

Entre Bugs e Emoções: *Como a Confusão pode Beneficiar a Aprendizagem de Programação*



Tiago Roberto Kautzmann

Este livro também está disponível para leitura no Kindle.

Encontre-o na Amazon (*amazon.com.br*) pesquisando pelo título.

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Kautzmann, Tiago Roberto

Entre bugs e emoções [livro eletrônico] : como a confusão pode beneficiar a aprendizagem de programação / Tiago Roberto Kautzmann. -- 1. ed. -- União da Vitória, PR : Ed. do Autor, 2025. -- (Série aprendizagem de programação)
PDF

Bibliografia.

ISBN 978-65-01-37136-8

1. Aprendizagem - Metodologia 2. Ciência da computação - Estudo e ensino 3. Programação (Computadores) - Estudo e ensino I. Título. II. Série.

25-258028

CDD-005.107

Índices para catálogo sistemático:

1. Programação : Informática : Estudo e ensino
005.107

Eliane de Freitas Leite - Bibliotecária - CRB 8/8415

Com o objetivo de aprimorar a clareza e a qualidade dos textos apresentados neste livro, utilizamos o ChatGPT como ferramenta auxiliar na revisão e no refinamento dos textos.

Sobre o autor

Tiago Roberto Kautzmann



Professor do Instituto Federal do Paraná (IFPR). Doutor em Computação Aplicada pela Unisinos - Universidade do Vale do Rio dos Sinos (2018 - 2022), com período sanduíche realizado em Marselha (França), junto à Aix-Marseille Université (2019). Mestre em Computação Aplicada pela Unisinos (2013 - 2015). Tecnólogo em Sistemas para Internet pela Universidade Feevale (2009 - 2011). Formação pedagógica pelo Programa Especial de Formação Pedagógica da Universidade Feevale (2012 - 2013). Realiza pesquisa científica em Inteligência Artificial aplicada à Educação. Professor há mais de 20 anos, ministra disciplinas sobre algoritmos, ciência de dados, inteligência artificial, estruturas de dados, programação orientada a objetos, desenvolvimento de software, banco de dados e lógica.

Contato: tkautzmann@gmail.com

Prefácio

O processo de aprendizagem é permeado por desafios cognitivos e emocionais. A **confusão** é uma das emoções que emergem durante esse percurso, ocupando um papel central, especialmente em contextos de aprendizagem complexa, como o de programação de computadores. Este livro explora a confusão como um fenômeno afetivo presente na interação de alunos com **ambientes de aprendizagem de programação**, trazendo discussões sobre sua relação com a aprendizagem e apresentando estratégias para que **tanto estudantes quanto professores possam utilizar essa emoção inevitável de maneira produtiva**.

O livro inicia com discussões sobre estados afetivos e emoções, trazendo à tona a importância do *appraisal* emocional na aprendizagem. A confusão é abordada sob a perspectiva de teorias de emoções relacionadas ao conhecimento, evidenciando como esse estado emocional pode levar tanto a desafios quanto a oportunidades no processo de aquisição de conhecimento. Em seguida, são analisados os impactos da confusão na aprendizagem em geral e, mais especificamente, na aprendizagem de programação de computadores.

A programação de computadores é um domínio que exige a integração de múltiplos conhecimentos, habilidades analíticas e

estratégias de resolução de problemas. Naturalmente, esse contexto gera momentos de incerteza e desequilíbrio cognitivo, muitas vezes manifestados como confusão. No entanto, quando bem gerenciada, a confusão pode atuar como um catalisador para a aprendizagem, impulsionando o estudante a desenvolver um raciocínio mais profundo e aprimorar suas habilidades metacognitivas e autorregulatórias.

Por fim, este livro apresenta estratégias para que os alunos possam transformar a confusão em um mecanismo de aprendizagem, ao invés de um obstáculo. Além disso, são discutidas práticas que professores poderiam adotar em suas aulas de programação para apoiar seus alunos nesse processo, auxiliando-os a enfrentar e a superar seus desafios cognitivos. Ao compreender a confusão não como um sinal de fracasso, mas como uma oportunidade de crescimento, educadores e estudantes podem ressignificar esse estado emocional e utilizá-lo como um elemento fundamental para a construção do conhecimento.

O Autor.

Conteúdo

1. Introdução.....	7
2. Estados afetivos e emoções.....	14
3. Appraisal de emoções.....	21
4. Dinâmica das emoções na aprendizagem.....	25
5. O que é a confusão.....	30
6. Como a confusão influencia a aprendizagem?.....	38
7. Como a confusão influencia a aprendizagem de programação de computadores?.....	41
8. Estratégias para o ALUNO se beneficiar da confusão durante a aprendizagem de programação.....	45
9. Estratégias para o PROFESSOR ajudar seus alunos a se beneficiar da confusão na aprendizagem de programação.....	51
10. Considerações finais.....	61
Referências.....	64

1. Introdução

As emoções desempenham um papel fundamental na aprendizagem, influenciando processos cognitivos como a memorização e a tomada de decisão, o que pode tanto favorecer quanto prejudicar a aquisição de conhecimentos e habilidades (PEKRUN, 2011). Quando um aluno progride bem em direção aos objetivos de uma tarefa de aprendizagem, emoções como engajamento e orgulho podem contribuir positivamente para o aprendizado (FREDRICKSON, 1998). Por outro lado, se o aluno encontrar dificuldades para avançar em uma tarefa e ficar estagnado, podem surgir emoções como a confusão, a frustração e o tédio, que tendem a prejudicar o estudante, especialmente se essas emoções persistirem por um longo período (D'MELLO; GRAESSER, 2014b).

Nas últimas décadas, computadores têm sido cada vez mais utilizados para promover a aprendizagem e as emoções também têm sido consideradas neste contexto. Pesquisadores das áreas de psicologia educacional e computação afetiva têm se interessado nos aspectos emocionais existentes na interação de usuários com computadores. Na área de pesquisa em Computação Afetiva, um dos principais objetivos é desenvolver computadores que detectam e respondem às emoções dos seus usuários (PICARD, 2010). Diversos estudos apresentaram

evidências sobre a frequência com que as emoções se manifestam na interação de usuários com computadores. Já foi encontrado que as emoções engajamento, tédio, confusão e frustração são mais frequentes do que as emoções básicas (felicidade, tristeza, medo, surpresa, raiva e nojo) quando usuários interagem com ambientes computacionais (D'MELLO; CALVO, 2013). No contexto de aprendizagem com computadores, uma meta-análise envolvendo 24 estudos mostrou que as seguintes emoções são mais frequentes: engajamento/flow, tédio, confusão, curiosidade, felicidade e frustração (D'MELLO, 2013).

Não há consenso entre os pesquisadores de que fenômenos afetivos frequentes na aprendizagem, como a confusão, a frustração, o tédio e o engajamento sejam emoções. Isso ocorre porque estes estados afetivos compartilham algumas, mas não todas as características comumente atribuídas às emoções básicas (D'MELLO; CALVO, 2013). No entanto, estes estados afetivos são comumente referenciados como emoções. Alguns autores as chamam de emoções não básicas (D'MELLO; CALVO, 2013; D'MELLO; KORY, 2012) ou emoções centradas na aprendizagem (AZEVEDO; TAUB; MUDRICK, 2019). Para simplificar, este livro usa o termo emoção para designar estes estados afetivos relacionados com a aprendizagem.

Este livro aborda a emoção **confusão**, no contexto de tarefas de aprendizagem de programação de computadores, durante a interação de alunos com um ambiente computacional, como quando os alunos

interagem com um software editor de programação (IDE) para desenvolver uma tarefa de aprendizagem. Estudar a confusão de alunos na aprendizagem de programação de computadores é interessante, pois é uma emoção propensa de ocorrer em ambientes de aprendizagem complexa, como o de programação de computadores (LEHMAN et al., 2013). Além de precisar entender o problema subjacente à tarefa de programação, o estudante precisa mobilizar esforços cognitivos para identificar e relacionar conhecimentos e estratégias de resolução de problemas e tomar decisões (CHI; OHLSSON, 2005; LEHMAN et al., 2013). Juntamente com engajamento, frustração, tédio e curiosidade, a confusão é uma das emoções mais frequentes durante a aprendizagem de programação (BOSCH; D'MELLO, 2017; BOSCH; D'MELLO; MILLS, 2013; COTO et al., 2021).

É possível tentar explicar a frequência com que a confusão é observada em ambientes de aprendizagem, através de teorias de emoções. A confusão pode ser abordada como uma emoção relacionada ao conhecimento, ou seja, uma emoção associada a metas de aprendizagem (SILVIA, 2010). Teorias de emoções que associam a confusão com o conhecimento, a consideram um fenômeno afetivo multicomponencial que inclui um componente cognitivo de *appraisal*. O capítulo 2 descreve esta abordagem multicomponencial das emoções.

O *appraisal* é um processo interno de um indivíduo, que detecta e avalia a significância do ambiente (satisfação ou não satisfação de

necessidades individuais, valores, metas e crenças) para o bem-estar pessoal (FRIJDA, 1988). Neste contexto teórico, a confusão é um estado afetivo que se supõe ocorrer quando o indivíduo avalia que a informação que chega não corresponde ao seu conhecimento anterior, ou quando a informação que chega não pode ser integrada ao modelo mental existente. Quando o aluno está confuso, ele entra em um estado de desequilíbrio cognitivo, resultante de contradições, conflitos ou informações erradas apresentadas a ele (D'MELLO; GRAESSER, 2014b). Quando este desequilíbrio ocorre, além de entrar em confusão, o indivíduo poderá experimentar um conjunto de outras emoções com efeitos negativos para a aprendizagem, como a frustração e o tédio, que o levam ao desengajamento (D'MELLO et al., 2014; SILVIA, 2010). Quando o equilíbrio é recuperado, mais emoções com efeitos positivos para a aprendizagem emergem, como o engajamento/flow (GRAESSER; D'MELLO, 2011).

