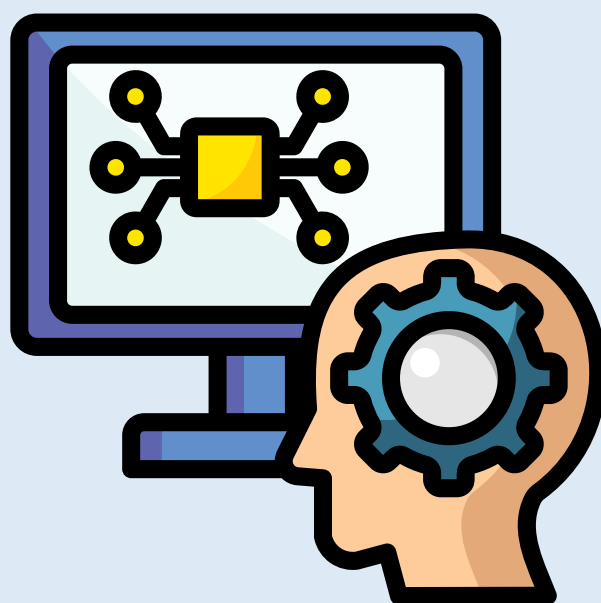




REUTILIZAR E RECOMBINAR: estratégias para o desenvolvimento do pensamento computacional no ensino técnico integrado



Autores:
Pedro Henrique Oliveira de Miranda
Júlio César Ferreira

Sobre os Autores

Pedro Henrique Oliveira de Miranda - ocupa o cargo de Técnico em Tecnologia da Informação no Instituto Federal Goiano - Campus Trindade. Possui Graduação em Análise e Desenvolvimento de Sistemas pelo Instituto Federal Goiano - Campus Urutaí, 2012; Especialista em Desenvolvimento Web com Interfaces Ricas pela Universidade Federal de Goiás, 2015 e mestrando do Programa de Pós-Graduação em Educação Profissional e Tecnológica (ProfEPT) no Instituto Federal de Educação, Ciência e Tecnologia Goiano (IF Goiano) - campus Ceres.

Orientador, Júlio Cesar Ferreira:

Professor efetivo no Instituto Federal Goiano e professor permanente do corpo docente dos mestrados ProfEPT e ENEB. Possui Graduação em Matemática pela Universidade Federal de Goiás, 2006; Técnico em telecomunicação pela antiga Escola Técnica Federal de Goiás, 1997; Mestrando em processamento da informação pela Universidade Federal de Uberlândia, 2011; Doutorado(cotutela) e PHD pela Escola de Doutores MathSTIC pela Universidade de Rennes 1, 2016.

REUTILIZAR E RECOMBINAR: estratégias para o desenvolvimento do pensamento computacional no ensino técnico integrado

Produto Educacional apresentado ao Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo campus Ceres do Instituto Federal Goiano, como parte dos requisitos para a obtenção do título de Mestre em Educação Profissional e Tecnológica.

Ceres-Go
2025

Ficha Técnica

Dissertação Vinculada: Práticas de reutilização e recombinação no desenvolvimento do pensamento computacional: uma sequência didática para o curso técnico em informática para a internet integrado ao médio

Mestrando: Pedro Henrique Oliveira de Miranda

Orientador: Júlio César Ferreira

Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo campus Ceres do Instituto Federal Goiano.

Linha de Pesquisa: Práticas Educativas em Educação Profissional e Tecnológica (EPT).

Ceres-Go
2025

Sumário

• Apresentação	6
• Objetivo	7
• Introdução	8
• Uma Discussão sobre a Sequência Didática	11
• Pensamento Computacional	12
• A Sequência Proposta	17
• Aula 1 – Diagnóstico Inicial e Introdução Crítica ao Pensamento Computacional	18
• Aula 2 – Estrutura e Pilar do Pensamento Computacional	19
• Aula 3 – Introdução ao MIT App Inventor 2: Primeiros Passos	20
• Aula 4 – Reutilizando e Remisturando Códigos: Remixagem Crítica	21
• Aula 5 – Desenvolvimento do Projeto Final com MIT App Inventor	23
• Aula 6 – Socialização e Avaliação Final dos Projetos	24
• Considerações Finais	25
• Referências	27

Apresentação

Caro/a professor/a

A presente sequência didática (SD) foi concebida a partir dos referenciais teóricos que fundamentam o desenvolvimento do pensamento computacional e do uso de práticas pedagógicas baseadas na reutilização e a recombinação, e da análise do contexto educacional do ensino técnico integrado ao médio. A seguir, serão apresentados fundamentos, a estrutura e a aplicação da sequência didática proposta.

Esta sequência didática foi elaborada para auxiliar professores e estudantes do primeiro ano do curso técnico em Informática para Internet, integrado ao ensino médio, que cursam a disciplina de lógica de programação.

Os objetivos das atividades da sequência didática buscam integrar as práticas de reutilização e a recombinação no contexto do ensino técnico integrado ao médio, referente às habilidades do pensamento computacional. Essa abordagem é apoiada pela programação por blocos, utilizando a plataforma MIT App Inventor 2 como meio facilitador.

