

# Por que estimular a Aprendizagem Significativa no ensino de Programação Orientada a Objetos?

Humberto A. P. Zanetti  
Faculdade de Tecnologia - FT  
Universidade Estadual de Campinas - UNICAMP  
Limeira - SP - Brasil  
h016304@dac.unicamp.com

Marcos A. F. Borges  
Faculdade de Tecnologia - FT  
Universidade Estadual de Campinas - UNICAMP  
Limeira - SP - Brasil  
marcosborges@ft.unicamp.com

## RESUMO

Este artigo apresenta uma análise crítico-reflexiva sobre a adoção da teoria da Aprendizagem Significativa no ensino-aprendizagem de Programação Orientada a Objetos. Este texto apresenta a visão dos autores de como a teoria de David Ausubel pode ser aplicada com resultados positivos no processo de construção do conhecimento, em especial no ensino de Programação Orientada a Objetos por meio de práticas e recursos que possam trazer mais significado ao aluno, como por exemplo, o uso Computação Física e outros recursos didáticos.

## PALAVRAS-CHAVE

Ensino de programação, Aprendizagem Significativa, Abstração.

## 1 Introdução

O ensino de programação para iniciantes possui muitos desafios. Essa discussão é foco de muitas pesquisas e ainda é uma área com muitas possibilidades de exploração. Butler e Morgan [5] argumentam que um dos principais problemas encontrados pelos alunos é combinar conceitos abstratos encontrados na programação e os processos de raciocínio lógico. Trabalhos como de Hadar [11] e Xinogalos [30], que abordam o ensino de Programação Orientada a Objetos (POO) para iniciantes, apontam que uma das principais deficiências dos aprendizes de POO é conseguir relacionar os conceitos de forma lógica e racional. No trabalho de Hadar [11] é possível notar que muitos alunos não conseguem associar “objetos” ao enunciado proposto, por não entender a função básica do conceito de classe e sua instância (o próprio objeto). Na pesquisa de Xinogalos [30] é mostrado que a falta de compreensão de um conceito e sua correlação com os demais faz com que as dúvidas não sejam sanadas ao longo de todo o processo de aprendizagem.

A Aprendizagem Significativa (AS) é uma teoria que defende que a aquisição de conhecimento tem como premissa básica a influência do conhecimento prévio do aprendiz como sendo um fator fundamental. Segundo a AS, o indivíduo, no processo de

aprendizagem, necessita organizar todo o conhecimento em uma estrutura cognitiva. Segundo Ausubel [4], essa estrutura cognitiva é uma organização, um construto, que se relaciona com algo concreto. A estrutura é particular para cada aprendiz, que, de uma forma individual, a cria para aprender de forma significativa. Moreira [16] afirma que essa estrutura é caracterizada pela interação entre os conhecimentos novos e aqueles relevantes já existente na estrutura cognitiva do sujeito que aprende. Para a AS, a interação do conhecimento prévio com novos conhecimentos e a relação entre eles deve ser organizada de maneira lógica para a composição da estrutura cognitiva do indivíduo.

Muitas pesquisas têm como objetivo buscar novos meios de proporcionar um ensino motivador. Este trabalho traz uma breve discussão sobre as motivações e desafios que podem ser associados à adoção da AS, sugerindo alguns recursos que podem ser utilizados para ajudar nesse propósito.

## 2 A Aprendizagem Significativa

A teoria da AS é pautada nos conhecimentos relevantes previamente adquiridos e existentes na estrutura cognitiva do sujeito. Alguns desses conhecimentos, relevantes para a aprendizagem de novos conhecimentos, podendo ser representados como um símbolo, imagem, conceito ou modelo mental, são definidos como ideia-âncora ou *subsunçor* [4] [16]. Moreira [17] define que “*subsunçor* é o nome que se dá a um conhecimento existente na estrutura de conhecimentos do indivíduo, que permite dar significado a um novo conhecimento que lhe é apresentado ou por ele descoberto”.

Na concepção do seu criador, David Ausubel, a AS é a aprendizagem por recepção significativa, ou seja, requer a apresentação de materiais e conceitos que sejam potencialmente significativos para o aprendiz. O material envolvido na aprendizagem tem que estar relacionado de forma não arbitrária (não aleatória, plausível e sensível), tendo um significado lógico ao aluno [4]. Moreira [16] complementa, afirmando que, no contexto da AS, deve haver uma interação de ideias e conhecimentos com aquilo que o aluno já sabe: não com qualquer conhecimento prévio, mas com algum conhecimento especificamente relevante existente na estrutura cognitiva do sujeito.