Quando resolvida, a confusão é benéfica para o estudante e pode levar a aprendizagem a níveis mais profundos. Isso ocorre porque o desequilíbrio cognitivo causado pela confusão pode levar o aluno a mobilizar esforços cognitivos, levando a uma aprendizagem mais profunda (D'MELLO; GRAESSER, 2014b). Mas quando o aluno não se engaja na resolução da confusão, os resultados para a aprendizagem podem ser negativos (D'MELLO; GRAESSER, 2012a). Outro aspecto que pode fazer a confusão levar o aluno a resultados de aprendizagem

negativos é quando ele possui defasagem de habilidades metacognitivas e autorregulatórias. Ao apresentar estas defasagens, o estudante pode não conseguir se engajar para resolver os impasses enfrentados em tarefas atribuídas em ambientes mais complexos de aprendizagem (LEHMAN; D'MELLO; GRAESSER, 2012), como o de programação de computadores.

O que vou aprender neste livro?

Este livro explora os temas introduzidos até aqui, abordando inicialmente o conceito de estados afetivos e emoções, seguido de discussões sobre o *appraisal* das emoções e sua importância no processo de aprendizagem. Em seguida, discute a confusão, suas características e seu impacto tanto na aprendizagem em geral quanto na aprendizagem de programação. Por fim, apresenta estratégias para que os alunos utilizem a confusão de maneira produtiva, além de sugestões para que os professores auxiliem seus alunos a transformar esse estado emocional em uma oportunidade de crescimento.

A quem se destina este livro?

Este livro é voltado principalmente para **estudantes e professores de programação**, oferecendo uma abordagem sobre o papel das

emoções no aprendizado dessa disciplina. O livro procura reforçar a ideia de que a confusão é um estado afetivo que pode ser tanto um desafio quanto uma oportunidade. **Para os estudantes**, o livro apresenta estratégias práticas para lidar com momentos de incerteza e dificuldade, ajudando-os a transformar a confusão em um processo ativo de aprendizado e desenvolvimento de habilidades em programação. **Para os professores**, o livro fornece insights baseados em pesquisas e na experiência do autor sobre como ajudar os alunos a se beneficiarem da confusão, permitindo que seus alunos naveguem melhor pelos desafios emocionais e cognitivos envolvidos nas tarefas de programação.

Esta leitura é ideal para quem busca compreender como as emoções influenciam o aprendizado e deseja adotar abordagens mais eficazes para melhorar tanto o ensino quanto a aprendizagem da programação.

Como estudar com este livro?

Para aproveitar ao máximo este livro, recomenda-se uma abordagem ativa e reflexiva durante a leitura. Os capítulos foram organizados de maneira progressiva, conectando conceitos teóricos sobre emoções e aprendizagem com estratégias práticas para lidar com a confusão na programação. **Para os estudantes**, uma boa estratégia é relacionar os conteúdos com suas próprias experiências de

aprendizagem, identificando momentos em que sentiram confusão e analisando quais das estratégias apresentadas poderiam ser úteis. **Para os professores**, a leitura pode ser complementada com a aplicação das estratégias em sala de aula, observando como seus alunos reagem e adaptando as abordagens conforme necessário. Além disso, é recomendável realizar anotações, testar as sugestões propostas e discutir os conceitos com colegas ou professores, pois o aprendizado se torna mais significativo quando há troca de ideias e aplicação prática dos conhecimentos adquiridos.

2. Estados afetivos e emoções

Este capítulo (2) e os próximos (3, 4, 5, 6 e 7) sempre começam com discussões sob uma perspectiva científica. Se você não estiver familiarizado com esse formato ou sentir dificuldade com os conceitos apresentados, ao final de cada um desses capítulos há uma seção que resume o capítulo em uma linguagem mais acessível, com exemplos práticos para facilitar a compreensão. Caso prefira, você pode ir diretamente para essa seção sem comprometer o entendimento dos conceitos principais que o autor deseja transmitir.

Antes de traçar qualquer definição sobre emoção, é necessário compreender o conceito de afeto. Na perspectiva psicológica, o afeto é um sentimento subjetivo que afeta a cognição do indivíduo (MELLO et al., 2014). A emoção é um tipo de afeto, referenciado na literatura como um estado afetivo, ou fenômeno afetivo. Estado afetivo tem sido utilizado na literatura como um termo mais abrangente para designar fenômenos afetivos como emoção, humor, atitudes, disposições afetivas e postura interpessoal, os quais podem ser diferenciados em função da duração, intensidade, impacto comportamental, eventos-gatilho e respostas (SCHERER, 2005).

A confusão vem sendo tema de estudos e reflexão de filósofos e cientistas há milhares de anos (JAQUES; VICARI, 2005; SCHERER, 1987). Acredita-se que existam expressões universais, inatas e interculturais para seis emoções, chamadas de emoções básicas: tristeza, alegria, raiva, surpresa, medo e nojo (EKMAN; FRIESEN, 1987).

Não há consenso entre os pesquisadores sobre um conceito para emoção, pois trata-se de um construto complexo, que já foi estudado sob diferentes perspectivas: expressão, fisiológica, cognitiva e social. A neurociência também tem contribuído recentemente para a compreensão sobre como funciona o circuito neural relacionado às experiências emocionais (CALVO; D'MELLO, 2010). Um exemplo de abordagem de emoção é a das emoções básicas de Ekman (1987), a partir da perspectiva das expressões. Nesta perspectiva, surgiu o FACS (Facial Action Coding System), um modelo que caracteriza as expressões faciais causadas pelas contrações musculares do rosto, chamadas de AUs (Action Units). O FACS tem sido usado em trabalhos de detecção de emoções básicas em fotos e vídeos (OLIVEIRA; JAQUES, 2008).

O presente livro utiliza, como teoria subjacente sobre emoções, uma perspectiva para uma teoria de emoções desenvolvida pelo pesquisador suíço Klaus Scherer, no qual diferentes perspectivas sobre as emoções são integradas em um modelo multicomponencial de emoções (MOORS et al., 2013; SCHERER, 2005). A teoria de Scherer tem sido amplamente aceita pelos pesquisadores em psicologia cognitiva

e computação afetiva. No seu modelo de componentes de emoções, Scherer apresenta uma definição para emoção como sendo um episódio de mudanças sincronizadas e inter relacionadas nos estados de todos ou quase todos os cinco subsistemas do organismo em resposta à avaliação de um evento de estímulo interno ou externo de grande importância para o organismo (SCHERER, 2005). O Quadro 1 apresenta os relacionamentos entre os componentes de emoção, suas funções emocionais e os subsistemas orgânicos. Neste modelo, o sentimento não pode ser confundido com a emoção, em si, pois ele é apenas um dos componentes de emoção, um componente de experiência emocional subjetiva (SCHERER, 2005).

Quadro 1 - Componentes de uma emoção

Componente de emoção	Função	Subsistema orgânico
Componente cognitivo (<i>appraisal</i>)	Avaliação de objetos e eventos	Processamento de informação (sistema nervoso central)
Componente neurofisiológico (sintomas corporais)	Regulação de sistema	Suporte (sistema nervoso central, sistema neuroendócrino, sistema nervoso autônomo)
Componente motivacional (tendências de ação)	Preparação e direção de ação	Executivo (sistema nervoso central)
Componente de expressão motora (expressão vocal e facial)	Comunicação da reação e intenção comportamental	Ação (sistema nervoso somático)
Componente de sentimento subjetivo (experiência emocional)	Monitoramento do estado interno e da interação do organismo com o ambiente	Monitor (sistema nervoso central)

Fonte: SCHERER, 2005

Nesta perspectiva multicomponencial, Scherer (2005) descreve algumas características de emoções que permitem discernir a emoção de outros estados afetivos.

Evento foco: A teoria de Scherer (2005) sugere que as emoções são geralmente provocadas por eventos de estímulo. O que o autor quer dizer é que algo (algum evento) acontece para o organismo que estimula ou desencadeia uma resposta após a significância do evento ter sido avaliada pelo indivíduo. Estes eventos podem ser externos ao organismo, como fenômenos naturais (trovoadas, ventanias, entre outros), o comportamento de outras pessoas, ou o comportamento da própria pessoa. Eventos internos também são provocadores de emoções, como mudanças fisiológicas ou memórias que surgem na mente do indivíduo.

Dirigido à *appraisal*: Um aspecto central da definição multicomponencial de Scherer (2005) é que o evento provocador da emoção e suas consequências deve ser relevante para as principais preocupações do organismo. A teoria sugere que diferentes emoções são produzidas por avaliações (*appraisals*) de eventos de estímulos internos ou externos. São estas avaliações sobre a relevância do evento de estímulo para o bem-estar pessoal que Scherer chama de *appraisals*. Discussões sobre o *appraisal* de emoções são ampliadas no capítulo 3.

Sincronização de resposta: Segundo Scherer (2005), se as emoções ocasionam respostas a eventos de estímulo, os padrões de resposta nos sistemas organísmicos devem corresponder à análise de

appraisal das implicações presumidas do evento. Dependendo do evento provocador da emoção, todos ou quase todos os subsistemas organísmicos devem contribuir para a preparação da resposta.

Rapidez de mudança: Os eventos de estímulo e, conseqüentemente, o *appraisal* sobre esses eventos, pode mudar rapidamente, considerando mudanças no ambiente ou, até mesmo, reavaliações (*reappraisals*) dos eventos de estímulo (SCHERER, 2005).

Impacto comportamental: As emoções têm forte efeito sobre o comportamento conseqüente dessas emoções, e frequentemente interrompem sequências de comportamento em andamento. Além disso, o componente de expressão motora do modelo de Scherer (2005) tem forte impacto na comunicação, podendo ter fortes conseqüências nas interações sociais.

Intensidade: Scherer (2005) sugere que a intensidade dos padrões de resposta das emoções e da experiência emocional são relativamente altas, e que esta é uma importante característica para distinguir emoções de outros estados afetivos, como o humor, por exemplo.

Duração: Uma discussão de Scherer (2005) sugere que a duração das emoções deve ser relativamente curta, em função de apresentarem uma forte resposta e sincronização dos subsistemas organísmicos que podem sobrecarregá-los. O autor utiliza como exemplo o estado afetivo de humor, que possui baixa intensidade e pouco impacto sobre o

comportamento, de forma que pode ser mantido por um período mais longo até apresentar efeitos adversos.

Scherer (2005) defende que essas características são importantes para discernir as emoções de outros estados afetivos como as atitudes, o humor, as disposições afetivas e as posturas interpessoais. Um exemplo é o humor, que é caracterizado por ser mais duradouro, podendo durar horas ou dias, mas possui baixa intensidade e pode emergir sem uma causa aparente. As emoções, por sua vez, são mais curtas, possuem geralmente intensidade mais alta e precisam de algum estímulo.

