

# Um Sistema para o Monitoramento das Atividades de Programação Para Alunos Iniciantes

Patricia Padula Lopes<sup>1</sup>, Thiago Ferreira Dantas<sup>1</sup>, Érico Marcelo Hoff do Amaral<sup>1</sup>

<sup>1</sup>Engenharia de Computação – Universidade Federal do Pampa (UNIPAMPA)  
Avenida Maria Anunciação Gomes de Godoy, 1650 – 96.413-172 – Bagé – RS – Brasil

{padulalopes, thiagodantas923, ericohoffamaral}@gmail.com}

**Abstract.** *This work presents a possible solution to the difficulties encountered by teachers in attending to the learning difficulties of students beginning in Algorithms and Programming, through a monitoring system, the CFacil+. For this purpose, the tool implemented by GOMES et al. (2016) that has been adapted to a client and server network environment in order to allow the teacher to monitor the students' performance in the tasks proposed in the classroom, providing assistance to those students who most need help during the activities. teachers' evaluations regarding the use of the system, pointed to the CFacil+ as an aid alternative to the teaching of the introductory programming disciplines.*

**Resumo.** *Este trabalho apresenta uma possível solução às dificuldades encontradas pelos professores no atendimento das dificuldades de aprendizagem dos alunos iniciantes em disciplinas de Algoritmos e Programação, através de um sistema de monitoramento, o CFacil+. Utiliza-se, para isto, a ferramenta implementada por GOMES et al. (2016) que foi adaptada a um ambiente de redes cliente e servidor, de forma a possibilitar que o professor acompanhe o desempenho dos alunos nas tarefas propostas em sala de aula, fornecendo assistência àqueles estudantes que mais necessitam de auxílio durante as atividades. As avaliações dos docentes em relação a utilização do sistema, apontaram para o CFacil+ como uma alternativa de auxílio ao ensino das disciplinas introdutórias de programação.*

## 1. Introdução

As disciplinas de Algoritmos e Programação constituem a base para o ensino de programação e abordam os princípios da lógica de programação, com o objetivo de desenvolver a capacidade de análise e resolução de problemas dos alunos através da descrição das soluções dos problemas na forma de algoritmos. Além disso, esta disciplina faz parte do plano curricular de vários cursos na área de tecnologia, com a finalidade de introduzir os conceitos de programação, exigindo dos alunos determinadas habilidades e desenvolvendo competências. Entre essas, pode-se mencionar a capacidade de pensar logicamente, a fim de organizar as ideias em forma de código e resolver problemas, tendo como propósito a construção de soluções algorítmicas (LOPES et al. 2017).

Como é possível perceber, para que os alunos possam desenvolver o raciocínio necessário para a construção de conhecimento sobre a lógica envolvida na implementação de soluções computacionais para problemas simples, é necessário conforme Raabe et al. (2005), que o professor possa atender, acompanhar, mediar e avaliar os alunos nas atividades que são propostas em sala de aula. Em muitas situações, esta demanda de interação é inviável de ser atendida, devido à quantidade de alunos e a diversidade de

dificuldades apresentadas por estes. Desta forma, muitas vezes as demandas com relação à aprendizagem não são detectadas e atendidas em tempo hábil, ocasionando problemas como desmotivação, reprovação ou evasão.

Com base neste contexto, e reconhecendo a finalidade do CFacil (GOMES *et al.* 2016), este estudo apresenta uma proposta de uma solução que permita ao professor acompanhar as atividades de programação desenvolvidas pelos alunos em ambientes de laboratório. Dessa forma, é esperado que o professor consiga identificar as dificuldades e os erros mais comuns dos discentes na aprendizagem inicial da programação, acompanhando os passos do estudante através do ambiente e registrando seu progresso por meio das atividades a serem desenvolvidas. A proposição e ajuste das estratégias didáticas então adotadas visam um aprimoramento da experiência de ensino e aprendizagem na disciplina de Algoritmos e Programação.

A fim de demonstrar este estudo, a presente pesquisa está estruturada da seguinte forma: na seção 2 é realizado o levantamento teórico sobre conceitos relevantes para o desenvolvimento do trabalho. A seção 3 apresenta a caracterização da pesquisa e os procedimentos metodológicos adotados. A seção 4 apresenta o desenvolvimento do sistema para monitoramento das atividades dos alunos iniciantes em programação. Na sequência descrevem-se as técnicas adotadas no processo da implementação, bem como os resultados da utilização dessa ferramenta. Por fim, a seção 5 apresenta as considerações finais.

