

Metodologia de resolução de problemas utilizando a linguagem LiVE*

Adriana Bordini¹, Simone André da Costa Cavalheiro¹,
Luciana Foss¹

¹Programa de Pós-Graduação em Computação – Universidade Federal de Pelotas (UFPel)
96.010-610 – Pelotas – RS – Brasil

{adriana.bordini, simone.costa, lfoss}@inf.ufpel.edu.br

Abstract. *Critical thinking, creativity, and problem-solving skills are considered essential to twenty-first century professionals. Computational Thinking is a reasoning process based on Computer Science capable of promoting such skills. Based on the pillars of abstraction, automation and analysis, several works propose the dissemination of this methodology to students of different levels of education. Differentiating from previous works, this paper presents methodological proposals for solving problems that systematize the use of techniques, such as composition and abstraction, for the description of solutions. The methodologies are designed for the LiVE language, a visual specification language based on concepts of Computational Thinking.*

Resumo. *Pensamento crítico, criatividade e capacidade de resolver problemas são competências consideradas essenciais aos profissionais do século XXI. O Pensamento Computacional é um processo de pensamento fundamentado na Ciência Computação capaz de promover tais habilidades. Alicerçados nos pilares da abstração, automação e análise, diversos trabalhos propõem a disseminação desta metodologia para estudantes de diferentes níveis de ensino. Diferenciando-se de trabalhos anteriores, este artigo apresenta propostas metodológicas para a resolução de problemas que sistematizam o uso de técnicas, como composição e abstração, para a descrição de soluções. As metodologias são concebidas para a linguagem LiVE, uma linguagem visual de especificação fundamentada em conceitos do Pensamento Computacional.*

1. Introdução

O Pensamento Computacional (PC) "...é um processo de pensamento envolvido na formulação de um problema e na expressão de sua solução de forma que um computador - humano ou máquina - possa efetivamente realizar" [Wing 2014]. O PC envolve habilidades, dimensões e conceitos [CSTA 2017] que estão diretamente relacionados com competências consideradas essenciais para os profissionais do século XXI, tais como: pensamento crítico, resolução de problemas, comunicação, colaboração, trabalho em equipe, alfabetização tecnológica, iniciativa, entre outros [P21.org 2018, Experience 2018].

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. O projeto também conta com o apoio da PREC e PRPPG / UFPel.

O PC é fundamentado em três pilares [Repenning et al. 2016]: abstração, automação e análise. Essas etapas são geralmente adotadas (iterativamente) por um profissional da computação para solução de problemas. Inicialmente, através de uma descrição verbal ou visual estabelece-se uma representação da realidade que descreve a formulação do problema; segundo, busca-se definir uma solução, preferencialmente que seja automatizada; e então executa-se e avalia-se a solução encontrada. Este processo, em geral, envolve a aplicação de uma série de técnicas da computação (como decomposição, generalização, composição, abstração, pensamento algorítmico,...) que levam ao "sucesso do processo".

Muitos trabalhos procuram desenvolver essas técnicas por meio da programação, robótica, computação desplugada, dentre outros [Cunha et al. 2018, Guarda et al. 2018, Melo et al. 2018, Costella et al. 2017, Oliveira et al. 2016], com alunos de diferentes níveis e áreas. Mas, na maioria das vezes, fica difícil identificar quais habilidades foram efetivamente desenvolvidas e o quanto os alunos conseguem reaplicar as técnicas e/ou conceitos aprendidos na resolução de outros problemas.

Diferentemente do que já foi proposto, este trabalho visa estabelecer metodologias que sistematizem a aplicação das técnicas do PC para a resolução de problemas. Para isto definiu-se uma linguagem de especificação visual com o objetivo de guiar os estudantes a construírem as suas próprias soluções. Optou-se por uma abordagem visual, por esta ser uma representação mais próxima do dia a dia dos estudantes.

De forma a sistematizar a proposição de soluções, são propostas metodologias de uso da linguagem, as quais fundamentam-se nas técnicas de decomposição, composição, abstração e generalização. Ao utilizar uma das metodologias propostas, já se está aplicando pelo menos uma destas técnicas no processo de solução. Em trabalho anterior [Bordini et al. 2018], foi apresentada e introduzida a linguagem visual. Neste artigo, a linguagem foi estendida para incluir construções condicionais e foram propostas as metodologias de solução de problemas que focam nas técnicas de composição e abstração.

