

## **Instrumentos de Visualização da Informação para Avaliação Diagnóstica em Curso de Programação a Distância**

**Márcia Gonçalves de Oliveira**<sup>1</sup>, **Helen França Medeiros**<sup>2</sup>, **Mônica F. Silva Lopes**<sup>2</sup>,  
**Anne Carolina Silva**<sup>2</sup>, **Ádler Oliveira Silva Neves**<sup>2</sup>

<sup>1</sup> Instituto Federal do Espírito Santo (Ifes) - Cefor

<sup>2</sup>Instituto Federal do Espírito Santo (Ifes) - Campus Serra

clickmarcia@gmail.com, helenfranca93@gmail.com

***Abstract.** Programming learning is a domain of knowledge considered difficult to learn. However, although the difficulties of teaching and learning programming are recurrent themes in academic discussions, few solutions have actually been developed to understand these difficulties and intervene accordingly. In order to perform a diagnostic assessment in a multidimensional perspective based on the careful analysis of source codes developed by students and performance history, this work presents some information visualization tools to assist teachers in the follow-up of their students' learning in the programming practice. The results of applying these instruments in a distance programming class demonstrate that it is possible to recognize, under different variables, indicators of learning difficulties and different student profiles classes.*

***Resumo.** A aprendizagem de programação é um domínio de conhecimento considerado de difícil aprendizagem. No entanto, embora as dificuldades de ensino e de aprendizagem de programação sejam temáticas recorrentes em discussões acadêmicas, poucas soluções de fato têm sido desenvolvidas para compreender essas dificuldades e nelas intervir de acordo. Com o objetivo de realizar uma avaliação diagnóstica em uma perspectiva multidimensional a partir da análise minuciosa de códigos-fontes desenvolvidos por alunos e de histórico de desempenhos, este trabalho apresenta alguns instrumentos de visualização de informação para auxiliar professores no acompanhamento da aprendizagem de seus alunos na prática da programação. Os resultados de aplicação desses instrumentos em uma turma de programação a distância demonstram que é possível reconhecer, por meio de diferentes variáveis, indicadores de dificuldades de aprendizagem e diferentes classes de perfis de alunos.*

### **1. Introdução**

As dificuldades de aprendizagem de programação têm sido uma temática recorrente na literatura acadêmica de Educação em Informática [Lahtinen et al. 2005, Almeida Neto et al. 2006, Tan et al. 2009], embora várias metodologias e tecnologias tenham sido desenvolvidas para apoio ao processo de ensino e de aprendizagem de programação [Mazza and Dimitrova 2007, Oliveira et al. 2015, Chao 2016]. Dessa forma, ainda representam um grande desafio educacional identificar, analisar e tratar as dificuldades de aprendizagem favorecendo o nivelamento de alunos e êxitos coletivos de aprendizagem em turmas de programação.

A Visualização da Informação, porém, pode ser uma opção para auxiliar professores no reconhecimento e na análise de dificuldades de aprendizagem de programação porque consiste em um conjunto de tecnologias que utilizam computação visual para amplificar a cognição humana com informações abstratas [Card et al. 1999, Jacko 2012].

Dessa forma, uma apresentação gráfica e compacta para manipular um grande número de itens dá possibilidades de professores realizarem descobertas, tomarem decisões e explicarem padrões (como, por exemplo, tendências, *clusters* e *outliers*) de grupos de alunos e de cada aluno individualmente [Plaisant et al. 2005].

Com o objetivo de auxiliar professores de programação no acompanhamento da aprendizagem de seus alunos, este trabalho apresenta alguns instrumentos de Visualização da Informação com a finalidade de realizar avaliação diagnóstica da aprendizagem de programação em uma perspectiva multidimensional por meio de diferentes variáveis de avaliação. Os primeiros resultados de aplicação desses instrumentos em um curso de programação a distância de programação mostram como podemos visualizar e comparar os estados de aprendizagem de vários alunos em uma única visualização gráfica.

