

Proposta de um Sistema para o Monitoramento das Atividades de Programação Para Alunos Iniciantes

Patricia Padula Lopes, Marina Silva Gomes, Thiago Ferreira Dantas, Érico Marcelo Hoff do Amaral

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

{padulalopes, gomes.marina93, thiagodantas923,
ericohoffamaral}@gmail.com

***Abstract.** This work presents a proposal of pedagogical solution that assists in the attendance to the learning difficulties of the students starting from the Algorithms and Programming disciplines. To this end, a monitoring system based on the integration of the CFacil tool (GOMES et al. 2016) is used in a client and server network environment, which will allow the teacher to monitor, in real time, the tasks proposed to the students. students in the classroom, guaranteeing, in this way, the proposition and adjustment of the didactic strategies adopted, aiming at an improvement of the experience of teaching and learning programming.*

***Resumo.** Este trabalho apresenta uma proposta de solução pedagógica que auxilia no atendimento às dificuldades de aprendizagem dos alunos iniciantes das disciplinas de Algoritmos e Programação. Para tanto, utiliza-se de um sistema de monitoramento a partir da integração da ferramenta CFacil (GOMES et al. 2016) em um ambiente de rede cliente e servidor, que irá permitir ao docente o acompanhamento, em tempo real, das tarefas propostas aos estudantes em sala de aula, garantindo, desta forma, a proposição e ajuste das estratégias didáticas adotadas, visando um aprimoramento da experiência de ensino e aprendizagem de programação.*

1. Introdução

O ensino de Algoritmos e Programação, em cursos de Computação ou afins, tem por objetivo estruturar o pensamento do aluno de maneira que ele seja capaz de conseguir utilizar a lógica de programação como ferramenta para a resolução de diversos problemas computacionais, fator importante e necessário para disciplinas mais avançadas (SCOLARI et al. 2007). A importância do aprendizado desses conceitos já nas disciplinas do primeiro semestre é percebida tão logo o aluno inicia o processo de aprendizado de alguma linguagem de programação e precise estruturar os passos para resolução de um problema, repassando-os ao computador, sob a forma de comandos ou instruções de linguagens de programação específicas (SANTOS et al. 2015).

No entanto, o processo de aprendizagem de programação é uma experiência nova, sendo permeada por diversos desafios relacionados principalmente à compreensão de abordagens abstratas e ao desenvolvimento do raciocínio lógico-dedutivo. Contudo, tais habilidades nem sempre são desenvolvidas no ensino básico, levando o aprendiz a

ter grandes dificuldades e conseqüentemente um desempenho não satisfatório durante o curso (GIRAFFA *et al.* 2016).

Aliado a isto, a utilização de aulas expositivas com baixos níveis de interação professor-aluno aumentam o nível de dificuldade para que se tenha êxito no ensino de algoritmos. Todos estes problemas em torno do ensino de programação acabam, segundo Schultz (2003), sendo agravados pela dificuldade dos próprios professores, em identificar e reconhecer nos alunos tais habilidades prévias de forma a aproveitá-las melhor para o desenvolvimento das competências de construção de algoritmos e programação, conjugada, muitas vezes, com turmas contendo um número elevado de aprendizes.

Como forma de amenizar estes problemas, este trabalho tem como objetivo o desenvolvimento de um recurso didático a partir da integração da ferramenta CFacil a um ambiente de rede cliente e servidor. O CFacil, foi proposto por Gomes *et al.* (2016) e é uma proposta focada em auxiliar o aluno, através de um sistema que permite a identificação e correção de erros de compilação em linguagem C mais simples e compreensível aos estudantes, visto que esta é a linguagem padrão onde o projeto foi desenvolvido. Todavia, devido à quantidade de alunos e a diversidade de dificuldades apresentadas durante as aulas de programação, não é possível por intermédio do CFacil possibilitar o acompanhamento individualizado ao professor quanto a identificação das dificuldades apresentadas pelos estudantes em ambientes de laboratório no desenvolvimento de atividades de programação. Reconhecendo esta lacuna, este estudo vislumbra implementar uma solução que visa aprimorar a forma de interação entre aluno e professor em ambientes de laboratório, através do acompanhamento individualizado dos alunos, permitindo, assim, uma maior eficiência na identificação e atendimento das dificuldades de aprendizagem de algoritmos e programação.

