculos e times formados;
- Feedback do professor para os estudantes;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em time;
Código do “menu de equações” onde se trata do código-fonte que o professor disponibilizará para os times testarem suas funções junto com as funções dos outros times, seguindo as etapas/critérios estabelecidos e trabalhados em aula.

É válido destacar que, para criar um menu de equações em um programa, você pode utilizar a modularização e uma estrutura de repetição, como um loop, para exibir o menu repetidamente até que o usuário escolha sair.

Aqui está um exemplo simples de como criar um menu de equações em Python:

```
def soma(a, b):  
    return a + b  
  
def subtracao(a, b):  
    return a - b  
  
def multiplicacao(a, b):  
    return a * b  
  
def divisao(a, b):  
    if b != 0:
```

```

        return a / b
    else:
        return "Erro: divisão por zero!"

# Loop principal do programa
while True:
    print("=== Menu de Equações ===")
    print("1. Soma")
    print("2. Subtração")
    print("3. Multiplicação")
    print("4. Divisão")
    print("0. Sair")

    opcao = int(input("Escolha uma opção: "))

    if opcao == 0:
        print("Saindo do programa...")
        break

    elif opcao >= 1 and opcao <= 4:
        num1 = float(input("Digite o primeiro número: "))
        num2 = float(input("Digite o segundo número: "))

        if opcao == 1:
            resultado = soma(num1, num2)
        elif opcao == 2:
            resultado = subtracao(num1, num2)
        elif opcao == 3:
            resultado = multiplicacao(num1, num2)
        elif opcao == 4:
            resultado = divisao(num1, num2)

        print("Resultado:", resultado)
        print()

    else:
        print("Opção inválida. Tente novamente.")
        print()

```

Neste exemplo, temos quatro funções que representam diferentes operações matemáticas (soma, subtração, multiplicação e divisão). O programa exibe um menu de opções e solicita ao usuário que escolha uma opção digitando o número correspondente.

Dependendo da opção selecionada, o programa solicita ao usuário que digite os números necessários para realizar a operação. Em seguida, a função apropriada é chamada para calcular o resultado e o resultado é exibido na tela.

O loop principal mantém o programa em execução até que o usuário escolha a opção "0" para sair. Quando essa opção é escolhida, o programa exibe uma mensagem de saída e o loop é interrompido, encerrando o programa.

Destaca-se, que, o professor, pode adicionar mais equações ou funcionalidades conforme necessário, bem como personalizar o formato de apresentação dos resultados. Esse exemplo é apenas uma base para você começar a criar um menu de equações em seu programa.

7) Descrição detalhada da estratégia da aula

Aula Modularização:

- Será proposto para que os estudantes, inicialmente, desenvolvam uma atividade individual, em que, de forma aleatória serão distribuídos problemas envolvendo equações simples, tais como:
 - **Cálculo do volume de um cubo:** Como neste exemplo em vamos criar uma função chamada "calcular_volume_cubo" que recebe o valor do lado do cubo como parâmetro e retorna o volume calculado. Neste exemplo, a equação, seguirá a seguinte estrutura:

```
def calcular_volume_cubo(lado):  
    volume = lado ** 3  
    return volume  
  
# Exemplo de uso da função  
lado_cubo = 5  
volume_cubo = calcular_volume_cubo(lado_cubo)  
print("O volume do cubo é:", volume_cubo)
```

Nesse exemplo, a função "calcular_volume_cubo" recebe o parâmetro lado, que representa o comprimento de um dos lados do cubo. O volume é

calculado pela fórmula $V = \text{lado}^3$, onde `**` é o operador de exponenciação em Python.

Dentro da função, calculamos o volume e o atribuímos à variável `volume`. Em seguida, retornamos esse valor usando a instrução `return volume`.

Fora da função, podemos usar a função `calcular_volume_cubo` passando o valor do lado desejado (no exemplo, usamos um lado de 5 unidades). O valor retornado é armazenado na variável `volume_cubo`. Por fim, exibimos o resultado na tela usando a função `print`.

Esse é apenas um exemplo básico de como usar a modularização para calcular o volume de um cubo em uma linguagem de programação. A modularização permite que você encapsule o cálculo do volume em uma função reutilizável, facilitando a organização e manutenção do código.

- **Cálculo da magnitude de um vetor:** Nesse caso, vamos criar uma função chamada `calcular_soma_vetor` que recebe um vetor como parâmetro e retorna a soma dos elementos desse vetor. Sendo representado na seguinte estrutura:

```
def calcular_soma_vetor(vetor):
    soma = 0
    for elemento in vetor:
        soma += elemento
    return soma

# Exemplo de uso da função
meu_vetor = [1, 2, 3, 4, 5]
soma_vetor = calcular_soma_vetor(meu_vetor)
print("A soma dos elementos do vetor é:", soma_vetor)
```

Nesse exemplo, a função `calcular_soma_vetor` recebe o vetor como parâmetro. Inicializamos uma variável `soma` com o valor zero para acumular a soma dos elementos.

Em seguida, utilizamos um loop `for` para percorrer cada elemento do vetor. Em cada iteração do loop, somamos o elemento atual à variável `soma`.

Após percorrer todos os elementos do vetor, retornamos o valor final da soma usando a instrução “return” soma “calcular_soma_vetor” passando “meu_vetor” como parâmetro. O valor retornado é armazenado na variável “soma_vetor”. Por fim, exibimos o resultado na tela utilizando a função print.

Ressalta-se, que este, é apenas um exemplo básico de como calcular a soma de um vetor em Python utilizando a modularização. Você pode adaptar essa lógica para atender às suas necessidades específicas de manipulação de vetores. A modularização permite encapsular o cálculo da soma em uma função reutilizável, tornando o código mais organizado e facilitando a manutenção.

- Seguindo estes exemplos, os professores poderão indicar ainda, situações como:
 - Cálculo da área de um quadrado:
 - Cálculo de Bhaskara:
 - Conversão monetária:
 - Cálculo da área de um círculo, entre outros, a critério do professor.
- E então, será indicado para os estudantes, individualmente, descobrir uma forma de transcrever essas equações em uma função, somente descrever em um documento word, como será transcrita a equação em uma função e como fará isso.
- Após o trabalho individual, será sugerida a formação de times, a partir da atividade individual acima (considerando que foi realizado o sorteio de uma das seis atividades para cada estudante, que fizeram a mesma equação formarão um time) para construção das funções a serem implementadas no código do professor no sistema "menu de equações".
 - Sua finalidade será:
 - Fomentar uma discussão entre o time da melhor forma de resolução desse problema, para a criação de uma função (de acordo com o resultado da atividade individual).
 - Desenvolver uma função final, da equação sorteada: que irá agregar em um sistema de “menu de equações”.
 - Testagem: Neste, cada time testará o “menu de equações” com as funções dos outros times, para ver o resultado do sistema “menu de equações”.

Ao final, do processo haverá um feedback do professor e dos estudantes.

8) Observações adicionais

Na atividade de elaboração de uma função em time, que será colocado no “menu de equações”, os times deverão explicar como chegaram ao resultado apresentado.

Desafios:

- Sequenciar e apresentar as atividades de avaliação.
- Seguir os quatro princípios essenciais da TBL

1ª Distribuição adequada dos grupos:

- Formar adequadamente os grupos, de modo que não formem times “panelinhas”, minimizando assim, possíveis barreiras durante a formação dos times. Sendo então, grupos de cinco e sete pessoas, que permaneçam nos mesmos times ao longo do semestre.

2ª Tornar o estudante responsável por seu progresso:

- Contribuir para que os estudantes, tornem-se protagonistas de suas ações por meio de responsabilidades prévias coletiva e individuais (planejamento).

3ª Promoção da interação por meio das atividades:

- Fomentar o diálogo entre os times e possíveis planos de ação futuras com os resultados das atividades (artigo, site ou game por exemplo).

4ª Constante feedback:

- O papel do professor é mediar as interações dos estudantes, neste sentido, é importante dar retorno quanto as potencialidades e fragilidades dos times ao longo de suas atividades.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Modularização, bem como sua aplicação;
- Auxiliar na formação dos times para construção da atividade avaliativa.

- Mediar as interações dos times.
- Gerar constante feedback

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Modularização, bem como sua aplicação;
- Auxiliar na formação de equipes para construção da atividade avaliativa.
- Apresentar constantes feedback aos times.

Ação dos estudantes:

- Construir a equação individual e posteriormente em time, apresentar as conclusões atingidas.

Feedback:

Entregar feedback dos trabalhos e atividades.

ESTRUTURAS DE REPETIÇÃO: ESTRATÉGIAS E APLICAÇÃO NA SALA DE AULA INVERTIDA

Em programação, as estruturas de repetição, também conhecidas como loops, são utilizadas para executar um bloco de código repetidamente até que uma condição seja satisfeita. Essas estruturas são fundamentais para automatizar tarefas repetitivas e reduzir a redundância no código.

Existem geralmente três tipos principais de estruturas de repetição:

1. *Loop "while"* (enquanto): Nesse tipo de loop, uma condição é verificada antes de cada iteração. Enquanto a condição for verdadeira, o bloco de código dentro do loop é executado repetidamente. Assim que a condição se tornar falsa, a execução continua para o próximo trecho de código após o loop.

Exemplo em pseudocódigo:

```
enquanto (condição) faça
    // bloco de código a ser repetido
fim do loop
```

2. Loop "for" (para): O *loop "for"* é usado quando o número de iterações é conhecido antecipadamente. Ele geralmente usa uma variável de controle que é incrementada ou decrementada a cada iteração, e a condição de término é baseada no valor dessa variável.

Exemplo em pseudocódigo:

```
para (inicialização; condição; atualização) faça
    // bloco de código a ser repetido
fim do loop
```

Essas são as estruturas de repetição mais comuns em programação. Elas fornecem maneiras flexíveis de controlar o fluxo de execução e executar tarefas repetitivas de forma eficiente. Assim, com base no conceito de Estruturas de Repetição, este capítulo divide-se em quatro diferentes metodologias ativas (Sala de aula Invertida, Aprendizagem baseada em Problemas, Gamificação e Aprendizagem baseada em Times), sendo uma metodologia para cada plano de

aula, que tem como objetivo, apresentar os fundamentos da estrutura de decisão simples e composta, utilizando vídeo explicativo disponibilizado antecipadamente e aula expositiva, para explorar o conteúdo.

Boa leitura!

Estruturas de Repetição: Estratégias e Aplicação na Sala de Aula Invertida

METODOLOGIA DE ENSINO: SALA DE AULA INVERTIDA

Conteúdo: Estruturas de Repetição

Metodologia de ensino: Sala de Aula Invertida

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino do conteúdo de Estrutura de Repetição, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos principais conceitos da estrutura de repetição.

PLANO DE AULA

Metodologia Ativa: Sala de Aula Invertida

Objetivo da aula:

- Desenvolver uma aula de 50 minutos por meio da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.
- Fixar o conteúdo de Estruturas de Repetição utilizando a metodologia ativa Sala de aula Invertida.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - Disponibilização do *link* para o aplicativo “Fábrica de jogos”: [Laços de Repetição \(fabricadeaplicativos.com.br\)](https://laçosde repetição.fabricadeaplicativos.com.br) , para promover a discussão do tema com os estudantes.

- Explicações sobre a estratégia da Sala de aula invertida por meio da disponibilização do aplicativo, para os estudantes terem acesso ao assunto antes da aula.

Avaliação: verificação dos estudantes que responderam ao questionário disponível no aplicativo “Fábrica de jogos”. As perguntas estarão disponíveis no *Google Form*, sendo acessível por meio do *link* <https://forms.gle/n1KxBwxWbM3StvGP7>

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à Estruturas de repetição.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de estruturas de repetição na Linguagem de Programação C:

- *while*;
- *do... while*;
- *for*.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de estruturas de repetição;
- Aprendizado cooperativo;
- Autonomia do estudante
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Aplicativo disponível na plataforma fabricadejogos.com;
- Apresentação em *Power Point* para o estudo dirigido, conforme disponível por Augusto (2018).
- Utilização do quadro negro para exemplificação dos estudos;
- Lista de exercícios para serem desenvolvidos em sala de aula, por meio de trabalho em grupo;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;

- Envio das resoluções via ambiente virtual de aprendizagem Via *Google Classroom*, e respostas ao questionário disponível no aplicativo “Fábrica de jogos”.

7) Descrição detalhada da estratégia da aula

Aula estruturas de repetição

Procedimento:

- Anteriormente ao dia aula, será apresentada a metodologia aos estudantes e informado sobre os materiais de estudos, disponíveis na sala no *Google Classroom*.
- Ainda, antes da aula, será disponibilizado por meio do *Google Classroom* três vídeos explicativos, sendo estes:
 - Um vídeo de apresentação do conteúdo, material didático e;
 - Um vídeo de visualização obrigatória, como no exemplo disponível no *link*: https://pwa.fabapp.com/lacos_de_repeticao/youtube/25863926/OLf1Q42ulPE e;
 - Um vídeo de visualização complementar, como no exemplo disponível no *link*: <https://www.youtube.com/watch?v=Fu3RBQ7y7Cw>
 - Ainda, será disponibilizado o *link* para o aplicativo via QR CODE:

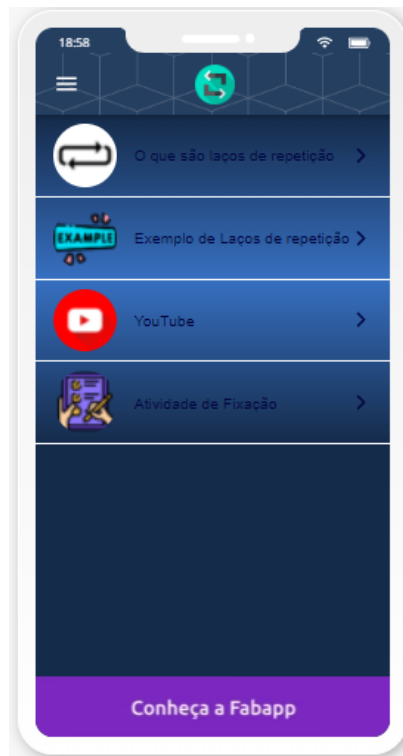
Figura 16 - Qr Code Fábrica de jogos



Fonte: Organizado pelo autor (2023)

- Também disponível em https://app.vc/lacos_de_repeticao, preparado pelo professor na plataforma Fábrica de jogos, com conteúdos relacionados a estruturas de repetição.
- Por meio do *link*, os estudantes terão acesso ao aplicativo, com a aparência exemplificada na Figura 17:

Figura 17 - Acesso ao aplicativo Fabrica de jogos



Fonte: Organizado pelo autor (2023)

- Por meio deste, o professor poderá disponibilizar quantos conteúdos forem necessários, em forma de textos, atividades, vídeos, podcasts entre outros.

Durante a aula:

- Será lembrado o aviso sobre a metodologia da aula, e por seguinte, por meio do livro de chamadas, de forma aleatória, serão formados grupos com até quatro estudantes. Cada grupo, receberá um conjunto de atividades de fixação, tais como:
 - Com base nos textos disponibilizados no fábrica de jogos, desenvolva e/ou explique:
 - O que são estruturas de repetição?
 - Quais são os tipos de estruturas de repetição?
 - De acordo com Mariano (2020): O laço _____ determina que enquanto uma determinada condição for válida, o bloco de código será executado.
 - O comando _____ permite que uma variável contadora seja testada e incrementada a cada iteração, sendo essas informações definidas na chamada do comando.

Ao logo das discussões nos grupos, o professor deverá passar de grupo em grupo para verificar as eventuais dúvidas.

8) Observações adicionais

Ao final da atividade os estudantes deverão explicar como chegaram ao resultado apresentado.

Desafios:

- Resolver em grupo a questão disponibilizada.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre estruturas de repetição, bem como sua aplicação;
- Auxiliar na formação dos grupos para resolução da atividade disponibilizada.

Ação dos estudantes:

- Resolver a questão em grupo e apresentar o resultado.

Feedback:

- Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

AUGUSTO, João. **Estrutura ou laços de repetição**. 2018. Disponível em: <http://docente.ifsc.edu.br/joao.augusto/MaterialDidatico/2018-1/Introdução%20à%20Programação/JavaScript/Laços%20de%20Repetição.pdf>
Acesso em 11 de agosto de 2022.

PLUSKOTA, Jackson W. Exercícios sobre laços de repetição. 2022. Disponível em: <https://forms.gle/n1KxBwxWbM3StvGP7> Desenvolvido em: 11 de agosto de 2022.

METODOLOGIA DE ENSINO: PROBLEM BASED LEARNING

Conteúdo: Laços de Repetição

Metodologia de ensino: Aprendizagem Baseada em Problemas

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Laços de Repetição, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de Estruturas de Repetição e a resolução de problemas, por meio de funções e aprendizagem ativa.

PLANO DE AULA

Metodologia Ativa: Aprendizagem Baseada em Problemas

Objetivo da aula:

- Desenvolver uma aula de 50 minutos por meio da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação.

Fixar o conteúdo de Estrutura de Repetição, utilizando a metodologia da aprendizagem baseada em problemas.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - *Sublime Text.*
 - Problema a ser compartilhado com a turma
- A atividade apresentada será realizada em uma aula de 50 minutos.
- Para o desenvolvimento da aula deverão estar preparados:

- Ao menos dois textos para leitura sobre o tema de Estruturas de Repetição (preferencialmente, a leitura deve ocorrer antes da aula).
- Utilizar o problema a seguir, para desenvolver a autonomia do estudante na resolução:
 - Desenvolver um programa que leia a altura de, pelo menos 15 pessoas da turma. Este programa deverá calcular e mostrar:
 - a. A menor altura do grupo;
 - b. A maior altura do grupo;

Especificamente, durante o dia da aula.

- Situar os estudantes sobre o que irão aprender e quais metodologias serão utilizadas para o processo de aprendizagem, através de apresentação que pode ser realizada via slides ou quadro negro.
- Em sala de aula, apresentar a atividade de fixação e solicitar que, em grupos, os estudantes resolvam o problema: Como desenvolver um programa que leia a altura de, pelo menos 15 pessoas da turma, apresentando, a menor e a maior altura do grupo?
- **Metodologia de ensino: Aprendizagem Baseada em Problemas**
- Por meio da narrativa, instigar os estudantes que, em grupo, resolvam o problema de:
 - Descobrir a altura dos estudantes de sua turma;
 - Desenvolver um algoritmo que leia a altura de ao menos 15 pessoas da sala/turma;
 - Apresentar o resultado em grupo ao professor;

A finalidade consiste em:

- Fomentar o entendimento dos conceitos e benefícios do uso de estrutura de repetição;

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos ao qual compreenderá em cinco etapas:

Etapa 1 - Estabelecendo relações com o problema: Dialogar com os estudantes sobre quais informações e conhecimentos prévios eles possuem a respeito dos conceitos e benefícios do uso de repetição.

Etapa 2 - Estabelecer um plano de trabalho para a resolução do problema: A partir das discussões realizadas com auxílio do professor e os estudantes listar informações que forem obtidas por meio da própria apresentação do problema, bem como dos seus conhecimentos prévios sobre o assunto.