Resumo para facilitar o entendimento do capítulo

As emoções fazem parte do nosso dia a dia e influenciam diretamente a forma como aprendemos e tomamos decisões. Antes de entender o que são as emoções, é importante conhecer o conceito de afeto, que é qualquer sentimento que afeta nossos pensamentos e ações. Por exemplo, quando estamos motivados e animados, aprendemos com mais facilidade, mas se estamos frustrados ou entediados, podemos ter dificuldade em continuar estudando.

A confusão é uma emoção comum durante o aprendizado, especialmente quando nos deparamos com algo novo ou difícil. Imagine que você está tentando resolver um problema de matemática e, no início, nada faz sentido. Esse momento de confusão pode ser frustrante, mas

também pode ser um passo importante para a aprendizagem, porque nos força a pensar mais e buscar soluções.

O presente livro adota um modelo de emoções criado pelo pesquisador Klaus Scherer, que explica que as emoções são reações do nosso corpo e mente a algo importante. Por exemplo, ao receber uma notícia inesperada, podemos sentir surpresa e nosso corpo reage de diferentes formas, como através do aumento dos batimentos cardíacos ou através de mudanças na expressão facial. Scherer identifica algumas características das emoções, como o fato de serem desencadeadas por eventos específicos, por mudarem rapidamente e por influenciarem nosso comportamento.

Além disso, Scherer diferencia emoções de outros estados afetivos, como o humor. Enquanto a emoção pode surgir rapidamente e durar pouco tempo, como um momento de alegria ao receber um elogio, o humor pode se estender por horas ou dias, como quando estamos de mau humor sem um motivo aparente. Essas diferenças são importantes para entender como as emoções afetam nossa aprendizagem e como podemos usá-las a nosso favor.

3. *Appraisal* de emoções

Um dos componentes de emoção presente na teoria de componentes de Scherer, e que tem sido fortemente considerado nos estudos dos pesquisadores em psicologia cognitiva é o componente cognitivo das emoções, responsável pelo *appraisal*, nome dado a uma avaliação cognitiva de situações e eventos que precedem uma emoção (SCHERER, 1999). *Appraisal* é um processo que avalia a significância do ambiente para o bem-estar pessoal. Significância para o bem-estar pode ser compreendida como a satisfação ou a não satisfação de necessidades pessoais, metas, crenças ou tudo aquilo com que o indivíduo se importa (MOORS et al., 2013). Scherer descreve que o *appraisal* pode ser intrínseco ou extrínseco (SCHERER, 2005). O *appraisal* é intrínseco quando ele avalia a característica de um objeto ou pessoa independente das metas ou necessidades do indivíduo, mas baseado em preferências genéticas ou aprendidas. O *appraisal* extrínseco avalia um evento de estímulo em relação a quanto ele vai de encontro com as necessidades, desejos e metas pessoais.

Segundo Moors e Scherer (2013), o termo *appraisal* foi usado pela primeira vez nos estudos de Magda Arnold nos anos 1960 (ARNOLD, 1960), mas foi o trabalho de Lazarus (1966) que teve influência direta para o desenvolvimento de uma teoria de *appraisal*. Nas décadas

seguintes, trabalhos que deram continuidade ao desenvolvimento deste conceito passaram a ser referenciados como teorias de *appraisal* (BARRETT, 2006; ORTONY; CLORE; COLLINS, 1988; RUSSELL, 2003; SCHERER, 2009).

Diferentes emoções são produzidas por diferentes *appraisals* de emoções (SCHERER, 2005). Uma das teorias mais consideradas nos estudos de emoções envolvendo computadores é o modelo OCC (ORTONY; CLORE; COLLINS, 1988), que assume que as emoções surgem como resultado da forma como o indivíduo avalia as situações. O modelo OCC resulta em um modelo psicológico que explica a origem de 22 tipos de emoções.

Talvez esse “cientifiquês” não esteja lhe ajudando muito a entender o conceito de *appraisal*. Imagine um aluno iniciante em programação tentando resolver um exercício de lógica que envolve laços de repetição. Ele lê o enunciado e acredita que entendeu a tarefa, mas ao escrever seu código e executar, percebe que a saída apresentada na tela não é a esperada. Neste momento, ocorre um *appraisal*, ou seja, uma avaliação cognitiva da situação pelo aluno. O aluno percebe um conflito entre o que ele esperava e o que realmente aconteceu, gerando confusão. Ele revisa o código e percebe que usou um operador errado na condição do laço, mas não consegue entender por que isso altera o resultado. Se o aluno persistir e buscar mais informações, seja consultando a documentação, testando pequenas mudanças no código ou pedindo ajuda

ao professor, ele pode transformar essa confusão em um aprendizado mais profundo. No entanto, se ele não conseguir resolver o problema e permanecer confuso por muito tempo, essa emoção pode evoluir para frustração e até desmotivação.

Imagine outra situação em que um aluno está tentando implementar um algoritmo de ordenação em Python, seguindo um exemplo apresentado em aula pelo professor. Ele digita o código exatamente como no material, mas ao executar, o programa retorna um erro inesperado: *"TypeError: '<' not supported between instances of 'str' and 'int'"*. Nesse momento, ocorre um *appraisal*, ou seja, uma avaliação cognitiva do aluno sobre a situação. Ele percebe que algo está errado, mas não consegue imediatamente relacionar o erro à causa. Isso gera um desequilíbrio cognitivo e o coloca em um estado de confusão. Ele se pergunta por que o código da aula não funciona no seu caso.

Se o estudante for capaz de refletir e buscar a origem do problema, pode perceber que, ao testar o algoritmo, acidentalmente misturou números e strings na lista de entrada, o que gerou o erro. Essa confusão, se resolvida, o levará a um aprendizado mais profundo sobre tipagem de dados em Python e erros comuns em comparação de valores. Por outro lado, se ele não conseguir superar essa confusão e não buscar ajuda ou testar outras abordagens, pode acabar frustrado e desmotivado com a tarefa. Isso mostra como o *appraisal* da situação pode determinar se a confusão será produtiva ou prejudicial para a aprendizagem.

Resumo para facilitar o entendimento do capítulo

O *appraisal* é um processo mental que acontece quando avaliamos algo e isso nos faz sentir uma emoção. Pense em quando você recebe a nota de uma prova: se for alta, você pode se sentir feliz e aliviado; se for baixa, pode ficar frustrado ou desmotivado. Esse processo de avaliar a situação antes de sentir a emoção é o que chamamos de *appraisal*.

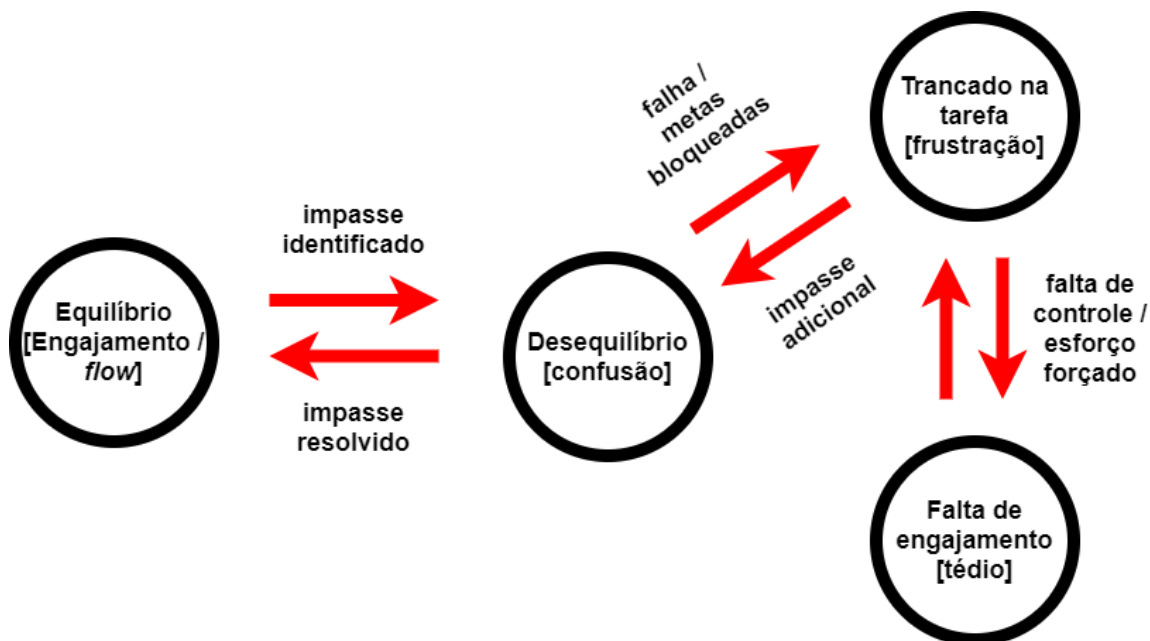
No aprendizado de programação, isso acontece o tempo todo, não é mesmo? Imagine que um aluno iniciante está tentando resolver um exercício de repetição em Python. Ele lê o enunciado e acha que entendeu, mas quando executa o código, percebe que o resultado não está certo. Nesse momento, seu cérebro faz um *appraisal*: ele percebe que algo deu errado e começa a se perguntar o motivo. Se ele insistir, testar variações no código ou procurar ajuda, pode aprender algo novo. Mas, se não conseguir resolver e ficar muito tempo sem entender, a confusão pode virar frustração.

O *appraisal* pode ser positivo ou negativo dependendo do que fazemos com a confusão. Se a usarmos para aprender, pode ser um trampolim para o conhecimento. Se ficarmos travados sem buscar solução, pode virar um bloqueio no aprendizado.

4. Dinâmica das emoções na aprendizagem

Os pesquisadores têm se preocupado em entender a dinâmica temporal das emoções na aprendizagem. O trabalho de D'Mello e Graesser (2012a) avaliou um modelo sobre as dinâmicas afetivas (ver Figura 1) em atividades de aprendizagem complexa.

Figura 1 - Dinâmica das emoções



Fonte: (D'MELLO; GRAESSER, 2012a)

Segundo o modelo, quando o aluno está em um estado de engajamento/flow e enfrenta impasses, como contradições,

incongruências e obstáculos para alcançar suas metas, ele poderá experimentar um desequilíbrio cognitivo que resultará em confusão. Quando o aluno consegue resolver esses impasses e o equilíbrio é restaurado, o aluno retorna a um estado de engajamento/flow. No entanto, se o aluno falha para restaurar o equilíbrio e fica bloqueado para alcançar suas metas, o estudante entra em um estado de frustração que, se não resolvida, pode levá-lo a um estado de tédio.