Darroz [7] afirma que a AS contrasta com a maneira tradicional, definida por Ausubel como sendo a aprendizagem mecânica, na medida em que, na primeira, uma nova informação interage com um *subsunçor* existente na estrutura cognitiva, e na segunda, a nova informação não interage com os conhecimentos já adquiridos, não contribuindo com a progressão e desenvolvimento

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

*EduComp '21, Abril 27–30, 2021, Jataí, Goiás, Brasil (On-line)*

©2021 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

do aprendizado. Segundo Moreira [16], mesmo que em alguns momentos a aprendizagem mecânica se faça necessária, ela deve objetivar uma aprendizagem mais significativa ao aprendiz.

### 3 A Aprendizagem Significativa e o ensino de Programação Orientada a Objetos

O paradigma POO pode ser todo organizado em conceitos, denotando diferentes *subsunçores* e suas relações. De acordo com a AS um problema que pode ocorrer no ensino de POO é não mostrar qual é a relação dos conceitos novos com aqueles que estão sendo apresentados no momento. Por exemplo, o *subsunçor* “classe” deve ser vinculado de maneira explícita ao *subsunçor* “objeto” no momento de sua apresentação, para que o aprendiz consiga correlacionar os conceitos de forma natural e significativa.

Conforme a AS, a exposição de um conteúdo novo, sem estar associado ao anterior, pode não fazer sentido ao aluno. Segundo Astolfi e Junior [2], o educador tem como função promover mecanismos para que o aprendiz possa progressivamente complementar e enriquecer sua estrutura cognitiva, além de reorganizar e relacionar seus conhecimentos. A atuação do educador deve ser focada em apresentar conceitos mais genéricos primeiro, depois descrevendo os mais específicos, mostrando qual é a estrutura hierárquica do conhecimento e fortalecendo a consolidação da estrutura cognitiva do aluno.

A diferenciação progressiva, um dos princípios fundamentais da AS, é um processo muito adequado para o ensino de POO. Este princípio estabelece que a aprendizagem significativa deve ser um processo contínuo, no qual novos conceitos adquirem mais significância à medida que são alcançadas novas relações [18]. Os conceitos não são aprendidos de uma forma completa, mas sim são enriquecidos constantemente e modificados ao longo de diversas interações. Esse processo se assemelha muito com que identificamos como “experiência”. Ser experiente em algo denota que o sujeito consegue adquirir novos conhecimentos e associá-los aos que já estão incorporados, tornando-os mais explícitos, significativos e amplos, em um processo de contínua reorganização cognitiva. Moreira [17] reforça que a diferenciação progressiva é o processo de atribuir novos significados a um dado *subsunçor*, resultando em um *subsunçor* mais elaborado e diferenciado, servindo como âncora para novos conhecimentos

Quando um estudante de Computação que está aprendendo conceitos fundamentais de POO, e já compreende o conceito de “classe” (um *subsunçor*), por exemplo, quando for abordado o conceito de “herança” (novo *subsunçor*), o *subsunçor* “classe” irá adquirir mais significado, continuando a se modificar e sofrendo uma diferenciação progressiva baseada nessa nova relação. A cada vez que um aluno avança nos estudos, o *subsunçor* “classe” irá incorporar novos significados, dado a sua aplicação em novos contextos, por meio da experiência (uso contínuo e interativo) do *subsunçor* por parte do aluno.

A experiência do aluno está diretamente relacionada ao que ele realmente aprende, ou seja, quando ele reconhece de maneira autônoma novas relações conceituais. A reconciliação integradora é um processo que se refere à dinâmica da estrutura cognitiva, junto com a diferenciação progressiva, que consiste em eliminar ambiguidades, resolver inconsistências e criar uma ordem hierárquica na construção do conhecimento [17]. Ausubel [4] explica que é fundamental que o aluno identifique que os conceitos anteriormente aprendidos e presentes na estrutura cognitiva são

semelhantes, ou essencialmente diferentes, aos novos conceitos que estão sendo apresentados. Por exemplo, é de grande valia compreender que o *subsunçor* “classe” agrega significado tanto ao *subsunçor* “herança” quanto ao *subsunçor* “classe abstrata”, mas que “herança” e “classe abstrata” possuem ideias que as diferenciam, e não são obrigatoriamente complementares.

### 4 Desafios na adoção da Aprendizagem Significativa

Toda tentativa de criar novos métodos de ensino possui inúmeros desafios. Na estrutura de uma disciplina de programação, sempre há um início no qual deve-se pressupor que os alunos ainda não saibam o fundamental. Este seria o momento de apresentar um primeiro *subsunçor* a ser compreendido pelos alunos [2]. Também se faz necessário apresentar aos alunos, por meio de algum instrumento, de uma maneira menos abstrata e lógica, todas as relações de conceitos, contribuindo para a evolução de sua estrutura cognitiva.