Etapa 3 – Abordagem do problema: Construir um método objetivo e eficiente para solucionar o problema apresentado a partir da prática de atividades de aprendizagem de forma colaborativa.

Etapa 4 - Reequacionando o problema: Na mesma aula, ou na aula seguinte, após os trabalhos (individual e coletivo) dos estudantes, cada grupo deverá apresentar um relatório do trabalho desenvolvido. A partir deste relatório, avaliar o desenvolvimento do problema e possíveis melhorias.

Etapa 5 - Elaborando e apresentando os produtos: Avaliação, nesta etapa, caberá ao professor, fomentar um processo diferenciado de avaliação, do qual, some as percepções do professor, dos estudantes e dos grupos em si, da atividade total.

Por exemplo:

Momento 1:

- **Retomada** – Resgatar os conhecimentos dos estudantes sobre estratégia de estruturas de repetição.
- **Ação:** Resgatar os conhecimento e estratégias de estruturas de repetição em programação;
- **Tempo:** 11min.

Momento 2:

- **Situação Problema:** Como desenvolver um algoritmo que leia a altura de ao menos 15 pessoas da sala/turma? Apresentando, a menor a maior altura do grupo?
- **Atividade principal** - Informar aos alunos sobre o funcionamento da atividade e fazê-lo utilizando a estratégia da ABP;
- **Ação:** Organizar os grupos, determinar as funções de seus componentes, seguido as regras propostas, por seguinte resolver o desafio.
- **Tempo:** 18 min.

Momento 3:

- **Discussão das soluções** – discutir e compartilhar os percursos trilhados na resolução do desafio e as estratégias utilizadas para resolver o desafio.
- **Ação:** Consolidar a estratégia da ABP e seus significados relacionados a estruturas de repetição.
- **Tempo:** 4 min.

Momento 4:

- **Raio X:** Observar e consolidar as aprendizagens dos estudantes sobre o uso estratégia da APP para efetuar o cálculo da estrutura de repetição;
- **Ação:** Utilizar os saberes da estratégia apreendida em sala de aula para revolucionar a questão aplicada pelo (a) professora e posteriormente comparar a resolução dentre os colegas sem sala de aula.

Tempo: 12 min

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente às estruturas de repetição.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de estruturas de repetição:

- *While*;
- *Do... while*;
- *For*.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de laços de repetição;
- Incentivar o trabalho em grupo;
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Editor de texto *Sublime Text*;
- Computador ou notebook (considera-se o trabalho em grupo, caso, na turma haja estudantes sem computadores);
- Fita métrica;
- Folha A4 (para possíveis anotações);
- Caneta/lapís para anotação;
- *Feedback* do professor para os estudantes.

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;

7) Descrição detalhada da estratégia da aula

Aula Estruturas de Repetição:

- Aplicação da questão problematizadora. Isto é, as questões exemplos:
 - Simule um conceito de contador em ***While*** em loop infinito;
 - Simule um conceito de contador em ***Do...while*** em loop infinito
 - Simule um conceito de contador em *for* em *loop* infinito
- Ao final, nas missões, haverá um *feedback* do professor e dos estudantes.

8) Observações adicionais

Na atividade de realização do problema, o professor deve acompanhar o desenvolvimento das atividades nos grupos. Os estudantes podem solicitar dicas do professor e de outros grupos, favorecendo o trabalho colaborativo.

Desafios:

- Explicar o conceito de laços de repetição com exemplos em sala de aula e direcionar exercícios que possam ser demandados pelos estudantes.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre laços de repetição, bem como sua aplicação;
- Auxiliar na formação de equipes para construção da atividade das missões.

Ação dos estudantes:

- Realizar o problema indicado pelo professor e apresentar em sala de aula.

Observações:

- Ainda cabe como ações dos estudantes e professores de acordo com Camargo (2019, p. 117):

Quadro 5 - Ações dos professores e estudantes em Estruturas de Repetição.

Estudantes	Professores
Reconhecer o sucesso no trabalho dos outros, além de se reconhecerem como colaboradores e fornecedores de recursos didáticos entre si (avaliação entre pares); Ser agente da própria aprendizagem, isto é, reconhecer o sucesso em seu próprio trabalho e se concentrar em como e o que estão aprendendo e onde ainda podem melhorar (autoavaliação).	Esclarecer e compartilhar as intenções de aprendizagem e critérios para obter sucesso: para que os alunos compreendam o que estão tentando aprender, por que e o que se espera deles; Organizar discussões e atividades efetivas em sala de aula de forma a evidenciar a aprendizagem: para criar um ambiente na sala de aula onde os alunos possam expor suas próprias ideias, pensar em voz alta e explorar a sua compreensão; Fornecer feedback que seja capaz de promover a aprendizagem: referenciando a qualidade de seu trabalho e o que os alunos podem fazer para torná-lo melhor;

Fonte: Organizada pelo autor (2023)

Feedback:

Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

CARVALHO, Melisse. **Estratégias para avaliação na aprendizagem baseada em problemas.** In LOPES, Renato Matos. **APRENDIZAGEM BASEADA EM PROBLEMAS: Fundamentos para a aplicação no ensino médio e na formação de professores.** 2019. Disponível em: <https://educapes.capes.gov.br/bitstream/capes/432641/2/APRENDIZAGEM%20BASEADA%20EM%20PROBLEMAS%20-%20fundamentos%20para%20a%20sua%20aplica%C3%A7%C3%A3o%20no%20Ensino%20M%C3%A9dio%20e%20na%20Forma%C3%A7%C3%A3o%20de%20Professores.pdf> Acesso em: 26 de janeiro de 2023.

MARIANO, Diego. **Laços de Repetição.** 2020. Disponível em: <https://diegomariano.com/lacos-de-repeticao-2/#:~:text=La%C3%A7os%20de%20repeti%C3%A7%C3%A3o%2C%20tamb%C3%A9m%20conhecidos,bloco%20sejam%20repetidos%20diversas%20vezes.> Acesso em: 02 de maio de 2022.

UNICAMP. **Estrutura de Repetição.** 2011. Disponível em: <https://www.ic.unicamp.br/~wainer/cursos/2s2011/Cap06-RepeticaoControle-texto.pdf> Acesso em: 02 de maio de 2022.

SUGESTÕES DE ATIVIDADES PARA EXEMPLOS EM SALA DE AULA, BASEADAS EM MARIANO (2020):

1. Simule um conceito de contador em **While** em loop infinito;

```
# Contador $i
my $i = 0;
while($i < 10){
    print "$i "; # 0 1 2 3 4 5 6 7 8 9
    $i++;
}
```

2. Simule um conceito de contador em **Do...while** em loop infinito

```
# Contador $i
my $i = 0; do
{
    print "$i "; # 0 1 2 3 4 5 6 7 8 9
    $i++; }while($i < 10);
```

3. Simule um conceito de contador em for em *loop* infinito

```
#                               Iteracao                               for
for(my $i = 0; $i < 10; $i++){
    print $i; # 0123456789
}
```

Estruturas de Repetição: Estratégias e Aplicação na Gamificação

METODOLOGIA DE ENSINO: GAMIFICAÇÃO

Conteúdo: Estruturas de Repetição

Metodologia de ensino: Gamificação

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Laços de Repetição, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de Repetição e a resolução de problemas por meio de funções e a aprendizagem ativa.

PLANO DE AULA

Metodologia Ativa: Gamificação

Objetivo da aula:

- Desenvolver uma aula de 50 minutos por meio da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação.
- Fixar o conteúdo de Estruturas de Repetição utilizando a metodologia ativa da Gamificação

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - *Sublime Text.*
 - *Google Classroom.*

- *App. Goosechase*³
- A atividade apresentada será realizada durante uma aula de 50 minutos.
- Para o desenvolvimento da aula deverão estar preparados no Google Classroom:
 - Ao menos dois textos para leitura antes da aula, sobre o tema de Estruturas de Repetição.
 - Grupos com até quatro alunos, organizados de forma aleatória, para a realização da missão.
 - Atividades pré-estabelecidas no aplicativo *Goosechase*.

Especificamente, durante o dia da aula.

- Em sala de aula, apresentar atividade disponível no aplicativo *Goosechase* (<https://www.goosechase.com/games/>), construídos pelo professor, solicitando que os alunos acessem o link disponibilizado e, por meio do código de acesso, entrem no app e em grupo, realizem as missões.

Metodologia de ensino: Gamificação

- A aplicação da metodologia Gamificação será realizada da seguinte forma:
 - Preparação do aplicativo *Goosechase* anteriormente a aula de Estruturas de repetição;
 - Explicação aos estudantes sobre a aula diferenciada que será realizada;
 - Solicitação aos estudantes para acessarem o aplicativo via *Play Store* (*Android*).
 - Repasse das informações para início das missões compostas no aplicativo (formação de grupo, selfie do grupo, missões no aplicativo, e missões distribuídas no bloco do curso);
 - Apresentação da atividade disponível no App *Goosechase*
 - Após a apresentação, o professor deverá dividir os estudantes em grupos para se cadastrarem no *Goosechase* e iniciarem as missões.

³ “*Goosechase*” é um App Criado em 2011 a partir de uma *Hackthoon*, que funciona como um mecanismo para construir jogos e missões. Geralmente utilizado para fomentar a interação entre grupos. Seu download é gratuito, bem como a criação de missões. Há opções para contas pró (pagas). Contudo, para o básico em educação, é uma possibilidade para Gamificar o ensino, ofertando missões e pontuação a cada missão realizada.

- Em seguida, o professor deverá acompanhar o desenvolvimento dos estudantes.
- A realização das missões, podem ser visualizadas em tempo real pelo professor.
- Cada equipe deverá realizar as missões o mais rápido possível.
- E por seguinte apresentar a função desenvolvida.

A finalidade da atividade consiste em:

- Fomentar o entendimento dos conceitos e benefícios do uso de uma estrutura de repetição, tais como:
 - *While;*
 - *Do... while;*
 - *For;*

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos:

Momento 1 – Momento antes da aula: Entrar no link: <https://www.goosechase.com/>

- Criar uma conta para “educação”, conforme o exemplo da imagem 18.

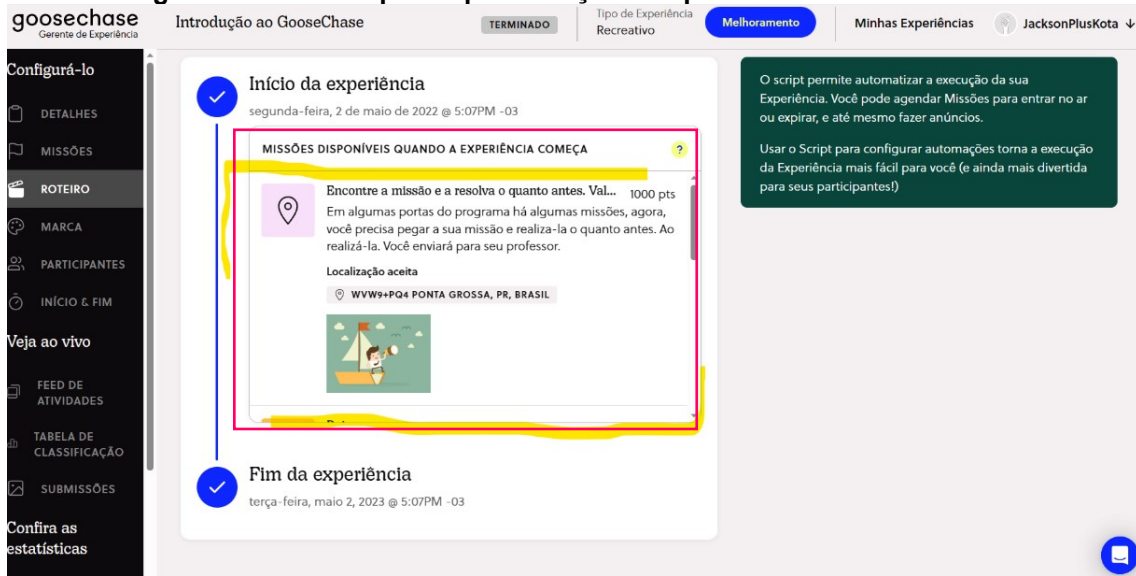
Figura 18 - Passo a passo para criação do aplicativo no Goosechase 1



Fonte: Organizado pelo autor (2023)

- Criar os conteúdos baseados na disciplina, como neste exemplo da Figura 19.

Figura 19 - Passo a passo para criação do aplicativo no Goosechase 2



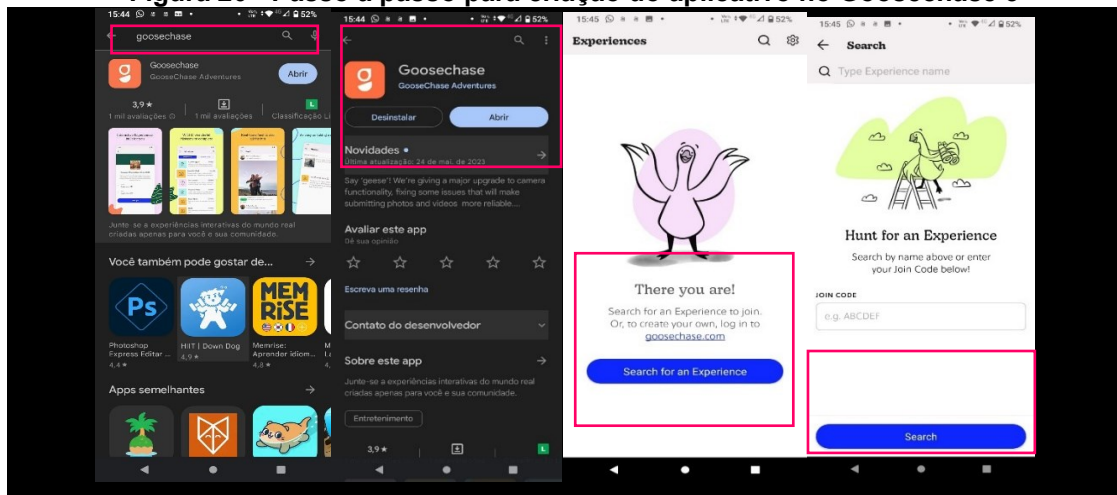
Fonte: Organizado pelo autor (2023)

- Preparar o campo para a realização da missão, por exemplo:
 - Recados distribuídos pelo bloco do curso na universidade, com as missões que devem ser realizadas (em locais que sejam acessíveis, seguros e que tenham tempo hábil de se realizar em 30 minutos).

Momento 2 – No dia da aula:

- Socializar com os estudantes o aplicativo: Indicando-os que baixem em seus celulares, via *Play Store*, conforme a imagem a seguir.

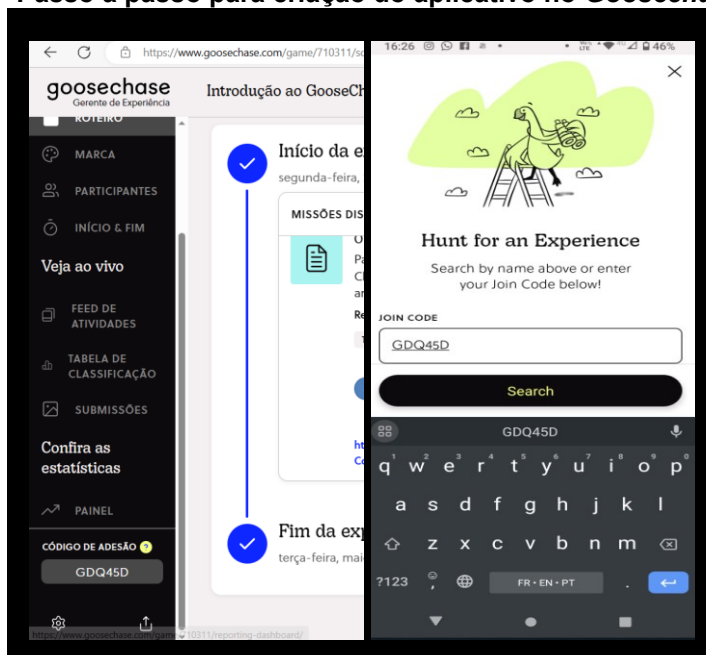
Figura 20 - Passo a passo para criação do aplicativo no Goosechase 3



Fonte: Organizado pelo autor (2023)

- Baixado o aplicativo, solicitar que os estudantes acessem via código gerado por sua atividade, por exemplo na atividade modelo: Código - GDQ45D;

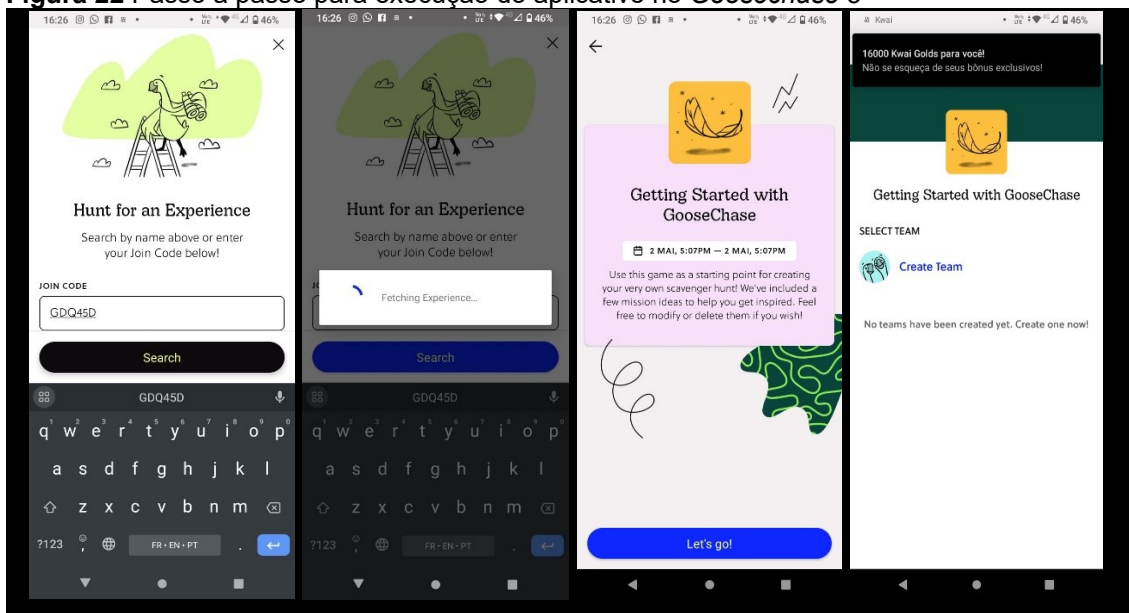
Figura 21 - Passo a passo para criação do aplicativo no Goosechase 4



Fonte: Organizado pelo autor (2023)

- Após inserir o código, indicar aos estudantes que aguardem o carregamento, cliquem no botão de iniciar e criem seus times, conforme a sequência a seguir:

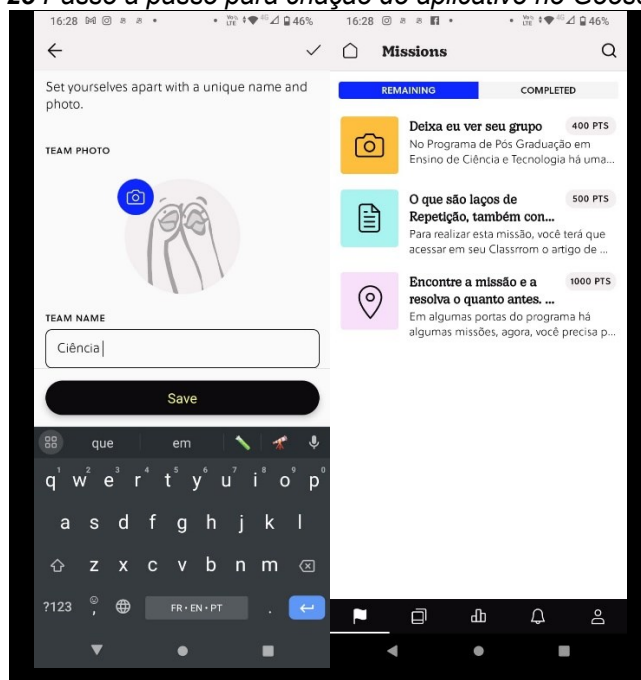
Figura 22 Passo a passo para execução do aplicativo no Goosechase 5



Nota: Organizado pelo autor.