A principal contribuição deste trabalho para a aprendizagem de programação é oferecer a professores visualizações que lhes informem o que está acontecendo no processo de aprendizagem de seus alunos de forma que eles possam reconhecer, analisar e compreender as dificuldades de aprendizagem de seus alunos e desenvolver ações formativas para remediá-las.

Este trabalho está organizado conforme a ordem a seguir. Na Seção 2, apresentamos uma revisão de literatura sobre a Visualização da Informação na Educação e os trabalhos relacionados. Na Seção 3, descrevemos como os códigos de programação desenvolvidos por alunos foram representados para visualização multidimensional bem como os instrumentos utilizados. Na Seção 4, relatamos uma experiência de aplicação de instrumentos de Visualização da Informação para avaliação diagnóstica de uma turma de programação a distância. Na Seção 5, concluímos com as considerações finais.

## **2. A Visualização da Informação na Educação**

As técnicas de Visualização da Informação podem ser utilizadas para melhorar várias atividades do processo de aprendizagem [Klerkx et al. 2014]. Para mostrar como isso tem se tornado possível, em especial para o domínio da programação, destacamos os seguintes trabalhos: como uma ferramenta de visualização pode direcionar a prática pedagógica de professores [Pissinati et al. 2014]; como a Visualização da Informação pode reduzir o esforço cognitivo dos usuários na obtenção de informação sobre a participação e a interação dos indivíduos nos cursos [Romani 2000]; como a ferramenta de Visualização da Informação *CourseVis* pode ser utilizada para monitoramento de estudantes em um curso de programação a distância [Mazza and Dimitrova 2007]; e como a Visualização da Informação pode ser utilizada para identificar e compreender dificuldades de aprendizagem em programação [Fu et al. 2017].

Visando um melhor acompanhamento da aprendizagem em turmas numerosas com alta diversidade de perfis de aprendizagem, o trabalho de [Pissinati et al. 2014] propõe uma ferramenta de visualização de desempenhos de vários alunos em várias atividades de forma ágil. Nessa proposta, foi possível visualizar, em um único

gráfico, os desempenhos de aproximadamente 600 alunos em quinze turmas diferentes [Pissinati et al. 2014].

Um exemplo do uso de visualização da informação em ambiente de Educação a Distância é a ferramenta *InterMap*, uma ferramenta desenvolvida por [Romani 2000]. Nessa ferramenta procurou-se utilizar técnicas de Visualização de Informação para diminuir o esforço cognitivo dos usuários na obtenção de informação sobre a participação e a interação dos indivíduos nos cursos.

O *CourseVis* é uma ferramenta desenvolvida por [Mazza and Dimitrova 2007] para monitoramento de estudantes. O objetivo dessa ferramenta é auxiliar instrutores de cursos a distância no acompanhamento de seus alunos. Para isso, o *Coursevis* gera representações gráficas multidimensionais para visualização de informações sobre desempenhos, características e comportamentos dos alunos de um curso a distância sintetizando um processo de aprendizagem em uma matriz cognitiva. As visualizações gráficas oferecidas pelo *Coursevis* representam mapas mentais de componentes individuais de aprendizagem. Através desses mapas, os instrutores podem identificar progressos e dificuldades de aprendizagem para tomada de decisões de reorientação do ensino.

Um trabalho mais recente de Visualização da Informação no domínio da programação é a proposta de [Fu et al. 2017]. Para facilitar o ensino e a aprendizagem da Linguagem C, [Fu et al. 2017] propõe um sistema, o *LAPLE* (*Learning Analytics in Programming Language Education*), que fornece um painel de aprendizado para capturar o comportamento dos alunos na sala de aula e identificar as diferentes dificuldades enfrentadas. De acordo com [Fu et al. 2017], através do *LAPLE*, os professores podem entender melhor a situação de aprendizagem de seus alunos em tempo real e melhorar o material instrucional conforme os resultados da análise. Os alunos, por sua vez, podem utilizar o *LAPLE* para localizar erros de sintaxe em programas C e obter recomendações de materiais educacionais sobre como corrigi-los [Fu et al. 2017].