A estrutura da SD foi delineada com base nas práticas de *reusing* e *remixing*, que envolvem a **reutilização** e a **recombinação** criativa de elementos já existentes para promover a construção ativa do conhecimento.

Boa Leitura!



Objetivo

O objetivo dessa sequência didática é apresentar atividades fundamentadas nos referenciais teóricos do desenvolvimento do pensamento computacional, com foco nas práticas pedagógicas de reutilização e recombinação, objetivando apoiar professores e estudantes do primeiro ano do curso técnico em informática para internet, integrado ao ensino médio, na disciplina de lógica de programação. Por meio do uso da plataforma *MIT App Inventor 2*, a sequência visa promover uma aprendizagem dinâmica, criativa e significativa, facilitando a construção ativa do conhecimento e a aplicação prática das habilidades do pensamento computacional.



Introdução



A contemporaneidade é marcada por uma profunda transformação impulsionada pelo avanço das tecnologias digitais. Essa transformação reflete em diversos campos da vida humana, como a educação, o trabalho, a comunicação e a cultura, e recria a forma como nos relacionamos com o conhecimento, com os outros e com o mundo. Contudo, embora esses avanços tenham possibilitado novas formas de criação e acesso à informação, também têm ampliado desigualdades, especialmente no que diz respeito à exclusão digital. A falta de acesso equitativo às tecnologias reforça as disparidades sociais existentes, limitando o pleno exercício da cidadania e a inserção de muitos sujeitos na cultura digital.

Diante desse cenário, a escola assume um papel central: por meio de processos de constituição de sujeitos capazes de compreender, criar e intervir criticamente no mundo digital. Essa tarefa exige que o processo de ensino e aprendizagem se incremente, incorporando as tecnologias de forma significativa, crítica e emancipadora. Como destacam autores como Gramsci (1968) e Saviani (2003), uma educação de qualidade deve estar comprometida com a formação omnilateral dos estudantes, uma formação que os prepare para atuar no mundo do trabalho, mas também para compreendê-lo e transformá-lo. Essa perspectiva reforça a importância de uma abordagem educacional que vá além do tecnicismo e promova o pensamento crítico, a criatividade e a autonomia.



Nesse contexto, o desenvolvimento do pensamento computacional se estabelece como uma ferramenta importante para o século XXI. Trata-se de uma forma de pensar que envolve a decomposição de problemas, a identificação de padrões, a abstração de informações relevantes e a construção de soluções algorítmicas. A Base Nacional Comum Curricular (BNCC) reconhece o valor do pensamento computacional no contexto escolar, articulando-o à formação de sujeitos capazes de compreender e atuar, de forma ética e crítica, em um mundo cada vez mais mediado por tecnologias digitais (Brasil, 2018).

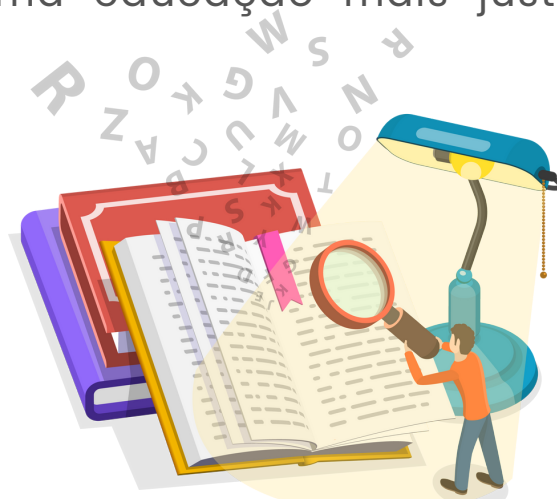
Entretanto, apesar do reconhecimento institucional, ainda são muitos os desafios para sua efetiva implementação. A ausência de um componente curricular específico voltado à computação na BNCC, por exemplo, dificulta a consolidação de propostas integradas. Como apontam Lucas, Moita e Viana (2023), o pensamento computacional aparece nos materiais didáticos e no currículo de forma fragmentada, muitas vezes tratado como conteúdo complementar, e não como um conhecimento estruturante. Isso demanda dos professores um domínio não apenas técnico, mas também didático e pedagógico, capaz de articular esse campo do saber às práticas escolares e aos conteúdos curriculares.



Nesse sentido, diferentes abordagens têm sido propostas para trabalhar o pensamento computacional na Educação Básica e na Educação Profissional e Tecnológica. Dentre elas, destacam-se as práticas de reutilização (*reusing*) e recombinação (*remixing*) de códigos e projetos, que, como afirmam Litts, Lewis e Mortensen (2019), representam estratégias criativas e colaborativas de aprendizagem, especialmente quando associadas a ferramentas de programação visual. Essas práticas promovem não apenas a apropriação técnica, mas também o engajamento coletivo, a autoria crítica e a troca de saberes entre os estudantes. Estudos como os de Brennan e Resnick (2012) mostram que práticas como testar e depurar, reutilizar e recombinaar fazem parte da lógica da programação contemporânea e devem ser valorizadas como experiências pedagógicas relevantes.