- Criado o time, caberá ao professor, indicar aos estudantes que iniciem as missões, indicadas no aplicativo, tais como: Fotografar o grupo, responder as questões da missão e encontrar as novas missões distribuídas no bloco do curso, conforme a imagem a seguir.

Figura 23 Passo a passo para criação do aplicativo no Goosechase 6



Nota: Organizado pelo autor

- Os grupos, poderão ser distribuídos aleatoriamente;
- O professor poderá acompanhar, via conta de professor, o andamento da missão.

Ao final do 'momento 2', nos últimos cinco minutos poderá ser dedicado para um feedback do professor e dos estudantes.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à Estruturas de repetição

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de laços de repetição:

- *While;*
- *Do... while;*
- *For.*

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de laços de repetição;
- Incentivar o trabalho em grupo;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Google Classroom para disponibilização dos textos bases e indicação de leituras (além do texto base da missão, se preferir);
- Editor de texto *Sublime Text*;
- *App Goosechase*

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;
Realização das missões

7) Descrição detalhada da estratégia da aula

Aula Estruturas de Repetição:

- Será proposto para os estudantes baixarem o *app Goosechase*, para realizarem as missões contidas no aplicativo.
- Ao final, nas missões, haverá um *feedback* do professor e dos estudantes.

8) Observações adicionais

Na atividade de realização das missões, o professor pode acompanhar o desenvolvimento dos grupos, via aplicativo. Os estudantes podem solicitar dicas do professor.

Desafios:

- Explicar o aplicativo, baixar o mesmo no celular e solicitar a criação de grupos.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre laços de repetição, bem como sua aplicação;
- Auxiliar na formação de equipes para construção da atividade das missões.

Ação dos estudantes:

- Realizar as missões e enviar ao professor.

Feedback:

- Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

GOOSECHASE. **Acesso desenvolvimento.** Disponível em: <https://www.goosechase.com/games/> Acesso em: 02 de maio de 2022.

MARIANO, Diego. **Laços de Repetição.** 2020. Disponível em: <https://diegomariano.com/lacos-de-repeticao-2/#:~:text=La%C3%A7os%20de%20repeti%C3%A7%C3%A3o%2C%20tamb%C3%A9m%20conhecidos,bloco%20sejam%20repetidos%20diversas%20vezes.> Acesso em: 02 de maio de 2022.

UNICAMP. **Estrutura de Repetição.** 2011. Disponível em: <https://www.ic.unicamp.br/~wainer/cursos/2s2011/Cap06-RepeticaoControle-texto.pdf> Acesso em: 02 de maio de 2022.

SUGESTÕES DE ATIVIDADES PARA MISSÃO BASEADAS EM MARIANO (2020):

- Simule um conceito de contador em **While** em loop infinito;

```
# Contador $i

my $i = 0;

while ($i < 10){

    print "$i "; # 0 1 2 3 4 5 6 7 8 9

    $i++;

}
```

- Simule um conceito de contador em **Do...while** em loop infinito

```
# Contador $i

my $i = 0; do

{print "$i "; # 0 1 2 3 4 5 6 7 8 9

    $i++; }while($i < 10);
```

- Simule um conceito de contador em **for** em loop infinito

```
# Iteracao for

for (my $i = 0; $i < 10; $i++){

    print $i; # 0123456789

}
```

Estruturas de repetição: Estratégias e Aplicação na Estratégia TBL

METODOLOGIA DE ENSINO: TEAM BASED LEARNING

Conteúdo: Laços de Repetição

Metodologia de ensino: Aprendizagem Baseada em Times

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Repetição, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de estrutura de Repetição e a resolução de problemas por meio de funções e a aprendizagem ativa.

PLANO DE AULA

Metodologia Ativa: Aprendizagem baseada em times

Objetivo da aula:

- Desenvolver uma aula de 50 minutos através da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação

Fixar o conteúdo de estruturas de Repetição utilizando a metodologia TBL

1) Metodologia a ser trabalhada

Procedimento:

Para o desenvolvimento da aula serão necessários:

- **Anteriormente a aula:**
 - Disponibilização de textos e vídeos explicativos aos estudantes através da plataforma do Google Classroom para leitura e conhecimento prévio do conteúdo a ser trabalhado em sala de aula.

- Explicações sobre a estratégia da TBL em aula, para que os alunos, compreendam a lógica da aula seguinte.
- **No dia da aula:**
 - Explicação sobre os comandos *While; Do... while; For* (20 min);
 - Disponibilização de atividades para os estudantes realizarem em sala de aula, em times, para fixação do conteúdo.

Avaliação: verificação dos estudantes que responderam ao questionário disponível no aplicativo “Fábrica de jogos”.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à Estruturas de repetição.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de estruturas de repetição:

- *While;*
- *Do... while;*
- *For;*

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de estruturas de repetição;
- Incentivar o trabalho em times;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- *Classroom;*
- *Power Point* para o estudo dirigido;
- Utilização do quadro negro para exemplificação dos estudos;
- Lista de exercícios em sala de aula para trabalho em grupo;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;
- Envio das resoluções via *Google Classroom*.

7) Descrição detalhada da estratégia da aula

Aula laços de repetição

Procedimento:

Anteriormente ao dia aula:

- Será apresentada a metodologia aos estudantes e informado sobre os materiais de estudos, disponíveis na plataforma *Google Classroom*.
- Ainda, antes da aula, serão disponibilizados, por meio do Google Classroom, dois vídeos explicativos, sendo estes:
 - um vídeo de apresentação da disciplina/aula/conteúdo, material didático e;
 - um vídeo de visualização obrigatória e; um vídeo de visualização complementar.

Durante a aula:

- Será lembrado o aviso sobre a metodologia da aula, e por seguinte, utilizando o livro de chamada, de forma aleatória, serão formados grupos com até quatro estudantes. Cada grupo receberá a descrição das atividades de fixação, tais como:
 - Com base nos textos disponibilizados no *classroom*, desenvolva e/ou explique em times de até 4 pessoas:
- **Escreva um programa que leia um número inteiro positivo ($n > 1$) e imprima sua fatoração em números primos.**

Possível resposta:

```
n = int(input("Digite um número inteiro positivo: "))
divisor = 2
while n != 1:
    if n % divisor == 0:
        print(divisor)
        n = n / divisor
    else:
        divisor = divisor + 1
```

- **Escreva um programa que leia um número inteiro positivo ($n > 1$) e determine se ele é primo.**

```
n = int(input("Digite um número inteiro positivo: "))
primo = True
for divisor in range(2, int(n**0.5)+1):
    if n % divisor == 0:
        primo = False
        break
if primo:
    print("Primo")
else:
    print("Composto")
```

- Ao logo das discussões dos alunos, o professor deverá passar de grupo em grupo para verificar se o grupo não está disperso, bem como as eventuais dúvidas.
- De acordo com a resolução das atividades, cada time será redirecionado para os outros grupos, para colaborar com a resolução do problema, para que assim, todos possam se ajudar.

Ao final, quando todos resolverem, os estudantes, deverão indicar um líder para explicar como resolveram seus problemas.

8) Observações adicionais

Na atividade os estudantes deverão explicar como chegaram ao resultado apresentado.

Desafios:

- Resolver em grupo a questão disponibilizada.
- Seguir os quatro princípios essenciais da TBL

1ª Distribuição adequada dos grupos:

- Formar adequadamente os grupos, de modo que não formem times “panelinhas”, minimizando assim, possíveis barreiras durante a formação dos times. Sendo então, grupos de cinco e sete pessoas, que permaneçam nos mesmos times ao longo do semestre.

2ª Tornar o estudante responsável por seu progresso:

- Contribuir para que os estudantes, tornem-se protagonistas de suas ações por meio de responsabilidades prévias coletiva e individuais (planejamento).

3ª Promoção da interação por meio das atividades:

- Fomentar o diálogo entre os times e possíveis planos de ação futuras com os resultados das atividades (artigo, site ou game por exemplo).

4ª Constante *feedback*:

- O papel do professor é mediar as interações dos estudantes, neste sentido, é importante dar retorno quanto as potencialidades e fragilidades dos times ao longo de suas atividades.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre estruturas de repetição, bem como sua aplicação;
- Auxiliar na formação dos times para construção da atividade avaliativa.
- Mediar as interações dos times.
- Gerar constante *feedback*.

Ação dos estudantes:

- Resolver a questão em time e apresentar o resultado.

Feedback:

Entregar *feedback* dos trabalhos e atividades.

VETORES: ESTRATÉGIAS E APLICAÇÃO COM METODOLOGIAS ATIVAS

Em programação, um vetor é uma estrutura de dados que armazena uma coleção de elementos do mesmo tipo, organizados de forma sequencial na memória. Também conhecido como *Array*, um vetor é uma maneira eficiente de armazenar e acessar conjuntos de dados relacionados.

Cada elemento em um vetor possui um índice que representa sua posição na sequência. O índice começa a partir de 0 para o primeiro elemento e vai até o tamanho do vetor menos 1 para o último elemento. Dessa forma, é possível acessar e manipular os elementos individuais do vetor usando seu índice.

Vetores podem ser utilizados para armazenar qualquer tipo de dados, como números inteiros, números de ponto flutuante, caracteres, *strings* ou objetos mais complexos. Eles oferecem uma maneira conveniente de organizar e manipular coleções de dados relacionados em uma única estrutura.

Ao utilizar um vetor, você pode:

- Armazenar uma lista ordenada de elementos.
- Acessar elementos individuais por meio de seus índices.
- Adicionar ou remover elementos (dependendo da linguagem de programação e suas características específicas).
- Percorrer o vetor usando loops para processar ou manipular os elementos.
- Realizar operações e cálculos em conjuntos de elementos do vetor.
- Realizar pesquisas, ordenações e outras operações comuns em estruturas de dados.

Vetores são amplamente utilizados em programação devido à sua simplicidade e eficiência. Eles são fundamentais para a organização e manipulação de dados em muitas aplicações, desde algoritmos básicos até sistemas mais complexos.

É importante ter em mente que a indexação dos vetores começa em 0 em muitas linguagens de programação populares, como *Python*, *C*, *C++* e *Java*.

No entanto, algumas linguagens podem permitir o uso de índices começando em 1 ou oferecer estruturas de dados semelhantes com diferentes nomes, mas com funcionalidades semelhantes.

Desta forma, com base no conceito de vetores, este capítulo divide-se em quatro diferentes metodologias ativas (Sala de aula Invertida, Aprendizagem baseada em Problemas, Gamificação e Aprendizagem baseada em Times), sendo uma metodologia para cada plano de aula, que tem como objetivo, apresentar os fundamentos da estrutura de decisão simples e composta, utilizando vídeo explicativo disponibilizado antecipadamente e aula expositiva, para explorar o conteúdo.

Boa leitura!

Vetores: Estratégias e Aplicação na Sala de Aula Invertida

METODOLOGIA DE ENSINO: SALA DE AULA INVERTIDA

1. **Descrição detalhada da estratégia:** Uma atividade em aula presencial no qual será realizada durante o ensino de Vetores, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos principais conceitos de Vetores.

PLANO DE AULA

Metodologia Ativa: Sala de Aula invertida.

Objetivo da aula: Desenvolver uma aula de 50 minutos por meio da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial. E fixar o conteúdo de vetores utilizando a metodologia ativa Sala de aula Invertida.

1) Metodologia a ser trabalhada:

Procedimento:

- Para o desenvolvimento da aula, serão necessários:

Anteriormente ao dia específico da disciplina de Vetores:

- Disponibilização dos conteúdos para estudos individuais aos estudantes por meio *Google Classroom*.
- Explicações sobre a estratégia da Sala de aula invertida por meio da disponibilização dos conteúdos para os estudantes na plataforma de aprendizagem, com *slides* explicativos e vídeo aula, como no exemplo disponível em: [Vetores em linguagem C partiner @thaispluskota6424 - YouTube](#)

No dia da aula:

- Revisão sobre os vídeos disponibilizados na plataforma de ensino e aprendizagem e do questionário online.
- Slides e atividades para revisão dos conceitos tratados no vídeo explicativo, disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf>

Avaliação:

- Verificação dos estudantes que responderam ao questionário disponível *Google Forms* que serão visualizados e corrigidos pelo professor, conforme disponível em: <https://forms.gle/tsBhgWSksU7SabLw8>.

2) Quando aplicar este plano?

Durante aplicar a disciplina de introdução a programação em linguagem C.

3) Qual (is) conteúdo (s) será (ão) trabalhado (s)?

Conteúdos de estruturas de dados na linguagem de programação C.

- *Arrays*.
- Vetores em pseudocódigo.
- Tipos de vetores.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Vetores;
- Incentivo a cultura do acesso a pré-aula.
- Aprendizagem ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentação lista de exercícios, vídeos, jogos, ...):

- Vídeo explicativo, disponível em: <https://youtu.be/e1RnZ66PCgE>
- Apresentação em *Power Point* para o estudo dirigido como o exemplo de Salamon (s/a): <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf>
- Lista de exercícios em sala de aula para trabalho em grupo, como o exemplo que poderá ser acessado em: <https://forms.gle/tsBhgWSksU7SabLw8>.

- *Feedback* do professor para os estudantes.

6) Forma de avaliação:

- Observação da participação dos estudantes nas atividades em sala de aula;

7) Descrição detalhada da estratégia da aula:

Aula de Vetores:

Procedimentos:

Para o desenvolvimento da aula, serão necessários:

Anteriormente ao dia específico da disciplina de Vetores:

- Criação de um grupo de *WhatsApp* para diálogo com os estudantes.
- Disponibilização dos conteúdos para estudos individuais aos estudantes por meio do grupo de *WhatsApp* aos estudantes.
- Criação de um vídeo explicativo, como no exemplo disponível em:
- Explicações sobre a estratégia da Sala de aula invertida aos estudantes.
- *Google Forms* para criação de atividades para os estudantes resolverem em casa ou em sala de aula, conforme exemplo disponível em: <https://forms.gle/tsBhgWSksU7SabLw8>

No dia da aula

Revisão do conteúdo, conforme disponibilizado no vídeo disponível em:

Aplicação dos Slides e atividades para revisão dos conceitos disponível em:

<http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf>

Discussão sobre o que são vetores, suas sequências e aplicação prática.

Avaliação: Verificação dos estudantes que responderam ao questionário disponível no *Google Forms* Criado pelo professor. Como no exemplo disponibilizado em: <https://forms.gle/tsBhgWSksU7SabLw8>

8) Observações adicionais:

Esta aula remete a uma aula de 50 minutos. Logo, todo material, deve ser preparado anteriormente ao dia da aula

Desafios: Desenvolver o vídeo explicativo e acompanhar o desenvolvimento dos estudantes.

Ação dos professores: Apresentar os conceitos e o conteúdo sobre vetores, bem como sua aplicação.

Ação dos estudantes: Resolver a questão disponibilizada no *Google forms* e apresentar ao professor.

Feedback: Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

PLUSKOTA, J. W. **Aula sobre Vetores**. 2023. Disponível em: [Vetores em linguagem C partiner @thaispluskota6424](#) Acesso em: 13 de março de 2023.

PLUSKOTA, J. W. **Aula sobre Vetores: Atividades de fixação**. 2023. Disponível em: <https://forms.gle/k3ARuBTp143B85SQ6> Acesso em: 13 de março de 2023.

SALAMON. Jordana S. **Programação básica de computadores**. S/A. Departamento de informática. Universidade Federal do Espírito Santos. Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf> Acesso em: 13 de março de 2023.

Vetores: Estratégias e Aplicação na Aprendizagem Baseada em Problemas

METODOLOGIA DE ENSINO: PROBLEM BASED LEARNING

Conteúdo: Vetores

Metodologia de ensino: Aprendizagem Baseada em Problemas

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Vetores, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de Vetores e a resolução de problemas por meio da aprendizagem baseada em problemas.

PLANO DE AULA

Metodologia Ativa: Aprendizagem Baseada em Problemas

- **Objetivo da aula:** Desenvolver uma aula de 50 minutos por meio da utilização de metodologias ativas de ensino e aprendizagem e, TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.

Fixar o conteúdo de Vetores utilizando a metodologia da aprendizagem baseada em problemas.

1. Metodologia a ser trabalhada

Procedimento:

Anteriormente ao dia da aula:

- Preparação da aula, com conteúdo explicativo, sobre o assunto que será trabalhado em sala de sobre Vetores via *Power Point* ou vídeo explicativo.

- Preparação de ao menos uma atividade de fixação a ser trabalhada em sala de aula.

No dia da aula:

- Em sala de aula, apresentar atividade de fixação das aulas anteriores e solicitar que individualmente, os estudantes resolvam as seguintes situações-problemas:

- a) Como fazer um programa que peça 3 números inteiros ao usuário, armazene em um vetor e depois mostre o valor de cada elemento do vetor, assim como seu índice?
- b) Como fazer um programa em C que peça ao usuário duas notas que ele tirou e mostre a média. Usando somente um vetor para essas três variáveis?

1) Metodologia de ensino: ABP

Através da atividade online disponível no *google forms*, instigar aos estudantes, que individualmente, resolvam o problema de:

- Realizar as situações-problemas e apresentar o resultado em sala de aula.

A finalidade consiste em: Fomentar o entendimento dos conceitos e aplicações dos vetores na linguagem C.

Detalhamento da metodologia: Seguindo o planejamento de uma aula de 50 minutos ao qual compreenderá as seguintes etapas:

Etapa 1 – Situar as relações com o problema: Dialogar com os estudantes sobre quais informações e conhecimentos prévios eles possuíam a respeito dos conceitos e aplicações dos vetores/*arrays*.

Etapa 2 - Estabelecer um plano de trabalho para a resolução do problema: A partir das discussões realizadas com auxílio do professor e os estudantes, será necessário listar as informações que forem obtidas por meio da própria apresentação do problema, bem como dos seus conhecimentos prévios sobre o assunto.

Etapa 3 – Abordar a situação problema: a partir das duas questões exemplo, professor e estudantes, deverão construir um método objetivo e eficiente para

solucionar o problema apresentado a partir da prática de atividades de aprendizagem de forma colaborativa.