Sobre características da confusão na dinâmica das emoções, um estudo encontrou que alunos com pouco conhecimento em álgebra e com alto nível de neuroticismo não conseguiram lidar muito bem com a confusão, passando menos tempo neste estado afetivo e entrando mais rapidamente para os estados de frustração e tédio (REIS et al., 2018). Pessoas com alto neuroticismo costumam ser mais sensíveis ao estresse, preocupadas com eventos negativos e podem ter maior tendência à depressão e à ansiedade. Já indivíduos com baixo neuroticismo são emocionalmente mais estáveis, resilientes e menos propensos a se preocupar excessivamente.

O tempo com que o aluno permanece na confusão também pode estar relacionado com suas habilidades metacognitivas autorregulatórias. Habilidades metacognitivas autorregulatórias são capacidades que permitem ao indivíduo planejar, monitorar e avaliar seu próprio processo de aprendizagem e resolução de problemas. Elas fazem parte da metacognição, que é o conhecimento e o controle sobre os próprios

processos cognitivos. Quando alunos com boas habilidades autorregulatórias entram em confusão, eles podem demorar mais tempo para passarem para os estados de tédio e frustração, ou, até mesmo, não chegarem a estes estados. Isso porque estes alunos conseguem mobilizar mais estratégias para resolver os impasses e contradições encontradas em uma tarefa de aprendizagem (DI LEO; MUIS, 2020).

Resumo para facilitar o entendimento do capítulo

Os pesquisadores estudam como as emoções mudam ao longo do aprendizado e descobriram que há um ciclo emocional que pode afetar o desempenho dos alunos.

Imagine um aluno programando e se sentindo engajado (ou no estado de flow, uma concentração profunda), porque tudo está funcionando bem. No entanto, se ele encontra um erro inesperado no código, como um loop infinito ou um erro de sintaxe difícil de entender, ele pode entrar em um estado de confusão. Nesse momento ocorre um desequilíbrio cognitivo, pois ele percebe que algo não está certo, mas ainda não sabe como resolver. Se o aluno conseguir superar esse obstáculo - seja testando soluções diferentes, lendo a documentação ou pedindo ajuda -, ele resolve o problema e volta ao estado de engajamento, aprendendo algo novo no processo. Porém, se ele tentar várias vezes sem sucesso e não souber como seguir, pode ficar frustrado.

Se essa frustração continuar sem solução, ele pode perder o interesse e entrar em um estado de tédio, onde desiste da tarefa e para de tentar aprender.

Nem todos os alunos lidam com a confusão da mesma forma. Pesquisadores descobriram que aqueles que possuem menos conhecimento prévio (por exemplo, alunos que têm dificuldade com matemática ou lógica) podem ficar presos na confusão por menos tempo, mas acabam desistindo mais rápido, caindo direto na frustração ou tédio. Isso pode acontecer, por exemplo, quando um aluno encontra um erro de compilação que não entende e, em vez de investigar, simplesmente abandona a tarefa.

Por outro lado, alunos que possuem boas habilidades de autorregulação (ou seja, sabem como lidar com desafios e buscar soluções) conseguem ficar confusos por mais tempo sem se frustrar. Eles testam possibilidades, procuram recursos e fazem perguntas, aumentando suas chances de resolver o problema e voltar ao engajamento. Por exemplo, se um aluno está implementando uma função recursiva e não entende por que a recursão nunca para, em vez de desistir rapidamente, ele pode adicionar comandos do tipo *print()*, um método de *debugging*, para verificar como a função está se comportando, fazendo ajustes até entender o erro.

A confusão pode ser uma parte normal do aprendizado de programação, mas o que o aluno faz com ela determina se ele aprende

ou se desiste. Quem desenvolve estratégias para resolver problemas (testando código, pesquisando, pedindo ajuda) tem mais chances de transformar a confusão em aprendizado. Já aqueles que não sabem como lidar com a confusão correm o risco de cair na frustração e perder o interesse. Assim, **aprender a enfrentar desafios é tão importante quanto aprender a programar.**

5. O que é a confusão

No capítulo 2, vimos que não há consenso entre os pesquisadores sobre o que é uma emoção. A mesma dificuldade de conceitualização ocorre sobre a confusão, que também tem recebido diferentes conceitualizações (D'MELLO; GRAESSER, 2014b). A confusão já foi classificada como uma emoção genuína (ROZIN; COHEN, 2003), uma emoção relacionada ao conhecimento (SILVIA, 2010), uma emoção epistêmica (PEKRUN; STEPHENS, 2012) e, até mesmo, uma “não emoção” (HESS, 2003; KELTNER; SHIOTA, 2003).

O trabalho de D'Mello e Graesser (2014b) apresenta uma discussão interessante sobre o que é a confusão. Segundo os autores, a falta de uma clara definição de emoção, as múltiplas perspectivas sobre o construto de emoção e a escassez de pesquisa básica sobre a confusão, leva à falta de uma definição mais precisa sobre a confusão e se a confusão é realmente uma emoção. Com o objetivo de encontrar uma correspondência entre o que se sabe sobre a confusão e as características de uma emoção, os autores (D'MELLO; GRAESSER, 2014b) sugerem iniciar a discussão a partir de um estudo de Carrol Izard (2010), que buscou identificar as características que definem uma emoção. Este estudo coletou respostas de 37 pesquisadores de emoções para seis questões, entre elas, “O que é uma emoção?”. O resultado do estudo

descreve a emoção como um estado afetivo que envolve seis características: a) circuitos neurais parcialmente dedicados ao processamento emocional, b) ativação de sistemas de respostas em preparação para alguma ação, c) ter estados de sentimentos distintos, d) desempenhar um papel no comportamento expressivo, e) surgir a partir de resultados de processos de *appraisal*, e f) poder envolver interpretação cognitiva de sentimentos.

Segundo D'Mello e Graesser (2014b), a confusão compartilha pelo menos quatro (“a”, “c”, “d” e “e”) das seis principais características de emoção identificadas pelos pesquisadores. Em relação ao item “a”, já foram encontradas evidências de que a confusão está ligada a atividades neurais, como no trabalho descrito por Halgren et al. (2002), que encontrou um alto valor para N400 em atividades de eletroencefalograma, quando um indivíduo foi confrontado com uma sentença que apresentou uma anomalia semântica, como “Maria estava com fome, então ela foi até uma loja comprar algumas cadeiras”. Em relação a característica “c”, já foi descrito um estado de sentimento distinto relacionado a confusão (ROZIN; COHEN, 2003). Em relação a “d”, já foi encontrado um componente expressivo para a confusão, a sobrelha franzida (CRAIG et al., 2008a; GRAFSGAARD; BOYER; LESTER, 2011; MCDANIEL et al., 2007). Em relação ao item “e”, foram encontradas evidências de que a confusão surge a partir de um processo de *appraisal* cognitivo relacionado a uma falta de

correspondência entre a informação que chega ao indivíduo e o seu conhecimento anterior relativo à informação (D'MELLO et al., 2014; SILVIA, 2010).

Sobre as demais características de emoções apontadas por Izard (IZARD, 2010), não se sabe ainda muito sobre mudanças corporais associadas com a confusão (característica “b”), mas o trabalho apresentado Alzoubi et al. (2012) conseguiu algum sucesso em distinguir a confusão do estado neutro e outras emoções, a partir de dados fisiológicos. Assim, este livro qualifica a confusão como sendo uma emoção, acompanhando o mesmo entendimento de D'Mello e Graesser (2014b), que consideraram evidências científicas relacionadas a características vinculadas a emoções, como ser um estado afetivo que surge de interações neurais, que tem um componente expressivo, que tem um estado de sentimento distinto e que está relacionado a um componente cognitivo de *appraisal*.

O *appraisal* da confusão

Como vimos no capítulo 3, o *appraisal* é um processo interno de avaliação de um indivíduo sobre como um evento de estímulo se relaciona com as próprias metas, valores, conhecimentos e habilidades. O *appraisal* é o componente cognitivo do modelo multicomponencial de emoções de Scherer (1999). O trabalho de Silvia (2010) descreve

emoções relacionadas ao conhecimento como emoções causadas pelas crenças dos indivíduos sobre seus próprios pensamentos e conhecimentos e decorrem de objetivos associados ao aprendizado, como o conhecimento, o pensamento e a compreensão. Seu trabalho encontrou evidências positivas para a hipótese de que a confusão está associada com *appraisals* de novidade (familiar) e de baixa compreensibilidade, no contexto da informação que chega para o indivíduo. Neste sentido, a teoria de discrepância de Macdowell e Mandler (1989) descreve que os indivíduos estão constantemente assimilando novas informações em um modelo mental de conhecimento. Quando uma informação discrepante é identificada, como um conflito com o conhecimento prévio, ocorre uma excitação no sistema nervoso autônomo, e o indivíduo experimenta uma variedade de possíveis emoções. A emoção experimentada depende do contexto e do processo de *appraisal*.

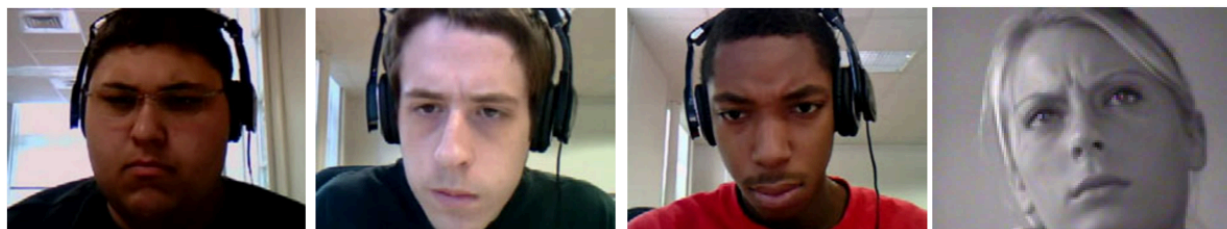
O trabalho de Silvia (2010) encontrou que a surpresa pode preceder a confusão quando um estímulo inesperado é avaliado no processo de *appraisal* como algo incompreensível. Com isso, os estudos que consideram o componente emocional de *appraisal* descrevem a confusão como uma emoção que ocorre quando há uma falta de correspondência entre a informação que chega e o conhecimento anterior, ou quando essa informação não pode ser integrada aos modelos

mentais existentes, iniciando um desequilíbrio cognitivo (D'MELLO; GRAESSER, 2014b).