Este trabalho combina ideias das propostas de [Mazza and Dimitrova 2007] e [Fu et al. 2017] ao oferecer instrumentos de Visualização da Informação para visualizar estados de aprendizagem de estudantes de programação em uma matriz cognitiva multidimensional com a finalidade de identificar e analisar dificuldades de aprendizagem bem como reconhecer classes de perfis de alunos.

### **3. Avaliação Diagnóstica da Aprendizagem de Programação**

Para apoiar a avaliação diagnóstica da aprendizagem de programação, este trabalho utilizou instrumentos de Visualização da Informação, conforme as etapas a seguir [Ware 2012]:

1. A coleta e armazenamento de dados
2. O pré-processamento planejado para transformar os dados em algo compreensível
3. O hardware de exibição e os algoritmos gráficos que produzem uma imagem
4. O sistema perceptivo e cognitivo humano (o perceptor)

Na Etapa 1, a coleta e o armazenamento de dados foram realizados através de um curso a distância de programação executado no ambiente virtual *Moodle*. Nesse curso, através do recurso *Tarefa*, alunos submetiam as soluções de exercícios de programação em três arquivos: *makefile*, arquivo de entrada e programa. O *makefile* contém as instruções

de execução do programa submetido. O arquivo de entrada contém as entradas de teste do programa. Já o programa é escrito em Linguagem C para resolver um problema proposto em um exercício de programação.

Na Etapa 2, o pré-processamento consiste em gerar as representações das soluções submetidas por cada aluno. Cada uma dessas representações consiste de um vetor multi-dimensional que mapeia um estudante e sua solução em um estado de aprendizagem ou perfil cujas dimensões são valores de componentes de habilidades [Oliveira et al. 2015] ou de métricas software ([Curtis et al. 1979, Berry and Meekings 1985]). Uma outra representação de perfil de estudante consiste de um vetor cujas dimensões são valores de desempenhos de um estudante ao longo de um curso de programação. Após a geração da representação de estado de aprendizagem ou de perfil, os vetores formados são reunidos em uma matriz cognitiva, conforme a Figura 1.

	25	104					
0.0	0.0	0.3333333	0.2973	0.0	0.0822	0.9304	0.5
1.0	1.0	0.3333333	0.4595	0.0	0.3748	0.8102	1.0
1.0	1.0	0.0	0.8649	1.0	0.2186	0.0	0.5
1.0	1.0	0.0	0.8649	1.0	0.2186	0.0	0.5
1.0	1.0	0.3333333	0.1081	0.0	0.2481	0.3736	0.5
1.0	1.0	0.3333333	0.2703	0.0	0.0619	0.7333	0.5
1.0	1.0	0.3333333	0.2703	0.1249	0.2406	1.0	0.5
1.0	1.0	0.3333333	0.0	0.0	0.0987	0.894	0.5
1.0	1.0	0.3333333	0.1351	0.0	0.2826	0.4344	1.0
1.0	1.0	0.0	0.2973	0.6457	0.4083	0.8048	0.0
1.0	1.0	0.3333333	1.0	0.7441	0.2388	0.5508	0.5
1.0	1.0	0.3333333	0.0541	0.1068	1.0	0.7346	0.5
1.0	1.0	0.3333333	0.5946	0.1068	0.0	0.1922	1.0
1.0	1.0	0.6666666	0.3514	0.0	0.2152	0.8438	0.5
1.0	1.0	0.3333333	0.5676	0.4735	0.3693	0.0481	0.5
0.0	0.0	0.0	0.0811	0.0	0.6746	0.3631	0.5