O texto está organizado como segue. A Seção 2 apresenta a linguagem LiVE e as metodologias propostas. Na Seção 3 descreve-se um exemplo de aplicação e na Seção 4 detalha-se um estudo de caso. As considerações finais são delineadas na seção 5.

2. Linguagem Visual LiVE para o desenvolvimento do PC

Nesta seção descreve-se brevemente a linguagem de especificação visual LiVE bem como propõe metodologias de uso desta linguagem. LiVE foi concebida para ser aplicada em qualquer área do conhecimento, podendo ser utilizada por não especialistas em computação. Soluções na linguagem podem ser descritas em um alto nível de abstração e a especificação de uma solução foca na técnica de construção em si (como composição, decomposição, abstração ou generalização) e não necessariamente em uma implementação.

A linguagem LiVE foi inspirada no diagrama de atividades UML [Pressman 2011], onde são representados os fluxos conduzidos pelas ações que descrevem a solução de um problema. Diferenciando-se do diagrama de atividades, o fluxo de uma solução é baseado em dados, os quais podem ser refinados (detalhados) ou abstraídos. Ademais, além de uma representação intuitiva, LiVE não tem como foco o desenvolvimento de projetos, tampouco a área da Computação de forma exclusiva, e sim a descrição de soluções dos mais diversos problemas relacionados a qualquer área.

As metodologias propostas na subseção 2.2 focam na composição e abstração. A primeira objetiva especificar a solução de um problema a partir da composição de ações já conhecidas. A segunda, objetiva-se identificar composições de ações que definem uma ação mais abstrata, permitindo que uma solução mais simples e intuitiva seja estabelecida.

2.1. Sintaxe

Uma especificação na linguagem visual descreve a solução de um problema. Ela inclui um conjunto de componentes elementares que apresenta de forma abstrata todas as ações que podem ser realizadas. Um destes componentes é o principal, o qual descreve a ação que soluciona o problema como um todo.

Um Componente Elementar (CE), ilustrado na Figura 1(a), descreve uma ação a ser executada sem detalhá-la, apenas especificando os tipos das informações de entrada necessárias para sua execução, bem como os tipos dos resultados obtidos. Ele é definido por um nome, descrevendo a ação (representada por um retângulo), um conjunto de portas de entrada e um conjunto de portas de saída (quadrados pretos nas bordas superior e inferior dos componentes, respectivamente). As portas de entrada/saída são os meios pelos quais as informações/resultados são recebidos/enviados pelo CE. Por sua vez, cada porta possui um tipo que restringe o tipo de elementos que pode ser associado a ela.

Caso sejam necessários mais detalhes, um CE pode ser associado a um Componente Decomposto (CD) que especifica tais detalhes. Um CD, ilustrado na Figura 1(b), é detalhado por um conjunto de instâncias de CE, podendo também incluir conjuntos de nós condicionais, nós de refinamento e nós de abstração. Estes elementos são ligados por conexões entre suas portas, estabelecendo caminhos por onde os dados circulam. Conexões só podem ser estabelecidas entre portas de mesmo tipo.

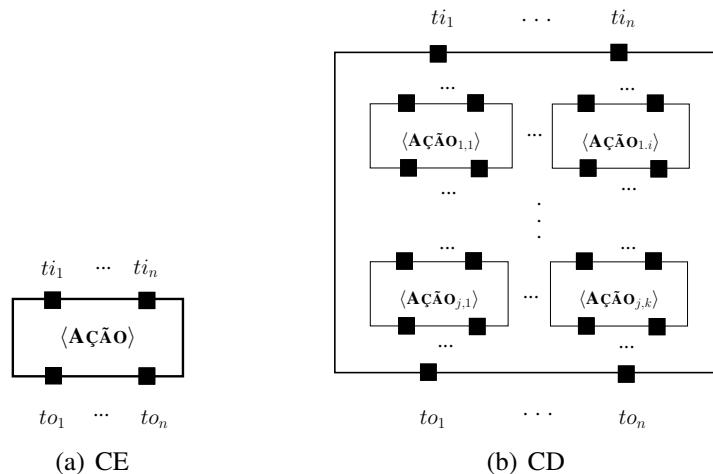


Figura 1. Componente Elementar e Componente Decomposto

Um nó condicional descreve uma escolha entre dois possíveis caminhos, pelos quais o dado pode seguir. Estes caminhos são delimitados por um nó de decisão (no início) e um nó de junção (no final), conforme as Figuras 2(a) e 2(b), respectivamente. Ao chegar em um nó de decisão, o dado é enviado por um dos caminhos, determinado por um valor booleano recebido (resultante de um teste). E ao chegar em um nó de junção, o dado chegará por um dos dois caminhos e seguirá o seu fluxo.