Este trabalho está estruturado em 5 seções. 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 de um sistema para monitoramento das atividades de programação para alunos iniciantes em programação e os resultados analisados a partir dele, a modelagem da ferramenta, as técnicas adotadas para o desenvolvimento e a sua implementação. Já a Seção 5 contém os resultados atingidos com os testes de validação realizados. Por fim, a Seção 6 apresenta as considerações parciais.

2. Referencial Teórico

Nesta Seção serão apresentadas as bases teóricas utilizadas para realização deste trabalho. A Seção 2.1 apontará as dificuldades apresentadas pelos alunos e professores no ensino de algoritmos e programação. E, na Seção 2.2, serão abordadas as tecnologias utilizadas para o auxílio ao docente no acompanhamento individualizado dos alunos.

2.1 Ensino de Algoritmos

As disciplinas de algoritmos constituem a base para o ensino de programação e abordam os princípios do desenvolvimento do raciocínio lógico, com o objetivo de desenvolver a capacidade de análise e resolução de problemas dos alunos através da descrição das

soluções na forma de algoritmos. Além disso, esta disciplina faz parte do projeto pedagógico 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.

Os estudantes, na maioria das vezes, não estão habituados com as novas formas de pensar requeridas por estas disciplinas, bem como, não possuem certas aptidões necessárias, encontram dificuldades que, por vezes, culminam na reprovação e evasão (BARBOSA, 2011). Tal problemática tem sido motivo de preocupação dos alunos e dos próprios docentes (VIEIRA *et al.* 2015). Dentre os problemas vivenciados em sala de aula, Ochoa (2011) aponta que professores têm demonstrado dificuldades na elaboração de estratégias no atendimento de alunos de forma individualizada, podendo também, ser uma das causas para a falta de motivação dos alunos na aprendizagem de algoritmos e programação (RAPKIEWICZ *et al.* 2006).

Reconhecendo que existe uma lacuna no processo de aprendizagem para compreensão da lógica de programação, causada pela distância entre professor e aluno durante atividades didáticas e, agravada pela dificuldade dos estudantes em entender as linguagens de programação, observa-se a adoção de recursos computacionais como ferramenta de apoio. Tais ferramentas permitem ampliar a possibilidade de interação entre docentes e discentes.

2.2 Tecnologias para auxílio ao docente no acompanhamento individualizado dos alunos

Diversas ferramentas vêm sendo criadas para o apoio ao processo de ensino-aprendizagem de programação. Várias delas objetivam propiciar aos alunos uma aprendizagem fácil e intuitiva através de sofisticadas tecnologias (TAVARES *et al.* 2013). No entanto, muitos destes recursos não atendem a demanda dos professores a fim de possibilitar o atendimento e o acompanhamento dos estudantes durante as atividades práticas de Algoritmos e Programação propostas em sala de aula. Muitas vezes as demandas com relação à aprendizagem não são detectadas e atendidas em tempo hábil, ocasionando problemas como desmotivação, reprovação ou evasão, devido à quantidade de alunos e a diversidade de dificuldades apresentadas por estes (RAABE *et al.* 2005).

Ao identificar o contexto apresentado, segundo Timmermann *et al.* (2016), quando a interação se torna dispensável para o professor, existe uma grande possibilidade deste não se aperceber das dificuldades ou incompreensões de alguns alunos, sejam elas sobre a aprendizagem de técnicas ou de elaboração de estratégias, limitando, por conseguinte, as possibilidades de mediação. É difícil para um professor levar em consideração o perfil, as metas, as necessidades, as expectativas, as preferências e o nível de conhecimento de cada aluno, de modo a proporcionar um ensino adaptado (FALCKEMBACH *et al.* 2013).

Visando auxiliar os professores das disciplinas iniciais de Algoritmos e Programação quanto ao tratamento personalizado dos discentes e o desenvolvimento de suas habilidades individuais, a ferramenta de software proposta neste trabalho se caracteriza como uma solução que permita ao docente acompanhar as atividades de

programação desenvolvidas pelos alunos em ambientes de laboratório, além de possibilitar ao discente a identificação e correção de erros de compilação em linguagem C mais simples e compreensível.