**Figura 1. Matriz cognitiva com representações de estados de aprendizagem**

Na Matriz Cognitiva da Figura 1, a primeira linha informa o número de alunos de uma turma e o número de variáveis de avaliação. Nesse exemplo temos a matriz cognitiva de uma turma de curso de programação a distância composta por 25 alunos representados em 104 métricas de código-fonte que contêm medidas de esforço e qualidade de programação ([Curtis et al. 1979, Berry and Meekings 1985]). As variáveis das duas primeiras colunas dessa matriz são indicadores de *compila* e *executa*. Dessa forma, se o programa de um estudante compila e executa, essas variáveis estarão setadas em 1 e em 0 (zero), caso contrário. Nas demais métricas, em geral, os valores altos (mais próximos de 1) indicam dificuldades e os valores mais baixos indicam boas práticas de programação.

Na Etapa 3, os algoritmos gráficos que produzem uma visualização são os algoritmos escritos em Linguagem R para geração de mapas de calor [Kolde 2015] e os algoritmos de *clustering* do software *Cluto 2.1.2* [Karypis 2002], para geração de gráficos de visualização de agrupamentos de perfis ou estados de aprendizagem. Para a geração dos *clusters* utilizamos o algoritmo *Bisecting K-means* tendo com medida de similaridade o coeficiente de correlação [Karypis 2002].

Na Etapa 4, o sistema perceptivo e cognitivo humano, a partir das visualizações geradas, visa identificar e analisar as dificuldades de aprendizagem individuais e comuns

em grupos de estudantes para remediá-las de acordo.

Na próxima seção, apresentaremos como as etapas 3 e 4 foram realizadas em nosso estudo de caso.

#### 4. Estudo de caso

Para experimentação dos instrumentos de Visualização da Informação utilizados na avaliação diagnóstica da aprendizagem de programação, escolhemos duas turmas de um curso de programação C a distância ofertado pelo Centro de Referência em Formação e Educação a Distância do Instituto Federal do Espírito Santo (Ifes). A coleta e a representação de dados dessas turmas foram realizadas conforme processos e recursos especificados na Seção 3.

Para a primeira turma, selecionamos um exercício de programação aplicado no meio do curso e utilizado por [Oliveira et al. 2015] para experimentação. A especificação do exercício consiste em obter, a partir do número de pontos obtidos por times de um campeonato de futebol, o campeão e o vice-campeão de um campeonato de futebol.

Esse exercício, que chamamos *Exercício do futebol*, abrange os conteúdos que, em geral, os alunos começam a ter dificuldades, que são: expressões lógicas, estruturas de controle condicional e estruturas de controle de repetição [Schildt and Mayer 2006]. Uma outra justificativa para utilizar esse exercício é o número de expressões lógicas. Um aluno que entende como será selecionado o campeão e o vice, utiliza no máximo três comparações. Por outro lado, o aluno que tem dificuldades ou programa para "dar certo" e não para "fazer certo" escreve mais de dez expressões lógicas, resultando em muitas comparações e muitas linhas de código. Dessa forma, a quantidade de comparações e de linhas de código são bons indicadores para evidenciar dificuldades de aprendizagem, principalmente na construção de expressões lógicas.

Para visualizar essas dificuldades, geramos um mapa de calor a partir de um *script* escrito em Linguagem R. Nesse mapa de calor apresentado na Figura 2, as linhas representam os alunos e as colunas, cerca de 104 métricas de códigos-fontes selecionadas entre 348 métricas de software ([Curtis et al. 1979, Berry and Meekings 1985, Oliveira et al. 2015]).

De acordo com a Figura 2, a primeira observação vem dos alunos que apresentam mais características de dificuldades de aprendizagem, que são os alunos com rótulos 1, 16 e 17. Observa-se, pela primeira coluna do gráfico na variável *compila*, que as soluções desses alunos não compilam, pois estão de cor azul, sinalizando que essa variável assume valor zero. Essas soluções apresentam-se também com altos valores em várias métricas, o que pode ser visualizado pela tonalidade amarela em várias métricas. Isso nos leva a concluir, que esses alunos escreveram muito código e não conseguiram desenvolver uma solução correta, o que caracteriza dificuldades de aprendizagem.

