

Nível educacional: Primeira infância, primária, secundária | Idade: 5-12
Autor: Michael Steiner e Hermann Morgenbesser, Future Learning Lab
Vienna



OBJECTIVOS DE APRENDIZAGEM

O Pensamento Computacional (PC) visa utilizar abordagens analíticas e algorítmicas para formular, analisar e resolver problemas. As práticas de PC incluem a concepção e desenvolvimento de artefactos computacionais, modelos, simulações; artefactos de fenómenos naturais e artificiais em colaboração e a implementação de técnicas computacionais para resolver problemas, tais como codificação, programação e robótica.



NARRATIVA

O pensamento só pode ser melhor desenvolvido se novos problemas estiverem constantemente a ser resolvidos. O pensamento computacional visa o desenvolvimento das seguintes competências:

- Familiaridade em lidar com a complexidade;
- Resistência em lidar com problemas difíceis;
- Tolerância à ambiguidade;
- Capacidade de lidar com questões abertas;
- Capacidade de comunicar e de resolver problemas em conjunto.

Estes são valores educativos que se estão a tornar cada vez mais importantes.

Para desenvolver o PC, os estudantes podem ser confrontados com os problemas que podem ser abertos, ambíguos, complexos e tão difíceis que só podem ser resolvidos em equipa e através de uma boa comunicação. Deve também salientar-se que o CT pode ser aprendido e ensinado em todas as idades, desde a primeira infância até aos exames de fim de estudos e, claro, para além destes; numa variedade de disciplinas: desde a ciência e matemática até à escrita e literatura.

Um exemplo é a construção de um pedómetro feito à mão. O professor designa-se a pensar num problema que um aluno quer resolver. É importante que os estudantes decidam com base nos seus próprios interesses. O aluno escolhe algo que o intriga e leva a ideia ao professor: A irmã do aluno afirma que ela anda 10.000 passos todos os dias. Ela trabalha como professora do jardim-de-infância e caminha diariamente de casa para o trabalho. Ela pensa que isso a mantém em forma. A estudante quer compreender o que significa caminhar 10.000 passos. Não é possível contar todos os seus passos num só dia. Além disso, ela provavelmente não anda o tempo todo, mas senta-se por vezes no seu trabalho. Isto é discutido com o professor e outros estudantes. Juntos, eles decidem construir um pedómetro que contaria os passos que a irmã do estudante anda. A tarefa é formulada como para construir um pedómetro que se prende ao pulso ou ao tornozelo e conta os passos à medida que se caminha. Com cada passo, um pulso é contado e depois mostrado no visor. Além disso, é necessário incluir a possibilidade de reiniciar o pedómetro (reset).



ABORDAGEM AO ENSINO & APRENDIZAGEM

- O ensino orientado para projetos e o trabalho de equipa são métodos adequados para lidar com problemas mais extensos de pensamento computacional.
- A Informática Unplugged envolve a resolução de problemas para atingir um objectivo, lidando com conceitos fundamentais das Ciências da Computação.
- Simulações computacionais para explorar fenómenos.
- Modelos de computador que podem ser testados, depurados e refinados.
- Um jogo de computador ou um projecto de construção de aplicação.

AVALIAÇÃO: "um sistema de avaliações". Por exemplo, o quadro de Análise de Padrões de Pensamento Computacional permite visualizar quais das nove competências específicas que os estudantes dominaram na concepção de jogos. O Dr. Scratch identifica sete dimensões de competência nesta área. Analisar os artefactos, cenários de resolução de problemas.



PAPÉIS

PROFESSORES: Os professores devem aprender a gerir uma sala de aula em que o computador serve tanto como meio primário para suporte ao desempenho como também como auxílio pedagógico ocasional. As atividades podem ser organizadas nas diferentes zonas de aprendizagem da sala de aula. Além disso, para apoiar a colaboração em grupos, é necessário criar 'andaimes' de suporte.

APRENDENTES: Os estudantes trabalham em colaboração (como equipa no planeamento e desenvolvimento de soluções), e assumem papéis (programador, analista, construtor, etc.) à medida que o grupo procura regular o seu trabalho.

OUTROS: Especialistas externos podem ser convidados através de ferramentas da Conferência para apoiar as actividades.



AMBIENTE DE APRENDIZAGEM

A interação entre o professor e os alunos é importante ao longo da experiência educacional, nesta o professor dá orientações, feedback e guias ao longo do processo. Os estudantes trabalham em equipa, e trocam ativamente opiniões sobre as tarefas do projeto, havendo divisão de papéis. Os estudantes desenvolvem o seu projeto e procuram soluções com o apoio do professor. As equipas criam soluções, partilhando diferentes tarefas na criação de um produto. Por exemplo, uma equipa programa, a outra trabalha com o micro:bit, e outros materiais. Em seguida, os alunos partilham o seu progresso com o professor, e o professor dá mais instruções sobre os próximos passos e dá possíveis pistas, discute também possíveis dificuldades ou erros - mais uma vez a interação acontece. Finalmente, os alunos apresentam o seu trabalho e refletem sobre a forma como este se desenvolveu e sobre o trabalho de equipa.



ATIVIDADES DE APRENDIZAGEM

As atividades de aprendizagem de PC são construídas em torno dos conceitos-chave tais como abstracção, pensamento algorítmico, automação, decomposição, depuração, e generalização. A codificação e programação é um componente da PC, uma vez que torna os conceitos de PC concretos e pode tornar-se uma ferramenta de aprendizagem. No entanto, ainda mais importante é o que as precede: o processo de análise e decomposição do problema.

As atividades chave para a história do Pedómetro seriam:

- O professor dá instruções em torno da concepção de pseudo-protótipos de código/UML, ou do desenvolvimento de uma solução.
- Desenvolvimento de projetos de soluções em equipas.
- Apresentação geral dos programas em execução ou peças de trabalho.
- Reflexão sobre o processo e realização das alterações necessárias.
- Há também espaço para melhorias. A última fase poderia ser em torno da reelaboração ou da melhoria da solução.



POSSÍVEIS DESAFIOS

Quando integrado no ensino obrigatório, há uma questão em aberto sobre que tipo de avaliação poderia suscitar a resolução de problemas por parte dos estudantes e competências de PC em contextos autênticos.

Há também a necessidade de desenvolver determinados comportamentos entre os estudantes - para trabalhar com outros e lidar com a frustração.

É necessário escolher atividades adequadas à idade, mas também é necessário responder aos interesses dos estudantes e não desmotivar as atividades das raparigas, que também devem ser propostas ao seu gosto.

É importante que as atividades de aprendizagem de PC exijam capacidade de ensino na concepção e avaliação de experiências em sala de aula se concentrando em conceitos de PC.



BIBLIOGRAFIA DE APOIO

- Wing, J. (2006): Computational thinking. Communications of the ACM, 49(3), 33-35.
- Computational Thinking Task Force: <https://csta.acm.org/Curriculum/sub/CompThinking.html>
- Computer Science Unplugged: <https://csunplugged.org/en/>
- K-12 Framework: <https://k12cs.org/wp-content/uploads/2016/09/Computer-Science-Framework.pdf>
- Coding with microbits: <https://padlet.com/eis/dlplwien>



RECURSOS

- Linguagem de programação, e.g, Python, Scratch, SNAP micro:bit
- agulha e linha de tecidos antigos
- fita de velcro



VÍDEO DO CENÁRIO DE APRENDIZAGEM

<https://www.youtube.com/watch?v=Z7xg1yZGeW0>