Com base nessas referências e na pertinência do tema para a formação de estudantes da Educação Profissional Técnica de Nível Médio, o presente produto educacional apresenta uma sequência didática voltada ao desenvolvimento do pensamento computacional, por meio do reuso e da remixagem de aplicativos no ambiente do MIT App Inventor 2. A proposta foi concebida como um guia pedagógico flexível, que pode ser adaptado a diferentes contextos escolares, e tem como objetivo apoiar o trabalho docente com atividades que integram teoria e prática, criatividade e criticidade.

Espera-se que este material auxilie os professores em sua prática pedagógica, oferecendo não apenas um roteiro, mas também um convite à reflexão sobre o papel da tecnologia na construção de uma educação mais justa, inclusiva e significativa.



Uma Discussão Sobre a Sequência Didática

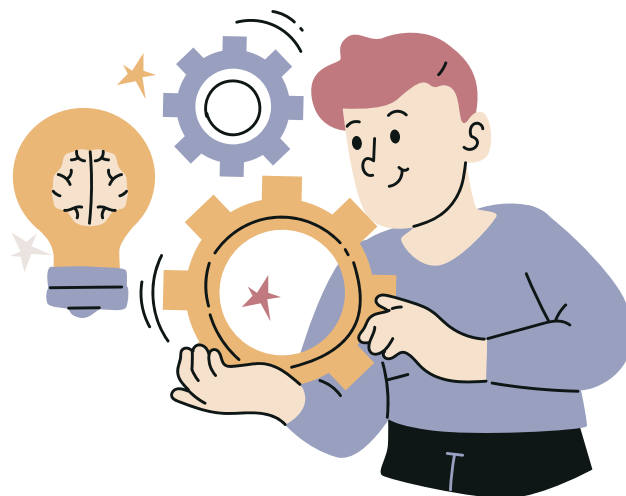
Esta sequência didática se ancora nos pressupostos da aprendizagem significativa a partir das ideias construcionistas, na perspectiva da construção de saberes em contextos reais e motivadores. O construcionismo, proposto por Seymour Papert, baseia-se na ideia de que a aprendizagem é mais eficaz quando os alunos estão ativamente envolvidos na construção de produtos ou na resolução de problemas (Papert, 1980). Esta abordagem é uma extensão do construtivismo de Piaget e enfatiza a importância de ambientes de aprendizagem onde os alunos possam experimentar e manipular objetos físicos e digitais (Harel; Papert, 1991).

A proposta está baseada nas categorias de conteúdos definidas por Zabala (1998):

- **Conteúdos factuais:** informações, nomenclaturas, dados e conceitos específicos sobre computação e programação.
- **Conteúdos conceituais:** princípios e fundamentos do Pensamento Computacional, da lógica algorítmica e do design de aplicativos.
- **Conteúdos procedimentais:** técnicas de uso do MIT App Inventor, estratégias de recombinação, etapas da prototipagem.
- **Conteúdos atitudinais:** cooperação, respeito às ideias dos colegas, autonomia e perseverança na resolução de problemas.



Pensamento Computacional



O pensamento computacional é uma habilidade essencial, e se refere à capacidade de formular e resolver problemas a partir de estratégias e conceitos oriundos da ciência da computação. Mais do que simplesmente programar, trata-se de uma forma de raciocínio lógico e estruturado que permite identificar problemas, analisar suas partes e propor soluções viáveis por meio de processos computacionais. Wing (2014) define esse pensamento como a atividade mental envolvida na formulação de problemas que admitem uma solução computacional, destacando sua relevância em diferentes contextos, muito além da área da informática.

Esse modelo mental é estruturado em quatro pilares interligados: decomposição, reconhecimento de padrões, abstração e algoritmos. A decomposição consiste em dividir um problema complexo em partes menores e mais simples, facilitando sua análise e resolução. O reconhecimento de padrões permite identificar semelhanças em problemas distintos, o que possibilita aplicar soluções já conhecidas. A abstração, por sua vez, auxilia na identificação dos aspectos mais relevantes de um problema, ignorando informações supérfluas. Já os algoritmos envolvem a criação de instruções sequenciais para resolver problemas de forma lógica e eficiente.

Além desses pilares, o pensamento computacional promove habilidades amplas, como criatividade, colaboração, autonomia e resolução de problemas, tornando-se uma competência transversal aplicável a todas as áreas do conhecimento. Brennan e Resnick (2012) destacam que o pensamento computacional não se limita ao domínio técnico, mas se apresenta como uma ferramenta cognitiva poderosa que pode transformar práticas educativas, potencializando o raciocínio e o protagonismo dos estudantes diante de desafios acadêmicos e da vida cotidiana.

Autores como Brackmann (2017) defendem que o pensamento computacional estimula o raciocínio crítico, criativo e estratégico, incentivando a resolução colaborativa de problemas a partir de etapas claras, executáveis tanto por pessoas quanto por máquinas. Para Ribeiro, Foss e Cavaleiro (2017), o pensamento computacional se diferencia do raciocínio lógico tradicional ao envolver a formalização de regras e instruções para transformar entradas em saídas, mesmo quando essas não são sentenças formais ou verdades absolutas, o que amplia suas possibilidades de aplicação em diversas áreas.