As expressões de confusão

Segundo Silvia (2010), a primeira discussão sobre expressões faciais da confusão foi realizada por Darwin (1872), no contexto de obstáculos para alcançar metas cognitivas. Darwin descreve o seguinte: “Um homem pode ser absorvido pelo pensamento mais profundo, e sua sobrancelha permanecerá serena até encontrar algum obstáculo em sua linha de raciocínio, ou ser interrompido por algum distúrbio, e então uma carranca passa como uma sombra sobre sua sobrancelha.”. O trabalho de Craig et al. (2008) encontrou uma correlação da confusão com as *action units* (AUs) AU4 (sobrancelha abaixada) (Figura 2 a, b e c), AU7 (pálpebras esticadas), AU4 + AU7 (Figura 2 d) e AU12 (canto dos lábios puxados), do sistema FACS (*Facial Acting Coding System*). O trabalho de McDaniel et al. (2007) também encontrou uma correlação da confusão com AU4.

Figura 2 - Expressões faciais de confusão



Fonte: (GRAFSGAARD; BOYER; LESTER, 2011; MCDANIEL et al., 2007).

Em relação a outros modos de expressão, estudos já encontraram evidências de correlação da confusão com a postura do corpo (D'MELLO; DALE; GRAESSER, 2012; K. D'MELLO; GRAESSER, 2010), mas esta forma de expressão necessita de mais investigações. Em relação a dados fisiológicos, a classificação da confusão como uma emoção relacionada ao conhecimento pode trazer o questionamento se realmente existem características fisiológicas com boa correlação com a confusão. O estudo descrito em (ALZOUBI; D'MELLO; CALVO, 2012) tentou encontrar estas características, como na atividade elétrica do coração (eletrocardiograma), para diferenciar emoções como confusão, curiosidade e o estado neutro. Os resultados foram apenas moderados. Também é bastante conhecido que conteúdo emocional pode ser expressado na linguagem, como nos trabalhos sobre análise de sentimentos em textos (PANG; LEE, 2008). Estudos já mostraram que respostas textuais de alunos e respostas em conversações podem carregar padrões que transmitem o estado de confusão (D'MELLO et al., 2008; D'MELLO; GRAESSER, 2014b).

Resumo para facilitar o entendimento do capítulo

Nem todos os pesquisadores concordam sobre o que é a confusão. Alguns a classificam como uma emoção genuína, outros dizem que está ligada ao aprendizado, e há até quem afirme que ela nem deveria ser considerada uma emoção. Essa incerteza existe porque a confusão

compartilha algumas características das emoções, mas não todas. Por exemplo, quando um aluno está programando e se depara com um erro que não entende, ele pode sentir confusão – um estado que ativa o cérebro, faz ele franzir a testa e até pode levá-lo a expressar essa emoção verbalmente, como dizendo "o que está acontecendo aqui?".

Os pesquisadores descobriram que a confusão geralmente acontece quando a informação nova não combina com o que o aluno já sabe. Imagine que um estudante está aprendendo sobre estruturas de repetição e vê um código em Python que contém *while(True):*. Se ele nunca viu essa sintaxe antes, pode se perguntar: "Por que esse loop nunca para?" Esse estranhamento inicial pode gerar confusão, mas se o aluno se dedicar a entender a lógica por trás do código, essa confusão pode ajudá-lo a aprender algo novo.

A confusão pode ser positiva ou negativa. Se um aluno consegue resolver o problema por conta própria, ele aprende e fortalece seu conhecimento. Mas se a confusão persistir por muito tempo sem solução, pode virar frustração e fazer o estudante desistir. Por exemplo, se alguém tenta rodar um código e recebe um erro como *"TypeError: 'int' object is not iterable"*, mas não sabe o que isso significa e não tenta pesquisar ou pedir ajuda, a frustração pode levá-lo a abandonar a atividade.

A confusão pode ser percebida em expressões faciais, como sobrancelhas franzidas, ou até na postura corporal, quando um aluno

começa a se mexer inquieto na cadeira. Além disso, estudos mostraram que alunos frequentemente expressam confusão ao falarem frases como “Isso não faz sentido” ou “Não entendi nada!”. Essas pistas podem ajudar professores a identificar quando um aluno está confuso e precisa de suporte.

A confusão faz parte do aprendizado, especialmente em áreas complexas como programação. O mais importante é saber lidar com ela: testar hipóteses, buscar ajuda e persistir na resolução do problema. Quando bem administrada, a confusão pode ser um sinal de que o cérebro está no caminho certo para aprender algo novo.

6. Como a confusão influencia a aprendizagem?

Este capítulo é curto, mas importante para reforçar como a confusão está relacionada com a aprendizagem, antes de avançarmos para os efeitos dessa emoção quando o aluno está aprendendo programação.

Apesar de a confusão ser comumente descrita como uma emoção negativa para o processo de aprendizagem, ela não deveria ser evitada, pois a partir dela o aluno pode engajar esforços para manter foco e atenção e buscar o conhecimento que lhe falta em materiais ou através da ajuda de colegas e professores (D'MELLO; GRAESSER, 2012b). Trabalhos já encontraram uma correlação da confusão com a aprendizagem (CRAIG et al., 2004; D'MELLO; GRAESSER; PICARD, 2007), inclusive em ambientes de programação de computadores (LEE et al., 2011). O estudo de Graesser et al. (2007) encontrou que a confusão foi a única emoção entre as investigadas (incluíram as demais emoções relacionadas a aprendizagem: frustração, tédio e engajamento) que apresentou uma correlação positiva com os resultados de testes de conhecimento, de forma que a confusão foi um bom preditor do sucesso ou fracasso da aprendizagem e um bom preditor de conhecimento.

Outros trabalhos também já encontraram evidências de uma correlação da confusão com a falta de conhecimento (D'MELLO; PERSON; LEHMAN, 2009; LEE et al., 2011; RODRIGO; BAKER; NABOS, 2010).

Os estudos descritos no capítulo 4 mostraram que quando o aluno passa muito tempo no estado de confusão, seu estado emocional pode mudar para frustração e tédio, emoções consideradas negativas para o processo de aprendizagem (D'MELLO; GRAESSER, 2012a). No entanto, quando resolvida a tempo, a confusão pode levar o aluno a adquirir níveis mais profundos de conhecimento (ARGUEL et al., 2019b; D'MELLO et al., 2014; LEHMAN et al., 2013; LEHMANN; GRAESSER, 2015), inclusive em ambientes de programação de computadores (LEE et al., 2011).

Resumo para facilitar o entendimento do capítulo

A confusão, embora muitas vezes vista como negativa, pode ser benéfica para a aprendizagem, pois incentiva os alunos a focar e buscarem soluções para suas dúvidas, seja em materiais de estudo, com colegas ou professores. Pesquisas mostraram que a confusão está associada a melhores resultados em testes de conhecimento, sendo um bom indicador de aprendizado, inclusive em programação. No entanto, se um aluno permanecer confuso por muito tempo sem resolver suas

dificuldades, pode acabar frustrado ou entediado, prejudicando o aprendizado. Por outro lado, quando superada a tempo, a confusão pode levar a um entendimento mais profundo do conteúdo.

7. Como a confusão influencia a aprendizagem de programação de computadores?

O trabalho apresentado por Kinnunen e Simon (2010) coletou relatos de estudantes de uma disciplina introdutória de programação em uma universidade, com o objetivo de entender as experiências emocionais envolvidas. O estudo encontrou que quando o aluno se depara pela primeira vez com enunciados de tarefas de programação, a falta de entendimento pode levá-lo de uma completa desorientação a um estado de confusão. Neste momento, o aluno pode se sentir travado, sem saber o que fazer. Os alunos que conseguiram obter um melhor entendimento dos enunciados foram aqueles que leram os enunciados mais de uma vez e discutiram as instruções das tarefas com um colega.

Quando o estudante passa por essa dificuldade inicial e começa a entender o problema, ele ainda pode não saber como prosseguir ou por onde começar a programar. Isso pode levá-lo novamente a um estado de confusão. Nesta situação, os alunos que obtiveram sucesso nas tarefas usaram estratégias como procurar por exemplos em livros, fazer pesquisa na internet ou começar a escrever algum código, mesmo sem saber exatamente se estava no caminho certo. Na medida que os alunos

foram programando e cometendo os primeiros erros em sequência, foram observadas fortes emoções como confusão, frustração e raiva. Estes alunos também passaram a refletir e a questionar sobre a própria capacidade de resolverem problemas, ou seja, começaram a questionar a sua autoeficácia. Estes relatos mostram que o contexto de tarefas de programação de computadores é fortemente emocional e propício para o desequilíbrio cognitivo e a confusão.

Estudos já mostraram que a confusão pode ser prejudicial, quando não resolvida a tempo, mas também pode ser benéfica, levando a níveis mais profundos de aprendizagem, quando os impasses que levam ao desequilíbrio cognitivo são resolvidos (D'MELLO et al., 2014; LEHMANN; GRAESSER, 2015). Estes resultados também parecem ser verdadeiros no contexto de programação de computadores. O trabalho de Lee et al. (2011) encontrou que a confusão prolongada em atividades de programação levou a uma piora de desempenho, mas quando resolvida, levou a resultados melhores do que se o aluno não tivesse estado confuso.

Sobre as transições emocionais no desenvolvimento de algoritmos, o trabalho de Bosch e D'Mello (2017) testou o modelo de dinâmicas afetivas (vimos no capítulo 4) no contexto de tarefas de programação de computadores com novatos (D'MELLO; GRAESSER, 2012a). Foram encontradas as mesmas transições do modelo de comparação. Segundo os autores (BOSCH; D'MELLO, 2017), os impasses que levam à

confusão costumam surgir quando os alunos se deparam com problemas não familiares, erros de sintaxe e saídas não esperadas dos algoritmos. Estes impasses podem, então, levar à frustração, que pode transitar para um estado de tédio.