Quando não há o conhecimento prévio ou não há *subsunçores* adequados, pode-se utilizar de organizadores prévios para fornecer o primeiro *subsunçor* necessário [4]. O professor, nesse caso, deve apelar para o meio tradicional de ensino, com o objetivo de tornar a aprendizagem significativa após a introdução desse primeiro *subsunçor* [15]. Na pesquisa de Pimentel e Osmar [20] são utilizados recursos de aprendizagem convencionais (aprendizagem mecânica) para dar uma aula introdutória de programação, para que os alunos, a partir disso, possam iniciar a estruturação e relação dos novos conceitos.

Outra importante etapa do processo de aplicação da AS no ensino de programação é definir como apresentar a relação e estrutura dos *subsunçores*. Novak e Gowin [18] baseiam sua pesquisa na adoção de mapas conceituais como instrumento para representar relações significativas no formato de proposições. A construção do conhecimento deve ser organizada e hierarquizada, de forma a oferecer uma compreensão mais simples dos relacionamentos entre os conceitos envolvidos [19].

Para um processo de aprendizagem de um novo paradigma de programação, ou mesmo de uma nova linguagem de programação, a organização dos conceitos envolvidos é fundamental para o entendimento de qual a relação de cada conceito aprendido em relação aos demais, facilitando a compreensão lógica. A percepção da relação entre os conceitos é fundamental para uma evolução progressiva e coerente por parte do aprendiz. Butler e Morgan [5] apontam uma fragilidade no processo de aprendizagem de programação, especificamente na dificuldade que o aluno tem quando é requerido o pensamento abstrato, correlação de conceitos e as aplicações lógicas em resolução de problemas. Astolfi e Junior [3] mostram a eficácia em identificar conhecimentos prévios dos alunos para conduzir as práticas pedagógicas, minimizando as deficiências conceituais, denotando a necessidade em criar meios que trazer aos alunos ferramentas e métodos possam auxiliar logo do início da apresentação dos novos conceitos.

### 5 Computação Física como instrumento didático

Uma maneira de trazer um maior significado ao aluno é promovendo a diminuição da complexidade e da abstração dos conceitos envolvidos, criando meios para que o aprendiz possa fazer correlações com o mundo real, analogias com o cotidiano,

interagindo com objetos reais. A Computação Física (CF) tem como objetivo proporcionar a conexão entre o mundo virtual e o real, por meio de interfaces intuitivas e programáveis entre objetos e seres humanos [21]. Segundo Stankovic et al. [25], a CF integra a computação com o mundo físico utilizando componentes eletrônicos, como sensores e atuadores, por meio de sistemas embarcados, comunicando-se por redes de computadores e criando ambientes automatizados. A Figura 1 ilustra um exemplo simples de um circuito contendo um LED, sendo controlado por uma placa Arduino.

O ambiente tangível da CF proporciona meios de diminuir as dificuldades de aprendizado, principalmente dos iniciantes em programação. Como mostra o trabalho de Zanetti e Oliveira [31], a utilização de um robô móvel e uma arena de obstáculos ajudou a compreensão em conceitos básicos de programação, como estruturas condicionais (*if-else*) e de repetição (*for* e *while*). Anfurrutia et al. [1], também apresentam um trabalho usando Robótica Pedagógica em ensino básico de programação e programação visual, utilizando a linguagem de programação Java e conceitos fundamentais da POO, com resultados positivos no estímulo a resolução de problemas e construção de algoritmos,

Jang, Kim e Lee [13] apresentam um experimento, utilizando a plataforma Arduino, para ensinar o paradigma da POO, por meio de programação de pequenos circuitos com componentes eletrônicos, como LEDs (Light Emitting Diode ou Diodo Emissor de Luz), botões e sensor de luminosidade, fazendo analogias entre os objetos reais e as classes desenvolvidas utilizando linguagem de programação. Kafai et al. [14] apresenta uma pesquisa em que usa componentes eletrônicos em vestuário (Tecnologia Vestível ou Wearable Technology) para ensinar programação, mostrando projetos de vestimentas, representado suas características e interagindo com elas por meio de uma linguagem visual (Scratch), fazendo uma transição entre o concreto (contendo uma placa Arduino) e linguagem de programação.

A POO tem, como princípio fundamental, a racionalização e construção do software por meio de uma analogia a objetos. Usando componentes eletrônicos, é possível fazer um paralelo direto entre o objeto real e sua “representação” virtual (o código-fonte que o descreve). Por exemplo, ao se criar uma classe LED, pode-se definir seus atributos e métodos com base em sua constituição real e nas ações que se pode fazer com o componente.