Etapa 4 - Reequacionar o problema: Na mesma aula, ou na aula seguinte, após os trabalhos individual sob a mediação do professor, cada estudante, deverá apresentar, à turma e ao professor, um relatório do trabalho desenvolvido. A partir deste relatório, avaliar o desenvolvimento do problema e possíveis melhoramentos.

Observação: As apresentações deverão ocorrer pela escolha aleatória (via livro de chamadas por exemplo) de três estudantes, para apresentar seu processo em sala de aula.

Etapa 5 - Elaborar e apresentar os produtos: Avaliação, nesta etapa, caberá ao professor, fomentar um processo diferenciado de avaliação, do qual, some as percepções do professor e dos estudantes sobre da atividade total.

Por exemplo:

Momento 1:

- **Retomada** – Resgatar os conhecimentos dos estudantes sobre Vetores.
- **Ação:** Resgatar os conhecimentos sobre Vetores em linguagem C;
- **Tempo:** 11min.

Momento 2:

Situação Problema:

a) Como fazer um programa que peça 3 números inteiros ao usuário, armazene em um vetor e depois mostre o valor de cada elemento do vetor, assim como seu índice?

b) Como fazer um programa em C que peça ao usuário duas notas que ele tirou e mostre a média. Usando somente um vetor para essas três variáveis

- **Atividade principal** - Informar aos alunos sobre o funcionamento da atividade e fazê-lo utilizando a estratégia da ABP;
- **Ação:** Organizar os grupos, determinar as funções de seus componentes, seguido as regras propostas, por seguinte resolver o desafio.

- **Tempo:** 18 min.

Momento 3:

- **Discussão das soluções** – discutir e compartilhar os percursos trilhados na resolução do desafio e as estratégias utilizadas para resolver o desafio.
- **Ação:** Consolidar a estratégia da ABP e seus significados relacionados a estruturas de repetição.
- **Tempo:** 4 min.

Momento 4:

- **Raio X:** Observar e consolidar as aprendizagens dos estudantes sobre o uso estratégia da aprendizagem baseada em problemas, para realizar a situação problemas dos vetores;
- **Ação:** Utilizar os saberes da estratégia apreendida em sala de aula para revolucionar a questão aplicada pelo (a) professora e posteriormente apresentar a resolução para o professor e os colegas em sala de aula.
- **Tempo:** 12 min.

5) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à vetores.

2. Qual (is) conteúdo (s) será (ão) trabalhado (s)?

Conteúdos de estruturas de dados na linguagem de programação C.

- *Arrays.*
- Vetores em pseudocódigo.
- Tipos de vetores.

3. Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Vetores;
- Incentivo a cultura do acesso a pré-aula.
- Aprendizagem ativa.

4. Objetos de aprendizagem a serem usados (documentos, apresentação lista de exercícios, vídeos, jogos, ...):

- Vídeo explicativo, disponível em:

- Apresentação em Power Point para o estudo dirigido como o exemplo de Salamon (s/a): <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf>

- Lista de exercícios em sala de aula para trabalho em grupo, como o exemplo que poderá ser acessado em: <https://forms.gle/tsBhgWSksU7SabLw8>.

- *Feedback* do professor para os estudantes.

5. Forma de avaliação:

- Observação da participação dos estudantes nas atividades em sala de aula;

6. Descrição detalhada da estratégia da aula:

Aula de Vetores em programação C:

Procedimentos:

Para o desenvolvimento da aula, serão necessários:

Anteriormente ao dia específico da disciplina de Vetores:

- Disponibilização dos conteúdos para estudos individuais aos estudantes (via grupo de WhatsApp por exemplo).
- Criação de um vídeo explicativo, como no exemplo disponível em: [\(417\) Vetores em linguagem C partiner @thaispluskota6424 - YouTube](#)
- Explicações sobre a PBL aos estudantes.
- *Google Forms* para criação de atividades para os estudantes resolverem em casa ou em sala de aula, conforme exemplo disponível em: <https://forms.gle/tsBhgWSksU7SabLw8>

No dia da aula

- Revisão do conteúdo, conforme disponibilizado no vídeo disponível em: <https://youtu.be/e1RnZ66PCgE>
- Aplicação dos *Slides* e atividades para revisão dos conceitos, como no exemplo disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf>

- Discussão sobre o que são vetores, suas sequencias e aplicação prática.

Avaliação: Observação da apresentação da situação-problema e da participação dos estudantes durante a apresentação da resolução do problema.

Observações adicionais:

Esta aula remete a uma aula de 50 minutos. Logo, todo material, deve ser preparado anteriormente ao dia da aula

Desafios: Desenvolver o vídeo explicativo e acompanhar o desenvolvimento dos estudantes.

Ação dos professores: Apresentar os conceitos e o conteúdo sobre vetores, bem como sua aplicação e apresentar os conceitos e o conteúdo sobre vetores, bem como sua aplicação.

Ação dos estudantes: Resolver a questão disponibilizada no *Google forms* e apresentar ao professor.

Feedback: Entregar *feedback* dos trabalhos e atividades.

Observações: Ainda cabe como ações dos estudantes e professores de acordo com Camargo (2019, p. 117):

Figura 24 - Ações dos professores e estudantes em Vetores em ABP

Estudantes	Professores
Reconhecer o sucesso no trabalho dos outros, além de se reconhecerem como colaboradores e fornecedores de recursos didáticos entre si (avaliação entre pares); Ser agente da própria aprendizagem, isto é, reconhecer o sucesso em seu próprio trabalho e se concentrar em como e o que estão aprendendo e onde ainda podem melhorar (autoavaliação).	Esclarecer e compartilhar as intenções de aprendizagem e critérios para obter sucesso: para que os alunos compreendam o que estão tentando aprender, por que e o que se espera deles; Organizar discussões e atividades efetivas em sala de aula de forma a evidenciar a aprendizagem: para criar um ambiente na sala de aula onde os alunos possam expor suas próprias ideias, pensar em voz alta e explorar a sua compreensão; Fornecer <i>feedback</i> que seja capaz de promover a aprendizagem: referenciando a qualidade de seu trabalho e o que os alunos podem fazer para torná-lo melhor;

Fonte: Organizado pelo autor (2023)

REFERÊNCIAS

CARVALHO, Melisse. **Estratégias para avaliação na aprendizagem baseada em problemas.** In LOPES, Renato Matos. **APRENDIZAGEM BASEADA EM PROBLEMAS: Fundamentos para a aplicação no ensino médio e na formação de professores.** 2019. Disponível em: <https://educapes.capes.gov.br/bitstream/capes/432641/2/APRENDIZAGEM%20BASEADA%20EM%20PROBLEMAS%20-%20fundamentos%20para%20a%20sua%20aplica%C3%A7%C3%A3o%20no%20Ensino%20M%C3%A9dio%20e%20na%20Forma%C3%A7%C3%A3o%20de%20Professores.pdf> Acesso em: 26 de janeiro de 2023.

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C.** 2023. Disponível em: <https://youtu.be/e1RnZ66PCgE> Acesso em: 13 de março de 2023.

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C: Atividades de fixação.** 2023. Disponível em: <https://forms.gle/k3ARuBTp143B85SQ6> Acesso em: 13 de março de 2023.

SALAMON. Jordana S. **Programação básica de computadores.** S/A. Departamento de informática. Universidade Federal do Espírito Santos. Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf> Acesso em: 13 de março de 2023.

Vetores: Estratégias e Aplicação na Gamificação

METODOLOGIA DE ENSINO: GAMIFICAÇÃO

Conteúdo: Vetores

Metodologia de ensino: Gamificação

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Vetores, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de vetor e sua aplicação por meio da gamificação.

PLANO DE AULA

Metodologia Ativa: Gamificação

Objetivo da aula:

- Desenvolver uma aula de 50 minutos através da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.
 - Fixar o conteúdo Vetores utilizando a metodologia ativa Gamificação.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - Listas de atividades sobre vetores;
 - *Power Point* para aula explicativa expositiva;
 - Vídeo explicativo para introdução ao conceito de Vetores.
- A atividade apresentada será durante uma aula de 50 minutos.
- Para o desenvolvimento da aula deverão estar preparados no *Google Classroom* (ou ambiente similar):

- Um vídeo sobre Introdução a programação na linguagem C: Vetores e Arrays.
- Atividades de fixação para instigar os estudantes, a verificarem o que será discutido em sala de aula.
- *Power Point* com o conteúdo que será tratado em sala de aula.

Especificamente, durante o dia da aula.

- Após introdução do tema. Explicar aos estudantes, a missão: “Quem Chega lá”. Ao qual, ocorrerá pela distribuição pelo professor, de atividades ao longo da sala de aula ou corredor do curso, por meio de *QR Code*.

Metodologia de ensino: Gamificação

- A aplicação da metodologia Gamificação será realizada da seguinte forma:
 - **Preparação do campo para a realização da missão:** como por exemplo, atividades de fixação distribuídas ao longo da classe, via imagem de *QR code* para os estudantes acessarem e resolverem no curto período de 20 minutos. As atividades remetem aos seguintes exemplos:
 - Faça um programa que peça 3 números inteiros ao usuário, armazene em um vetor, depois mostre o valor de cada elemento do vetor, assim como seu índice:
 - Faça um programa em C que peça ao usuário duas notas que ele tirou e mostre a média. Use somente um vetor para essas três variáveis.
 - **Apresentação da atividade:** Apresentação da atividade, compreendendo a metodologia ativa da gamificação. E exemplificando com a resolução de ao menos uma atividade.
 - **Acompanhamento do desenvolvimento dos estudantes:** Em seguida, o professor deverá acompanhar o desenvolvimento dos estudantes. A realização das missões, podem ser visualizadas em tempo real pelo professor via Google Forms. Cada estudante deverá:
 - Realizar o máximo de questões o mais rápido possível.
 - E por seguinte apresentar a função realizada ao professor (os estudantes poderão formar duplas ou trio para resolver as questões, mas a iniciativa deverá partir dos mesmos (contudo, o professor não poderá deixar

claro que os estudantes poderão fazer individualmente ou não). Após a resolução das questões, os estudantes deverão justificar porque decidiram ou não se unir).

A finalidade consiste em:

- Fomentar o entendimento dos conceitos e benefícios do uso de Vetores;
- Incentivar a competição saudável;
- Apresentar a importância da ajuda mútua.

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos:

Momento 1: Preparação da aula.

- Desenvolver atividades de fixação no *Google Forms* os estudantes resolverem em sala de aula.
- Criar QR Code para os estudantes acessarem as atividades.
- Distribuir ao longo do corredor da universidade ou na sala de aula, os QR Code para os estudantes realizam as atividades/missões.

Momento 2: No dia da aula

- Socializar com os estudantes a forma em que a aula irá ocorrer, baseada na metodologia ativa da gamificação. Apresentar os conceitos de Vetores a partir do vídeo explicativo, como no exemplo disponível em: <https://youtu.be/e1RnZ66PCgE>
- Dialogar de forma expositiva sobre os conceitos via atividade prática, por meio do Power Point Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf>
- Explicar as razões dos *QR Code* Disponibilizado ao longo da sala. Instigar a realização das missões.
- Realizar uma das missões para os estudantes compreenderem o processo. Tais como:

- Faça um programa que peça 3 números inteiros ao usuário, armazene em um vetor, depois mostre o valor de cada elemento do vetor, assim como seu índice:
- Faça um programa em C que peça ao usuário duas notas que ele tirou e mostre a média. Use somente um vetor para essas três variáveis.
- Disponibilizar o tempo de 20 minutos para os estudantes realizarem o máximo de questões que puderem durante este período.
- Ao final dos 20 minutos, o professor poderá corrigir com os estudantes as atividades e apresentar o feedback a eles. Com prêmio simbólico (chocolate por exemplo).

Vale ressaltar que a atividade disponibilizada no *google forms* deverá ser identificada, e na identificação, descrever “nome(s)” indicando que o estudante poderá fazer a atividade com mais pessoas, mas sem mencionar a situação

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente à Vetores

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de estruturas de dados na linguagem de programação C.

- *Arrays*.
- Vetores em pseudocódigo.
- Tipos de vetores.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Vetores;
- Incentivar o trabalho e apoio mútuo bem como a competição saudável;
- Aprendizagem Ativa

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- *Google Classroom* para disponibilização dos textos bases e indicação de leituras, vídeos explicativos e os slides.

- Google Forms para disponibilização das atividades avaliativas.

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;
- Correção das atividades realizadas pelos estudantes.
- Realização das missões.

7) Descrição detalhada da estratégia da aula

Vetores

Momento 1: Preparação da aula.

- Desenvolver atividades de fixação no *Google Forms* os estudantes resolverem em sala de aula, como no exemplo disponível em: <https://forms.gle/LvsTYXo9VZmXZ3LN8>
- Criar QR Code para os estudantes acessarem as atividades como no exemplo da imagem:

Figura 25 - Qr Code atividade de vetores em Gamificação



Fonte: Organizado pelo autor (2023)

- Distribuir ao longo do corredor da universidade ou na sala de aula, os QR Code para os estudantes realizam as atividades/missões.

Momento 2: No dia da aula

- Socializar com os estudantes a forma em que a aula irá ocorrer, baseada na metodologia ativa da gamificação. Apresentar os conceitos de Vetores a partir do vídeo explicativo, como no exemplo disponível em: <https://youtu.be/e1RnZ66PCgE>
- Dialogar de forma expositiva sobre os conceitos via atividade prática, por meio do Power Point Disponível em: <http://www.inf.ufes.br/~jssalamon/wp->

<content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf>

- Explicar as razões dos *QR Code* Disponibilizado ao longo da sala. Instigar a realização das missões.
- Realizar uma das missões para os estudantes compreenderem o processo.
- Disponibilizar o tempo de 20 minutos para os estudantes realizarem o máximo de questões que puderem durante este período.
- Ao final dos 20 minutos, o professor poderá dispensar os estudantes, corrigir as atividades e apresentar o feedback para a turma. Com prêmio simbólico (chocolate, por exemplo ou bonificação na nota).

Vale ressaltar que a atividade disponibilizada no *google forms* deverá ser identificada, e na identificação, descrever “nome(s)” indicando que o estudante poderá fazer a atividade com mais pessoas.

8) Observações adicionais

Na atividade de realização das missões, o professor pode acompanhar o desenvolvimento dos estudantes de forma individual ou em grupos, durante a interação em aula e via desenvolvimento da atividade, via *Google Forms*. Os estudantes podem solicitar também, dicas do professor.

Desafios:

- Desenvolver as atividades no pré aula, para preparação dos estudantes, bem como a divisão de tarefas para a construção das missões. Instigar o sentimento de competição e colaboração entre a turma, para que busquem realizar o máximo de atividades em menor tempo, mas que se colaborem entre si.

Ação dos professores:

- Apresentar os conceitos e o conteúdo de Vetores, bem como sua aplicação;
- Auxiliar na formação de realização das atividades, para completar a missão no menor tempo possível e de forma correta.
- Preparar para a aula, um prêmio simbólico (chocolates ou botton de algum personagem que a turma geralmente discute, como anime ou jogador de futebol).

Ação dos estudantes:

- Realizar o máximo de questões das missões, dentro do período de 20 minutos.

Feedback:

- Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C**. 2023. Disponível em: <https://youtu.be/e1RnZ66PCgE> Acesso em: 13 de março de 2023.

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C: Atividades de fixação**. 2023. Disponível em: <https://forms.gle/k3ARuBTp143B85SQ6> Acesso em: 13 de março de 2023.

SALAMON. Jordana S. **Programação básica de computadores**. S/A. Departamento de informática. Universidade Federal do Espírito Santos. Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf> Acesso em: 13 de março de 2023.

Vetores: estratégias e Aplicação na Aprendizagem Baseada em Times

METODOLOGIA DE ENSINO: TEAM BASED LEARNING

Conteúdo: Vetores

Metodologia de ensino: Aprendizagem Baseada em Times

Descrição detalhada da estratégia: Uma atividade em aula presencial na qual será realizada durante o ensino do conteúdo de Vetores, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos principais conceitos que envolvem Vetores e sua aplicação por meio de uma estratégia de aprendizagem ativa.

PLANO DE AULA

Metodologia Ativa: Aprendizagem baseada em times

Objetivo da aula:

- Desenvolver uma aula de 50 minutos por meio da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.
- Fixar o conteúdo de Vetores utilizando a metodologia TBL.

1) Metodologia a ser trabalhada

Procedimento:

Para o desenvolvimento da aula serão necessários:

- Anteriormente a realização da aula:
 - Disponibilização de textos e vídeos explicativos aos estudantes utilizando a plataforma do *Google Classroom* (ou similar), para leitura e conhecimento prévio do conteúdo a ser trabalhado em sala de aula.
 - Explicações sobre a estratégia da TBL em aula, para que os alunos, compreendam a lógica da aula seguinte.

- No dia da aula:
 - Explicação sobre Vetores e *arrays*, por meios dos materiais disponibilizados no ambiente virtual.
 - Disponibilização de atividades para os estudantes realizarem em sala de aula, em times, para fixação do conteúdo.
 - Explicação sobre o processo de apresentação da questão resolvida (utilizando um vídeo explicativo que será desenvolvido pelos próprios estudantes e disponibilizados ao professor e aos colegas).

Avaliação: verificação dos estudantes que desenvolveram o vídeo explicativo e enviaram o conteúdo ao professor e aos colegas da turma.

2) Quando aplicar este plano?

Em conjunto com o conteúdo de vetores.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de estruturas de repetição:

- *Arrays*.
- Vetores em linguagem C.
- Tipos de vetores.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Vetores;
- Incentivar o trabalho em times;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- *Classroom* ou grupo de *WhatsApp*;
- Apresentação em Power Point, para estudo dirigido;
- Utilização do quadro negro para exemplificação dos estudos;
- Lista de exercícios para desenvolver em sala de aula, por meio de trabalho em grupo via *Google Forms*;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;
- Envio dos vídeos explicativos da resolução das questões, via *WhatsApp* ou *Google Classroom*.

7) Descrição detalhada da estratégia da aula

Procedimentos:

Para o desenvolvimento da aula, serão necessários:

Anteriormente ao dia específico da disciplina de Vetores:

- Criação de um grupo de *WhatsApp* ou atividade no *Google Classroom*, para diálogo com os estudantes por meio de um fórum.
- Disponibilização dos conteúdos para estudos individuais aos estudantes, por meio do grupo de *WhatsApp* ou *Classroom* aos estudantes.
- Criação de um vídeo explicativo, como no exemplo disponível em: <https://youtu.be/e1RnZ66PCgE>
- Explicações sobre a estratégia da TBL aos estudantes.
- Utilização do *Google Forms* para criação de atividades para os estudantes resolverem em sala de aula, conforme exemplo disponível em: <https://forms.gle/tsBhgWSksU7SabLw8>

No dia da aula

Revisão do Conteúdo: Revisão do conteúdo, conforme disponibilizado no vídeo disponível em: [\(417\) Vetores em linguagem C partiner @thaispluskota6424 - YouTube](https://www.youtube.com/watch?v=417Vetores)

Aula Expositiva: Utilização dos slides e atividades para revisão dos conceitos, confirme o exemplo disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%20%20-%20Vetores%20em%20C.pdf>

Discussão sobre os conceitos: Discussão sobre o que são vetores, suas organizações, índices e aplicação prática.