Em relação ao comportamento dos alunos em tarefas de programação de computadores, um trabalho que identificou a confusão, o engajamento, a frustração e o tédio como as emoções mais proeminentes neste contexto, encontrou que quando o estudante está confuso, ele fica mais tempo ocioso no sistema (BOSCH; D'MELLO; MILLS, 2013).

Resumo para facilitar o entendimento do capítulo

Um estudo com estudantes de programação mostrou que, ao se depararem com tarefas pela primeira vez, muitos entram em estado de confusão, sentindo-se travados e sem saber por onde começar. Os alunos que conseguiram superar essa fase foram aqueles que releram o enunciado, discutiram com colegas e buscaram referências em livros e na internet. Erros ao programar geraram emoções como frustração e raiva, levando alguns a questionarem sua própria capacidade. No entanto, a confusão pode ser benéfica quando resolvida, aprofundando o aprendizado. Por outro lado, se prolongada, pode levar a queda de desempenho e desmotivação. Estudos indicam que erros inesperados,

problemas de sintaxe e dificuldades em interpretar resultados são gatilhos comuns para a confusão, que pode evoluir para frustração e até tédio, impactando o engajamento do aluno.

No próximo capítulo, são apresentadas estratégias que os alunos poderiam utilizar nos seus estudos de programação para se beneficiar da confusão.

8. Estratégias para o ALUNO se beneficiar da confusão durante a aprendizagem de programação

A confusão é frequentemente vista como um obstáculo no aprendizado de programação, mas, quando bem gerenciada, pode se tornar uma ferramenta poderosa para o desenvolvimento do conhecimento, como já discutimos em capítulos anteriores. Em vez de sinalizar fracasso, a confusão pode indicar que o aluno está no limiar de uma nova compreensão, desafiando-o a explorar diferentes abordagens, testar hipóteses e aprofundar seu raciocínio lógico. Neste capítulo, são apresentadas 12 estratégias que o aluno poderia adotar para transformar a confusão em um aliado no processo de aprendizado. As estratégias foram pensadas para ajudar o estudante a lidar com momentos de incerteza de forma produtiva, promovendo autonomia e resiliência na jornada de aprendizagem em programação. Ao aplicar essas estratégias, o aluno poderia transformar a confusão de um obstáculo em um trampolim para um aprendizado mais profundo e eficaz.

As estratégias apresentadas neste capítulo foram elaboradas com base na experiência do autor, que há mais de 15 anos atua como professor de programação, acompanhando de perto os desafios

enfrentados pelos estudantes ao aprender a codificar. Além da vivência prática em sala de aula, essas estratégias foram alinhadas aos conceitos discutidos nos capítulos anteriores, que exploram a relação entre emoções e aprendizagem, o papel do *appraisal* na confusão e no que foi discutido sobre as habilidades metacognitivas autorregulatórias. Vamos às estratégias:

1. Reconhecer e aceitar a confusão

Uma primeira etapa importante é o aluno compreender que sentir-se confuso é normal e que este pode ser um indicativo de que está no caminho do aprendizado. Em vez de evitar a confusão, o aluno deveria aceitá-la como parte do processo.

2. Praticar a metacognição

A metacognição envolve a capacidade de o aluno refletir sobre o próprio pensamento. O aluno poderia se perguntar o seguinte:

- O que exatamente está me confundindo?
- Que partes do problema eu entendi e quais partes estão nebulosas?
- Já enfrentei algo semelhante antes?

Fazer questionamento como esses pode ajudar o aluno a organizar o pensamento e encontrar estratégias para superar os obstáculos.

3. Desconstruir o problema em partes menores

Em programação, problemas complexos podem ser divididos em partes menores e mais gerenciáveis. Ao codificar, o aluno poderia:

- Focar em resolver primeiro os elementos mais básicos do problema.
- Testar pequenos trechos de código separadamente antes de integrá-los.

4. Buscar exemplos e analogias

O aluno poderia procurar na internet ou em livros por exemplos resolvidos e compará-los com o problema em questão. Isso poderia ajudá-lo a entender padrões e abordagens de solução.

5. Explorar diferentes recursos

O aluno poderia buscar conhecimento em diversas fontes: documentação oficial; fóruns como Stack Overflow; tutoriais em vídeo ou livros; discussões com colegas e professores; e IAs generativas como o ChatGPT. A exposição a diferentes explicações pode oferecer novas perspectivas e facilitar a compreensão.

6. Praticar o "Rubber Duck Debugging" (Depuração do Pato de Borracha)

Essa técnica consiste em explicar o problema para um objeto inanimado - como um pato de borracha -, para o espelho ou para outra pessoa. Esse processo força o aluno a estruturar seu pensamento e muitas vezes resulta em perceber onde está o erro ou lacuna na compreensão.

7. Escrever notas e diagramas

O estudante poderia criar resumos, fluxogramas ou mapas mentais para visualizar melhor o problema e suas possíveis soluções.

8. Testar hipóteses e experimentar

Em vez de ficar preso a uma abordagem que não está funcionando, o aluno poderia testar diferentes soluções, alterando pequenos trechos de código e observando o que acontece.

9. Ajustar o ritmo de aprendizagem

O aluno poderia fazer pausas e revisar o problema mais tarde. Isso poderia ajudar a mente a processar a informação subconscientemente. Às vezes, o insight vem quando menos se espera.

10. Usar a confusão como indicador de aprendizado profundo

Se um conceito de programação está gerando confusão, poderia ser um sinal de que o aluno está prestes a entender algo importante. Em vez de desistir, o aluno deveria persistir e ver a confusão como uma etapa que surge antes da compreensão.

11. Desenvolver resiliência e persistência

A programação também envolve tentativa e erro. Encarar desafios com uma mentalidade de crescimento ajuda o aluno a ver cada erro como uma oportunidade de aprendizado, em vez de um fracasso.

12. Ensinar alguém

Explicar um conceito para outra pessoa é um dos métodos mais eficientes para aprender algo de maneira sólida. Ensinar alguém solidifica o entendimento e pode revelar lacunas na própria compreensão.

9. Estratégias para o PROFESSOR ajudar seus alunos a se beneficiar da confusão na aprendizagem de programação

Nos capítulos anteriores, foram apresentadas várias discussões sobre a confusão ser um estado emocional inevitável na aprendizagem de programação, e que ao invés de ser vista como um obstáculo, deve ser vista pelos alunos como uma ferramenta para impulsionar o raciocínio e a autonomia. O papel do professor nesse contexto é fundamental, pois sua abordagem pode determinar se a confusão levará ao aprendizado ou à frustração. Neste capítulo, são apresentadas **12 estratégias que os professores poderiam considerar em suas aulas para incentivar seus alunos a transformar a confusão em uma oportunidade de crescimento**. Essas estratégias incluem desde técnicas de questionamento e estímulo à metacognição até o uso de práticas pedagógicas que incentivam a autonomia e a resiliência no aprendizado de programação.

Ao implementar essas abordagens, o professor não apenas auxilia os alunos a superar dificuldades momentâneas, mas também contribui para o desenvolvimento de habilidades essenciais, como pensamento crítico, persistência e resolução de problemas. Ao adotar essas

estratégias, o professor ajuda os alunos a transformar a confusão em um mecanismo de aprendizado ativo, desenvolvendo sua capacidade de resolução de problemas e promovendo maior independência no processo de aprendizagem.

Da mesma forma como no capítulo anterior, as estratégias aqui apresentadas também foram pensadas a partir da experiência do autor no ensino de programação, alinhadas aos conceitos discutidos anteriormente sobre emoções, aprendizagem e autorregulação metacognitiva. Vamos a elas:

1. Normalizar e valorizar a confusão

O professor poderia trabalhar com os alunos a ideia de que a confusão é um estágio natural da aprendizagem e não um sinal de fracasso. Poderia também compartilhar histórias de programadores experientes que passaram por desafios semelhantes e criar um ambiente seguro, onde os alunos se sintam confortáveis para expressar dúvidas e dificuldades. Além disso, o professor pode relatar suas próprias dificuldades quando era aluno e quando aprendeu a programar, mostrando que errar e enfrentar obstáculos faz parte do processo de aprendizagem e desenvolvimento.

2. Estimular a metacognição

O professor poderia incentivar os alunos a refletirem sobre suas dificuldades, através dos seguintes questionamentos que os alunos poderiam fazer para si mesmos:

- O que exatamente está causando a confusão?
- Que estratégias já tentei para resolver esse problema?
- O que eu já sei que pode me ajudar aqui?

O professor também poderia sugerir aos alunos que mantenham um diário de aprendizagem para registrar dúvidas e insights ao longo do tempo. Esse registro pode ajudá-los a identificar padrões de dificuldades, acompanhar sua evolução e refletir sobre as estratégias que funcionam melhor para eles. Além disso, o diário pode servir como um recurso para revisões futuras e incentivar a autonomia no aprendizado.

3. Aplicar o método da pergunta socrática

Uma pergunta socrática é uma pergunta aberta e reflexiva que incentiva o pensamento crítico e a exploração mais profunda de um tema. Esse tipo de pergunta é inspirado no método socrático, uma

técnica atribuída ao filósofo grego Sócrates, que buscava estimular o raciocínio e a autodescoberta por meio do questionamento.

Por exemplo, imagine um aluno que pergunta: *'Por que precisamos usar estruturas de loops em programação?'* Em vez de dar uma resposta direta, o professor poderia responder com uma pergunta socrática, como: *'O que aconteceria se você precisasse repetir a mesma instrução 100 vezes sem uma estrutura de loop? Como isso afetaria o seu código?'*. Isso poderia levar o aluno a refletir sobre a necessidade e a utilidade dos *loops*, ajudando-o a compreender o conceito de forma mais profunda, sem receber uma resposta pronta do professor.

Em vez de dar respostas diretas e finais, o professor poderia conduzir os alunos por meio de perguntas que os ajudassem a encontrar soluções por conta própria. Por exemplo, se um aluno não entende um erro de código, o professor poderia fazer as seguintes perguntas:

- O que esse erro significa?
- O que acontece se você mudar esse trecho do código?

4. Utilizar problemas progressivos

O professor poderia apresentar aos alunos desafios gradualmente mais difíceis, de modo a ajudar os alunos a desenvolver resiliência e confiança. Poderia projetar tarefas que contenham pequenos obstáculos

para incentivar a experimentação e a exploração, e fornecer exercícios que promovam debugging ativo, onde os alunos precisem encontrar e corrigir erros.