## 2. Referencial Teórico

### 2.1 Dificuldades dos professores no ensino de Algoritmos e Programação

O ensino de Algoritmos e Programação tem como propósito permitir que os alunos desenvolvam um conjunto de competências necessárias para conceber programas e sistemas computacionais capazes de resolver problemas reais. Segundo Biggs e Tang (2011), o professor deve ter a capacidade de instruir os alunos a fim de que eles consigam aprender a desenvolver programas, de forma a garantir uma metodologia de análise de problemas através da construção de algoritmos, capacitando-os a traduzir estes algoritmos em programas, a partir de uma linguagem em alto nível.

Por ser o primeiro componente curricular que trabalha com os conceitos de resolução de problemas via computador, as disciplinas de programação são ministradas no início dos cursos relacionados às tecnologias da informação e comunicação. É comum que essas turmas alcancem um número consideravelmente elevado de alunos, visto que há um alto nível de reprovação e muito dos alunos são repetentes (WATSON; LI, 2014).

Todos estes fatores dificultam uma avaliação individualizada do professor com relação às atividades propostas em laboratório, além de limitar a quantidade de avaliações que se pode realizar em um semestre: é muito difícil para o educador avaliar individualmente todas as soluções de um determinado algoritmo, e pontuá-lo de acordo com a quantidade de acertos ou erros dos estudantes (MOREIRA *et al.* 2009).

Para produzir melhores resultados no processo de aprendizagem, torna-se necessário que o docente consiga compreender como as experiências relacionadas à programação de algoritmos estão relacionadas aos conceitos estudados e como podem ser organizadas para alcançar os objetivos esperados. No entanto, conforme Tardif (2002), para muitos professores, acompanhar os alunos, individualmente, se traduz como algo

bastante difícil, principalmente quando realizam atividades práticas, pois este tipo de atividade requer ao professor acompanhá-los de forma individual ou em pequenos grupos no processo de aprendizagem. Com base nestas características, percebe-se que a disponibilização de um recurso computacional, que permita ao docente acompanhar e identificar os alunos com maiores dificuldades em relação as atividades propostas em sala de aula, irá possibilitar um maior contato entre alunos e professores, proporcionando o acompanhamento constante do processo de ensino e aprendizagem.

## 2.2 Tecnologias e práticas pedagógicas para o acompanhamento dos alunos em atividades de programação

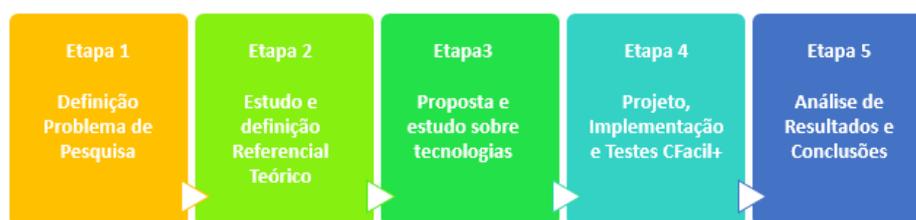
A cada ano diversas tecnologias são desenvolvidas como auxílio ao processo de ensino-aprendizagem de algoritmos e programação. Porém, muitas das práticas de ensino adotadas não conseguem suprir a necessidade dos docentes em acompanhar todos os estudantes durante as aulas de programação realizadas em laboratórios, devido ao quantitativo elevado de alunos. Neste sentido, devido ao seu caráter técnico-formal, a disciplina carece de modelos pedagógicos que facilitem o acompanhamento efetivo das atividades dos estudantes (AMARAL, 2015). Diante dessa problemática, novas ferramentas computacionais vêm, ao longo dos anos, sendo incorporadas para auxiliar os professores no monitoramento das turmas em aulas de algoritmos e programação.