Nesse contexto, é fundamental considerar a integração do pensamento computacional ao currículo escolar de forma interdisciplinar. Valente (2016) argumenta que ensinar apenas o uso de “softwares de escritório” não desenvolve as competências digitais necessárias à cidadania na cultura digital. Por isso, propõe-se o uso do pensamento computacional para promover o entendimento das tecnologias digitais, sua aplicação em diferentes contextos e seu impacto na sociedade, favorecendo o desenvolvimento do pensamento lógico, crítico e criativo dos estudantes.



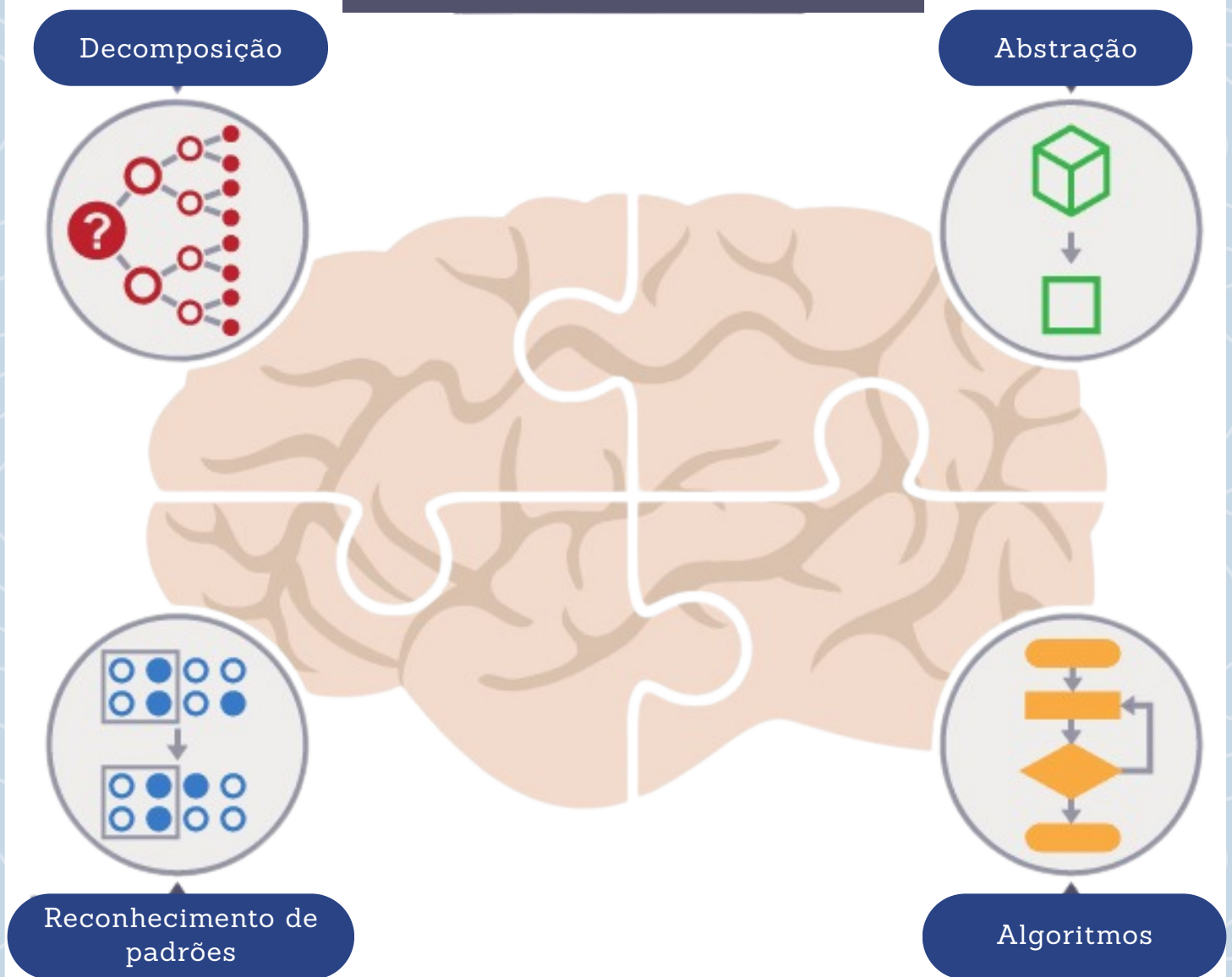
No entanto, essa integração ainda enfrenta desafios, sobretudo na formação inicial de professores. Raabe, Couto e Blikstein (2020) alertam para a necessidade de rever a forma como a computação é ensinada nos cursos de licenciatura, geralmente limitada à abordagem de ferramentas digitais. Estudos como os de Resnick et al., (2009) demonstram que a aplicação do pensamento computacional em contextos construcionistas, que valorizam a aprendizagem ativa, colaborativa e baseada em projetos, tem potencial para desenvolver habilidades cognitivas superiores. Assim, repensar práticas pedagógicas para incluir o pensamento computacional de maneira crítica e criativa pode contribuir significativamente para a formação integral dos estudantes.