3. Metodologia

De acordo com os métodos descritos na literatura, a pesquisa realizada neste trabalho constitui-se como aplicada e exploratória, a qual teve por objetivo propor uma solução para auxiliar o docente a acompanhar as atividades de programação desenvolvidas em laboratório, permitindo observar/registrar de forma individual as dificuldades apresentadas pelos alunos em sala de aula. A partir do projeto piloto implementado, realizou-se uma análise quantitativa com o objetivo de avaliar suas funcionalidades. Quanto aos procedimentos técnicos, a pesquisa pode ser classificada em bibliográfica, já que implica no estudo de artigos, teses, livros e outras publicações usualmente disponibilizadas por editoras e indexadas (FONSECA, 2002).

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 para o monitoramento das atividades de programação pelos professores em sala de aula. Na Figura 1 é apresentado um resumo de cada uma destas etapas.

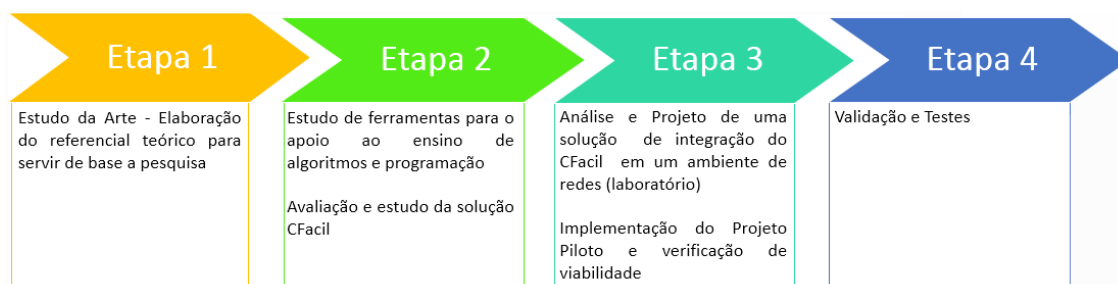


Figura 1. Etapas da Metodologia

Inicialmente, foi realizado um estudo do estado arte com base em alguns repositórios e periódicos, dentre eles, destacam-se: IEEE, ACM, SCIELO, RENOTE, RBIE, SBIE, WALGPROG entre outros. Em relação à segunda etapa, foi feito um levantamento das ferramentas voltadas ao ensino de algoritmos. Paralelo a isso, uma análise da solução CFacil foi realizada, a fim de comparar sua funcionalidade perante a estas tecnologias. A partir deste estudo, na terceira etapa, foi elaborado um projeto de uma solução de integração da ferramenta CFacil em um ambiente de laboratório. Também foram avaliadas e estudadas as formas de integração desta solução em um ambiente de rede (cliente-servidor) para a implementação de um projeto piloto utilizando uma linguagem orientada a objetos - Java. Por fim, o protótipo foi implementado e para que pudesse ser validado, seguiu-se um modelo de Plano de Teste fundamentado na adoção de um processo de teste e nos artefatos sugeridos pela Norma IEEE 829.

4. Desenvolvimento

O objetivo deste tópico é demonstrar como foi concebida a proposta da aplicação para o monitoramento das atividades dos estudantes, com base no projeto CFacil. Na Seção 4.1

é apresentado uma análise da ferramenta CFacil e um melhor detalhamento sobre a integração ao projeto piloto apresentado neste estudo. Na Seção 4.2 é apresentado o desenvolvimento do projeto piloto e os resultados analisados a partir dele.

4.1. Análise da ferramenta CFacil e a proposta da solução de monitoramento

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. Com esta solução, o estudante, em seus primeiros contatos com a linguagem de programação, pode ter maior facilidade na abstração da complexidade envolvida na correção de erros durante o processo de programação, estimulando, desta forma, a construção de conhecimento sobre a lógica envolvida na implementação de soluções computacionais para problemas simples. Foi possível perceber que o CFacil está focado nas dificuldades encontradas pelo aluno na disciplina de algoritmos e programação, não dando atenção aos problemas enfrentados pelo professor no ensino individualizado necessário nesta disciplina.