Pesquisas envolvendo a utilização deste tipo de ferramenta foram apontadas por Guoli *et al.* (2010), como alternativa para ampliar o processo de ensino e aprendizagem na área educacional. Os pesquisadores descrevem que essas aplicações podem fornecer maneiras mais fáceis e eficientes para os professores monitorarem o progresso dos alunos de forma individual. Assim sendo, a proposta de ambiente apresentada neste trabalho tenta englobar estas características, para que o professor consiga acompanhar os alunos continuamente, por intermédio da visualização dos resultados das atividades. Enfim, pretende-se que a ferramenta proposta no presente trabalho possua um diferencial das já existentes, podendo agregar um fator motivacional aos alunos, e apresentar-se como uma alternativa inovadora para auxiliar o professor no ensino dos algoritmos e programação.

## 3. Metodologia

De acordo com a forma de abordagem, este estudo, segundo Wainer *et al.* (2007), constitui-se numa análise qualitativa com o objetivo de avaliar a aplicabilidade do sistema no monitoramento do desempenho dos estudantes durante as práticas de programação, validando as propostas apresentadas neste trabalho. Quanto aos procedimentos técnicos, a pesquisa classifica-se como experimental, pois baseia-se em um estudo de caso aplicado a professores e alunos de turmas de Algoritmos e Programação na Universidade Federal do Pampa. A pesquisa segundo Fonseca (2002) pode também ser considerada como bibliográfica, já que implica no estudo de artigos, teses, livros e outras publicações usualmente disponibilizadas por editoras e indexadas.

Tendo em vista a metodologia adotada para o desenvolvimento deste trabalho, propôs-se um conjunto de ações, organizadas formalmente em etapas, por meio de atividades específicas, com o intuito de apontar um método efetivo para o acompanhamento das atividades de programação pelos professores. Um resumo de cada uma destas etapas é apresentado a seguir, na Figura 1.



**Figura 1. Estrutura de Desenvolvimento da Pesquisa**

O planejamento das atividades partiu da definição do problema de pesquisa na etapa 1, a etapa 2 contemplou estudo e levantamento do referencial teórico para servir de base a construção da proposta do sistema de monitoramento, a definição do CFacil em um ambiente de redes (laboratório), as tecnologias adotadas, o projeto, a implementação e os testes de software foram realizados nas etapas 3 e 4 respectivamente. Por fim a avaliação dos resultados, com base nos experimentos de utilização da ferramenta em sala de aula com professores da Universidade Federal do Pampa em turmas de algoritmos e programação foi realizada na quinta etapa.

## 4. Desenvolvimento

### 4.1 Análise da ferramenta CFacil e a proposta de integração a um ambiente de redes cliente e servidor

A ferramenta desenvolvida por Gomes *et al.* (2016) é um analisador de códigos em linguagem C, e tem como objetivo tornar a depuração dos mesmos mais atraente ao aluno, através da apresentação de mensagens de erros no idioma português. Para isto, a ferramenta realiza uma análise recursiva descendente sobre o código a ser analisado. Um analisador recursivo descendente é uma coleção de funções recursivas que processam uma expressão. Ao final desta análise a ferramenta de *software* mostra a saída com os erros na tela para o aluno, ao mesmo tempo gera um arquivo de *logs*, capturando a saída que foi enviada para o terminal.

Com os arquivos de *logs* que são gerados durante as compilações com o CFacil e a organização e disposição interna das informações nestes arquivos, torna-se possível gerar estatísticas sobre as dificuldades que os alunos estão encontrando na prática de programação e apresentar ao professor, de maneira que ele consiga acompanhar as atividades individuais realizadas por cada estudante e, também de forma centralizada, de toda a turma. Desta forma, o professor pode analisar o desempenho da turma e também de cada aluno, podendo intervir em atividades ou conteúdos que um aluno, ou mais de um, estejam com dificuldades.

A proposta de solução para o monitoramento das atividades dos alunos tem como propósito coletar os logs de erros que são emitidos através da ferramenta CFacil e analisar estes *logs* para realizar uma correlação geral da quantidade de compilações, erros e acertos bem como a descrição de cada erro cometido em cada exercício que será compilado pelos alunos durante as tarefas de programação, permitindo que o professor consiga acompanhar as dificuldades dos estudantes através do monitoramento destas informações que pode ser realizado através da configuração de uma rede cliente e servidor, disponibilizando uma infraestrutura para que o docente consiga acompanhar as tarefas propostas aos estudantes em sala de aula.



cfacilplus\_cliente será iniciada e a conexão com o servidor (computador do professor) será estabelecida para monitorar os alunos. Este endereço é disponibilizado ao docente, por meio da aplicação cfacilplus\_servidor, conforme é mostrado na Figura 4, ao clicar no botão “Obter IP” e deve ser informado aos alunos para que seja estabelecida a conexão com a sua máquina. Isso foi planejado para que o professor possa utilizar a ferramenta em qualquer laboratório, já que se pode disponibilizar a ferramenta para mais de uma instituição de ensino e conseqüentemente atender um número maior de alunos.