A reutilização e a recombinação são ferramentas poderosas que podem melhorar significativamente a eficiência e a qualidade do desenvolvimento do pensamento computacional. Essas técnicas também têm o potencial de reduzir o “valor de troca”, acelerar o lançamento de produtos e fomentar a inovação. A aplicação de reutilização e a recombinação no desenvolvimento do pensamento computacional não apenas acelera o aprendizado, mas também prepara os alunos para enfrentar desafios complexos com uma perspectiva criativa e adaptável, habilidades cruciais em um ambiente tecnológico em constante evolução (Coletto; Braga, 2022).

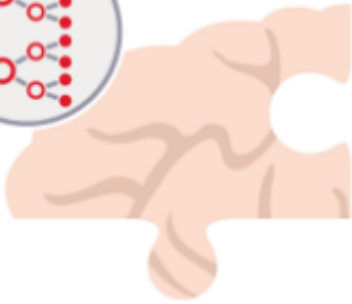
Durante a sequência didática, os quatro pilares do pensamento computacional (segundo a abordagem de Wing, 2006) são trabalhados de forma transversal:



Eixos do Pensamento Computacional

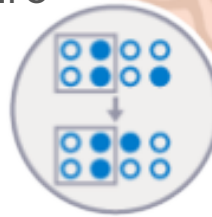


Fonte: BBC Learning (2019).



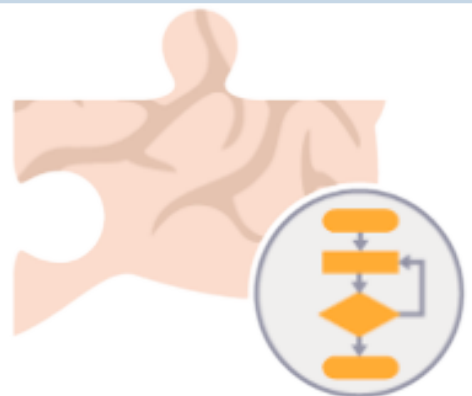
Decomposição: dividir o problema em partes menores durante a análise de apps e desenvolvimento dos próprios.

Reconhecimento de padrões: identificar elementos comuns entre aplicativos e funcionalidades.



Abstração: selecionar os dados essenciais para a resolução do problema proposto.

Algoritmos: criação de sequências lógicas de comandos usando blocos de programação.



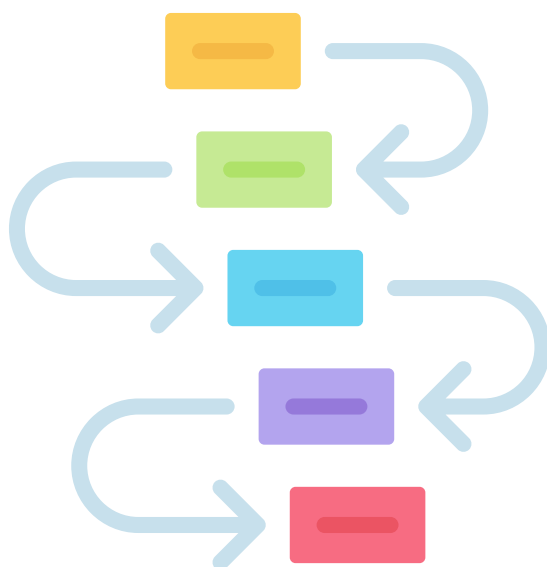
Sequência Didática Proposta

Professor/a, a seguir apresentamos a proposta da sequência didática que você poderá utilizar como ponto de partida para desenvolver com seus estudantes práticas significativas de pensamento computacional, por meio das estratégias de reutilização e recombinação de aplicativos no ambiente do *MIT App Inventor 2*.

Este material foi pensado como um guia flexível e adaptável, que pode (e deve!) ser ajustado conforme a realidade da sua escola, os interesses da turma e os recursos disponíveis. Ao longo da sequência, sugerimos atividades que promovem não só habilidades técnicas, mas também reflexão crítica sobre o papel da tecnologia na sociedade e na formação cidadã.

Para te apoiar nesse processo, organizamos os conteúdos com base na proposta de Zabala (1998), que considera quatro dimensões essenciais da aprendizagem: os conteúdos factuais, conceituais, procedimentais e atitudinais. Além disso, cada passo está alinhado aos pilares do pensamento computacional, como decomposição, abstração, reconhecimento de padrões e algoritmos.

Esperamos que esta proposta possa inspirar novas práticas e caminhos no seu fazer docente!



Aula 1

Diagnóstico inicial e introdução crítica ao Pensamento Computacional



Objetivos Específicos:

- Mapear os conhecimentos prévios dos estudantes sobre programação e tecnologia.
- Refletir criticamente sobre o papel da tecnologia na educação e na sociedade.
- Contextualizar o pensamento computacional (PC) como ferramenta de mediação e não apenas técnica.

Conteúdos:

- Pensamento computacional: conceitos iniciais
- Tecnologia e tecnocentrismo (Pinto, 2005)
- Educação omnilateral (Saviani, 2003)

Estratégias Didáticas:

- Aplicação de questionário diagnóstico (Google Forms ou papel)
- Discussão orientada: 'Afim, para que serve programar?'
- Dinâmica em grupos: mapas conceituais sobre 'tecnologia' e 'educação'

Recursos e Ferramentas:

- Quadro branco, projetor, questionário impresso/digital, papel kraft e canetões

Avaliação:

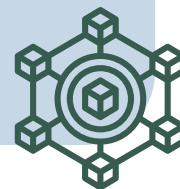
- Análise dos mapas conceituais e participação nas discussões reflexivas.
- Observação e registro no diário de campo sobre concepções tecnológicas dos estudantes.