O CFacil 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. O arquivo de *logs* gerado apresenta padrões em sua construção, como armazenamento de data e hora da compilação, a frase “Erro na linha x”, e um contador de erros que é apresentado ao final de cada compilação. Ao total 32 mensagens de erros podem ser apresentadas. Porém, por ser uma ferramenta para alunos iniciantes, não contempla análises para ponteiros e estruturas, conteúdos que não são apresentados nas disciplinas de algoritmos e programação.

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. Dessa forma, o professor conseguirá reconhecer as dificuldades e os erros mais comuns dos discentes na aprendizagem

inicial da programação, acompanhando os passos do estudante dentro do ambiente e registrando seu progresso por meio das atividades a serem desenvolvidas, garantindo, desta forma, a proposição e ajuste das estratégias didáticas adotadas, visando um aprimoramento da experiência de ensino e aprendizagem na disciplina de Algoritmos e Programação.

4.2. Projeto piloto

Tendo como propósito fornecer ao professor um método que permitisse o monitoramento das atividades desenvolvidas pelos alunos durante as aulas práticas de algoritmos e programação, realizou-se a integração da ferramenta CFacil a um ambiente de rede (cliente e servidor) exemplificado a partir da Figura 2. Para concepção desta ferramenta, utilizou-se linguagem de programação Java e a API de sockets java.net para realizar a coleta, análise e envio dos *logs* de erros via rede, emitidos pelo CFacil após a depuração de códigos em linguagem C realizada pelos alunos durante as práticas de programação.

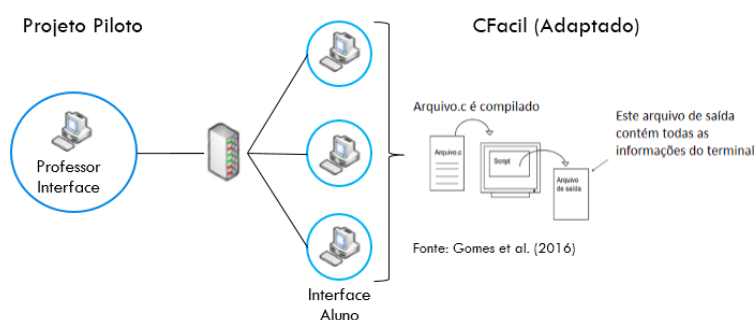


Figura 2. CFacil integrado a um ambiente de rede

A partir da análise dos *logs* de erros emitidos ao sistema através da compilação dos arquivos “.c” pelos alunos, o professor, no lado servidor da aplicação, poderá verificar as dificuldades que os estudantes estão apresentando durante a programação de algoritmos em linguagem C em cada uma das tarefas propostas em sala de aula. Isto é possível, através da apresentação de um relatório individualizado de cada estudante presente na turma, demonstrado a partir da Figura 3.

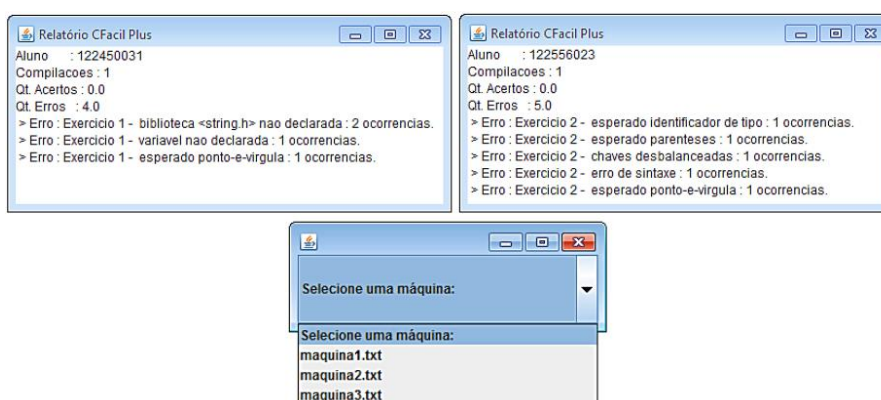


Figura 3. Tela de apresentação dos relatórios no CFacil+ ao professor

Neste relatório, a aplicação do servidor (através da qual o professor observa aos logs de compilação dos alunos) identifica cada estudante através do seu número de