Figura 3. Interação do Aluno com o CFacil+



Figura 4. Interação do Professor com o CFacil+

Após a conexão ser estabelecida, a aplicação permite o envio de relatórios que contém a análise dos *logs* de erros coletados e analisados pelo CFacil+. Posteriormente, conforme é apresentado na Figura 4, o professor no lado servidor da aplicação, precisa clicar no botão “Iniciar Servidor” por meio desta ação, o servidor será iniciado e ficará à espera da conexão dos clientes (alunos) na rede.

Por último, ao clicar no botão “Monitorar Alunos” será disponibilizada uma tabela (Figura 5) que contém as informações sobre o nome dos alunos, a identificação da máquina ao qual ocupam em sala de aula, a quantidade de erros e acertos e um *status* de desempenho que irá determinar a situação de cada um dos alunos com relação aos exercícios compilados.

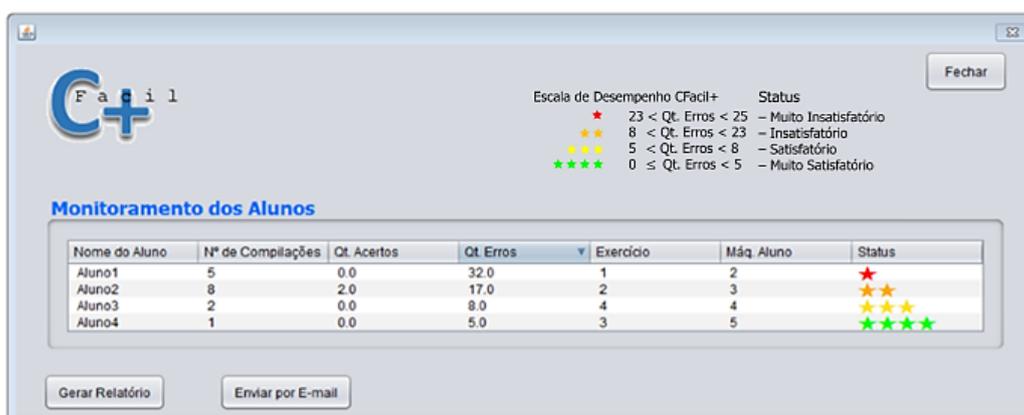


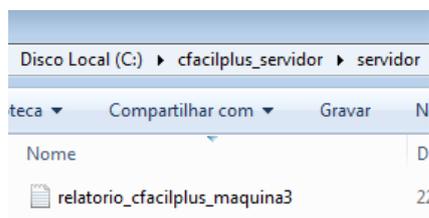
Figura 5. Interface para Monitoramento dos Alunos no CFacil+

Aos alunos que obtiveram um desempenho “Muito Insatisfatório” com uma quantidade de erros maior do que 25 erros foram classificados com uma estrela na cor vermelha e apareciam no topo da tabela. Enquanto que os alunos que apresentaram um

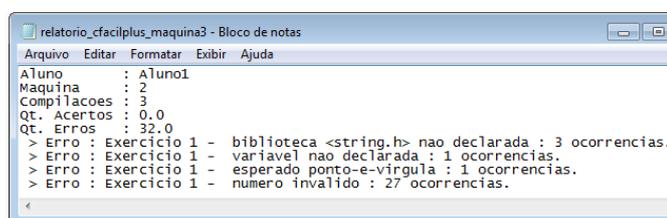
desempenho “Insatisfatório”, com uma quantidade de erros entre 8 a 23 erros, eram classificados com 2 estrelas na cor laranja, logo abaixo. Aqueles alunos que apresentassem uma quantidade de erros entre 5 a 8 erros, eram classificados com 3 estrelas na cor amarela. Já os alunos que obtiverem um desempenho “Muito Satisfatório”, apresentando 5 erros durante a resolução das suas atividades, eram classificados com 4 estrelas de cor verde. Os estudantes que apresentarem um menor desempenho na realização de suas atividades, ou seja, aqueles com uma maior quantidade de erros acumulada na realização dos exercícios ficaram no topo da tabela. Isso permite o acompanhamento contínuo de cada aprendiz e o tratamento personalizado a cada estudante para reverter o quadro de dificuldades constatadas.