Sugestão de adaptação:

Caso sua escola tenha acesso limitado à internet, o questionário pode ser aplicado em papel e as discussões realizadas oralmente, com anotações no quadro. Em turmas mais jovens ou com pouca familiaridade com debates, use vídeos curtos como disparadores e organize rodas de conversa mais direcionadas.

Aula 2

Estrutura e pilares do Pensamento Computacional



Objetivos Específicos:

- Compreender os pilares do PC: decomposição, abstração, padrões e algoritmos.
- Discutir criticamente a aplicação do PC na EPT e EMI.
- Relacionar o PC ao mundo do trabalho e à formação cidadã.

Conteúdos:

- Pilares do pensamento computacional (Wing, 2006)
- Educação Profissional e Tecnológica (Saviani, 2003; Frigotto, 2005)
- Dimensões conceitual, factual e atitudinal.

Conteúdos sob o olhar de Zabala (1998)

- Factual: Pilares do pensamento computacional (Wing, 2006);
- Conceitual: Relações entre PC e EPT (Saviani, Frigotto);
- Procedimental: Fichamento e síntese reflexiva;
- Atitudinal: Valorização da tecnologia com intencionalidade social.

Estratégias Didáticas:

- Aula expositiva-dialogada com infográficos
- Estudo de caso: 'Como resolver um problema da escola com tecnologia?'
- Debate estruturado: 'Pensamento computacional é só para programadores?'

Recursos e Ferramentas:

- Apresentação em slides, vídeos, papel para fichamentos reflexivos

Avaliação:

- Registro de hipóteses sobre PC antes e depois da aula.
- Produção de síntese individual com base no debate e estudo de caso.

Sugestão de adaptação:

Em contextos onde o tempo é reduzido, os pilares podem ser apresentados em forma de cartazes fixados pela sala para uma atividade de “ciranda reflexiva” com post-its. Também é possível substituir o estudo de caso por uma situação problema local, como o uso do celular na escola ou descarte de lixo eletrônico.

Aula 3

Introdução ao MIT App Inventor 2: Primeiros Passos



Objetivos Específicos:

- Familiarizar os estudantes com a interface e as funcionalidades básicas do *MIT App Inventor 2*.
- Compreender a lógica de blocos visuais na programação.
- Construir um aplicativo simples como ponto de partida.

Conteúdos:

- Ambiente gráfico do *MIT App Inventor 2*
- Componentes básicos de interface: botões, textos, imagens
- Blocos de controle e ações simples

Conteúdos sob o olhar de Zabala (1998)

- Factual: Interface do *App Inventor*; componentes básicos;
- Conceitual: Relação entre interface e lógica de programação;
- Procedimental: Criação do app "Hello Escola";
- Atitudinal: Entusiasmo, autonomia, espírito explorador.

Estratégias Didáticas:

- Aula demonstrativa com uso projetado do *Mit App Inventor 2*
- Atividade prática: criação de um app 'Hello Escola' com interação básica
- Exploração livre com orientação do professor

Recursos e Ferramentas:

- Laboratório de informática, projetor, contas Google para login no *Mit App Inventor 2*

Avaliação:

- Criação do primeiro *app* funcional.
- Observação da autonomia e curiosidade durante a experimentação.

Sugestão de adaptação:

Se não houver laboratório com internet disponível, existem várias iniciativas relacionadas a *computação desplugada*, por exemplo, o professor pode utilizar vídeos de demonstração do *App Inventor* e realizar uma simulação em papel dos blocos de programação (tipo "pseudocódigo visual"). Aplicativos simples podem ser projetados usando quadros magnéticos ou cartolinas com ícones móveis.

Aula 4

Reutilizando e remixando códigos: remixagem crítica



Objetivos Específicos:

- Introduzir o conceito de reutilização e remixagem de projetos.
- Estimular a adaptação de códigos existentes a novos contextos.
- Promover autoria e pensamento crítico no desenvolvimento de soluções tecnológicas.

Conteúdos:

- Reutilização e recombinação em ambientes de programação visual (Brennan & Resnick, 2012)
- Cultura digital colaborativa e ética da autoria
- Pensamento computacional e autoria crítica

Conteúdos sob o olhar de Zabala (1998)

- Factual: Blocos de controle, lógica condicional, variáveis;
- Conceitual: Remixagem (Brennan & Resnick); cultura digital colaborativa;
- Procedimental: Adaptação técnica e contextual de *apps*;
- Atitudinal: Respeito à autoria, criatividade, empatia social.

Estratégias Didáticas:

- Apresentação de projetos prontos para análise em grupo
- Tarefa em dupla: adaptar um *app* existente para resolver um novo problema (ex.: transporte escolar, coleta de lixo, biblioteca local)
- Orientação técnica e ética durante o processo de adaptação

Recursos e Ferramentas:

- Modelos de *apps* prontos, laboratório, *MIT App Inventor*, guia impresso de remixagem

Aula 4

Reutilizando e remixando códigos: remixagem crítica



Avaliação:

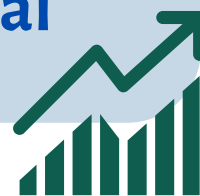
- Apresentação da versão adaptada com justificativas técnicas e éticas.
- Autoavaliação e coavaliação entre duplas com base em rubrica.