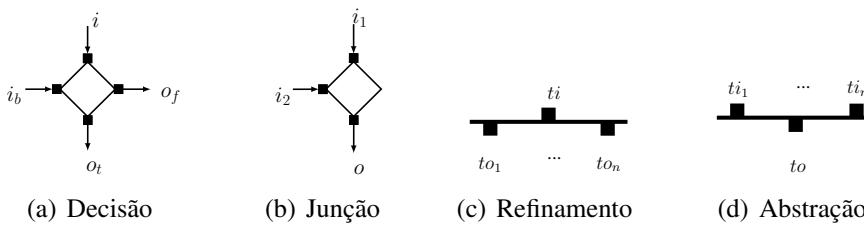


Figura 2. Detalhamento dos Nós de Decisão, Junção, Refinamento e Abstração

Além disso, as informações de entrada ou saída do componente decomposto também podem ser mais detalhadas. Tais detalhes são descritos por nós de refinamento e de abstração (representadas por barras horizontais), os quais têm a função de detalhar e abstraírem os dados, conforme as Figuras 2(c) e 2(d), respectivamente. Um nó de refinamento permite que um tipo de dado mais abstrato (t_i) seja detalhado em um conjunto de tipos mais concretos ($t_{o_1} \dots t_{o_n}$). Um nó de abstração permite que um conjunto de tipos mais concretos ($t_{i_1} \dots t_{i_n}$) sejam abstraídos num tipo mais alto nível (t_o).

2.2. Metodologias de resolução de problemas utilizando a linguagem LiVE

Para o uso da linguagem visual LiVE, desenvolveram-se metodologias de resolução de problemas fundamentadas nas técnicas da composição e da abstração.

Na **metodologia da composição**, quer-se descrever a solução de um problema a partir da composição de componentes (ações) elementares. A solução do problema é dada por uma especificação, obtida por meio da construção de um componente decomposto a partir de instâncias de CE e conexões entre esses componentes. Dado um conjunto de CEs, o processo de solução deve seguir as seguintes etapas (sintetizado na Figura 3):

- C1 Adicionar um conjunto de instâncias de CE cujas entradas sejam independentes (isto é, cujas entradas não são resultantes de nenhuma outra ação, e portanto não serão conectadas a nenhuma porta de saída).
- C2 Adicionar/selecionar componentes (instâncias de CE, nós condicionais, de refinamento ou de abstração) cujas as entradas se conectem com saídas dos componentes selecionados na etapa/iteração anterior, estabelecendo as conexões.
- C3 Repita a etapa C2 até que os resultados esperados para o conjunto selecionado em C1 sejam alcançados.
- C4 Repetir as etapas C1 e C2 até que todos os resultados esperados sejam alcançados.
- C5 Construir (estabelecer) um CD contendo todas as instâncias dos CE, nós condicionais e conexões definidos nas etapas anteriores. Cria-se para o CD: uma porta de entrada para cada entrada independente dos CE, estabelecendo as conexões entre tais entradas; e, uma porta de saída para cada saída desconectada dos CE, estabelecendo as conexões com tais saídas.
- C6 Define-se o CE cuja decomposição foi estabelecida nas etapas anteriores. Esse componente elementar terá as mesmas portas de entrada/saída do CD definido na etapa C5, porém não mostrará os detalhes da solução. Além disso, um nome deve ser escolhido para este CE que será o componente principal da especificação.

Na **metodologia da abstração**, objetiva-se simplificar a solução de um problema a partir da identificação de estruturas (conjuntos de componentes) que podem ser omitidas.