Atividade de fixação: Realização de atividades avaliativas para fixação do conteúdo com os estudantes, seguindo o exemplo disponível em: <https://forms.gle/tsBhgWSksU7SabLw8>

Avaliação: Explicação sobre o processo de avaliação, no qual consistirá no desenvolvimento de um vídeo explicativo sobre o conteúdo VETORES – o que são? Como devem ser declarados? Como se dá o processo de utilização?

Exemplos. (o professor poderá indicar softwares para elaboração do vídeo explicativo, tais como: *Video Scribe, Canva, Power Point, OBS, Camtasia e ou DaVince Resolve*).

Avaliação final: disponibilização de um questionário no Google Forms, como no exemplo: [Vetores em linguagem C \(google.com\)](#)

Ressalta-se que o vídeo explicativo criado pelos estudantes, deverá ser compartilhado com a turma via Google Classroom ou WhatsApp.

8) Observações adicionais

Na atividade os estudantes deverão explicar como chegaram ao resultado apresentado por meio do vídeo explicativo, que deverá ser compartilhado com o professor e a turma.

Desafios:

- Resolver em grupo uma questão/problema disponibilizada.
- Desenvolver e compartilhar um vídeo explicativo.
- Seguir os quatro princípios essenciais da TBL, conforme segue:

1ª Distribuição adequada dos grupos:

- Formar adequadamente os grupos, de modo que não formem times “panelinhas”, minimizando assim, possíveis barreiras durante a formação dos times. Sendo então, grupos de cinco e sete pessoas, que permaneçam nos mesmos times ao longo do semestre.

2ª Tornar o estudante responsável por seu progresso:

- Contribuir para que os estudantes, tornem-se protagonistas de suas ações por meio de responsabilidades prévias coletiva e individuais (planejamento).

3ª Promoção da interação por meio das atividades:

- Fomentar o diálogo entre os times e possíveis planos de ação futuras com os resultados das atividades (artigo, site ou game por exemplo).

4ª Constante *feedback*:

- O papel do professor é mediar as interações dos estudantes. Neste sentido, é importante dar retorno quanto as potencialidades e fragilidades dos times ao longo de suas atividades.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Vetores, bem como sua aplicação;
- Auxiliar na formação dos times para construção da atividade avaliativa.
- Mediar as interações dos times.
- Realizar constante *feedback* com os estudantes

Ação dos estudantes:

- Resolver a questão em time e apresentar o resultado utilizando um vídeo explicativo.

Feedback:

- Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C**. 2023. Disponível em: Acesso em: 13 de março de 2023.

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C: Atividades de fixação**. 2023. Disponível em: <https://forms.gle/k3ARuBTp143B85SQ6> Acesso em: 13 de março de 2023.

SALAMON. Jordana S. **Programação básica de computadores**. S/A. Departamento de informática. Universidade Federal do Espírito Santos. Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf> Acesso em: 13 de março de 2023.

MATRIZES: ESTRATÉGIAS E APLICAÇÃO COM METODOLOGIAS ATIVAS

Em programação, uma matriz é uma estrutura de dados retangular composta por elementos do mesmo tipo, organizados em linhas e colunas. Uma matriz pode ser pensada como uma tabela bidimensional em que cada elemento é referenciado por um par de índices, representando sua posição na matriz.

As matrizes são usadas para armazenar e manipular conjuntos de valores relacionados, como números, caracteres ou objetos. Elas são amplamente utilizadas em diversas áreas da programação, como processamento de imagens, manipulação de dados, jogos, simulações, entre outros.

Uma matriz é definida pelo número de linhas e colunas que possui. Por exemplo, uma matriz de tamanho 3x3 tem 3 linhas e 3 colunas, totalizando 9 elementos.

Os elementos de uma matriz podem ser acessados utilizando os índices de linha e coluna correspondentes. Por exemplo, considere a seguinte matriz de tamanho 2x3:

1	2	3
4	5	6

Para acessar o valor 5 nessa matriz, você usaria os índices (1, 1), onde o primeiro número representa a linha e o segundo número representa a coluna. A indexação começa a partir de zero, então o elemento na segunda linha e segunda coluna é o valor 5.

As matrizes podem ser usadas para realizar várias operações, como adição, multiplicação, transposição, entre outras. Elas são uma ferramenta poderosa para lidar com grandes conjuntos de dados estruturados de forma organizada e eficiente.

Desta forma, com base no conceito de matrizes, este capítulo divide-se em quatro diferentes metodologias ativas (Sala de aula Invertida, Aprendizagem baseada em Problemas, Gamificação e Aprendizagem baseada em Times),

sendo uma metodologia para cada plano de aula, que tem como objetivo, apresentar os fundamentos da estrutura de decisão simples e composta, utilizando vídeo explicativo disponibilizado antecipadamente e aula expositiva, para explorar o conteúdo.

Boa leitura!

METODOLOGIA DE ENSINO: SALA DE AULA INVERTIDA

Conteúdo: Matrizes

Metodologia de ensino: Sala de Aula Invertida

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino de Matrizes, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos principais conceitos de Matrizes.

PLANO DE AULA

Metodologia Ativa: Sala de Aula Invertida

Objetivo da aula:

- Desenvolver uma aula de 50 minutos, por meio da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.
- Fixar o conteúdo de Matrizes utilizando a metodologia ativa Sala de aula Invertida.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:

Anteriormente ao dia específico da disciplina de Matrizes.

- **Disponibilização do aplicativo para acesso aos conteúdos didáticos:** Disponibilização dos conteúdos para estudos individuais aos estudantes por meio do “Matriz-X” - App Desenvolvido pelo MIT App Inventor. (isso deve ocorrer em uma aula anterior da aula debate). Disponível em: [Matrizes UTFPR.apk](#)

- **Construção e publicação do blog (Blogger – Blog da Google):** O blog criado, será base para o aplicativo, considerando que o conteúdo disponível no blog, será capturado pelo App Inventor.
- **Apresentação da metodologia ativa:** Explicações sobre a estratégia da Sala de aula invertida, por meio da disponibilização do aplicativo para os estudantes terem acesso ao assunto antes da aula.
- **Preparação das atividades de fixação:** Google Forms para criação de atividades para os estudantes resolverem em casa ou em sala de aula.

No dia da aula

- **Discussão sobre os conteúdos a partir do “Matriz-X”:** Para revisão do app. Como disponível em: [Matrizes UTFPR.apk](#)
- **Aula Expositiva:** Slides e atividades para revisão dos conceitos tratados no App (disponível em: <https://docs.google.com/presentation/d/1js3kXdtCmJQnJVjEs7Cp9DFmHnyHPvUgQtdVsj-xZK0/edit?usp=sharing>).

Avaliação: verificação dos estudantes que responderam no questionário disponível no aplicativo “App Inventor – MatrizX”. Que poderão ser acessados pelo Google Forms Criado pelo professor.

2) Quando aplicar este plano?

Posteriormente a disciplina de vetores.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de estruturas de Matrizes na Linguagem de Programação C:

- Estrutura de repetição

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de Matrizes;
- Incentivo do aprendizado pré-aula.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- App Inventor, disponível em: [Matrizes UTFPR.apk](#)
- Apresentação em Power Point para o estudo dirigido (exemplo disponível em:
<https://docs.google.com/presentation/d/1js3kXdtCmJQnJVjEs7Cp9DFmHnyHPvUgQtdVsj-xZK0/edit?usp=sharing>)
- Utilização do quadro negro para exemplificação dos estudos;
- Lista de exercícios em sala de aula para trabalho em grupo;
- Feedback do professor para os estudantes

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em sala de aula;
- Verificação dos estudantes que responderam ao questionário disponível no aplicativo “App Inventor – MatrizX” ([Matrizes UTFPR.apk](#)). Que poderão ser acessados pelo Google Forms Criado pelo professor. Disponível em: [Exercícios-de-Linguagem-C-resolvidos.pdf](#)

7) Descrição detalhada da estratégia da aula

Aula de Matriz em Programação em C

Procedimento:

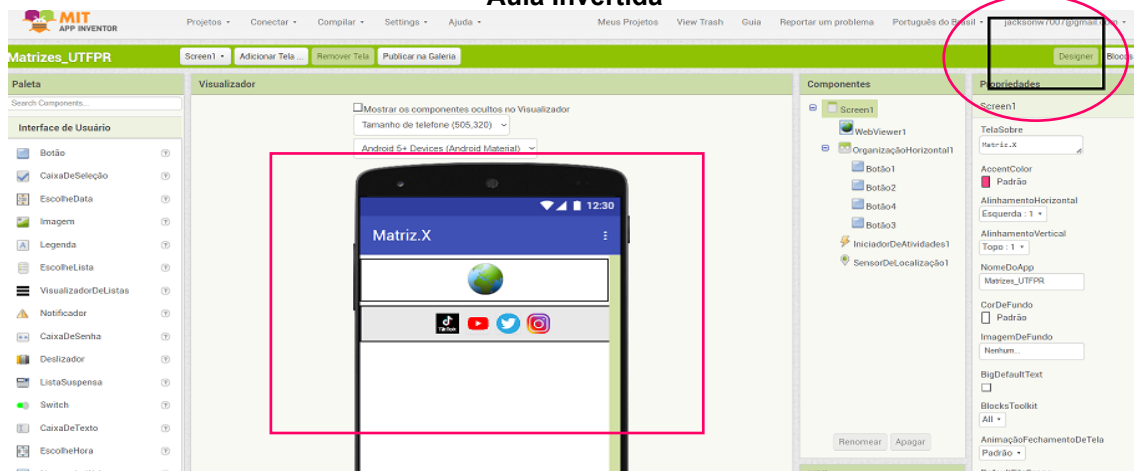
Para o desenvolvimento da aula serão necessários:

Anteriormente ao dia específico da disciplina de Matrizes.

- Anteriormente ao dia da aula, é importante que a aula para trabalhar com os estudantes esteja previamente pronta. Isso significa, no caso desta aula, estar desenvolvido um aplicativo com conceitos da aula, slides e ou vídeos e exercícios de fixação.
- Neste caso, para desenvolver um aplicativo, sugere-se para esta aula, o desenvolvimento de um aplicativo nomeado como “MatrizX”, ao qual, foi desenvolvido no Software App Inventor. Assim, os passos para este processo, são os seguintes:
 - **Desenvolver o aplicativo MatrizX:** Acessar o App Inventor no site do *App Inventor*, disponível em: <http://appinventor.mit.edu>, e criar sua conta de professor.

- **Desenvolvimento do Design:** Para iniciar o desenvolvimento do aplicativo, caberá ao professor acessar com sua conta e iniciar o desenvolvimento do aplicativo, pela aba “designer”. Como no seguinte exemplo:

Figura 26 - Desenvolvimento do design do aplicativo para aula de matrizes na Sala de Aula Invertida



Fonte: Organizado pelo autor (2023)

- **Construir o background:** Além do design do aplicativo, caberá ao professor, construir o background do aplicativo, no qual, remete a codificação dos comandos que serão necessários para acessar o conteúdo do Blog e demais redes sociais que se considera necessário ao professor, conforme o exemplo a seguir:

Figura 27 Background aplicativo para aula de Matrizes na Sala de Aula Invertida

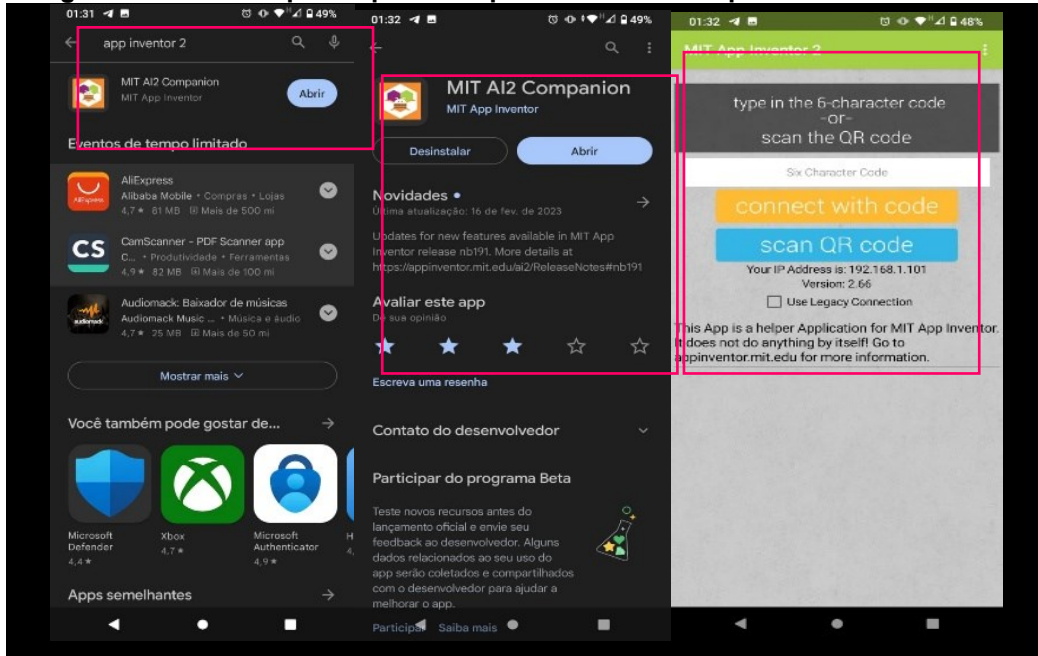


Fonte: Organizado pelo autor (2023)

- **Compartilhar o aplicativo com os estudantes:** Terminada a programação no aplicativo, o professor deverá solicitar que os estudantes,

baixem em seus celulares, via Play Stores, o aplicativo App Inventor 2. Instalem em seus celulares e cliquem o ícone QR Code, conforme a sequência a seguir:

Figura 28 - Passo a passo para compartilhamento do aplicativo Mit Inventor 1



Fonte: Organizado pelo autor (2023)

- Por sua vez, o QR Code deverá ser compartilhado pelo professor, aos estudantes, por meio dos seguintes passos:
 - **Conectando o Assistente “Artificial Intelligence”:**

Figura 29 - Passo a passo para compartilhamento do aplicativo Mit Inventor 2 - Assistente AI

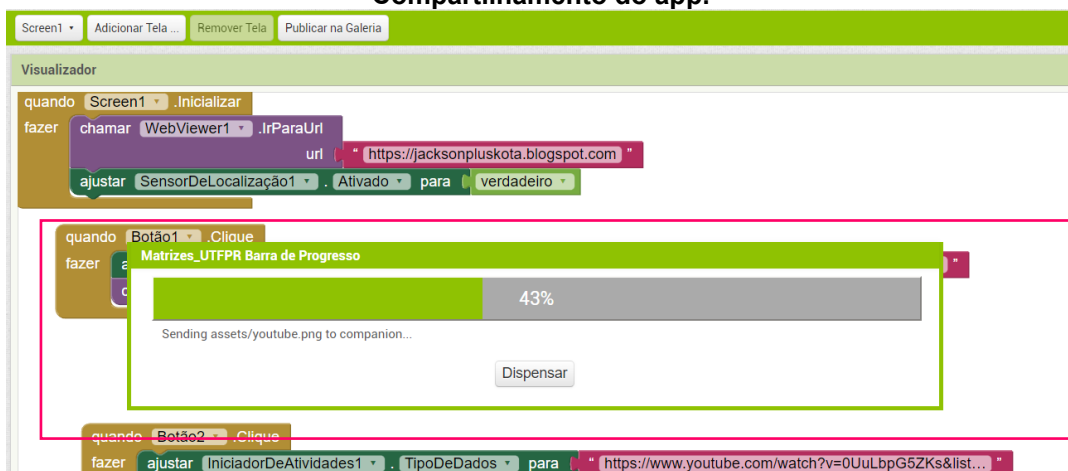


Fonte: Organizado pelo autor (2023)

- **Ler o QR Code Gerado pela AI:** Ao abrir o programa no computador do professor em sala de aula. A inteligência artificial gerará um código, em letras e

números e em Qr Code. Ao apontar o celular para a leitura do Qr Code, os estudantes poderão acessar todo o conteúdo desenvolvido pelo professor, conforme o exemplo a seguir:

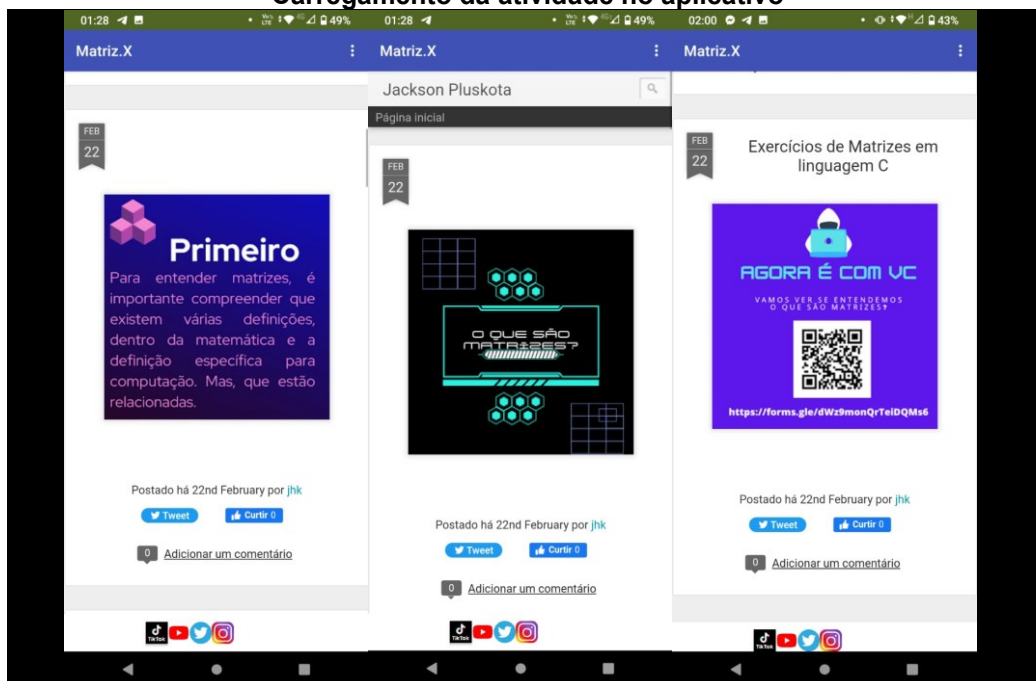
Figura 30 Passo a passo para compartilhamento do aplicativo Mit *Inventor* - Compartilhamento do app.



Fonte: Organizado pelo autor (2023)

- Terminado o carregamento, o estudante terá acesso as seguintes informações, conforme exemplo a seguir:

Figura 31 Passo a passo para compartilhamento do aplicativo Mit *Inventor* - Carregamento da atividade no aplicativo



Fonte: Organizado pelo autor (2023)

- Por meio deste aplicativo, é possível, disponibilizar os conteúdos para estudos individuais aos estudantes por meio do “Matriz-X”. Certamente,

que esta atividade, pode ser realizada por meio do Moodle ou Google Classroom. Contudo, a sugestão deste App, se dá para demonstração das possibilidades de utilização de outras TICs e objetos de aprendizagem.