Para que o professor consiga identificar os alunos em sala de aula, sugere-se que os alunos sejam dispostos em máquinas previamente numeradas. Dessa forma, a partir da interface de monitoramento será possível identificar os estudantes e auxiliar aqueles que apresentam problemas com base na identificação dos discentes em cada máquina.

Para identificar quais são os erros cometidos pelos estudantes, é possível ter acesso aos relatórios de cada um dos alunos em sua sala de aula adentrando o diretório que hospeda o servidor. Em uma situação hipotética, suponha que o “Aluno1” esteja ocupando a máquina 3 e ele é quem possui maior dificuldade, então o professor deve acessar nesta pasta o seguinte relatório apresentado na Figura 6. Para que isto seja possível, no momento da instalação da ferramenta é preciso configurar um arquivo.cfg e indicar a numeração das máquinas dos alunos.



**Figura 6. Acesso ao relatório de Erros dos Alunos**



**Figura 7. Detalhes do Monitoramento das Atividades dos Alunos Apresentados no Relatório de Erros**

Então, conforme mostra a Figura 7, obter-se-á em detalhes a descrição de cada um dos erros que este aluno está cometendo na resolução de suas atividades. Assim sendo, o professor poderá auxiliar esse estudante. Quando todos os alunos estiverem finalizado suas atividades, um relatório personalizado contendo a quantidade de erros, acertos, compilações e descrição de cada erro em cada exercício pelos alunos poderá ser gerado e enviado via *e-mail* para o professor. No botão “Gerar Relatório”, um relatório geral da turma será gerado e armazenado no diretório raiz da aplicação cfacilplus\_servidor. Para enviar este relatório para o *e-mail* do professor basta selecionar a opção de envio de relatórios no CFACIL+.

A mensagem enviada ao professor é apresentada na Figura 8, através da qual o docente terá acesso ao relatório geral do monitoramento das atividades dos alunos realizado durante as suas aulas.

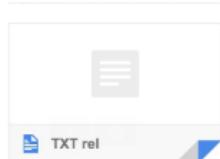
Seu Relatório Geral do CFacil Plus está pronto!

cfacilplus@gmail.com

para mim

O relatório segue em anexo

Bem vindo(a)



**Figura 8. Acesso ao Relatório Geral do Monitoramento dos alunos**

```
Relatorio_Geral - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda

Aluno      : Aluno 1
Maquina    : 10
Compilacoes : 1
Qt. Acertos : 0.0
Qt. Erros  : 12.0
> Erro : Exercício 1 - numero invalido : 4 ocorrencias.
> Erro : Exercício 1 - erro de sintaxe : 1 ocorrencias.
> Erro : Exercício 1 - variavel nao declarada : 3 ocorrencias.
> Erro : Exercício 1 - esperado parenteses : 2 ocorrencias.
> Erro : Exercício 1 - esperado ponto-e-virgula : 2 ocorrencias.

Aluno      : Aluno 2
Maquina    : 12
Compilacoes : 6
Qt. Acertos : 0.0
Qt. Erros  : 37.0
> Erro : Exercício 1 - numero invalido : 8 ocorrencias.
> Erro : Exercício 1 - esperado ponto-e-virgula : 6 ocorrencias.
> Erro : Exercício 1 - esperado parenteses : 6 ocorrencias.
> Erro : Exercício 1 - chaves desbalanceadas : 4 ocorrencias.
> Erro : Exercício 2 - esperado ponto-e-virgula : 2 ocorrencias.
> Erro : Exercício 2 - tipos incompativeis na funcao/expressao : 2 ocorrencias.
> Erro : Exercício 3 - numero invalido : 2 ocorrencias.
> Erro : Exercício 3 - esperado ponto-e-virgula : 3 ocorrencias.
> Erro : Exercício 3 - esperado parenteses : 1 ocorrencias.
> Erro : Exercício 3 - erro de sintaxe : 2 ocorrencias.
> Erro : Exercício 3 - variavel nao declarada : 1 ocorrencias.

Aluno      : Aluno 3
Maquina    : 13
Compilacoes : 5
Qt. Acertos : 0.0
Qt. Erros  : 26.0
> Erro : Exercício 1 - numero invalido : 8 ocorrencias.
> Erro : Exercício 1 - esperado ponto-e-virgula : 4 ocorrencias.
> Erro : Exercício 1 - esperado parenteses : 4 ocorrencias.
> Erro : Exercício 1 - chaves desbalanceadas : 5 ocorrencias.
> Erro : Exercício 2 - esperado ponto-e-virgula : 3 ocorrencias.
> Erro : Exercício 2 - tipos incompativeis na funcao/expressao : 2 ocorrencias.
```