O professor poderia começar com um exercício simples, como pedir aos alunos que criem um programa que soma dois números. Depois, pode aumentar a complexidade, pedindo que façam um programa que some dois números e subtraia por um terceiro número. Em seguida, poderia introduzir mais operações. Poderia, depois, aos poucos, introduzir o tratamento de entrada do usuário e a validação de dados. A ideia é propor vários exercícios, em que o próximo exercício é apenas um pequeno desafio a mais.

5. Promover a aprendizagem cooperativa

O professor poderia incentivar os alunos a trabalharem em pares ou pequenos grupos para resolver problemas. Poderia implementar metodologias como *pair programming* (programação em pares), onde um aluno escreve o código enquanto o outro revisa e sugere melhorias. Poderia também criar momentos para que alunos expliquem conceitos uns aos outros, consolidando seu aprendizado.

6. Incorporar estratégias de "Productive Failure" (Fracasso Produtivo)

O professor poderia permitir aos alunos tentarem resolver problemas antes de fornecer a explicação teórica completa. Poderia criar atividades em que os alunos testam diferentes abordagens antes de serem guiados para a solução ideal. Por fim, o professor poderia enfatizar que errar faz parte do processo e que cada tentativa traz um aprendizado.

O professor também poderia incentivar discussões entre os alunos para que compartilhem suas diferentes abordagens e raciocínios, promovendo a aprendizagem colaborativa. Ao invés de apenas corrigir os erros, o professor pode questionar os alunos sobre por que determinada solução não funcionou e o que poderia ser ajustado, estimulando o pensamento crítico. Também pode propor desafios onde os alunos revisam e melhoram suas próprias soluções, reforçando a ideia de que a programação é um processo iterativo e que a melhoria contínua é mais importante do que encontrar a resposta correta de imediato.

7. Ensinar técnicas de *debugging*

O professor poderia apresentar métodos sistemáticos para depuração de código, como os seguintes:

- **Dividir para conquistar:** Ao enfrentar um erro em um código complexo, os alunos podem adotar a estratégia de "dividir para conquistar", isolando partes específicas para testar individualmente. Isso pode ser feito separando funções, testando trechos menores de código antes de integrá-los e verificando cada etapa de execução. Por exemplo, se um programa que calcula a média de notas não retorna o valor esperado, o aluno pode primeiro testar a leitura dos dados, depois a conversão para números e, por fim, a operação matemática, identificando assim onde está o problema.
- **Inserir prints/logs:** Uma técnica eficaz de depuração é inserir prints ou logs para visualizar valores intermediários e acompanhar a execução do programa. Adicionar comandos como *print(f"Valor de x: {x}")* ajuda os alunos a entender como as variáveis estão sendo modificadas ao longo do código. Para projetos maiores, é recomendado o uso de bibliotecas de logging, que permite diferentes níveis de mensagens (*debug*, *warning*, *error*), tornando a depuração mais estruturada e eficiente.
- **Usar ferramentas de depuração:** Os depuradores (*debuggers*) das IDEs permitem que os alunos “pausem” a execução do programa, inspecionem variáveis e avancem linha por linha sem precisar

modificar o código com prints. No VS Code, por exemplo, é possível adicionar breakpoints e utilizar os botões "Step Over" e "Step Into" para navegar pelo código enquanto ele é executado. Essa abordagem facilita a identificação de valores inesperados, loops infinitos e exceções, tornando o processo de depuração mais preciso e eficiente.

- **Criar desafios de caça a bugs:** Uma forma interativa de ensinar depuração é criar desafios de "caça a bugs", onde os alunos recebem códigos propositalmente com erros e precisam identificar e corrigir os problemas. O professor pode incluir erros comuns, como índices fora dos limites de uma lista, divisão por zero ou variáveis não inicializadas. Além de encontrar os erros, os alunos devem explicar por que ocorreram e propor soluções corretas, desenvolvendo assim um olhar mais crítico e habilidades de depuração de maneira prática e engajante.

8. Utilizar Estratégias Visuais e Analógicas

O professor poderia explicar conceitos abstratos com diagramas, fluxogramas e pseudocódigo antes de mergulhar no código. Poderia utilizar analogias para facilitar a compreensão de conceitos complexos (por exemplo, comparar variáveis com caixas de armazenamento).

9. Implementar revisões baseadas em erros comuns

O professor poderia coletar erros frequentes cometidos pelos alunos e transformá-los em atividades de revisão. Um exemplo seria criar um "museu de bugs", onde os alunos compartilham erros interessantes que cometeram e discutem como os corrigiram.

10. Incentivar a autonomia e a exploração

O professor poderia incentivar os alunos a buscarem respostas em documentações, fóruns e tutoriais antes de pedir ajuda. Poderia ensinar como fazer boas pesquisas no Google, no Stack Overflow, ou nas IAs generativas como o ChatGPT. Poderia sugerir desafios extras para que os alunos explorem novas soluções por conta própria.

11. Criar um ambiente de feedback contínuo

O professor poderia oferecer feedback detalhado e contínuo, não apenas sobre se a solução está certa ou errada, mas sobre o raciocínio por trás dela.

12. Introduzir o ensino baseado em projetos

O professor poderia propor projetos práticos que desafiem os alunos a conectar conceitos e resolver problemas reais. Poderia permitir que os alunos escolham temas de interesse para aumentar o engajamento e dar suporte à criação de projetos colaborativos, onde diferentes alunos contribuem com partes do código.

10. Considerações finais

A confusão, frequentemente vista como um obstáculo na aprendizagem de programação, pode ser transformada em um poderoso aliado tanto para alunos quanto para professores. Os capítulos anteriores demonstraram que, quando gerenciada adequadamente, a confusão pode impulsionar o aprendizado, incentivando a exploração de diferentes estratégias, o pensamento crítico e o desenvolvimento da autonomia. Este capítulo final discute os principais pontos abordados e destaca a importância de abordagens pedagógicas que consideram a confusão como parte essencial do processo de aprendizagem.

Tanto para alunos quanto para professores, a confusão representa uma oportunidade de crescimento. Para os estudantes, reconhecer e aceitar a confusão como uma etapa natural do aprendizado é o primeiro passo para superá-la. Práticas como a metacognição, a desconstrução de problemas em partes menores e a exploração de diferentes recursos permitem que o aluno enfrente desafios com maior segurança. O desenvolvimento da resiliência e da persistência também se mostra essencial, pois a programação envolve um processo contínuo de tentativa e erro.

Para os professores, a confusão pode ser um indicativo de aprendizado ativo, exigindo uma abordagem que normalize e valorize

esse estado afetivo. O uso de estratégias pedagógicas como perguntas socráticas, aprendizagem cooperativa e feedback contínuo permite que os alunos desenvolvam habilidades para superar momentos de incerteza. O incentivo à autonomia, através da busca por soluções em documentações, fóruns e ferramentas como IAs generativas, também pode ampliar o repertório de estratégias dos estudantes.

A confusão desempenha um papel crucial no desenvolvimento do raciocínio computacional e na capacidade de resolução de problemas, pois, quando bem administrada, promove um aprendizado mais profundo e duradouro. Alunos que aprendem a lidar produtivamente com a confusão tornam-se mais autônomos, confiantes e preparados para desafios mais complexos. No entanto, se não for tratada adequadamente, pode levar à frustração e à desmotivação, tornando essencial a atuação do professor. Com um suporte pedagógico adequado, momentos de incerteza podem ser transformados em oportunidades valiosas de reflexão e crescimento.

A confusão não deve ser encarada como uma barreira ao aprendizado, mas sim como um indicador de progresso e um estímulo para a exploração ativa do conhecimento. Tanto alunos quanto professores têm um papel fundamental na transformação da confusão em uma ferramenta de aprendizagem. Enquanto os alunos podem adotar estratégias para lidar melhor com momentos de incerteza, os professores

devem criar um ambiente que favoreça a resolução produtiva da confusão.

Ao longo deste livro, discutimos não apenas a natureza da confusão, mas também formas de abordá-la de maneira construtiva. Através de estratégias bem estruturadas, é possível transformar desafios em oportunidades de aprendizado significativo, preparando os estudantes para um futuro onde a capacidade de resolver problemas e se adaptar a novas situações é cada vez mais valorizada.

Referências

- ALZOUBI, O.; D'MELLO, S. K.; CALVO, R. A.** Detecting Naturalistic Expressions of Nonbasic Affect Using Physiological Signals. *IEEE Transactions on Affective Computing*, v. 3, n. 3, p. 298–310, jul. 2012.
- ARGUEL, A. et al.** Seeking optimal confusion: a review on epistemic emotion management in interactive digital learning environments. *Interactive Learning Environments*, v. 27, n. 2, p. 200–210, 2019b.
- ARNOLD, M. B.** Emotion and personality. *New York, NY, US: Columbia University Press*, 1960.
- AZEVEDO, R.; TAUB, M.; MUDRICK, N. V.** Understanding and Reasoning about Real-Time Cognitive, Affective, and Metacognitive Processes to Foster Self-Regulation with Advanced Learning Technologies. *Handbook of Self-Regulation of Learning and Performance*, p. 254–270, 2019.
- BARRETT, L. F.** Solving the Emotion Paradox: Categorization and the Experience of Emotion. *Personality and Social Psychology Review*, v. 10, n. 1, p. 20–46, 2006.
- BOSCH, N.; D'MELLO, S.** The Affective Experience of Novice Computer Programmers. *International Journal of Artificial Intelligence in Education*, v. 27, n. 1, p. 181–206, 2017.
- BOSCH, N.; D'MELLO, S.; MILLS, C.** What emotions do novices experience during their first computer programming learning session? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 7926 LNAI, p. 11–20, 2013.

CALVO, R. A.; D'MELLO, S. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, v. 1, n. 1, p. 18–37, 2010.

CHI, M. T. H.; OHLSSON, S. Complex Declarative Learning. In: *The Cambridge handbook of thinking and reasoning*. New York, NY, US: Cambridge University Press, 2005. p. 371–399.

COTO, M. et al. Emotions and programming learning: systematic mapping. *Computer Science Education*, v. 00, n. 00, p. 1–36, 2021.

CRAIG, S. et al. Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media*, v. 29, n. 3, p. 241–250, 2004.