matrícula e, a cada nova compilação, é especificada a quantidade total de erros e acertos cometidos pelos discentes e o tipo de erro referente a cada exercício que foi compilado para que o professor consiga monitorá-los. No lado cliente, o processo de registro do aluno mostrado na Figura 04, implica nele informar ao sistema o seu nº da matrícula e, para a compilação dos algoritmos implementados, é preciso que o estudante especifique o nome do arquivo “.c” e a numeração do exercício que está sendo solucionado. A cada nova compilação, o aluno deve seguir informando ao sistema em qual exercício ele está, se houve troca ou não de atividade. No terminal de compilação, inicialmente, é realizada a análise do código fonte enviado pelos alunos, para avaliar se foi encontrado algum erro durante a compilação dos arquivos “.c”. Se forem encontrados erros, eles serão apresentados no terminal através de mensagens como apresentado na Figura 04 e através de um relatório de erros como mostrado na Figura 5, que permitem a identificação imediata das falhas que foram cometidas na implementação.

```

1 #include <stdio.h>
2 #include<stdlib.h>
3
4 int main(){
5     float nota1, nota2, media;
6     puts ("Digite a primeira nota\n");
7     scanf ("%f", &nota1);
8
9     puts("Digite a segunda nota\n");
10    scanf("%f", &nota2);
11    media = (nota1 + nota2) / 2;
12    printf("A media do aluno foi %f\n", &media);
13
14    system("pause");
15 }

```

```

Informe o arquivo.c a ser compilado, o nr do exercicio e a sua matricula:
testel.c 1 122450031
Com o CFACILPLUS
Erro 1: Na Funcao main - Erro na linha 5: biblioteca <string.h> nao declarada
LOCAL DO ERRO:
  puts
Erro 2: Na Funcao main - Erro na linha 8: biblioteca <string.h> nao declarada
LOCAL DO ERRO:
  puts(
Erro 3: Na Funcao main - Erro na linha 11: variavel nao declarada
LOCAL DO ERRO:
  o aluno foi %f\n", &
Erro 4: Na Funcao main - Erro na linha 13: esperado ponto-e-virgula
LOCAL DO ERRO:
  system(

```

Figura 4. Interação dos alunos com a solução de monitoramento

Dentre as funcionalidades previstas para o software estão a capacidade de classificar as informações de *logs* coletadas, gerar relatórios a partir da análise das informações recebidas pelo sistema e realizar o monitoramento das atividades desenvolvidas pelos alunos.

```

<compilacao metodo='CFACILPLUS' data='qua, 17 de mai de 2017 18:00:05' exercicio='1' aluno='122450031' >
Erro 1: Na Funcao main - Erro na linha 5: biblioteca <string.h> nao declarada
LOCAL DO ERRO:
  puts
Erro 2: Na Funcao main - Erro na linha 8: biblioteca <string.h> nao declarada
LOCAL DO ERRO:
  puts(

```

Figura 5. Relatório de erros emitido ao aluno

Para que estes recursos possam ser oferecidos por intermédio da ferramenta proposta, após ser estabelecida uma conexão com o servidor, o sistema envia ao professor as informações do cliente (aluno), onde contabiliza o total de erros, acertos, número de compilações, tipo de erro, nome e matrícula do aluno a partir de cada um dos relatórios emitidos pelos estudantes. Além de apresentar em telas separadas estes relatórios, como apresentado anteriormente, o sistema também gera um arquivo de saída para ser enviado ao docente. Na Figura 6, é mostrado a arquivo que é emitido ao professor, e que contém uma relação final de todos os erros e acertos e ocorrências de cada erro apresentados por cada um dos alunos conectados na rede.

```

Aluno      : 122450031
Compilacoes : 1
Qt. Acertos : 0.0
Qt. Erros   : 4.0
> Erro : Exercício 1 - biblioteca <string.h> nao declarada : 2 ocorrencias.
> Erro : Exercício 1 - variavel nao declarada : 1 ocorrencias.
> Erro : Exercício 1 - esperado ponto-e-virgula : 1 ocorrencias.