**Figura 9. Relatório Geral CFacil+**

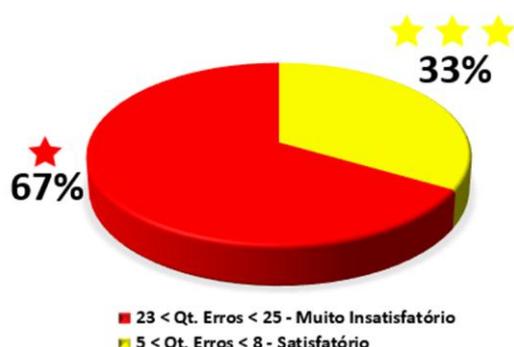
De posse deste relatório (Figura 9) o professor pode observar se os alunos estão tendo dificuldades e onde essas se encontram em cada um dos exercícios do conteúdo programático. Por meio das informações coletadas durante o acompanhamento das atividades no sistema, o docente poderá identificar precocemente aqueles alunos que podem vir a não obter aprovação na disciplina, no intuito de melhorar o seu desempenho.

Ao identificar que um grande número de alunos apresenta dificuldades em um mesmo conteúdo, é possível trabalhar neste com mais profundidade ou com estratégias diferentes para propor novas formas de intervir nesta realidade.

## 4.2 Verificação e Validação

A fim de verificar a usabilidade e efetividade do CFacil+, no processo de ensino aprendizagem de algoritmos e programação, foi preparado um ambiente para testes, através da configuração de uma rede de computadores, no qual a ferramenta foi instalada nos laboratórios de informática geralmente utilizados pela disciplina. Com isso, a ferramenta foi utilizada em duas turmas de alunos da Universidade Federal do Pampa. O universo amostral dos testes era de aproximadamente 20 alunos por aula de cursos de engenharia. Cada encontro, ocorrido nas diferentes turmas, teve duração de dois períodos de 50 minutos subsequentes.

No primeiro teste com o CFacil+, foi proposto pelo professor que os alunos resolvessem 4 exercícios, variando do nível mais fácil ao mais difícil, sendo abordado o conteúdo de estruturas de repetição. Conforme é apresentado no gráfico da Figura 10, constatou-se que 67% dos alunos apresentaram um desempenho “Muito Insatisfatório”, sendo estes classificados com uma estrela na cor vermelha na interface de monitoramento. Enquanto que 33% apresentaram um desempenho “Satisfatório”, sendo classificado com 3 estrelas na cor amarela.



**Figura 10. Análise do Desempenho dos Alunos no 1º Teste**



**Figura 11. Análise do Desempenho dos Alunos no 2º Teste**

Na interface apresentada ao docente, foi possível que o professor observasse durante o experimento realizado, que os estudantes conseguiram resolver até o terceiro exercício, apresentando em torno de 5 ou mais compilações e um total de 55 ou mais erros. Com relação ao conteúdo que foi abordado, neste caso sobre estruturas de repetição, no primeiro exercício constatou-se uma média de 27 erros. Enquanto que no segundo exercício a média total de erros encontrada foi de 6 erros e no terceiro foi de 16 erros.

No segundo teste que foi realizado, foi proposto aos alunos a realização de 4 exercícios sobre o conteúdo de vetores, também variando em nível de complexidade, como no teste anterior. Considerando os resultados obtidos no gráfico da Figura 11, de acordo com o *status* sobre o progresso dos alunos nas atividades, observou-se que 50% dos alunos, apresentaram um desempenho “Muito Insatisfatório”. Enquanto que 37,5% apresentaram um desempenho “Insatisfatório” e 12,5% apresentaram um desempenho “Muito satisfatório”. A partir das observações constatadas com o CFacil+, verificou-se que os alunos apresentaram uma maior dificuldade no primeiro exercício, apresentando uma média de 16 erros para resolução desta atividade. Ao passo que, a média de erros observadas para resolução do segundo exercício foi de 8 erros e do terceiro de 9 erros.