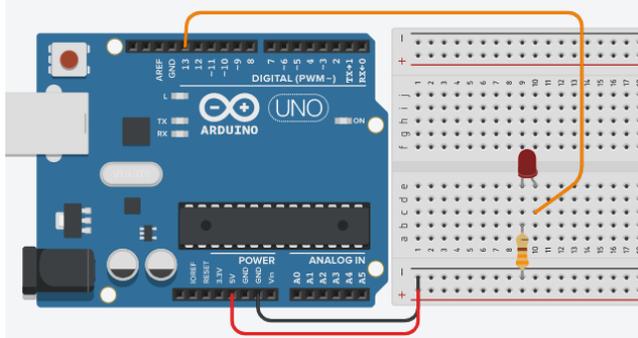


Figura 1: Circuito simples com Arduino e LED

Uma possível descrição de classe para o controle do LED presente no circuito da Figura 1 está apresentada na Figura 2. Os

atributos cor, pino (pino que está conectado ao Arduino) e estado (ligado/desligado) representam valores que representam características reais do objeto junto ao sistema. Os métodos acender(), apagar() e piscar() representam ações reais que podem ser feitas com o LED.

LED
- Pino: int - Cor: String - Estado: boolean
- Acender() - Apagar() - Piscar()

Figura 2: Exemplo de uma classe LED

O uso de CF em educação tem seu primeiro registro nos anos 1970, tendo sido usado para explorar conceitos de Inteligência Artificial, pela exploração de um mundo físico e tangível [21] [23]. Os recursos da CF oferecem aos alunos várias maneiras de aprender computação (software, hardware, comunicação em rede e interação humano-computador), além de proporcionar práticas mais dinâmicas e motivadoras [22].

## 6 A teoria de Concreteness Fading e o ensino de programação

A teoria do *Concreteness Fading* (CoFa) de Bruner [6] propõe apoiar a transferência do conhecimento, facilitando as conexões entre as representações de conceitos a serem alcançados ao longo de uma progressão do mais concreto para o mais abstrato. Na CoFa, as representações concretas são utilizadas no primeiro momento, para que possa tornar o conceito mais acessível e tangível ao aluno, fornecendo uma conexão com o conhecimento prévio e com cenários realistas [12].

Além da transição do concreto para o abstrato, o uso de representações concretas pode servir como referência após a transição para as representações mais abstratas. Essas abstrações podem ser mais ambíguas ou simplificadas. Poder estar em contato com a representação concreta pode auxiliar na compreensão [8]. A transição de uma representação concreta para uma mais abstrata pode diminuir equívocos que poderiam surgir na análise do aluno se fosse baseada apenas na representação abstrata, pois com o objeto concreto, sua análise é mais direta e empírica [9].

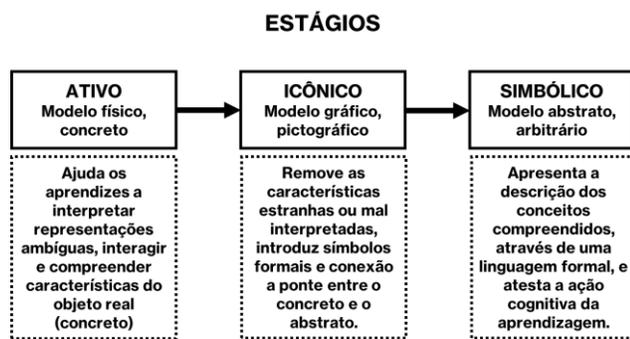
Uma das principais habilidades demandas do aluno iniciante em POO é a capacidade de abstração e a resolução de problemas. A CoFa pode proporcionar um meio de explorar o uso de representações concretas que sejam significativas aos alunos, promovendo uma maior compreensão sobre os conceitos aprendidos e dando mais autonomia na análise de problemas. Giraffa, Moraes e Uden [10] afirmam que, no processo de aprendizagem de programação, os alunos precisam imaginar e compreender termos abstratos que muitas vezes não possuem equivalentes na vida real. Wrenn e Krishnamurthi [29] utilizam em sua pesquisa ambientes de programação que mostram, em tempo

real (por meio de uso de gráficos e animações), o que ocorre durante a execução de um código, criando um meio de relacionar o modelo mais abstrato de um programa com um modelo mais real e concreto.

No trabalho de Suh, Lee e Xia [27] é discutido que a utilização da teoria de CoFa no ensino de programação pode ser efetiva, ajudando na compreensão de conceitos abstratos, desde fundamentos de lógica como variáveis e estruturas condicionais, até conceitos mais complexos, como recursividade e algoritmos de ordenação, que são comumente abordados em aulas relacionadas à programação.

A proximidade com o construto real pode estimular a independência e confiança do aluno em explorar o objeto conseguindo, desta forma, ter maior domínio sobre o problema ou conceito. Trory, Howland e Good [28] realizaram um experimento com crianças entre 9 e 10 anos para ensinar conceitos fundamentais para algoritmos de problema de caminho mínimo, utilizando uma maquete com linhas dispostas em localidades diferentes e arestas indicando as distâncias entre elas. É relatado que a presença de um cenário “real” análogo a um grafo possibilitou que crianças pudessem compreender a aplicação desses algoritmos, considerados complexos até mesmo dentro de cursos de Computação.