```

Figura 6. Arquivo de saída enviado para o professor

A Figura 7 apresenta a arquitetura (cliente servidor) da aplicação apresentada. Na aplicação, o sistema monitora as compilações a partir de um módulo denominado monitor, e gera estatísticas de erros através de um analisador, que permitem o professor acompanhar a evolução dos alunos. O analisador identifica o aluno através do seu número de matrícula, computa a quantidade total de compilações realizadas, a quantidade total de erros e acertos, informa a descrição e a ocorrência dos erros cometidos em cada exercício e gera um relatório de erros do aluno. Este arquivo é atualizado a cada nova compilação a partir de seu monitoramento.

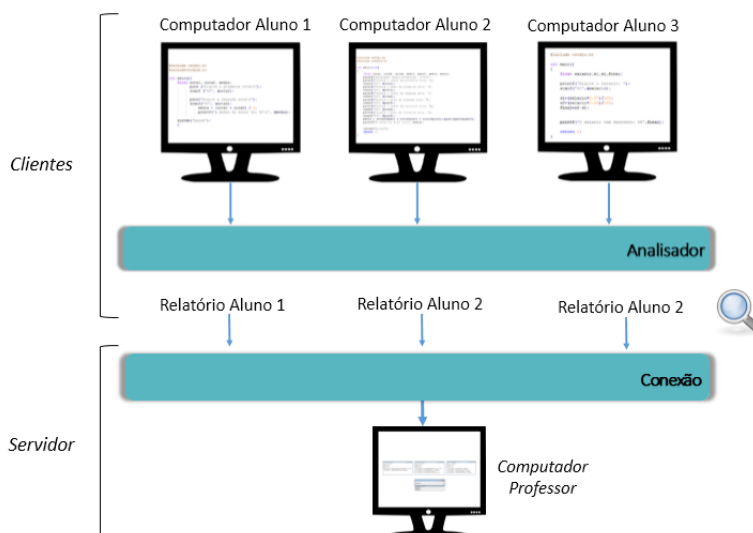


Figura 7. Arquitetura (cliente servidor) do projeto piloto

A fim de permitir a conexão de todos os alunos ao sistema de monitoramento (Cliente/Servidor), adotou-se a tecnologia de threads, garantindo desta forma um padrão eficiente e confiável de comunicação entre as aplicações.

5. Resultados Parciais

A fim de comprovar a viabilidade do projeto piloto, será apresentado o comportamento da ferramenta quando a mesma é executada. Inicialmente, é preciso rodar a aplicação cliente em uma máquina e o servidor em outra. Neste teste, foram executados três clientes simultaneamente. Quando um arquivo.c é compilado em um dos clientes, o sistema detecta a partir da alteração do relatório de erros que é gerado. Por meio da análise, que faz a contagem do número de compilações, erros e acertos por exercícios compilados, o sistema confirma que o relatório para o servidor será enviado. Na Figura 8 é possível observar este comportamento, a partir dos registros dos processos de cada um dos clientes no envio do relatório geral ao servidor após o monitoramento e análise dos dados coletados.

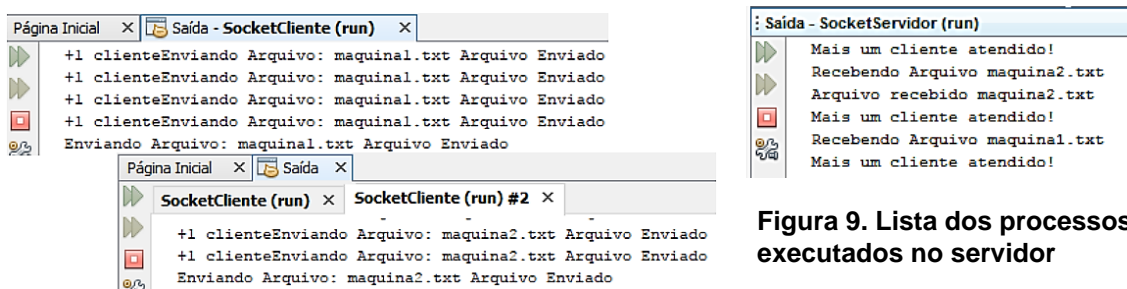


Figura 9. Lista dos processos executados no servidor

Figura 8. Lista dos processos executados no cliente

O servidor, rodando em paralelo, em outro computador, identifica o arquivo e confirma seu recebimento, informando que um novo cliente está sendo atendido, conforme é mostrado na Figura 9.