No gráfico de *clustering* da Figura 3, podemos visualizar os perfis de estudantes de programação reunidos em sete grupos (ou *clusters*), conforme similaridades. Nesse gráfico, as linhas representam cada um dos 25 alunos, as colunas são as métricas de código-fonte e as linhas pretas horizontais separam os grupos de alunos. Nas colunas, a cor vermelha denota valores positivos, a cor verde denota valores negativos e as cores próximas de branco indicam valores próximos de 0. Quanto mais forte for a tonalidade



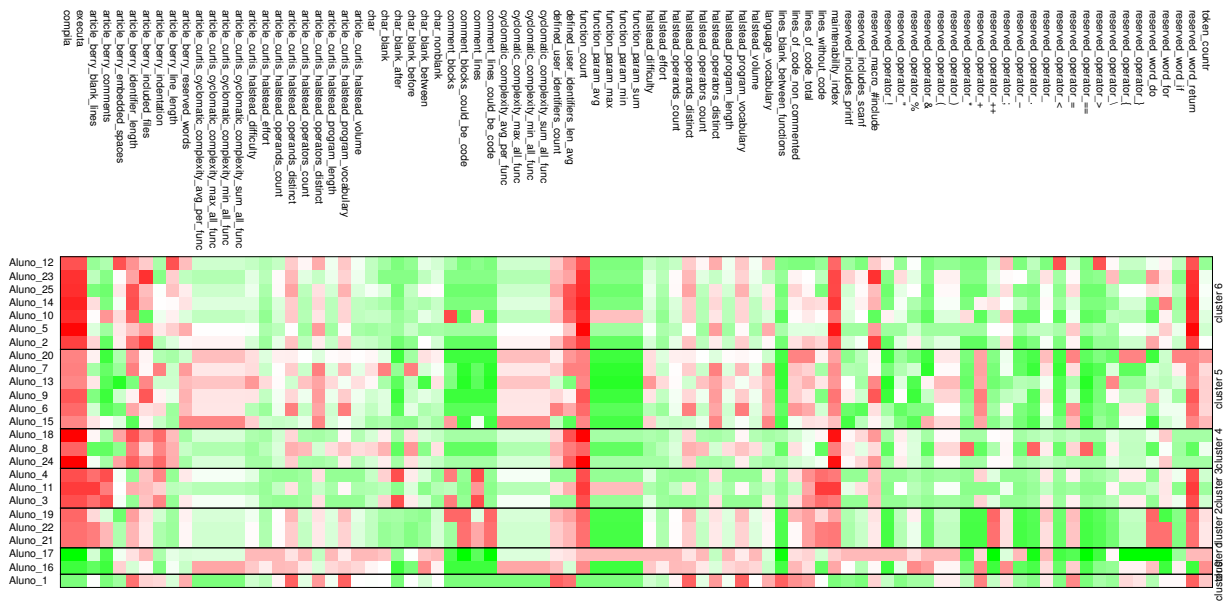


Figura 3. Grupos de estados de aprendizagem

entre 0 e 1 das métricas apresentam-se altos, conforme a presença marcante da cor vermelha nas amostras desse *cluster*. Uma característica comum das soluções desse *cluster* foi o uso do comando de seleção *switch* da Linguagem C. Nesse exercício, esperava-se apenas o uso de expressões lógicas, estrutura *if* aninhada e uma estrutura de repetição [Schildt and Mayer 2006]. Dessa forma, o uso do *switch* na resolução desse exercício seria considerada uma manobra para fazer o programa “dar certo” e não um recurso correto para resolver o problema especificado.

A Figura 4 é um instrumento de Visualização da Informação também baseado em gráfico de mapa de calor para visualização de desempenhos de uma turma em todas as atividades ao longo do Curso de Programação C a distância. Nesse gráfico, as linhas representam os alunos e as colunas, cada atividade do curso de programação.