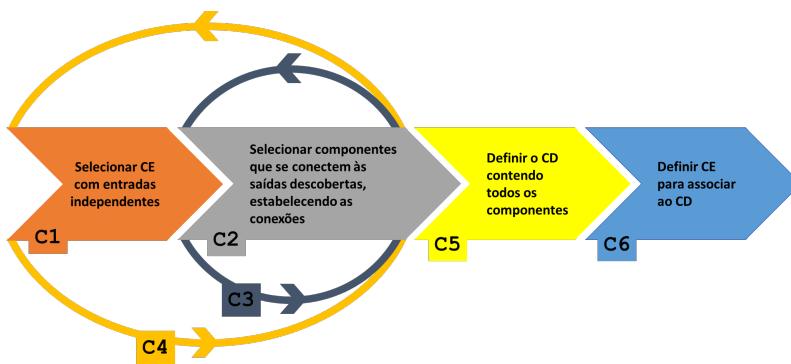


Figura 3. Síntese da Metodologia da Composição

Para isso, definem-se CEs que abstraem as estruturas selecionadas e substitui-se todas as ocorrências destas estruturas por suas respectivas abstrações. Dado um CD C de uma especificação S, o processo de abstração deve seguir as etapas (sintetizadas na Figura 4):

- A1 Selecionar dois ou mais componentes (instancias de CE, nós condicionais, de refinamento ou de abstração) de C que serão abstraídos. Esta seleção deve respeitar as seguintes restrições: (i) Para cada nó condicional selecionado, todos os componentes que estão nos caminhos que partem do nó de decisão e chegam no nó de junção devem ser selecionados; (ii) Para quaisquer dois componentes selecionados, para os quais existe um caminho que os conecta, todos os componentes deste caminho, também devem ser selecionados; (iii) Pelo menos um componente elementar de C não deve fazer parte da seleção.
- A2 Identifique as portas de entrada/saída externas (portas de entrada/saída que não estão conectadas com portas de componentes selecionados) ao conjunto de componentes selecionados em A1.
- A3 Defina o componente elementar E que abstrai os selecionados, criando uma porta de entrada/saída para cada porta externa identificada em A2, respeitando os tipos.
- A4 Adicione E em S e uma instância de E em C. "Altere" o destino/origem de toda conexão ligada a uma porta de entrada/saída externa, redefinindo-o/redefinindo-a para a respectiva porta de entrada/saída de E.
- A5 Defina um CD D, transfira os componentes abstraídos e suas respectivas conexões internas (não ligadas a portas externas) de C para D, conecte as portas de entrada/saída do D (as mesmas portas definidas em E) com as respectivas portas externas dos componentes abstraídos, associando-o ao componente elementar E.
- A6 Verificar na especificação se algum CD contém a mesma estrutura abstraída. Em caso afirmativo, substitua todas as estruturas pelo CE abstrato.



Figura 4. Síntese da Metodologia da Abstração

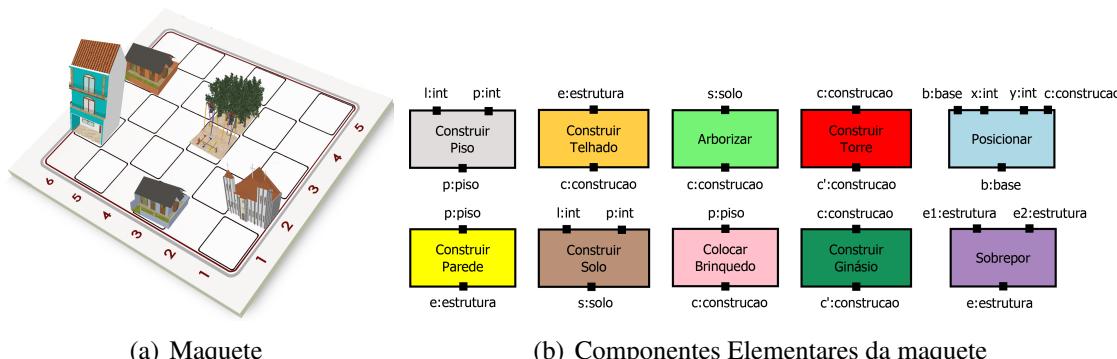


Figura 5. Exemplo para a Aplicação da Metodologia da Composição

3. Exemplo de aplicação

Para exemplificar a aplicação da metodologia proposta, descreve-se a solução do problema de especificar generalização da construção de praças.

A maquete é obtida a partir do posicionamento das seguintes construções em um tabuleiro base dado como entrada: duas casas, um edifício de 3 andares, uma igreja de duas torres, uma praça arborizada e uma praça com brinquedos. Os componentes elementares descrevem as ações que podem ser utilizadas na especificação da solução. Em particular, a ação *Construir Piso*, recebe duas entradas do tipo inteiro (int), indicando respectivamente largura e profundidade, e resulta numa saída do tipo piso. Já a ação *Construir Parede* recebe como entrada um piso e retorna uma estrutura.