. No dia da aula

- Revisão do conteúdo, conforme disponibilizado no App “Matriz-X”
- Explicações sobre a estratégia da Sala de aula invertida por meio da disponibilização do aplicativo para os estudantes terem acesso ao assunto antes da aula.
- Aplicação dos Slides e atividades para revisão dos conceitos tratados no App (disponível em: <https://docs.google.com/presentation/d/1js3kXdtCmJQnJVjEs7Cp9DFmHnyHPvUgQtdVsj-xZK0/edit?usp=sharing>).
- Discussão sobre o que são matrizes, seus conceitos principais, como tratar os índices, como manipular os dados e aplicação prática.

Avaliação: verificação dos estudantes que responderam ao questionário disponível no aplicativo “App Inventor – MatrizX”. Que poderão ser acessados pelo Google Forms Criado pelo professor. Como no exemplo disponibilizado no aplicativo MIT: por meio do QR Code:

Figura 32 Passo a passo - Compartilhamento via QR Code



Nota: Organizado pelo autor.

O aplicativo poderá ser compartilhado via QR Code ou por meio do *link*: <https://forms.gle/dWz9monQrTeiDQMs6> que levará diretamente ao exercício.

8) Observações adicionais

O aplicativo é um modelo, que poderá ser desenvolvido pelo professor, de acordo com suas demandas, de forma gratuita.

Desafios:

- **Professor:** Desenvolver o app e acompanhar o desenvolvimento dos estudantes. Cabe ressaltar que o que contribuirá com o aprendizado dos estudantes, não será somente o aplicativo em si e sim, o conteúdo disposto no mesmo. O aplicativo é somente uma ferramenta diferenciada, para conhecimento.
- **Estudantes:** Acessar o app e resolver as questões disponibilizadas no mesmo.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre matrizes, bem como sua aplicação;
- Auxiliar interpretação de algum conteúdo que os estudantes apontem antes e durante a aula.

Ação dos estudantes:

- Resolver a questão disponibilizada no *App* e apresentar ao professor.

Feedback:

Entregar *feedback* dos trabalhos e atividades.

Considerações:

Para este, vale ressaltar as seguintes questões:

- Para desenvolver o app, é importante acessar o endereço <http://appinventor.mit.edu> e criar uma conta a partir do endereço de seu próprio Gmail se preferir.
- Após, é preciso desenvolver o app a partir do Designer, observando as funções que deseja conter no App e por seguinte, é necessário iniciar a programação via blocos pelo comando *Blocks* disponível no canto direito do *site* MIT.

- Atenção: Após desenvolver o aplicativo no site, pelo computador, é importante testá-lo antes de enviar aos estudantes.
- Este teste, deverá ocorrer pelos seguintes passos:
- Baixando a AI através do aplicativo *Android MIT AI 2 Companion* disponível no endereço: <https://play.google.com/store/apps/de...>
- Caso, não possua celular *Android* para teste, é importante, instalar no computador o emulador do MIT disponível em: <http://appinventor.mit.edu/explore/ai...>
- Porém, aconselha-se utilizar o celular *Android* para o teste mais ágil.
- Ao instalar a AI no celular *Android*, o mesmo abrirá uma página com a mensagem “*type in the 6-character code or scan the QR code*”, desta forma no computador, será necessário clicar no topo da página em *CONNECT* e após em *AI COMPANION*, neste irá gerar um *QR code* e um código que poderá ser lido pelo *QR code* do App MIT 2.
- Após, somente aguardar o programa renderizar e o App estará disposto no celular *Android*.
- Este processo poderá ser realizado com o seguinte código, para testar o aplicativo *Matriz-X*: no caso deste exemplo, o Código:

Figura 33 QR Code MatrizX



Fonte: Organizado pelo autor (2023)

- Vale ressaltar que para *TESTE* é importante o computador e o celular estarem conectados na mesma rede *Wi-fi*.
- Teste, ok, é importante copiar o *APP* para disponibilizar aos estudantes através do *APK* para poderem acessar o App em casa.
- Para *Compilar* é importante acessar a opção *COMPILAR* disponível no topo do site do MIT local onde esteja desenvolvendo o aplicativo e clicar em “(App

Salvar. Apk salvar em meu computador)” e posteriormente, poderá enviar via *WhatsApp* ou e-mail para os estudantes instalarem no celular. Este processo também irá gerar um QR Code que pode ser disponibilizado aos estudantes:

Figura 34 Compartilhamento APK MatrizX



Fonte: Organizado pelo autor (2023)

- Sugere-se realizar este procedimento em sala de aula.

REFERÊNCIAS

PLUSKOTA, Jackson W. **Exercícios de Matrizes resolvidos**. 2023. Disponível em: [Exercícios-de-Linguagem-C-resolvidos.pdf](#) Desenvolvido em: 16 de março de 2023.

PLUSKOTA, Jackson W. **Matrizes UTFPR APK**. 2023. Disponível em: [Matrizes UTFPR.apk](#) Desenvolvido em: 16 de março de 2023.

Matrizes: Estratégias e Aplicação na aprendizagem baseada em problemas

METODOLOGIA DE ENSINO: PROBLEM BASED LEARNING

Conteúdo: Matrizes

Metodologia de ensino: Aprendizagem Baseada em Problemas

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Matrizes, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de Matrizes e a resolução de problemas por meio de funções e a aprendizagem baseada em problemas.

PLANO DE AULA

Metodologia Ativa: Aprendizagem Baseada em Problemas

Objetivo da aula:

- Desenvolver uma aula de 50 minutos através da utilização de metodologias ativas de ensino e aprendizagem e, TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.
- Fixar o conteúdo de Matrizes utilizando a metodologia da aprendizagem baseada em problemas.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - Livro de chamada ou aplicativo de sorteio aleatório.
 - *Sublime Text*.

- A atividade será desenvolvida durante uma aula de 50 minutos.
- Para o desenvolvimento da aula deverão estar preparados:
 - Slides sobre o conteúdo de matrizes.
- Além do seguinte exercício como exemplo (podem ser organizados outros desafios, de acordo com a necessidade do (a) professor (a):
 - Faça um programa que lê uma matriz de 3 x 3 elementos usando um comando for, multiplique cada elemento por 5 e imprima o resultado.

Especificamente, durante o dia da aula.

- Situar os estudantes sobre o que irão aprender e quais metodologias serão utilizadas para o processo de aprendizagem, por meio de exposições que podem ser realizadas via slides ou quadro negro.
- Em sala de aula, apresentar atividade de fixação das aulas anteriores e solicitar que em grupos, os estudantes resolvam o seguinte problema: Como desenvolver um programa capaz de ler uma matriz de 3 x 3 elementos usando um comando for, e que multiplica cada elemento por 5 e imprima este resultado?
- **Metodologia de ensino: A BP**
- Através da narrativa apresentada, instigar aos estudantes, que em grupo, resolvam os seguintes problemas:
 - Capaz de ler uma matriz 3x3;
 - Utilizar o comando FOR;
 - Imprimir os resultados obtidos pelo programa;

A finalidade consiste em:

- Fomentar o entendimento dos conceitos e benefícios do uso das Matrizes;

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos ao qual compreenderá em cinco etapas:

Etapa 1 - Estabelecendo relações com o problema: Dialogar com os estudantes sobre quais informações e conhecimentos prévios eles possuem a respeito dos conceitos e benefícios do uso das Matrizes.

Etapa 2 - Estabelecendo um plano de trabalho para a resolução do problema: A partir das discussões realizadas com auxílio do professor e os estudantes listar informações que forem obtidas por meio da própria apresentação do problema, bem como dos seus conhecimentos prévios sobre o assunto.

Etapa 3 – Abordagem do problema: Construir um método objetivo e eficiente para solucionar o problema apresentado a partir da prática de atividades de aprendizagem de forma colaborativa.

Etapa 4 - Reequacionando o problema: Na mesma aula, ou na aula seguinte, após os trabalhos individual e coletivo dos estudantes, cada grupo deverá apresentar, à turma e ao professor, um relatório do trabalho desenvolvido. A partir deste relatório, avaliar o desenvolvimento do problema e possíveis melhorias.

Etapa 5 - Elaborando e apresentando os produtos: Avaliação, nesta etapa, caberá ao professor, fomentar um processo diferenciado de avaliação, do qual, some as percepções do professor, dos estudantes, da atividade total.

Por exemplo:

Momento 1:

- **Retomada** – Resgatar os conhecimentos dos estudantes sobre vetores, para facilitar a introdução dos conceitos sobre matrizes.
- **Ação:** Resgatar o conhecimento e estratégias das Matrizes
- **Tempo:** 11min.

Momento 2:

Situação Problema: Como desenvolver um programa capaz de ler uma matriz de 3 x 3 elementos usando um comando for, e que multiplica cada elemento por 5 e imprima este resultado?

- **Atividade principal** - Informar aos alunos sobre o funcionamento da atividade e fazê-lo utilizando a estratégia da ABP;
- **Ação:** Organizar os estudantes, determinar as funções de seus componentes, seguido as regras propostas, por seguinte resolver o desafio.
- **Tempo:** 18 min.

Momento 3:

- **Discussão das soluções** – discutir e compartilhar os percursos trilhados na resolução do desafio e as estratégias utilizadas para resolver o desafio.

- **Ação:** Consolidar a estratégia da ABP e seus significados relacionados a matrizes.
- **Tempo:** 4 min.

Momento 4:

- **Raio X:** Observar e consolidar as aprendizagens dos estudantes sobre o uso estratégia da PBL para o programa utilizando as matrizes 3X3;
- **Ação:** Utilizar os saberes da estratégia apreendida em sala de aula para revolucionar a questão aplicada pelo professor e posteriormente comparar a resolução com os colegas em sala de aula.
- **Tempo:** 12 min.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo Matrizes.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Matrizes:

- Dimensões e índices;

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de matrizes;
- Incentivar o trabalho a autoaprendizagem;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- Editor de texto *Sublime Text*;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades;
- Realização da missão.

7) Descrição detalhada da estratégia da aula

Aula - Matrizes:

- Aplicação da questão problematizadora.

Ao final, na missão, haverá um feedback do professor e dos estudantes.

8) Observações adicionais

Na atividade de realização do problema, o professor pode acompanhar o desenvolvimento dos estudantes em sala de aula e, os estudantes podem solicitar dicas do professor.

Desafios:

- Explicar o conceito de Matrizes com exemplos em sala de aula e direcionar exercícios que possam ser demandados pelos estudantes.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Matrizes, bem como sua aplicação;
- Auxiliar na formação de equipes para construção da atividade das missões.

Ação dos estudantes:

Realizar o problema indicado pelo professor e apresentar em sala de aula.

Observações:

- Ainda cabe como ações dos estudantes e professores de acordo com Camargo (2019, p. 117):

Quadro 6 - Ação dos professores e estudantes em na aula de Matrizes em ABP

Estudantes	Professores
Reconhecer o sucesso no trabalho dos outros, além de se perceberem como colaboradores e fornecedores de recursos didáticos entre si (avaliação entre pares); Ser agente da própria aprendizagem, isto é, reconhecer o sucesso em seu próprio trabalho e se concentrar em como e o que estão aprendendo e onde ainda podem melhorar (autoavaliação).	Esclarecer e compartilhar as intenções de aprendizagem e critérios para obter sucesso: para que os alunos compreendam o que estão tentando aprender, por que e o que se espera deles; Organizar discussões e atividades efetivas em sala de aula de forma a evidenciar a aprendizagem: para criar um ambiente na sala de aula onde os alunos possam expor suas próprias ideias, pensar em voz alta e explorar a sua compreensão; Fornecer feedback que seja capaz de promover a aprendizagem: referenciando a qualidade de seu trabalho e o que os alunos podem fazer para torná-lo melhor;

Fonte: Organizado pelo autor (2023)

FEEDBACK:

Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

CARVALHO, Melisse. **Estratégias para avaliação na aprendizagem baseada em problemas.** In LOPES, Renato Matos. **APRENDIZAGEM BASEADA EM PROBLEMAS: Fundamentos para a aplicação no ensino médio e na formação de professores.** 2019. Disponível em: <https://educapes.capes.gov.br/bitstream/capes/432641/2/APRENDIZAGEM%20BASEADA%20EM%20PROBLEMAS%20-%20fundamentos%20para%20a%20sua%20aplica%C3%A7%C3%A3o%20no%20Ensino%20M%C3%A9dio%20e%20na%20Forma%C3%A7%C3%A3o%20de%20Professores.pdf> Acesso em: 26 de janeiro de 2023.

LUZERNA, Ricardo. **Matrizes.** 2014. Disponível em: <https://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2014/11/Exercícios-de-Linguagem-C-resolvidos.pdf> Acesso em: 02 de maio de 2022.

ANEXO

Sugestões de atividades para exemplos em sala de aula, baseadas em Luzerna (2014):

Faça um programa que lê uma matriz de 3 x 3 elementos usando um comando for, multiplica cada elemento por 5 e imprime o resultado.

```
#include <studio.h>

#include <stdlib.h>

int main () {

    int i , j , m[3] [3] ;

    //captura os elementos

    for (i=0; i<3; i++) {

        for (j=0 ; j<3 ; j++) {

            printf("Elemento[%d] [%d]=", i , j) ;

            scanf("%d" , &m [i] [j] ) ;

        }

        //EXIBIR VALORES ORIGINAIS

        printf ("\n : : : Valores Originais : : :") ;

        for (i=0 ; i<3; i++) {

            for (j=0 ; j<3; j++)

                printf ("%d " ,m [i] [j] ) ;

            printf("\n")

        }

        //multiplica por 5

        for (i=0; i<3; i++)

            for (j=0; j<3; j++)
```

```
        m[i][j]=m[i][j]*5;

//EXIBIR VALORES MULTIPLICADOS

printf ( "\n :: : Valores Multiplicados por 5 :: : \n ");

for ( i=0; i<3 ; i++ ) {

    for ( j=0; j<3 ; j++ )

        printf ( " %d " , m [i] [j] );

    printf ("\n");

}

return 0 ;

}
```

Matrizes: Estratégias e Aplicação na Gamificação

METODOLOGIA DE ENSINO: GAMIFICAÇÃO

Conteúdo: Matrizes

Metodologia de ensino: Gamificação

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Matrizes, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos de Matrizes e a resolução de problemas por meio de funções e a aprendizagem ativa.

PLANO DE AULA

Metodologia Ativa: Gamificação

Objetivo da aula:

- Desenvolver uma aula de 50 minutos através da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.
- Fixar o conteúdo de Matrizes utilizando a metodologia ativa de Gamificação.

1) Metodologia a ser trabalhada

Procedimento:

- Para o desenvolvimento da aula serão necessários:
 - *Sublime Text*.
 - Turma criada no *Google Classroom* (ou similar)
 - *App Kahoot*
- A atividade apresentada será realizada em uma aula de 50 minutos.
- Para o desenvolvimento da aula deverão estar preparados no Classroom:

- Ao menos dois textos para leitura antes da aula sobre o tema de Matrizes.
- Uma missão disponível por meio de *link* criado no aplicativo *Kahoot*.
- Atividades pré-estabelecidas no aplicativo *Kahoot*

Especificamente, durante o dia da aula.

- Em sala de aula, apresentar a atividade disponível no *Kahoot*, pré-construídos pelo professor, solicitando que os alunos acessem o link: [Kahoot!](#) e, entrem no App e em grupos, realizem as missões definidas na plataforma.

Metodologia de ensino: Gamificação

- A aplicação da metodologia Gamificação será realizada da seguinte forma:
- Formação dos grupos para competição;
- Solicitação para criação do nome do grupo para a competição;
- Disponibilização do *link*: <https://play.kahoot.it/v2/?quizId=818d31d3-27ab-45a4-b524-72f278d31415> para os estudantes acessarem e começarem o a competição;
- A competição, remete a resolução das equações/funções que os estudantes deverão realizar sobre o conteúdo tratado em aula sobre Estruturas de Repetição.
 - Apresentação da atividade disponível no “*Kahoot*”, disponível em: <https://create.kahoot.it> no Google e *Play Store (Android)*.
 - Após a apresentação, o professor deverá indicar aos estudantes que acessem o link das atividades no Kahoot e iniciem as missões.
 - Em seguida, o professor deverá acompanhar o desenvolvimento dos estudantes. A realização das missões, podem ser visualizadas em tempo real pelo professor.
 - Cada estudante deverá realizar as missões o mais rápido possível.
 - E por seguinte apresentar ao professor a explicação da função desenvolvida.