O instrumento de Visualização da Informação da Figura 4 é uma importante ferramenta de avaliação diagnóstica porque um professor tem em mãos uma visualização de toda uma turma durante todo o curso em um só gráfico que lhe permite comparar os alunos, identificar em quais atividades estão apresentando mais dificuldades e quais estão progredindo melhor durante um curso de programação.

As experiências apresentadas aplicando instrumentos de Visualização da Informação contemplando o processo de aprendizagem de programação através de 104 variáveis de avaliação mapeadas em métricas de código-fonte mostram que esses instrumentos podem ser muito úteis para professores de programação porque dão as seguintes possibilidades: reconhecimento automático das melhores soluções de uma exercício, identificação dos alunos que estão apresentando mais dificuldades de aprendizagem, detecção de casos de plágios e acompanhamento dos desempenhos de uma turma ao longo de um curso.

Em resumo, os instrumentos de Visualização da Informação apresentados podem ser utilizados para auxiliar professores de programação em suas práticas de ensino e de







Aprendizagem de Programação por Tecnologias de Reconhecimento de Padrões”. Esta publicação é resultante da execução desses projetos.

## Referências

- Almeida Neto, F. A., de Castro, T. H. C., and de Castro Júnior, A. N. (2006). Utilizando o método clínico piagetiano para acompanhar a aprendizagem de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 1, pages 527–536.
- Berry, R. E. and Meekings, B. A. (1985). A style analysis of c programs. *Communications of the ACM*, 28(1):80–88.
- Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann.
- Chao, P.-Y. (2016). Exploring students’ computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, 95:202–215.
- Curtis, B., Sheppard, S. B., Milliman, P., Borst, M., and Love, T. (1979). Measuring the psychological complexity of software maintenance tasks with the Halstead and McCabe metrics. *IEEE Transactions on software engineering*, (2):96–104.
- Fu, X., Shimada, A., Ogata, H., Taniguchi, Y., and Suehiro, D. (2017). Real-time learning analytics for c programming language courses. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 280–288. ACM.
- Jacko, J. A. (2012). *Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications*. CRC press.
- Karypis, G. (2002). CLUTO:Clustering Toolkit. Technical report, Minnesota Univ Minneapolis Dept of Computer Science.
- Klerkx, J., Verbert, K., and Duval, E. (2014). Enhancing learning with visualization techniques. In *Handbook of research on educational communications and technology*, pages 791–807. Springer.
- Kolde, R. (2015). Pheatmap: Pretty heatmaps. R package version 1.0.8.
- Lahtinen, E., Ala-Mutka, K., and Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. In *Acm Sigcse Bulletin*, volume 37, pages 14–18. ACM.
- Mazza, R. and Dimitrova, V. (2007). Coursevis: A graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human-Computer Studies*, 65(2):125–139.
- Oliveira, M., Nogueira, M., and Oliveira, E. (2015). Sistema de apoio à prática assistida de programação por execução em massa e análise de programas. In *CSBC 2015 - Workshop de Educação em Informática (WEI), Recife-PE*.
- Pissinati, E., Oliveira, M., and Oliveira, E. (2014). Direcionando a atuação do professor através de uma ferramenta de visualização. In *Anais da XIV Escola Regional de Computação*, Bahia.

- Plaisant, C. et al. (2005). Information visualization and the challenge of universal usability. *Exploring Geovisualization*, pages 53–82.
- Romani, L. (2000). Intermap: Ferramenta para visualização da interação em ambientes de educação a distância na web. Master's thesis, Universidade Estadual de Campinas, São Paulo.
- Schildt, H. and Mayer, R. C. (2006). *C completo e total*. Pearson.
- Tan, P.-H., Ting, C.-Y., and Ling, S.-W. (2009). Learning difficulties in programming courses: undergraduates' perspective and perception. In *Computer Technology and Development, 2009. ICCTD'09. International Conference on*, volume 1, pages 42–46. IEEE.
- Ware, C. (2012). *Information visualization: perception for design*. Elsevier.