Com base na análise destes resultados, foi possível perceber que os erros relacionados aos conteúdos no primeiro e no segundo teste, se concentraram, na espera de *tokens* como abre e fecha chaves, abre e fecha parênteses e ponto e vírgula no final de comandos.

Considerando-se os resultados obtidos nos processos de avaliação da ferramenta pelos professores, com o uso do CFacil+ a medida que os alunos foram realizando os primeiros exercícios, era possível que o professor acompanhasse os alunos que realizavam as atividades, identificando dentre estes, aqueles que apresentavam maiores dificuldades em sala de aula através da interface de monitoramento que era apresentada pela ferramenta.

## 5. Considerações Finais

Neste trabalho foi apresentada uma ferramenta para o acompanhamento das atividades dos alunos na disciplina de programação. O ferramental implementado foca em oferecer informações úteis sobre o acompanhamento das atividades dos estudantes a partir das compilações que os usuários realizam, numa visão global, mas dando oportunidade para análise individual.

O desenvolvimento do CFacil+ permitiu visualizar um relatório detalhado de cada estudante e de sua produção de exercícios de programação. Esta abordagem retrospectiva, permite que seja possível identificar quadros típicos de alunos que estão adotando comportamentos de risco, por exemplo, realizando poucos exercícios, realizando exercícios errados sem retomá-los, etc. Assim, a aplicação auxilia na orientação dos alunos de forma mais embasada e na reflexão de suas atitudes frente às atividades propostas em aula práticas de Algoritmos e Programação. Ao passo que é possível confrontá-los com as suas próprias dificuldades, o CFacil+ também permite o contraste com a produção da turma.

Dessa forma, pode-se concluir que, a partir do desenvolvimento da ferramenta, é possível o professor atuar de forma rápida nos alunos que mais necessitam de intervenção. Ao mesmo tempo, o professor pode rever práticas próprias de ensino, como por exemplo, ser capaz de se adequar a diferentes ritmos de produção de exercícios dos alunos.

## Referências

- AMARAL, Érico Marcelo Hoff do. Processo de ensino e aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual. 2015.
- BIGGS, J.; TANG, C. Teaching For Quality Learning At University. McGraw-Hill Education, 2011. ISBN 9780335242757.
- FONSECA, J. J. S. Metodologia da pesquisa científica. Apostila. Fortaleza: UEC, 2002.
- GOMES, Marina S.; DO AMARAL, Érico MH. Simplificando a depuração de códigos na Linguagem C - Uma solução para alunos iniciantes. RNOTE, v. 14, n. 1. 2016.
- GUOLI, Zhang; WANJUN, Liu. The applied research of cloud computing platform architecture in the E-Learning area. In: Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on. IEEE, 2010. p. 356-359.
- JAVA. Obtenha Informações sobre a Tecnologia Java, 2017. Disponível em: <[https://www.java.com/pt\\_BR/about/](https://www.java.com/pt_BR/about/)>. Acesso em: 24 Maio 2017.
- LOPES, Patrícia Padula et al. Proposta de um Sistema para o Monitoramento das Atividades de Programação de Alunos Iniciantes. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2017. p. 942.
- MOREIRA, Mireille Pinheiro; FAVERO, Eloi Luiz. Um ambiente para ensino de programação com feedback automático de exercícios. In: Workshop sobre Educação em Computação (WEI 2009). 2009.
- RAABE, André Luís Alice; SILVA, JMC da. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. In: XIII Workshop de Educação em Computação (WEI'2005). São Leopoldo, RS, Brasil. 2005.
- TARDIF, M. Saberes docentes e formação profissional. 3. ed. Petrópolis: Vozes, 2002.
- WAINER, Jacques et al. Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação. Atualização em informática, v. 1, p. 221-262, 2007.
- WATSON, C.; LI, F. W. B. Failure rates in introductory programming revisited. Proceedings of the 2014 conference on Innovation & technology in computer science education. Uppsala, Sweden: ACM: 39-44 p. 2014.