Usando a metodologia da composição, a construção da solução pode ser obtida por meio das etapas a seguir (Figura 6). Cabe observar que existem diferentes maneiras de aplicar a metodologia para obter o mesmo resultado, dependendo da quantidade de CEs selecionados na primeira ocorrência de C1. Na apresentada, todas as ações independentes foram selecionadas já na primeira ocorrência, tornando desnecessária a sua repetição.

- C1 São adicionadas 7 instâncias de *Construir Piso* e uma instância de *Construir Solo*.

C2 São adicionadas: 6 instâncias de *Construir Parede*, que são conectadas às instâncias 1 à 6 de *Construir Piso*; uma instância de *Arborizar*, a qual é conectada a *Construir Solo*; e uma instância de *Colocar Brinquedo*, que é conectada a *Construir Piso*₇.

C3 A etapa C2 foi repetida 5 vezes (C3.1 à C3.5 na Figura 6). Na primeira repetição são adicionadas: as instâncias *Construir Telhado*₁, *Construir Telhado*₂ e *Construir Telhado*₃, conectadas a *Construir Parede*₁, *Construir Parede*₅ e *Construir Parede*₆, respectivamente; a instância *Sobrepor*₁ conectada às instâncias *Construir Parede*₂ e *Construir Parede*₃; a instância *Sobrepor*₂, conectada à instância *Construir Parede*₄; e as instâncias *Posicionar*₁ e *Posicionar*₂, conectadas às instâncias *Colocar Brinquedo* e *Arborizar*, respectivamente. Na segunda repetição são adicionadas: as instâncias *Posicionar*₃ e *Posicionar*₄, conectadas às instâncias *Construir Telhado*₁ e *Construir Telhado*₃, respectivamente; a instância *Construir Telhado*₄ conectada a *Sobrepor*₂; e a instância *Construir Torre*₁, conectada a *Construir Telhado*₂. Nesta repetição ainda são selecionadas algumas ações (adicionadas em etapas anteriores) para estabelecer as conexões (coloridas em vermelho na Figura 6) com as portas de saída que ainda estão desconectadas. As instâncias

selecionadas são: *Sobrepor*₂ que é conectada a *Sobrepor*₁; e *Posicionar*₂ que é conectada a *Posicionar*₁. Na terceira repetição foram adicionadas: a instância *Posicionar*₅, conectada a *Posicionar*₃ e a *Construir Telhado*₄; e a instância *Construir Torre*₂ conectada a *Construir Torre*₁. Nesta repetição é selecionada a instância *Posicionar*₁ que é conectada a *Posicionar*₄. Na quarta repetição é adicionada a instância *Posicionar*₆ que é conectada a *Posicionar*₅ e a *Construir Torre*₂. Na última repetição, é apenas selecionada a instância *Posicionar*₄ para conectar com a saída da instância *Posicionar*₆.

C4 Esta etapa não se aplica, pois todos os resultados são alcançados na etapa anterior.

C5 Constrói-se o CD conforme a Figura 7.

C6 Define-se o CE, denominado *Construir Maquete*, esquecendo toda a estrutura interna do CD apresentado na Figura 7.