Ambos os experimentos supracitados, utilizam a abordagem do *framework* de 3 estágios definida por Bruner [6], na adoção do CoFa: ativo (*enactive*), que proporciona a interação com o ambiente; icônico (*iconic*), que utiliza uma representação mais visual e pictográfica, e; simbólico (*symbolic*), que utiliza uma linguagem e a apresentação abstrata. Fyfe et al. (2014) apresentam um modelo teórico da CoFa, definindo ações básicas que devem ocorrer em cada estágio. A Figura 3 apresenta o modelo resumido e adaptado do original.



**Figura 3: Modelo teórico do *Concreteness Fading* (adaptado de Fyfe et al. [8])**

O trabalho de Suh, Lee e Law [26] apresenta uma revisão sistemática da literatura, na qual umas das questões que fundamentam a pesquisa está relacionada a adoção do *framework* de Bruner [6] e suas derivações, indicando que a adoção dos três estágios pode proporcionar melhores resultados. Este *framework* pode ser relacionado ao ensino de POO. Por exemplo, o estágio de apresentação do objeto concreto poderia discutir e apresentar todas as características e ações possíveis que aquele objeto pode fazer, possibilitando ao aluno interagir com o objeto (estágio ativo). Em um segundo momento, pode-se levar esse levantamento preliminar das características do objeto concreto para o desenvolvimento de modelos gráficos, como um diagrama de classes (estágio icônico).

Por fim, após o desenvolvimento desses modelos gráficos, poderia ser iniciado o trabalho em código-fonte, utilizando uma linguagem de programação (estágio simbólico).

## 7 Computação Física e Concreteness Fading para promover a Aprendizagem Significativa

Recursos da CF são baseados em objetos tangíveis e interação com usuário, por meio da utilização de componentes e plataformas eletrônicas, podendo ser ferramentas a ser utilizadas como representações concretas. Usar construtos interativos e programáveis tem a potencialidade de criar atividades que sejam dinâmicas e motivadoras e que possam apoiar a transição à conceitos mais abstratos, como afirma o trabalho de Przybilla et al. [21]. Nesse mesmo trabalho são apresentadas as etapas da construção de uma prática envolvendo CF:

1. Configuração: preparar componentes de hardware, ferramentas e demais materiais;
2. Introdução e Inspiração: introduzir os conceitos fundamentais do hardware, apresentar o tema da prática e os conceitos básicos que serão usados para a programação;
3. Criação: formar grupos, incentivar brainstorms entre os alunos e idealizar projetos, com o professor apresentando sugestões sobre o hardware e programação;
4. Compartilhar e Refletir: apresentar os projetos elaborados, estimular os participantes a trocar experiências e relatar erros e acertos durante a realização da prática.

As etapas descritas acima podem ser adotadas em uma aplicação do primeiro estágio da CoFa (estágio ativo), para que o aprendiz tenha o contato com representação concreta. Em uma aula de introdução a POO, componentes da CF poderiam ser materiais de estudo para definição de conceitos como objetos e seus atributos e métodos. A facilidade em interagir com esses artefatos eletrônicos e robóticos pode colaborar com a exploração e motivação, despertando o interesse do aluno, como corrobora os estudos realizados por Qiu et al. [24] e Przybilla [22].

Atualmente, existem plataformas acessíveis que facilitam o controle e programação de componentes eletrônicos, como as placas de prototipagem Arduino e Raspberry Pi. Um professor pode criar experimentos que se apoiem com construtos reais e concretos, podendo evoluir a apresentação dos conceitos para o meio mais abstrato. Com o uso da programação é possível criar uma maneira na qual o aluno veja os dois modelos (concreto e abstrato) juntos, com a transição sendo feita de imediato.

Em uma proposta didática, por exemplo, poderia ser criado um pequeno circuito no qual o professor poderia debater e expor as características do objeto, acendendo e apagando o LED, mostrando quais são seus atributos, como cor, estado (ligado/desligado) entre outros. Após isso, o professor apresenta um modelo mais abstrato, o diagrama de classe, referenciando o que foi apresentado na etapa anterior. Por fim, são usados recursos de uma linguagem de programação para transcrever e formalizar o modelo mais abstrato, que surge nessa transição gradual.

Considerando as premissas da AS, a influência do conhecimento prévio do aluno, o recurso de subçunçores e a interação com novos conhecimentos para formar a estrutura cognitiva do indivíduo, unir os recursos da CF e a estratégia proposta pela CoFa pode proporcionar um método eficiente para

promover a AS no ensino de POO. As barreiras que surgem para os aprendizes de POO, como a necessidade de abstrair conceitos e entender a relação entre objetos em código (como em um sistema com um ou mais objetos interagindo), podem ser desmotivadores para o aluno [10]. A analogia com componentes eletrônicos interagindo em um cenário real pode ser melhor compreendida pelo aluno.