CRAIG, S. D. et al. Emote aloud during learning with AutoTutor: Applying the Facial Action Coding System to cognitive - Affective states during learning. *Cognition and Emotion*, v. 22, n. 5, p. 777–788, 2008a.

DI LEO, I.; MUIS, K. R. *Confused, now what? A Cognitive-Emotional Strategy Training (CEST) intervention for elementary students during mathematics problem solving*. *Contemporary Educational Psychology*, v. 62, n. May, p. 101879, 2020.

DARWIN, C. *The expression of the emotions in man and animals*. John Murray, 1872.

D'MELLO, S. A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *Journal of Educational Psychology*, v. 105, n. 4, p. 1082–1099, 2013.

D'MELLO, S. K. et al. Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*, v. 18, n. 1–2, p. 45–80, 2008.

D'MELLO, S.; et al. Confusion can be beneficial for learning. *Learning and Instruction*, v. 29, p. 153–170, 2014.

D'MELLO, S.; CALVO, R. A. Beyond the basic emotions. *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*, p. 2287, 2013.

D'MELLO, S.; DALE, R.; GRAESSER, A. Disequilibrium in the mind, disharmony in the body. *Cognition and Emotion*, v. 26, n. 2, p. 362–374, 2012.

D'MELLO, S.; GRAESSER, A.; PICARD, R. W. Toward an affect-sensitive autotutor. *IEEE Intelligent S.*, v. 22, n. 4, p. 53–61, 2007.

K. D'MELLO, S.; GRAESSER, A. Mining Bodily Patterns of Affective Experience during Learning. 2010.

D'MELLO, S.; GRAESSER, A. Dynamics of affective states during complex learning. *Learning and Instruction*, v. 22, n. 2, p. 145–157, 2012a.

D'MELLO, S.; GRAESSER, A. AutoTutor and affective autotutor. *ACM Transactions on Interactive Intelligent Systems*, v. 2, n. 4, p. 1–39, 1 dez. 2012b.

D'MELLO, S.; GRAESSER, A. Confusion and its dynamics during device comprehension with breakdown scenarios. *Acta Psychologica*, v. 151, p. 106–116, 2014a.

D'MELLO, S.; GRAESSER, A. Confusion. In: *International handbook of emotions in education*. Routledge, 2014b. p. 299–320.

D'MELLO, S. K.; KORY, J. Consistent but Modest: A Meta-Analysis on Unimodal and Multimodal Affect Detection Accuracies from 30 Studies. *Proceedings of the 14th ACM international conference on Multimodal interaction - ICMI '12*, p. 31–38, 2012.

D'MELLO, S.; PERSON, N.; LEHMAN, B. Antecedent-consequent relationships and cyclical patterns between affective states and problem solving outcomes. *Frontiers in Artificial Intelligence and Applications*, v. 200, n. 1, p. 57–64, 2009.

EKMAN, P.; FRIESEN, W. V. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of Personality and Social Psychology*, 1987.

FREDRICKSON, B. L. What good are positive emotions? *Review of General Psychology*, v. 2, n. 3, p. 300–319, 1998.

FRIJDA, N. H. The Laws of Emotion. *American Psychologist*, v. 43, n. 5, p. 349–358, 1988.

GRAESSER, A. et al. Emotions and Learning with AutoTutor. *Proceedings of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, p. 569–571, 2007.

GRAFSGAARD, J. F.; BOYER, K. E.; LESTER, J. C. Predicting facial indicators of confusion with hidden Markov models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 6974 LNCS, n. PART 1, p. 97–106, 2011.

GRAESSER, A.; D'MELLO, S. K. Theoretical Perspectives on Affect and Deep Learning. In: *New Perspectives on Affect and Learning Technologies*. New York, NY: Springer New York, 2011. p. 11–21.

HALGREN, E. et al. N400-like Magnetoencephalography Responses Modulated by Semantic Context, Word Frequency, and Lexical Class in Sentences. *NeuroImage*, v. 17, n. 3, p. 1101–1116, nov. 2002.

HESS, U. Now you see it, now you don't--the confusing case of confusion as an emotion: Commentary on Rozin and Cohen (2003). *Emotion*, v. 3, n. 1, p. 76–80, 2003.

IZARD, C. E. The many meanings/aspects of emotion: Definitions, functions, activation, and regulation. *Emotion Review*, v. 2, n. 4, p. 363–370, 2010.

JAQUES, P. A.; VICARI, R. M. Estado da Arte em Ambientes Inteligentes de Aprendizagem que Consideram a Afetividade do Aluno. *Informática na Educação*, v. 8, n. 1, p. 15–38, 2005.

KINNUNEN, P.; SIMON, B. Experiencing programming assignments in CS1: The emotional toll. *ICER'10 - Proceedings of the International Computing Education Research Workshop*, p. 77–85, 2010.

KELTNER, D.; SHIOTA, M. N. New displays and new emotions: A commentary on Rozin and Cohen (2003). *Emotion*, v. 3, n. 1, p. 86–91, 2003.

LAZARUS, R. S. Psychological stress and the coping process. *New York, NY, US: McGraw-Hill*, 1966.

LEE, D. M. C. et al. Exploring the relationship between novice programmer confusion and achievement. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 6974 LNCS, n. PART 1, p. 175–184, 2011.

LEHMAN, B. et al. Inducing and Tracking Confusion with Contradictions during Complex Learning. *International Journal of Artificial Intelligence in Education*, v. 22, n. 1–2, p. 85–105, 2013.

LEHMAN, B.; D'MELLO, S.; GRAESSER, A. Confusion and complex learning during interactions with computer learning environments. *Internet and Higher Education*, v. 15, n. 3, p. 184–194, 2012.

LEHMANN, B.; GRAESSER, A. To Resolve or Not to Resolve? That is the big question about confusion. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 9112, n. June, p. 883, 2015.

MCDANIEL, B. et al. Facial Features for Affective State Detection in Learning Environments. *Proceedings of the Annual Meeting of the Cognitive Science Society*. Anais... 2007.

MELLO, S. D. et al. I Feel Your Pain: A Selective Review of Affect-Sensitive Instructional Strategies. *Design Recommendations for Intelligent Tutoring Systems*, v. 2, p. 35–48, 2014.

MOORS, A. et al. Appraisal theories of emotion: State of the art and future development. *Emotion Review*, v. 5, n. 2, p. 119–124, 2013.

MOORS, A.; SCHERER, K. R. The Role of Appraisal in Emotion. In: *Handbook of cognition and emotion*. New York: Guilford Press, 2013. p. 135–155.

OLIVEIRA, E. DE; JAQUES, P. A. Inferindo as emoções do usuário pela face através de um sistema psicológico de codificação facial. *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems*, n. January 2008, p. 156–165, 2008.

ORTONY, A.; CLORE, G. L.; COLLINS, A. *The Cognitive Structure of Emotions*. New York: Cambridge University Press, 1988.

PANG, B.; LEE, L. Opinion Mining and Sentiment Analysis: Foundations and Trends in Information Retrieval. [s.l: s.n.]. v. 2.

PEKRUN, R. *Emotions as Drivers of Learning and Cognitive Development*. In: *New Perspectives on Affect and Learning Technologies*. New York, NY: Springer New York, 2011. p. 23–39.

PEKRUN, R.; STEPHENS, E. J. Academic emotions. In: *APA educational psychology handbook, Vol 2: Individual differences and cultural and contextual factors*. Washington: American Psychological Association, 2012. v. 2, p. 3–31.

PICARD, R. W. *Affective Computing: From laughter to IEEE*. IEEE Transactions on Affective Computing, v. 1, n. 1, p. 11–17, 2010.

REIS, H. et al. *Analysis of Permanence Time in Emotional States: A Case Study Using Educational Software*. International Conference on Intelligent Tutoring Systems (ITS), p. 180–190, 2018.

RODRIGO, M. M. T.; BAKER, R. S. J.; NABOS, J. Q. The Relationships Between Sequences of Affective States and Learner Achievement. *Proceedings of the 18th International Conference on Computers in Education*, n. December 2016, p. 56–60, 2010.

ROZIN, P.; COHEN, A. B. High frequency of facial expressions corresponding to confusion, concentration, and worry in an analysis of naturally occurring facial expressions of Americans. *Emotion*, v. 3, n. 1, p. 68–75, 2003.

RUSSELL, J. A. *Core affect and the psychological construction of emotion*. *Psychological Review*, v. 110, n. 1, p. 145–172, 2003.

SCHERER, K. R. Toward a dynamic theory of emotion. *Geneva Studies in Emotion*, v. 1, p. 1–96, 1987.

SCHERER, K. R. Appraisal Theory. In: DALGLEISH, T.; POWER, M. J. (Eds.). *Handbook of Cognition and Emotion*. [s.l.] John Wiley & Sons, 1999. p. 637–663.

SCHERER, K. R. What are emotions? and how can they be measured? *Social Science Information*, v. 44, n. 4, p. 695–729, 2005.

SCHERER, K. R. *The dynamic architecture of emotion: Evidence for the component process model*. *Cognition & Emotion*, v. 23, n. 7, p. 1307–1351, 2009.

SILVIA, P. J. Confusion and interest: The role of knowledge emotions in aesthetic experience. *Psychology of Aesthetics, Creativity, and the Arts*, v. 4, n. 2, p. 75–80, 2010.

Entre Bugs e Emoções:

1ª edição

Como a Confusão pode Beneficiar a Aprendizagem de Programação



Você já se sentiu perdido durante a codificação de um exercício de programação? Ficou encarando a tela sem entender o que está acontecendo? E se eu disser que essa confusão pode ser sua maior aliada no aprendizado?

Este livro explora o papel das emoções na aprendizagem e mostra como a confusão, quando bem aproveitada, pode impulsionar a evolução do aprendizado de programação. Você vai entender por que se sentir confuso faz parte do processo, como isso influencia sua forma de pensar e aprender, e como transformar essa sensação em um trampolim para dominar a programação. Além de estratégias para o aluno se beneficiar da sua confusão, o livro também apresenta estratégias que os professores poderiam adotar para ajudar seus alunos a se beneficiarem da confusão durante as tarefas de programação.

Prepare-se para enxergar a confusão de um jeito completamente novo e transformar suas dificuldades em progresso. O próximo bug que aparecer pode ser exatamente o que você precisa para dar um salto no seu aprendizado!



Tiago Roberto Kautzmann

Série Aprendizagem de Programação