Sugestão de adaptação:

Em locais com poucos computadores, organize revezamentos por grupo, com atividades preparatórias no papel antes de acessar a plataforma. Se não for possível o uso do *App Inventor*, proponha que os alunos redesenhem os **apps** no papel, discutindo como seria o remix e quais problemas ele resolver.

Aula 5

Desenvolvimento do projeto final com MIT App Inventor



Objetivos Específicos:

- Consolidar os conhecimentos adquiridos sobre PC e programação por blocos.
- Desenvolver um aplicativo funcional e socialmente útil.
- Fomentar o protagonismo, a autonomia e o trabalho colaborativo.

Conteúdos:

- Integração de conceitos computacionais e procedimentais.
- Planejamento, execução e teste de aplicativos.
- Reflexão crítica sobre o papel social da tecnologia.

Conteúdos sob o olhar de Zabala (1998)

- Factual: Integração de comandos e lógica;
- Conceitual: Tecnologia como instrumento social;
- Procedimental: Planejamento, desenvolvimento e testes;
- Atitudinal: Autonomia, trabalho colaborativo, responsabilidade.

Estratégias Didáticas:

- Organização de duplas para desenvolvimento livre de um *app*;
- Definição de problema real a ser resolvido (ex.: agenda de estudos, orientações de segurança na escola).
- Ciclos de desenvolvimento e testes com orientação do professor.

Recursos e Ferramentas:

- Laboratório, *App Inventor*, formulários de planejamento de projeto, painel para esboços.

Avaliação:

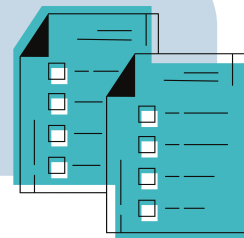
- Entrega do protótipo funcional.
- Preenchimento de relatório reflexivo e justificativa do projeto.
- Avaliação por rubrica com critérios técnicos, criativos e críticos.

Sugestão de adaptação:

Se o tempo disponível for reduzido, foque no planejamento do *app* e em uma maquete digital ou física do protótipo (por exemplo, no Canva, no papel ou com materiais recicláveis). Em escolas com dificuldade de acesso à plataforma, o produto final pode ser uma apresentação multimodal com desenhos, mapas de blocos e proposta de uso.

Aula 6

Socialização e Avaliação Final dos Projetos



Objetivos Específicos:

- Apresentar publicamente os aplicativos desenvolvidos.
- Refletir sobre os aprendizados, desafios e aplicações futuras do PC.
- Estimular a avaliação por pares e a autoavaliação crítica.

Conteúdos:

- Socialização de produtos educacionais.
- Apresentação oral e argumentação técnica.
- Cidadania digital e responsabilidade ética.

Conteúdos sob o olhar de Zabala (1998)

- Factual: Funcionalidades dos *apps*;
- Conceitual: Cidadania digital e responsabilidade ética;
- Procedimental: Apresentação oral e via *QR Code*;
- Atitudinal: Respeito, criticidade, argumentação.

Estratégias Didáticas:

- Organização de feira tecnológica na escola (exposição de *apps* via *QR Codes*).
- Sessão de apresentações com banca avaliadora (docente e estudantes).
- Debate final: o que aprendemos com o pensamento computacional?

Recursos e Ferramentas:

- Computadores, datashow, formulários de avaliação, cartazes.

Avaliação:

- Apresentação do projeto com argumentação.
- Avaliação por pares (formulário online).
- Autoavaliação escrita final.

Sugestão de adaptação:

Caso não seja possível realizar uma feira, organize uma “aula aberta” com outra turma ou professores convidados. Os alunos podem criar vídeos explicativos dos seus projetos e compartilhá-los via *QR Codes* em murais pela escola. Outra alternativa é a apresentação em círculos de conversa, em sala comum, com fichas de avaliação coletiva.

Considerações Finais



A proposta apresentada neste guia busca contribuir para a inserção do pensamento computacional no contexto escolar, reconhecendo-o não apenas como um conjunto de habilidades técnicas, mas como uma competência essencial para a formação integral dos estudantes. O desenvolvimento do raciocínio lógico, da criatividade, da capacidade de resolução de problemas e do trabalho colaborativo é cada vez mais valorizado nas práticas pedagógicas contemporâneas.

O Pensamento computacional, conforme reconhecido pela Base Nacional Comum Curricular (BNCC), é um componente importante na formação dos estudantes para o mundo contemporâneo, marcado pela presença constante das tecnologias digitais. Trata-se de uma forma de pensar e resolver problemas que envolve a decomposição de tarefas, o reconhecimento de padrões, a abstração e a criação de soluções algorítmicas, habilidades que se articulam com diferentes áreas do conhecimento. Ao integrar essa abordagem ao currículo escolar, a BNCC reafirma a importância de promover uma educação que prepare os estudantes para compreender e interagir com as tecnologias de maneira crítica, ética e significativa, contribuindo para sua autonomia intelectual e participação ativa na sociedade.