Um exemplo do uso de analogias, seria um pequeno circuito como o apresentado na Figura 4, com um LED e um botão (tipo *push button*). A partir desse circuito, é possível sugerir como *subsunçores* a apresentação das características e ações que os componentes possuem, mostrando os componentes reais e uma apresentação oral. Em um segundo momento, pode-se apresentar alguma relação entre os dois componentes, como o acender do LED por meio da interação do botão. Mesmo que o aluno não tenha conhecimento em eletrônica, ele pode entender os princípios básicos de cada um dos objetos envolvidos e compreender que eles podem “conversar”, “trocar mensagem”, interagir. Essa percepção, quando for apresentada em modelos mais abstratos, pode ter uma retenção mais adequada e clara por parte do aluno, por já ter experimentado e compreendido essas relações em um cenário real. Em uma prática usando esse cenário, seria possível compreender todos os estágios definidos pela CoFa e ainda utilizar os conceitos fundamentais da AS.

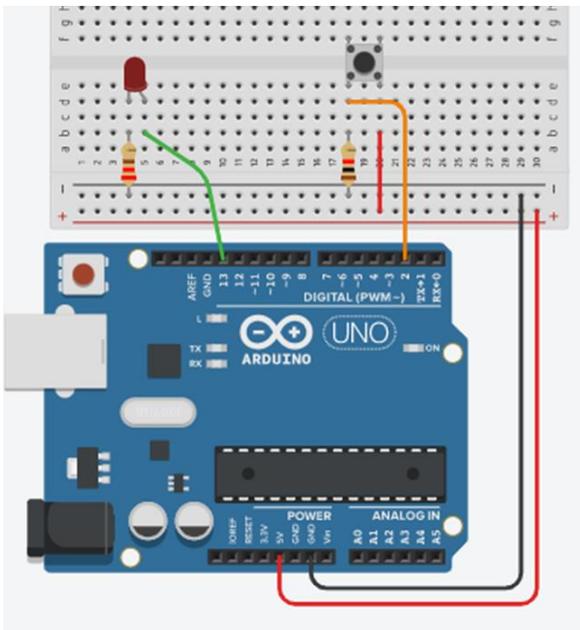


Figura 4: Exemplo de uma montagem com LED e botão

A Figura 5 apresenta um diagrama que ilustra um exemplo de prática e o processo de apresentação ao aluno. No estágio “ativo” pode-se abordar os *subsunçores* básicos (com os objetos reais) e interagir com um cenário real. No estágio icônico, podem ser discutidos os conceitos em diagramas e representações mais pictográficas. Por fim, no estágio simbólico, é introduzida a linguagem de programação e as estruturas básicas para o desenvolvimento do código-fonte.

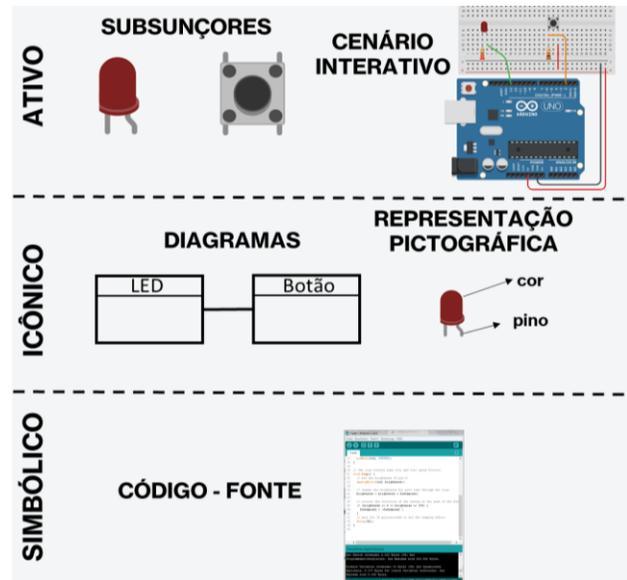


Figura 5: A relação dos estágios do *Concreteness Fading* e artefatos didáticos

## 8 Considerações finais

Há muitos desafios que envolvem o ensino de POO, principalmente aos iniciantes na área de Computação, com várias pesquisas sendo feitas sobre o tema. São discutidos desde aspectos motivacionais, até capacidade de abstração e construção cognitiva. Promover a AS em um cenário de ensino de programação pode ser adequado, se bem planejado e com recursos que possam incentivar o aluno a interagir com novos conhecimentos.