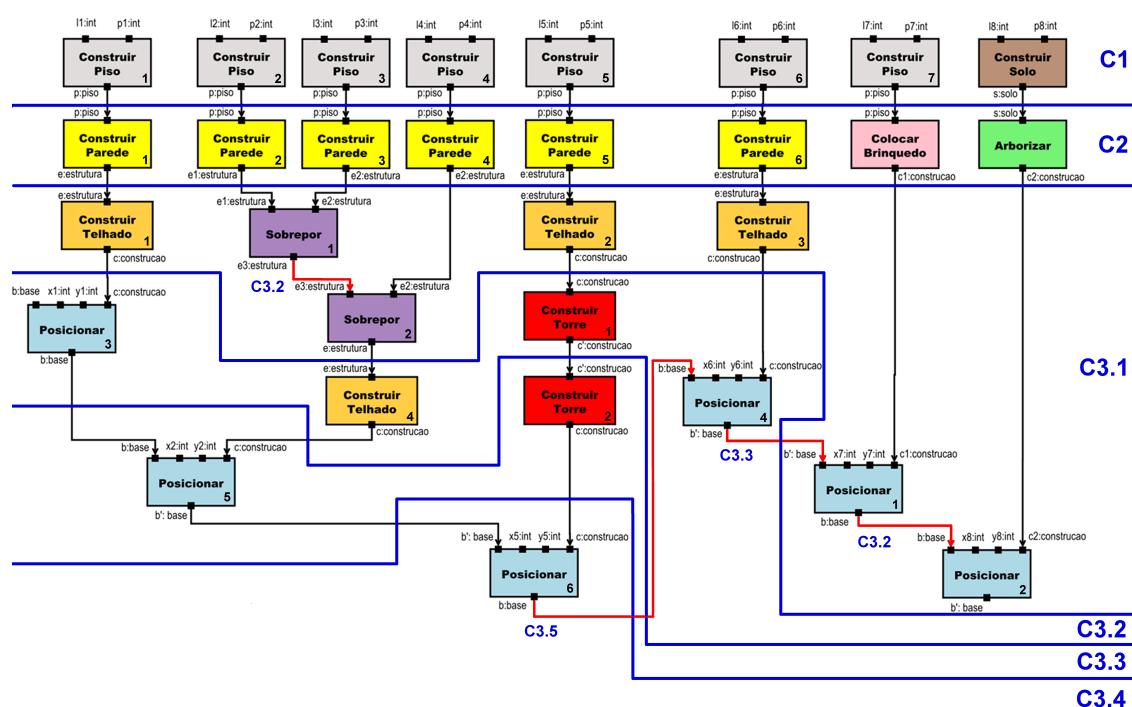


Figura 6. Exemplo: Composição Resultante das Etapas C1, C2 e C3

Agora, pode-se simplificar a solução do problema anterior, identificando-se conjuntos de componentes a serem abstraídos. Em particular, é possível definir as seguintes abstrações: *Construir Casa*, *Construir Edifício*, *Construir Igreja* e *Construir Praças*. A **metodologia da abstração** deve ser utilizada para obter cada uma das construções. Por exemplo, para abstrair a construção das praças, seguem-se as seguintes etapas:

- A1 Nesta etapa foram selecionados os componentes delimitados pelo retângulo pontilhado da Figura 7.
- A2 Nesta etapa foram identificadas as portas de entrada externas l7, p7, l8 e p8 e as portas de saída externas c1 e c2.
- A3 Cria-se o CE *Construir Praças* com as portas identificadas na etapa A2, conforme ilustrado na Figura 8(a).
- A4 Substitui-se a região pontilhada na Figura 7 pelo CE obtido na etapa A3.
- A5 Define-se o CD ilustrado na Figura 8(b) que contém a estrutura abstraída.
- A6 Não é aplicada, já que não tem outra ocorrência da construção de praças.