Ao longo da sequência didática, buscamos contemplar os diferentes tipos de conteúdos propostos por Zabala (1998):

- Os conteúdos factuais, relacionados às nomenclaturas e conhecimentos específicos sobre programação e tecnologias educacionais;
- Os conteúdos conceituais, que tratam dos fundamentos teóricos e das reflexões críticas sobre tecnologia e sociedade;
- Os conteúdos procedimentais, voltados ao uso prático do *MIT App Inventor 2*, à criação de aplicativos e à lógica de programação em blocos;
- E os conteúdos atitudinais, que dizem respeito às atitudes, valores e disposições que favorecem uma aprendizagem cooperativa, autônoma e socialmente engajada.

Dessa forma, espera-se que esta sequência didática sirva como apoio concreto ao trabalho docente, oferecendo não apenas um roteiro de atividades, mas também possibilidades de reflexão, adaptação e inovação na prática pedagógica. Mais do que replicar modelos prontos, a proposta aqui apresentada convida o professor a ser autor de seus próprios caminhos metodológicos, articulando teoria e prática, tecnologia e educação crítica, sempre com foco na formação humana e cidadã dos estudantes.



Referências

- BBC LEARNING, B. What is computational thinking?, 2015. Disponível em: <<http://www.bbc.co.uk/education/guides/zp92mp3/revision>>. Acesso em: 2 de jun 2025.
- BLIKSTEIN, P. O Pensamento Computacional e a reinvenção do computador na educação. 2008. Disponível em: http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html.
- BRACKMANN, C. Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica. Tese (Doutorado) – Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS, 2017. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/172208/001054290.pdf?sequence=1&isAllowed=y>. Acesso em: 20 dez. De 2023.
- BRASIL. Base Nacional Comum Curricular (BNCC). Educação é a Base. Brasília, MEC/CONSED/UNDIME, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf . Acesso em: 19 Jul. 2022
- BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada, 2012, p. 1–25.
- COLETTI, C. J.; BRAGA, J. C.. Remix de Recursos Educacionais Abertos–Revisão Sistemática de Literatura. Anais do XXXIII Simpósio Brasileiro de Informática na Educação, p. 149–161, 2022.

HAREL, I.; PAPERT, S. (Eds.). Construcionismo. Ablex Publishing, 1991.

GRAMSCI, Antonio. Os intelectuais e a organização da cultura. Rio de Janeiro: Civilização Brasileira, 1968.

LITTS, B. K.; LEWIS W. E.; MORTENSEN C. K. (2019): Engaging youth in computational thinking practices through designing place-based mobile games about local issues, Interactive Learning Environments, DOI: 10.1080/10494820.2019.1674883

LUCAS, M. L.; MOITA, F. M. G. da S. C.; VIANA, L. H. O pensamento computacional no novo ensino médio: Uma análise das obras didáticas da área de matemática e suas tecnologias. Educ. Matem. Pesq., São Paulo, v.25, n.3, p. 049-078, 2023. Disponível em: <https://doi.org/10.23925/1983-3156.2023v25i3p049-078>

PAPERT, S. Mindstorms: children, computers, and powerful ideas. New York: Basic Books, 1980.

PINTO, A. V. O conceito de tecnologia. 2º ed. Rio de Janeiro: Contraponto, 2005.

RAABE, A; COUTO, N. E. R; BLIKSTEIN, P; Diferentes abordagens para a computação na educação básica. In: RAABE, A., ZORZO, A. F; BLIKSTEIN. (Org) Computação na Educação Básica: Fundamentos e Experiências. Porto Alegre: Penso, 2020.

RIBEIRO, L.; FOSS, L.; CAVALHEIRO, S. A. da C. Entendendo o pensamento computacional. Nova Iorque: Cornell University Library, 2017. Disponível em: <https://arxiv.org/pdf/1707.00338.pdf> . Acesso em: 02 jan. de 2024.

SAVIANI, D. D. O choque teórico da politecnia. Trabalho, educação e saúde. Rio de Janeiro, v.1, n. 1, p. 131 - 152, 2003. Disponível em:
<https://www.scielo.br/j/tes/a/zLgxprrzCX5GYtgFpr7VbhG/?format=pdf&lang=pt>.

VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. Revista e-Curriculum, Pontifícia Universidade Católica de São Paulo, v. 14, n. 3, p. 864-897, 2016. Disponível em:
<https://revistas.pucsp.br/%20curriculum/article/view/29051/20655> .

WING, J. Computational Thinking. Communications of the ACM. New York, v. 49, nº 3, p. 33-35, 2006. Disponível em:
<https://doi-org.ez369.periodicos.capes.gov.br/10.1145/1118178.1118215>

WING, J. Computational Thinking Benefits Society. Social Issues in Computing, New York: Academic Press. 2014. Disponível em:
<http://socialissues.cs.toronto.edu/2014/01/computational-thinking> . Acesso em: 08 de dez. de 2023.

ZABALA, A. A prática educativa- como ensinar. Porto Alegre: Artmed, 1998.