O objetivo deste artigo foi discutir temas relacionados à AS e sua aplicação no contexto de ensino de POO. O trabalho apresenta em quais aspectos a teoria de David Ausubel poderia ser adequada para fornecer um amparo metodológico e instrumental de como apresentar conceitos de programação. O trabalho também discute como potencializar a motivação do aluno diante de uma solução que pode trazer conceitos muito abstratos, por meio de uma correlação mais lógica e alcançável, baseada em conhecimentos pré-existentes.

Há desafios que devem ser superados na aplicação da AS. Os educadores e pesquisadores devem buscar novas soluções que permitam integrar os conceitos fundamentais da teoria da AS em suas aulas, seguindo o conteúdo didático para a formação do aprendiz. Recursos como os organizadores prévios e os mapas conceituais, podem ser instrumentos eficazes para estruturar um ambiente favorável para essa integração.

Cabe ao professor buscar a maneira mais adequada à uma aprendizagem mais significativa. Por exemplo, os materiais apresentados para discutir algum *subsunçor* podem ter um caráter menos abstrato, do que, por exemplo, quando apresentados diretamente em uma linguagem de programação (código-fonte). Essa prática não precisa ser feita de maneira isolada, podendo agregar outros métodos de ensino como CoFa, e utilizando recursos tecnológicos mais dinâmicos e interativos, como artefatos da CF.

As discussões apresentadas neste texto objetivam mostrar que a CF e a CoFa podem auxiliar na adoção da AS, oferecendo um cenário didático atraente ao aluno, seguindo as principais premissas definidas pela AS. A união dessas 3 linhas pode gerar instrumentos interessantes e efetivos para a dinâmica de ensino de POO, ensino que demanda meios de estimular e apoiar a capacidade de abstração dos alunos.

Baseado no framework de Bruner [6], os recursos da CF podem ser efetivamente adotados como instrumentos de criação dos objetos concretos necessários no início do processo (estágio ativo). Além disso, provêm recursos interativos e várias plataformas que facilitam sua programação e implementação de pequenos circuitos ou projetos.

Explorar o paradigma Orientado a Objetos diretamente em um modelo muito abstrato (como a linguagem de programação), pode ser frustrante ao aluno iniciante. Adotar um processo que se apoie na CoFa permite uma elaboração de uma prática pedagógica relevante e estruturada em etapas em que o objetivo seja a gradual compreensão de conceitos. O exercício de relacionar objetos reais, com suas características físicas e ações pode ser um suporte interessante para que se possa levar esses conceitos aos níveis mais abstratos.