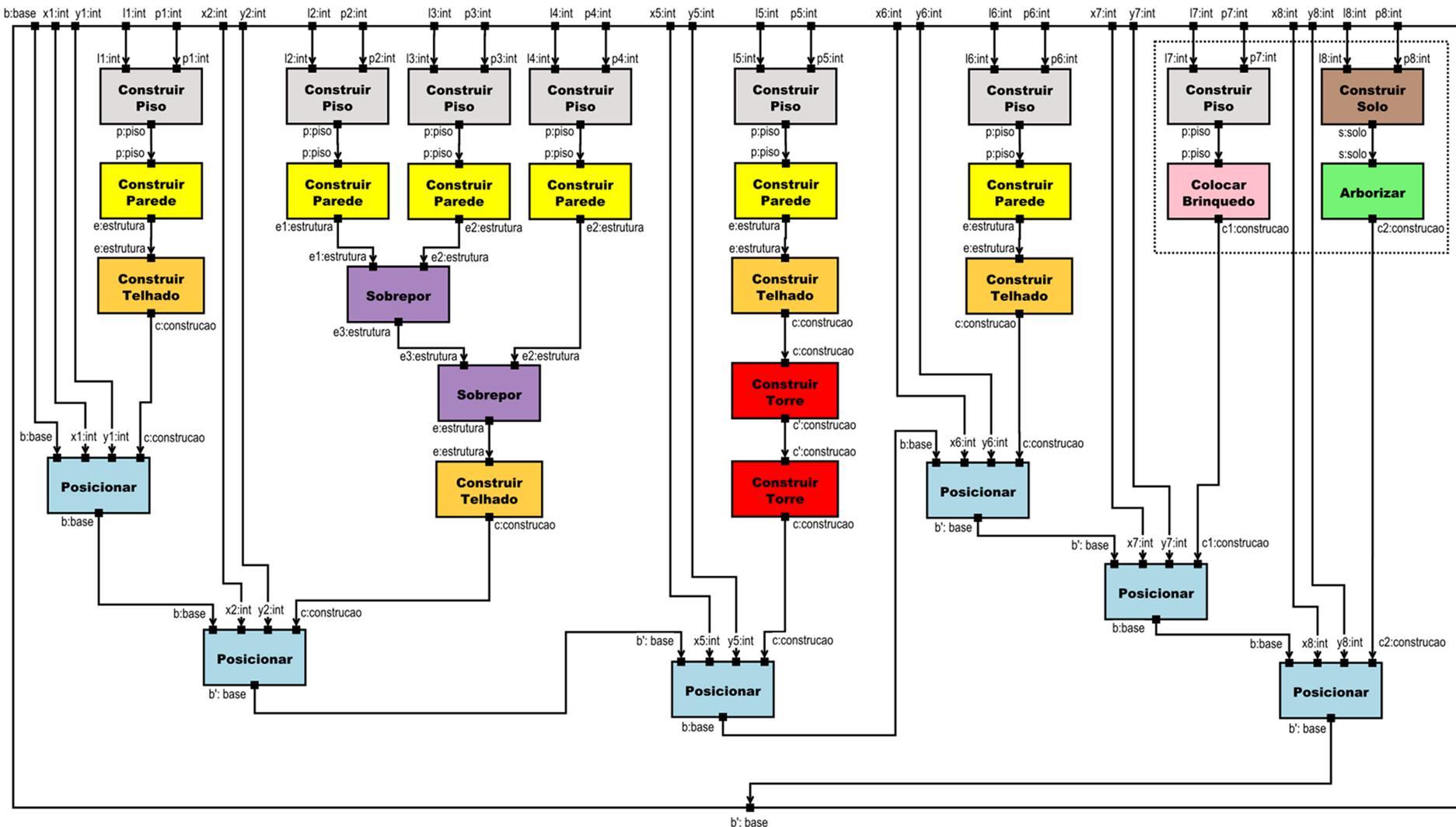


Figura 7. Exemplo: Componente Decomposto Obtido na Etapa C5

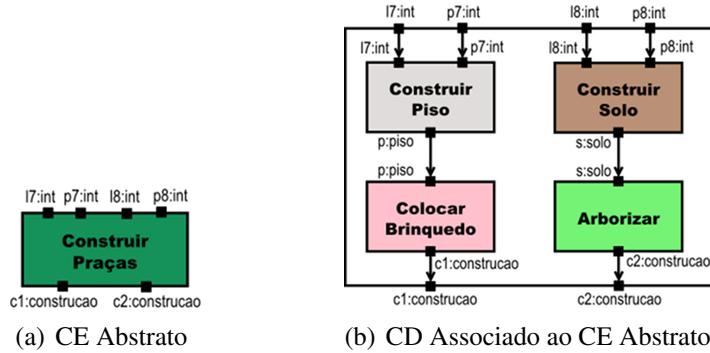


Figura 8. Exemplo de Utilização da Metodologia da Abstração

4. Estudo de caso

Um estudo de caso foi realizado com 28 alunos do primeiro semestre de um curso superior em Ciência da Computação. Foi solicitado que os alunos utilizassem o método da composição para resolver o problema detalhado na seção anterior. A turma foi dividida em 15 grupos, dos quais 13 duplas e 2 individuais. Cada grupo recebeu um kit, contendo: uma folha de papel tamanho 42cm × 89cm, onde deveria ser construída a solução, uma folha com a síntese da metodologia e a maquete, um tabuleiro com os CEs impressos e 21 instâncias de cada um dos CEs. A atividade teve duração de 2h30min. A solução deveria ser dada a partir da colagem das instâncias dos CEs na folha de solução, seguido pela identificação das portas de entrada, saída e conexões. Além disso, solicitou-se que os grupos identificassem em cada CE, em qual etapa da metodologia ele foi adicionado.