A finalidade consiste em:

- Fomentar o entendimento dos conceitos e benefícios do uso de Matrizes, tais como:
 - Dimensões;

- Principais conceitos de Matrizes em programação;

Detalhamento da metodologia, seguindo planejamento de uma aula de 50 minutos:

Anteriormente ao dia da aula:

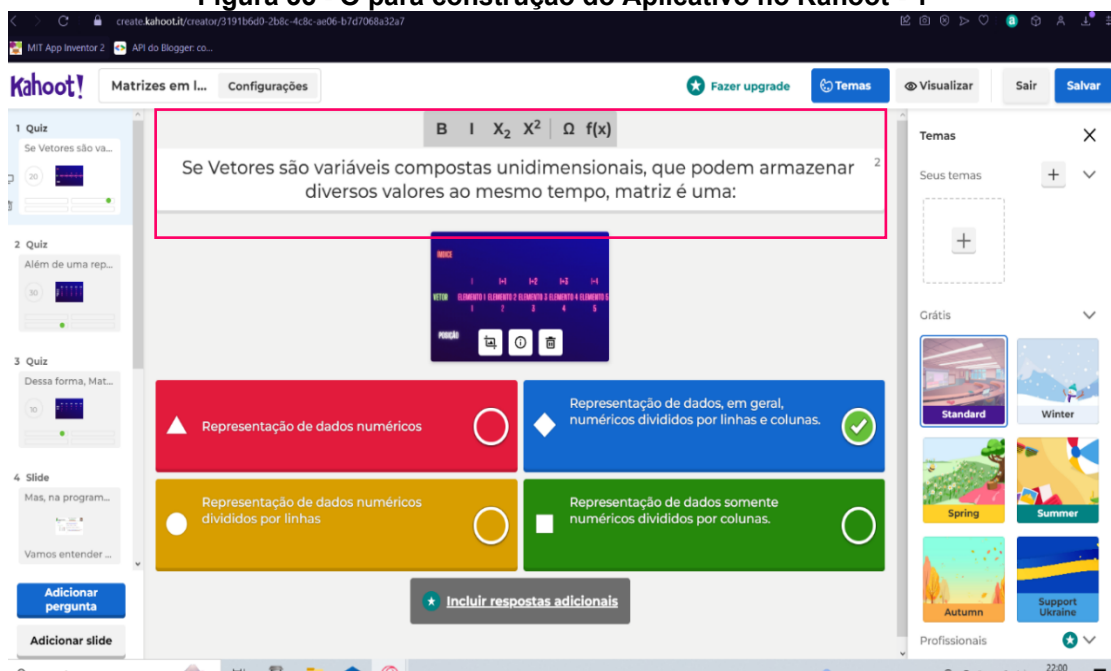
Anteriormente ao dia da aula, caberá ao professor, seguir os seguintes passos:

- **Criar a conta de professor:** Ao entrar no link: <https://play.kahoot.it/v2/?quizId=3191b6d0-2b8c-4c8c-ae06-b7d7068a32a7>

caberá ao professor:

- Criar uma conta para “Professor”;
- Criar os conteúdos baseados na disciplina;
- Preparar *quizzes* com perguntas direcionadas para o conteúdo e vídeos que complementem o que foi trabalhado em sala de aula. Como representado na Figura 35:

Figura 35 - O para construção do Aplicativo no Kahoot - 1

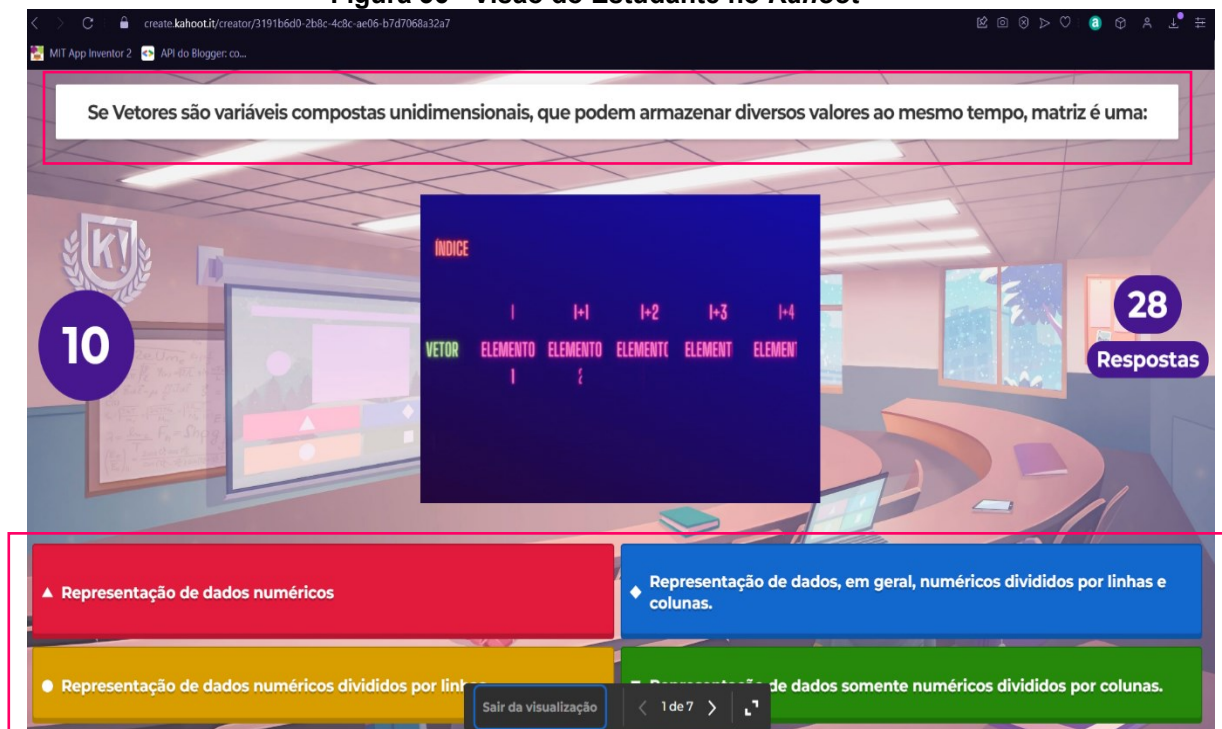


Fonte: Organizado pelo autor (2023)

A visão do estudante é a registrada na Figura 36. Diferente do professor, que pode elaborar uma atividade e compartilhar o *link* para acesso, o estudante, terá acesso apenas para resolver os problemas que tiverem sido elaborados em determinada estratégia.

Por se tratar de uma plataforma gamificada, os participantes ganham pontos à medida em que forem acertando as questões. O jogo vai sendo desenvolvido e é feito o ranqueamento das pontuações, que pode ser acompanhada em tempo real pelo professor.

Figura 36 - Visão do Estudante no Kahoot



Fonte: Organizado pelo autor (2023)

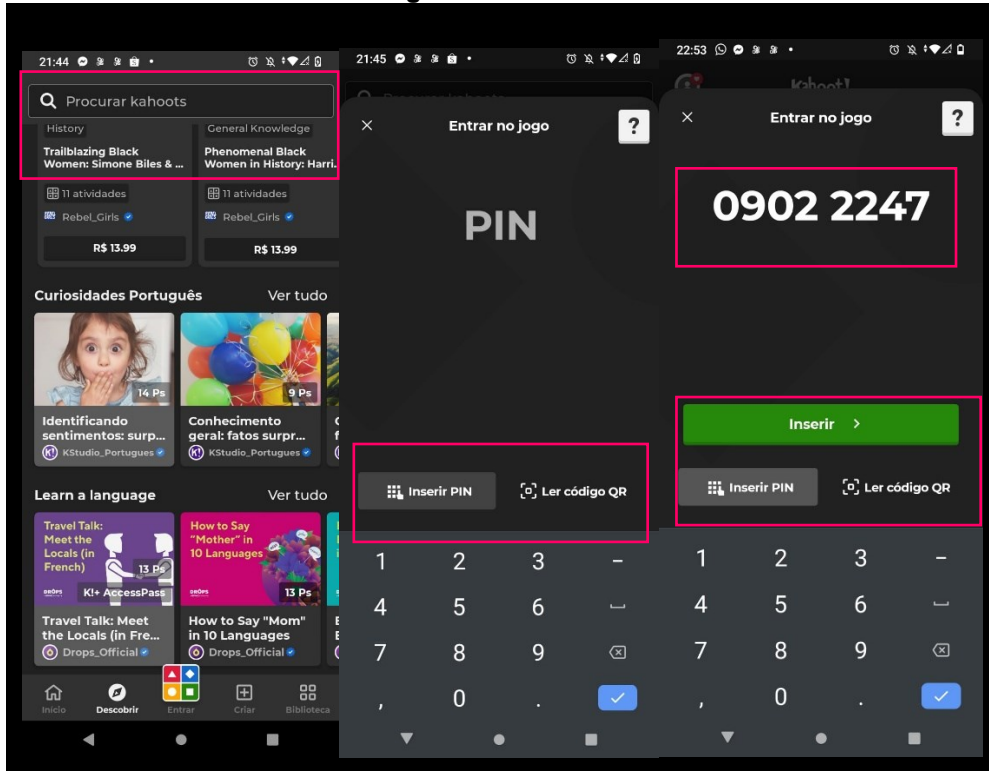
Este processo de instalação e criação da aula, pelo aplicativo, pode ser observado no seguinte vídeo disponível no seguinte [link: Create kahoot - Kahoot! - Opera 2023-02-23 22-05-21.mp4](#)

No dia da aula:

Socialização dos estudantes com o aplicativo: Solicitar que os estudantes acessem o [link da atividade no Kahoot: Escolher modo – Kahoot!](#) disponível na plataforma do *Google Classroom* (pode ser tanto pelo computador, quanto pelo celular). Ao entrar como estudante, surgirá a mensagem “já tem um PIN de jogo? Insira-o aqui para participar desse *Kahoot!*”.

- A seguir o estudante, deverá acessar o link: [Escolher modo – Kahoot!](#) ou inserir o PIN de sua aula. Como no exemplo registrado na 37 com o PIN – **09022247**.

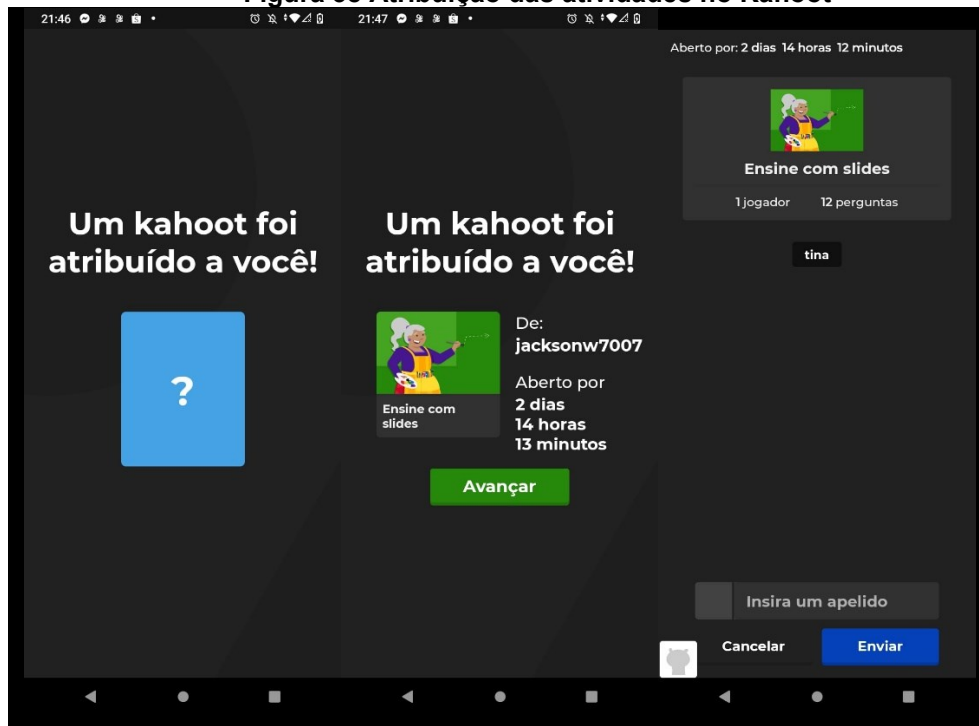
Figura 37 - do Modo – Kahoot



Fonte: Organizado pelo autor (2023)

Inserido o PIN, caberá ao professor, solicitar/acompanhar os estudantes para as missões que lhes foram atribuídas.

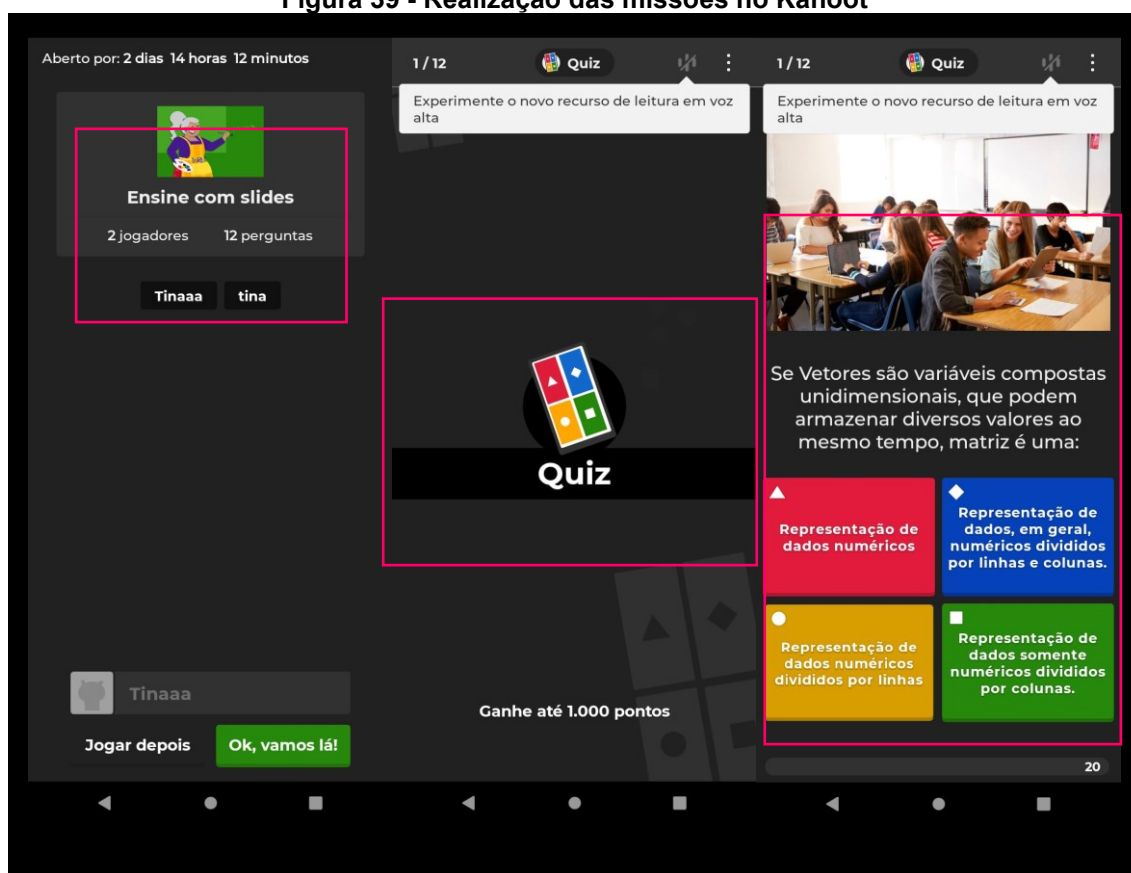
Figura 38 Atribuição das atividades no Kahoot



Fonte: Organizado pelo autor (2023)

Atribuído as atividades, caberá aos estudantes realizar as missões dentro dos Quiz.

Figura 39 - Realização das missões no Kahoot



Fonte: Organizado pelo autor (2023)

- Observe que no *quizz* é possível entrar mais de um estudante. Logo, a turma inteira poderá participar da estratégia desenvolvida pelo professor.
- Vale ressaltar que o aplicativo permite a opção “ler em voz alta”, para favorecer a acessibilidade.
- Após indicado aos estudantes que acessem o aplicativo, é importante engajar os acadêmicos que participem da missão;
Ao iniciar a missão caberá ao professor;
- Acompanhar, por meio da conta de professor, o andamento da missão e o ranqueamento dos grupos.
Ao final do ‘momento 2’, nos últimos cinco minutos deverá ocorrer um *feedback* com a turma, sobre o conteúdo trabalhado.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente a matrizes.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

- Estruturas de repetição;
- Definição de matrizes;
- Conceitos principais e exemplos de aplicação;

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos de matrizes;
- Aprendizagem Ativa.
- Trabalho em equipe

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- *Google Classroom* para disponibilização dos textos-base e indicação de leituras (além do texto-base da missão, se preferir);
- Editor de texto *Sublime Text* ou qualquer outro editor de texto da preferência do professor;
- *App Kahoot*.

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;
- Realização das missões.

7) Descrição detalhada da estratégia da aula

Aula sobre Matrizes:

- Será proposto para os estudantes acessarem a plataforma Kahoot, utilizando o *link* disponibilizado pelo professor, para realizarem as missões contidas no aplicativo.
- Ao final de cada missão, haverá um feedback em conjunto com a turma.

8) Observações adicionais

Na atividade de realização das missões, o professor pode acompanhar o desenvolvimento dos estudantes, via plataforma. A qualquer momento, durante a realização da atividade, os estudantes podem solicitar dicas para o professor.

Desafios:

- Explicar o aplicativo, baixar o mesmo no celular (isto é opcional) e indicar a criação de grupos.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre matrizes, bem como sua aplicação;
- Acessar do aplicativo do *Kahoot* e montar as aulas em formato de quizz;
- Auxiliar na formação de equipes para construção da atividade das missões.

Ação dos estudantes:

- Acessar ou baixar o App do *Kahoot*
- Realizar as missões desenvolvidas pelo professor.

Feedback:

Caberá ao professor, a realização de *feedback* dos trabalhos e atividades desenvolvidos.

REFERÊNCIAS

KAHOOT. **Endereço do aplicativo Kahoot**. 2023. Disponível em: <https://create.kahoot.it> Acesso em: 23 de fevereiro de 2023.

LUZERNA, Ricardo. **Matrizes**. 2014. Disponível em: <https://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2014/11/Exercícios-de-Linguagem-C-resolvidos.pdf> Acesso em: 02 de maio de 2022.

PLUSKOTA, J. W. **Vídeo explicativo**. Disponível em: [Create kahoot - Kahoot! - Opera 2023-02-23 22-05-21.mp4](#) Desenvolvido em: 23 de fevereiro de 2023.

METODOLOGIA DE ENSINO: TEAM BASED LEARNING

Conteúdo: Matrizes

Metodologia de ensino: Aprendizagem Baseada em Times

Descrição detalhada da estratégia: Uma atividade em aula presencial no qual será realizada durante o ensino da matéria de Introdução a programação no conteúdo de matrizes, considerando uma aula de 50 minutos. Terá como objetivo fomentar a compreensão dos conceitos Matrizes e a resolução de problemas por meio de funções e a aprendizagem ativa.

PLANO DE AULA

Metodologia Ativa: Aprendizagem baseada em times

Objetivo da aula:

- Desenvolver uma aula de 50 minutos através da utilização de metodologias ativas de ensino e aprendizagem e TICs, para o fomento da aprendizagem ativa no ensino de programação durante o ensino presencial.
- Fixar o conteúdo de Matrizes utilizando a metodologia TBL.

1) Metodologia a ser trabalhada

Procedimento:

Para o desenvolvimento da aula serão necessários:

- Anteriormente a aula:
 - Disponibilização de textos e vídeos explicativos aos estudantes por meio da plataforma *Google Classroom* (ou similar), para leitura e conhecimento prévio do conteúdo a ser trabalhado em sala de aula.

- Explicação sobre a estratégia TBL a ser desenvolvido em aula, para que os alunos, compreendam o que se pretende com a atividade que será proposta.
- **No dia da aula:**
 - Explicação sobre Matrizes (20 min) utilizando slides (ou outro material de preferência do professor).
 - Disponibilização de atividades para os estudantes se organizarem em grupos e realizarem em sala de aula, para fixação do conteúdo.
 - **Avaliação:** verificação dos estudantes que responderam ao questionário que será entregue em sala de aula.

2) Quando aplicar este plano?

Em conjunto, com o conteúdo referente a Matrizes.

3) Qual(is) conteúdo(s) será(ão) trabalhado(s)?

Conteúdos de Matrizes:

- Conceitos de matrizes.
- *Arrays* bidimensionais.

4) Quais habilidades serão desenvolvidas na turma?

- Compreensão dos conceitos matrizes;
- Incentivar o trabalho em grupos;
- Resolução de problemas por meio de funções.
- Aprendizagem Ativa.

5) Objetos de aprendizagem a serem usados (documentos, apresentações, lista de exercícios, vídeos, jogos, ...):

- *Classroom*;
- *Power Point* para o estudo dirigido;
- Utilização do quadro negro para exemplificação dos estudos;
- Lista de exercícios em sala de aula para trabalho em grupo;

6) Forma de avaliação

- Observação da participação dos estudantes nas atividades em grupo;
- Envio das resoluções via *Google Classroom*.