## REFERÊNCIAS

- [1] Anfurrutia, F. I. et al. (2016). Incorporating educational robots and visual programming environments in introductory programming courses. In *2016 International Symposium on Computers in Education (SIIE)* (pp. 1-4). IEEE. DOI: 10.1109/SIIE.2016.7751835.
- [2] Astolfi, G., Junior, D. L. (2015). Investigação sobre conhecimentos prévios de alunos do curso Técnico em Informática a partir da aplicação de organizadores prévios. *Aprendizagem Significativa*. In *Revista/Meaningful Learning Review*, pp. 15-28.
- [3] Astolfi, G., Junior, D. L. (2016). Ensino de linguagem de programação com ênfase na aprendizagem significativa. In *Anais do XXIV Workshop sobre Educação em Computação* (pp. 111-120). SBC. DOI: <http://dx.doi.org/10.5753/wei.2016.9654>.
- [4] Ausubel, D. P. (2003). *Aquisição e retenção de conhecimentos: uma perspectiva cognitiva*. Lisboa: Plátano, 1.
- [5] Butler, M., e Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. *Proceedings ascilite Singapore*, (99-107).
- [6] Bruner, J. S. (1966). *Toward a theory of instruction*. Cambridge, MA: Belknap Press of Harvard University Press.
- [7] Darroz, L. M. (2018). Aprendizagem significativa: a teoria de David Ausubel. *Revista Espaço Pedagógico*, 25(2), 576-580. DOI: <http://dx.doi.org/10.5335/rep.v25i2.8180>
- [8] Fyfe, E. R., McNeil, N. M., Son, J. Y., e Goldstone, R. L. (2014). Concreteness fading in mathematics and science instruction: A systematic review. *Educational psychology review*, 26(1), 9-25. DOI: <http://dx.doi.org/10.1007/s10648-014-9249-3>
- [9] Gentner, D., e Smith, L. A. (2013). *Analogical learning and reasoning*. Oxford University Press. DOI: <http://dx.doi.org/10.1093/oxfordhb/9780195376746.013.0042>
- [10] Giraffa, L. M., Moraes, M. C., e Uden, L. (2014). Teaching object-oriented programming in first-year undergraduate courses supported by virtual classrooms. In *The 2nd International Workshop on Learning Technology for Education in Cloud* (pp. 15-26). Springer, Dordrecht. DOI: [http://dx.doi.org/10.1007/978-94-007-7308-0\\_2](http://dx.doi.org/10.1007/978-94-007-7308-0_2).
- [11] Hadar, I. (2013). When intuition and logic clash: The case of the object-oriented paradigm. *Science of Computer Programming*, 78(9), 1407-1426. DOI: <http://dx.doi.org/10.1016/j.scico.2012.10.006>.
- [12] Jaakkola, T., e Veermans, K. (2020). Learning electric circuit principles in a simulation environment with a single representation versus “concreteness fading” through multiple representations. *Computers & Education*, 148. DOI: <https://doi.org/10.1016/j.compedu.2020.103811>.
- [13] Jang, Y.; Lee, W.; Kim, J. (2015). Assessing the usefulness of object based programming education using Arduino. *Indian Journal of Science and Technology*, 8(S1), 89-96. DOI: 10.17485/ijst/2015/v8iS1/57701.
- [14] Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., e Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1-20. DOI: <http://dx.doi.org/10.1145/2576874>.
- [15] Moreira, M. A. (2003). Linguagem e aprendizagem significativa. In *Conferência de encerramento do IV Encontro Internacional sobre Aprendizagem Significativa*, Maragogi, AL, Brasil (Vol. 8).
- [16] Moreira, M. A. (2006). Aprendizagem Significativa: da visão clássica à visão crítica (Meaningful learning: from the classical to the critical view). In *Conferência de encerramento do V Encontro Internacional sobre Aprendizagem Significativa*, Madrid, Espanha.
- [17] Moreira, M. A. (2010). O que é afinal aprendizagem significativa? Aula Inaugural do Programa de Pós-Graduação em Ensino de Ciências Naturais, Instituto de Física, Universidade Federal do Mato Grosso, Cuiabá, MT.
- [18] Novak, J. D. e Gowin, D. B. (1995). *Aprendendo a aprender*. Lisboa: Plátano Edições Técnicas.
- [19] Novak, J. D. (2002). Meaningful learning: The essential factor for conceptual change In *limited or inappropriate propositional hierarchies leading to empowerment of learners*. *Science education*, 86(4), 548-571.
- [20] Pimentel, E., Omar, N. (2008). Ensino de algoritmos baseado na aprendizagem significativa utilizando o ambiente de avaliação Netedu. SBC, 79.
- [21] Przybylla, M.; Henning, F.; Schreiber, C.; Romeike, R. (2017). Teachers' Expectations and Experience in Physical Computing. In *International Conference on Informatics in Schools: Situations, Evolution, and Perspectives* (pp. 49-61). Springer, Cham. DOI: [http://dx.doi.org/10.1007/978-3-319-71483-7\\_5](http://dx.doi.org/10.1007/978-3-319-71483-7_5).
- [22] Przybylla, M. (2018). From embedded systems to physical computing: Challenges of the “digital world” in secondary computer science education. Tese de doutorado. Universität Potsdam, Postdam, Alemanha.
- [23] Perlman, R. (1974). TORTIS (Toddler's Own Recursive Turtle Interpreter System). Massachusetts Institute of Technology, LOGO-9, v. 11, 1974.
- [24] Qiu, K., Buechley, L., Baafi, E., e Dubow, W. (2013). A curriculum for teaching computer science through computational textiles. In *Proceedings of the 12th international conference on interaction design and children* (pp. 20-27). DOI: <http://dx.doi.org/10.1145/2485760.2485787>.
- [25] Stankovic, J. A. et al. (2005). Opportunities and obligations for physical computing systems. *Computer*, 38(11), 23-31. DOI: 10.1109/MC.2005.386.
- [26] Suh, S., Lee, M., e Law, E. (2020). How do we design for concreteness fading? survey, general framework, and design dimensions. In *Proceedings of the Interaction Design and Children Conference* (pp. 581-588).
- [27] Suh, S., Lee, M., e Xia, G. (2020). Coding strip: A pedagogical tool for teaching and learning programming concepts through comics. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 1-10). IEEE.
- [28] Trory, A., Howland, K., & Good, J. (2018, June). Designing for concreteness fading in primary computing. In *Proceedings of the 17th ACM Conference on Interaction Design and Children* (pp. 278-288). DOI: <http://dx.doi.org/10.1145/3202185.3202748>.
- [29] Wrenn, J., e Krishnamurthi, S. (2019, July). Executable examples for programming problem comprehension. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 131-139). DOI: <http://dx.doi.org/10.1145/3291279.3339416>.
- [30] Xinogalos, S. (2015). Object-oriented design and programming: an investigation of novices' conceptions on objects and classes. *ACM Transactions on Computing Education (TOCE)*, 15(3), 1-21. DOI: <http://dx.doi.org/10.1145/2700519>.
- [31] Zanetti, H., e Oliveira, C. (2015). Práticas de ensino de programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 4, No. 1, p. 1236). DOI: <http://dx.doi.org/10.5753/cbie.wcbie.2015.1236>.