Ao final foi aplicado um questionário individual, disponível em <https://bit.ly/2FTBHzA>, com 7 questões objetivas e 3 expositivas, que tinham por objetivo identificar a impressão dos alunos sobre o uso da linguagem e aplicação da metodologia. A grande maioria dos alunos, de 84,6%, concorda que a linguagem é simples e intuitiva de ser utilizada e que a metodologia auxiliou no processo de solução do problema. Um total de 61,5% respondeu que sabia o que deveria fazer em cada passo da metodologia. Todos os estudantes afirmaram que conseguiram identificar facilmente os tipos de entradas e saídas dos componentes, assim como os componentes independentes. A maioria também afirmou que conseguiu identificar facilmente tanto a dependência entre componentes (96,2%), quanto as conexões (88,5%). Nas respostas expositivas, observou-se que a linguagem foi bem aceita pela grande maioria dos estudantes, já que 22 alunos identificaram algumas vantagens na sua utilização: abordagem visual, intuitiva, fácil, simples, prática, entre outras. Apenas 10 alunos identificaram algumas desvantagens na sua utilização: trabalhosa, demorada, requer bastante espaço, entre outras. Dentre os comentários realizados, alguns alunos citaram a atividade como divertida, interessante e intuitiva.

Analisando as especificações, percebeu-se que todos os alunos conseguiram chegar à solução do problema, sendo que: 2 grupos não identificaram as etapas de inclusão dos componentes; 6 grupos não desenharam as portas de entrada/saída do CD e do CE nas etapas C5 e C6; 5 grupos fizeram confusão entre o nome e o tipo das portas de entrada e de saída do CE da etapa C6; e 4 grupos não nomearam a ação do CE na etapa C6.

5. Considerações Finais

Resolver problemas é uma das habilidades essenciais para o século XXI. Este trabalho propõe metodologias que sistematizam a resolução de problemas fundamentadas em técnicas da computação (composição e abstração). Estas metodologias fazem uso da linguagem LiVE, a qual se propõe a ser intuitiva e de fácil aplicação.

Por meio de um estudo de caso, a partir de uma breve explicação da linguagem e da metodologia, observou-se que os alunos conseguiram utilizá-las corretamente e chegaram a solução do problema proposto. Das vantagens citadas, as que se destacaram foram o caráter visual e intuitivo da linguagem. Os problemas levantados pelos estudantes não são decorrentes do processo metodológico em si, mas com relação a dificuldade em lidar com a resolução de problemas grandes. Esta dificuldade pode ser contornada por meio do desenvolvimento de uma ferramenta que dê suporte ao uso da linguagem.

Como trabalho futuro pretende-se definir novas metodologias fundamentadas em outros conceitos da computação, tais como, generalização e recursão, além do desenvolvimento de uma ferramenta de edição para linguagem.

Referências

- Bordini, A. et al. (2018). Linguagem visual para resolução de problemas fundamentada no pensamento computacional: uma proposta. In *Anais do SBIE*, pages 81–90.
- Costella, L. et al. (2017). Construção de ambiente de ensino de robótica remota: Democratizando o desenvolvimento do pensamento computacional em alunos da educação básica. In *Anais do SBIE*, pages 354–363.
- CSTA (2017). Csta k-12 computer science standards. <https://www.csteachers.org/page/standards>.
- Cunha, F. et al. (2018). Uma abordagem baseada em robótica e computação desplugada para desenvolver o pensamento computacional na educação básica. In *Anais do SBIE*, pages 1845–1849.
- Experience, E. (2018). 13 essential 21st century skills for todays students. <https://goo.gl/xnSrXE>.
- Guarda, G. et al. (2018). Jogos lúdicos sob a ótica do pensamento computacional: Experiências do projeto logicamente. In *Anais do SBIE*, pages 486–495.
- Melo, D. et al. (2018). Robô euroi: Game de estratégia matemática para exercitar o pensamento computacional. In *Anais do SBIE*, pages 685–694.
- Oliveira, E. et al. (2016). Pensamento computacional e robótica: Um estudo sobre habilidades desenvolvidas em oficinas de robótica educacional. In *Anais do SBIE*, pages 530–539.
- P21.org (2018). Framework for 21st century learning. <http://goo.gl/VS7HGD>.
- Pressman, R. (2011). *Engenharia de Software*. Rio de Janeiro: McGraw, 7 edition.
- Repenning, A. et al. (2016). Computational thinking tools. In *Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 218–222. IEEE.
- Wing, J. M. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing*.