7) Descrição detalhada da estratégia da aula

Aula sobre Matrizes

Procedimento:

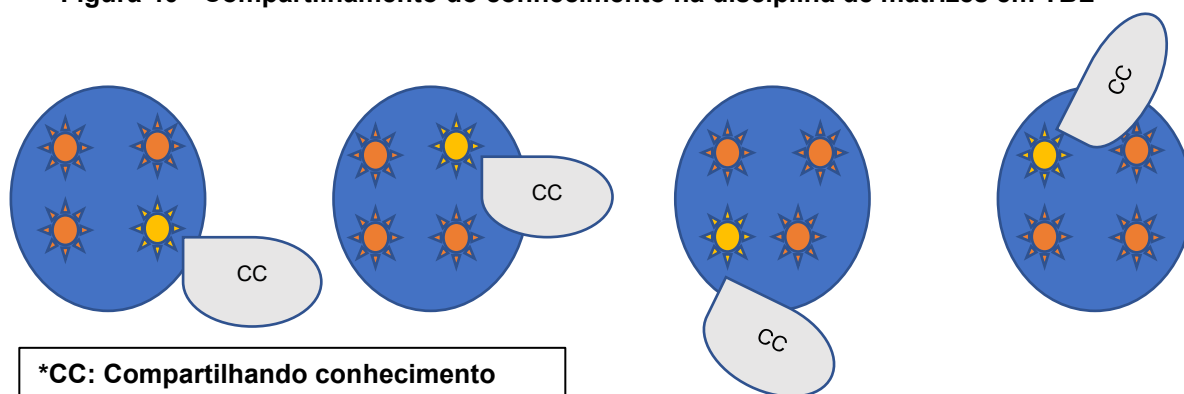
Anteriormente ao dia aula:

- Será apresentada a metodologia aos estudantes e informado sobre os materiais de estudos, disponíveis na sala criada no Google Classroom.
- Adicionalmente, antes da aula, será disponibilizado por meio do Google Classroom dois vídeos explicativos, sendo:
- Matrizes em linguagem C disponível em:
<https://www.youtube.com/watch?v=8d1bH8bkj5Q>

Durante a aula:

- Apresentação de *Power Point* para a aula expositiva, como no exemplo disponível em:
https://docs.google.com/presentation/d/1js3kXdtCmJQnJVjEs7Cp9DFmHnyHPvUgQtdVsj-xZK0/edit#slide=id.g1dcec63b09d_1_6
- Apresentação da metodologia da aula;
- Sorteio aleatório de times para compor a atividade da aula (utilizando o livro de chamadas ou aplicativo de sorteio online) - com até quatro estudantes
- Cada grupo, receberá a descrição das atividades de fixação, tais como as disponíveis em: <https://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2014/11/Exercícios-de-Linguagem-C-resolvidos.pdf>
- Ao logo das discussões dos alunos, o professor deverá passar em cada grupo para verificar as eventuais dúvidas e como está o andamento da atividade.
- De acordo com a resolução das atividades, cada time será redirecionado para os outros times, para colaborar com a resolução do problema, para que assim, todos possam se ajudar, conforme ilustrado na 40.

Figura 40 - Compartilhamento do conhecimento na disciplina de matrizes em TBL



Fonte: Organizado pelo autor (2023)

- Ao final, quando todos os grupos resolverem os problemas propostos, os estudantes deverão indicar um líder do grupo para explicar como encontraram a solução para os problemas.

8) Observações adicionais

Na atividade os estudantes deverão explicar como chegaram ao resultado apresentado.

Desafios:

- Resolver em grupo a questão disponibilizada.
- Seguir os quatro princípios essenciais da TBL

1ª Distribuição adequada dos grupos:

- Formar adequadamente os grupos, de modo que não formem times “panelinhas”, minimizando assim, possíveis barreiras durante a formação dos times. Sendo então, grupos de cinco e sete pessoas, que permaneçam nos mesmos times ao longo do semestre.

2ª Tornar o estudante responsável por seu progresso:

- Contribuir para que os estudantes, tornem-se protagonistas de suas ações por meio de responsabilidades prévias coletiva e individuais (planejamento).

3ª Promoção da interação por meio das atividades:

- Fomentar o diálogo entre os times e possíveis planos de ação futuras com os resultados das atividades (artigo, site ou game por exemplo).

4ª Constante *feedback*:

- O papel do professor é mediar as interações dos estudantes, neste sentido, é importante dar retorno quanto as potencialidades e fragilidades dos times ao longo de suas atividades.

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Matrizes, bem como sua aplicação;
- Auxiliar na formação dos times para construção da atividade avaliativa.
- Mediar as interações dos times.
- Gerar constante *feedback*

Ação dos professores:

- Apresentar os conceitos e o conteúdo sobre Matrizes, bem como sua aplicação.
- Auxiliar na formação dos times para resolução da atividade disponibilizada.

Ação dos estudantes:

- Resolver a questão em time e apresentar o resultado.

***Feedback*:**

Entregar *feedback* dos trabalhos e atividades.

REFERÊNCIAS

LUZERNA, Ricardo. **Exercício de Matrizes**. 2014. Disponível em: <https://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2> Acesso em: 24 de janeiro de 2023.

MARTINS, Pietro. **Matrizes e Arrays Multidimensionais em C**. 2022. Disponível em: <https://www.youtube.com/watch?v=8d1bH8bkj5Q> Acesso em: 23 de novembro de 2023.

PLUSKOTA, J. W. **Aula de Matrizes em linguagem C**. 2022. Disponível em: <https://docs.google.com/presentation/d/1js3kXdtCmJQnJVjEs7Cp9DFmHnyHPvUgQtdVsj-xZK0/edit?usp=sharing> Acesso em: 23 de novembro de 2022.

CONSIDERAÇÕES

Este e-book parte do pressuposto de que planos de aula e estratégias de ensino, são ferramentas essenciais no campo da educação e possuem diversas finalidades e benefícios. E dentre as principais razões de suas necessidades, estão a organização e estruturação de estratégias de ensino diferenciadas que permitam um aprendizado ativo do estudante.

Isto porque, as estratégias de ensino, devem fornecer uma estrutura organizada para o ensino. Permitindo que os educadores planejem e organizem o conteúdo, as atividades e os recursos necessários para alcançar os objetivos de aprendizagem. Ajudando o professor, estabelecer uma sequência lógica de instruções e a definindo um cronograma para as atividades.

Além da estrutura e organização destaca-se que as estratégias de ensino devem ser objetivas e claras, pois, estas estratégias contribuem para que os educadores estabeleçam objetivos claros de aprendizagem para os estudantes. Isso significa identificar o que os estudantes devem saber, entender ou serem capazes de fazer no final da aula ou da disciplina. Pois, os objetivos auxiliam na definição das expectativas de aprendizagem e orientam o processo de instrução.

Vale destacar que as estratégias de ensino, também necessitam apresentar um alinhamento curricular, considerando que as elas permitem que os educadores alinhem seus ensinamentos aos currículos e padrões educacionais estabelecidos. Garantindo que o conteúdo a ser ensinado esteja em conformidade com os requisitos e as diretrizes educacionais relevantes.

Neste sentido, são apresentados neste e-book, planos de aulas que visam focalizar de forma clara os temas relacionados a aprendizagem de programação por meio das metodologias ativas. Especificamente das metodologias ativas da sala de aula invertida, aprendizagem baseada em problemas, gamificação e aprendizagem baseada em times.

Este e-book, fornece por meio de suas estratégias, uma base para a consistência entre diferentes sessões de ensino, podendo ser aplicado em diferentes cursos de graduação que atuem com a disciplina de programação. Buscando ainda, facilitar a comunicação entre educadores, permitindo que estes, compartilhem e discutam as estratégias e recursos para melhorar a

qualidade de ensino por meio das metodologias ativas e da utilização das tecnologias de informação e comunicação.

Considera-se desta forma, que as estratégias que compõe este e-book, servem como guias estruturados que auxiliam os educadores a planejar, organizar e entregar a instrução sobre programação, de forma eficaz, garantindo uma aprendizagem ativa e facilitando o processo de ensino-aprendizagem.

Por meio das metodologias ativas, estas estratégias de ensino, tem como finalidade, tornar as aulas de programação mais ativa, divertida e diferenciada. Instigando nos estudantes a colaboração entre equipes, capacidade de problematização e resolução de problemas e, fomentando o processo de criatividade.

REFERÊNCIAS

AUGUSTO, João. **Estrutura ou laços de repetição**. 2018. Disponível em: <http://docente.ifsc.edu.br/joao.augusto/MaterialDidatico/2018-1/Introdução%20à%20Programação/JavaScript/Laços%20de%20Repetição.pdf> Acesso em 11 de agosto de 2022.

CARVALHO, Melissa. **Estratégias para avaliação na aprendizagem baseada em problemas**. In LOPES, Renato Matos. **APRENDIZAGEM BASEADA EM PROBLEMAS: Fundamentos para a aplicação no ensino médio e na formação de professores**. 2019. Disponível em: <https://educapes.capes.gov.br/bitstream/capes/432641/2/APRENDIZAGEM%20BASEADA%20EM%20PROBLEMAS%20-%20fundamentos%20para%20a%20sua%20aplica%C3%A7%C3%A3o%20no%20Ensino%20M%C3%A9dio%20e%20na%20Forma%C3%A7%C3%A3o%20de%20Professores.pdf> Acesso em: 26 de janeiro de 2023.

COSTA, Yanko Yanez Keller da. **O desenvolvimento de jogos digitais como metodologia ativa no ensino de programação de computadores no ensino superior**. 2020. 96 f. Dissertação - Mestrado Profissional em Educação e Novas Tecnologias, 2020. Disponível em: <https://repositorio.uninter.com/handle/1/469> Acesso em 21 de fev. de 2022.

CHAGAS, Joselito M. **Aprendizagem Ativa do Estudante – Aplicação de Project Based Learning nos Cursos de Engenharia**. Universidade Estadual Paulista. 2020. Disponível em: <https://www.revistasuninter.com/intersaberes/index.php/revista/article/view/1821> Acesso em: 24 de fev. de 2022

FERNANDES, Gustavo Q. **Exercícios sobre estruturas de decisão em algoritmos**. 2020. Disponível em: <https://mecatronicacemporcento.com.br/exercicios-sobre-estruturas-de-decisao-em-algoritmos/> Acesso em: 26 de janeiro de 2023.

Freire, Paulo. **Educação como prática da liberdade**. 18ª Ed. Rio de Janeiro: Paz e Terra, 1987.

GOOSECHASE. **Acesso desenvolvimento**. Disponível em: <https://www.goosechase.com/games/> Acesso em: 02 de maio de 2022.

GOMES, Eduardo Savino. **Narrativas OC2-RD2 e PBL: uma proposta para o ensino da programação de computadores**. Pontifícia Universidade Católica de São Paulo. 2021. Disponível em: <https://tede.pucsp.br/handle/handle/24304> Acesso em: 24 de fev. de 2022.

GOMES, Maria Solange dos Santos; MAZUIM, Cleusa Helena Rockembach. **Teorias e Vivências de Metodologias Ativas**. 1º Edição. Editora Perse. 2019

KAHOOT. **Endereço do aplicativo Kahoot**. 2023. Disponível em: <https://create.kahoot.it> Acesso em: 23 de fevereiro de 2023.

KRASSMANN, Aliane Loureiro. **JOGO SÉRIO UBÍQUO INTEGRADO A MUNDO VIRTUAL OPENSIM PARA O ENSINO DE REDES DE COMPUTADORES**

(JASPION). Dissertação. Santa Maria/RS. 2016. Disponível em: <https://repositorio.ufsm.br/handle/1/17702> Acesso em: 23 de janeiro de 2022.

LIMA, Árlon Chaves. **Metodologia 7Cs: uma proposta de ensino e aprendizagem para disciplinas introdutórias à programação**. Orientador: Marcos Monteiro Diniz. Coorientadora: Marianne Kogut Eliasquevici. 2020. 169 f. Dissertação (Mestrado Profissional em Ensino) - Programa de Pós-Graduação Criatividade e Inovação em Metodologias de Ensino Superior, Núcleo de Inovação e Tecnologias Aplicadas a Ensino e Extensão, Universidade Federal do Pará, Belém, 2020. Disponível em: <http://repositorio.ufpa.br/jspui/handle/2011/12990> Acesso em 21 de fev. de 2022.

Lima, André. **Aula Modularização de código no portugol studio**. Disponível em: <https://quizizz.com/join/quiz/61939d914b6e10001e46987f/start> Acesso em: 05 de maio de 2022.

Lima, André. **Aula 12 de funções e modularização**. Disponível em: <https://quizizz.com/join/quiz/60ac0b3ceb63c5001bf986f8/start?from=admin> Acesso em: 05 de maio de 2022.

LIMA, Layara K O S; SANTOS, Ernani M. Metodologias ativas e suas contribuições para os processos de ensino e aprendizagem. **7 Congresso de educação CONEDU**. 2020. Disponível em: https://editorarealize.com.br/editora/anais/conedu/2020/TRABALHO_EV140_M D1_SA19_ID5564_01092020215918.pdf Acesso em: 28 de fev. de 2022.

LUZERNA, Ricardo. **Exercício de Matrizes**. 2014. Disponível em: <https://professor.luzerna.ifc.edu.br/ricardo-antonello/wp-content/uploads/sites/8/2> Acesso em: 24 de janeiro de 2023.

MARIANO, Diego. **Laços de Repetição**. 2020. Disponível em: <https://diegomariano.com/lacos-de-repeticao-2/#:~:text=La%C3%A7os%20de%20repeti%C3%A7%C3%A3o%2C%20tamb%C3%A9m%20conhecidos,bloco%20sejam%20repetidos%20diversas%20vezes.> Acesso em: 02 de maio de 2022.

MARTINS, Pietro. **Matrizes e Arrays Multidimensionais em C**. 2022. Disponível em: <https://www.youtube.com/watch?v=8d1bH8bkj5Q> Acesso em: 23 de novembro de 2023.

MARQUEZ, Mercedes R. Gonzalez. **Modularização**. 2013. Disponível em: <http://www.comp.uems.br/~mercedes/disciplinas/2013/AEDI/AEDI-modularizacao.ppt> Acesso em: 27 de janeiro de 2023.

MECATRONICA. **Exercícios sobre estrutura de decisão em algoritmos**. 2023. Disponível em: <https://mecatronicacemporcento.com.br/exercicios-sobre-estruturas-de-decisao-em-algoritmos/> Acesso em: 18 de janeiro de 2023.

MAZUIM, C. H. R; GOMES. M. S dos S. **Teorias e Vivências de Metodologias ativas**. São Paulo: Perse. 1º Ed. 2019.

PAPERT, Seymour. **A máquina das crianças: repensando a escola na era da informática**. Porto Alegre: Artmed, 2008.

PEREIRA, Jonathan. **Lista de exercício de algoritmo**. S/A. Disponível em: <https://docente.ifrn.edu.br/jonathanpereira/disciplinas/algoritmos/lista-de-exercicios-2/view> Acesso em: 23 de dezembro de 2022.

PLUSKOTA, Jackson William. **Decisão Simples e Composta**. 2022. Vídeo Aula. Disponível em: <https://www.youtube.com/watch?v=vFEuFYUZTRw> Acesso em: 23 de dezembro de 2022.

PLUSKOTA, Jackson W. **Modelo aula gamificada de Estrutura de Decisão Simples e Composta**. 2023. Disponível em: <https://forms.gle/kqdfa3C8mGRbTb2h6> Acesso em: 18 de janeiro de 2023.

PLUSKOTA, Jackson W. **WebQuest – Sites Google**. 2023. Disponível em: <https://sites.google.com/view/jacksonpluskota/p%C3%A1gina-inicial> Desenvolvido em: 26 de janeiro de 2023

PLUSKOTA, Jackson W. **Google Forms. 2023**. Disponível em: <https://forms.gle/Y83k3qZefJj36BfF7> Desenvolvido em: 26 de janeiro de 2023.

PLUSKOTA, Jackson W. Texto APK Twine, 2022. Disponível em: [A casa inteligente.html](#) Desenvolvido em maio outubro de 2022.

PLUSKOTA, Jackson W. APK Twine, 2022. Disponível em: [Twine](#) Desenvolvido em maio outubro de 2022

PLUSKOTA, Jackson W. **Exercícios sobre laços de repetição**. 2022. Disponível em: <https://forms.gle/n1KxBwxWbM3StvGP7> Desenvolvido em: 11 de agosto de 2022.

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C**. 2023. Disponível em: Acesso em: 13 de março de 2023.

PLUSKOTA, J. W. **Aula sobre Vetores em linguagem C: Atividades de fixação**. 2023. Disponível em: <https://forms.gle/k3ARuBTp143B85SQ6> Acesso em: 13 de março de 2023.

PLUSKOTA, Jackson W. **Exercícios de Matrizes resolvidos**. 2023. Disponível em: [Exercícios-de-Linguagem-C-resolvidos.pdf](#) Desenvolvido em: 16 de março de 2023.

PLUSKOTA, Jackson W. **Matrizes UTFPR APK**. 2023. Disponível em: [Matrizes UTFPR.apk](#) Desenvolvido em: 16 de março de 2023.

PLUSKOTA, J. W. **Vídeo explicativo**. Disponível em: [Create kahoot - Kahoot! - Opera 2023-02-23 22-05-21.mp4](#) Desenvolvido em: 23 de fevereiro de 2023.

PLUSKOTA, J. W. **Aula de Matrizes**. 2022. Disponível em: <https://docs.google.com/presentation/d/1js3kXdtCmJQnJVjEs7Cp9DFmHnyHPvUgQtdVsj-xZK0/edit?usp=sharing> Acesso em: 23 de novembro de 2022.

Ribeiro, Fabrício de S. **Avaliação do impacto de ambientes gamificados no processo de ensino-aprendizagem da lógica de programação de computadores: uma comparação entre elementos monousuário e multiusuários**. Universidade Federal do Pará. 2019.

RODRIGUES, Claudio C. **Linguagem C - Estrutura de Seleção Condicional**. UFU. 2021. Disponível em: <https://www.facom.ufu.br/~crlopes/IC/8-C-Condicionais.pdf> Acesso em: 11 de janeiro de 2023.

SALAMON. Jordana S. **Programação básica de computadores**. S/A. Departamento de informática. Universidade Federal do Espírito Santos. Disponível em: <http://www.inf.ufes.br/~jssalamon/wp-content/uploads/disciplinas/prog/slides/Slide%208%20-%20Vetores%20em%20C.pdf> Acesso em: 13 de março de 2023.

SANTANA, Max. **Exercícios de fixação sobre modularização**. 2013. Disponível em: <http://www.univasf.edu.br/~max.santana/material/introprog/listaExercicios04.pdf> Acesso em: 27 de janeiro de 2023.

SILVA; Adilson da; GARCIA, Ana Flávia G. G; RIBEIRO, Selma F. da C.; JESÚS, Sonia F. de. **Metodologias ativas: um desafio para o trabalho da orientação**. In. SILVA, A. R. L. da; BIEGING, P.; BUSARELLO; R. I. **Metodologia ativa na educação**. São Paulo: Pimenta Cultural, 2017. 150p.

UNICAMP. **Estrutura de Repetição**. 2011. Disponível em: <https://www.ic.unicamp.br/~wainer/cursos/2s2011/Cap06-RepeticaoControle-texto.pdf> Acesso em: 02 de maio de 2022.

JACKSON W. PLUSKOTA

JOÃO PAULO AIRES

MARIA JOÃO VARANDA PEREIRA

Metodologias ativas

Estratégias de ensino em
Programação