Nestes testes foram compilados três códigos implementados em linguagem C. Os arquivos foram executados na máquina em que os clientes estavam sendo executados. No lado servidor da aplicação era mostrado o monitoramento da máquina cliente a partir da apresentação do arquivo de saída que era enviado periodicamente. Uma lista com os clientes que estavam em atividade no momento deveria ser exibida. Enfim, ao analisar os resultados obtidos com o processo de monitoramento, é possível perceber que o projeto piloto implementado satisfaz as expectativas dos possíveis usuários do sistema, onde os alunos poderão interagir com a ferramenta a partir do módulo cliente e o professor através do módulo servidor.

6. Considerações Finais

O desenvolvimento do projeto piloto da ferramenta 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 as 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 parcial 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. Como objetivo futuro, pretende-se finalizar o projeto piloto e aplicá-lo a uma turma de Algoritmos e Programação para validar as propostas apresentadas neste trabalho.

Referências

- BARBOSA, Leônidas da Silva. **Aprendizado significativo aplicado ao ensino de algoritmos**. Dissertação de Mestrado, UFRN, 2011.
- FALCKEMBACH, Gilse Morgental; DE ARAUJO, Fabrício Viero. **Aprendizagem de algoritmos: dificuldades na resolução de problemas**. Anais Sulcomp, v. 2, 2013.

- FONSECA, J. J. S. **Metodologia da pesquisa científica**. Apostila. Fortaleza: UEC, 2002.
- GIRAFFA, Maria Martins; DA COSTA MORA, Michael. **Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno**. In: Congresso CLABES, 2016.
- GOMES, M.S; AMARAL E.M.H. **Uma Proposta de Ferramenta para Simplificar a Depuração de Códigos em C, por Alunos Iniciantes**. Anais dos Workshops do Congresso Brasileiro de Informática na Educação. Vol. 5. No. 1. 2016.
- IEEE Computer Society. IEEE Std 829: **Standard for Software Teste Documentation**. 1998.
- JAVA. Obtenha Informações sobre a Tecnologia Java, 2017. Disponível em: <https://www.java.com/pt_BR/about/>. Acesso em: 24 Maio 2017.
- 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.
- RAPKIEWICZ, E.C; FALKEBACH G; SEIXAS, L; ROSA, N.S; CUNHA, V.V; KLEMMANN, M. **Estratégias Pedagógicas No Ensino De Algoritmos e Programação Associadas Ao Uso De Jogos Educacionais**. RENOTE, Rio Grande do Sul, v.4 n.2, 2006.
- SANTOS, Antunes et al. **A Importância do Fator Motivacional no Processo Ensino-Aprendizagem de Algoritmos e Lógica de Programação para Alunos Repetentes**. In: WEI-Workshop sobre Educação em Computação. (2015). 2015. p. 1-10
- SCOLARI, Angélica Taschetto; BERNARDI, Giliane; CORDENONSI, Andre Zanki. **O desenvolvimento do raciocínio lógico através de objetos de aprendizagem**. RENOTE, v. 5, n. 2, 2007.
- TAVARES, O. L. et al. **Uma arquitetura pedagógica auxiliada por tecnologias para ensino e aprendizagem de programação**. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), p. 778-787, 2013.
- TIMMERMANN, Glaucia Keidann; GONZÁLEZ, Fernando. **Mediações que os professores e alunos estabelecem com o conteúdo da disciplina de Algoritmos de cursos superiores: estudo de caso**. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação, p. 1295, 2016.
- VIEIRA, Carlos Eduardo Costa et. al. **Dificuldades no Processo de Aprendizagem de Algoritmos: uma Análise dos Resultados na Disciplina de AL1 do Curso de Sistemas de Informação da FAETERJ–Campus Paracambi**. Cadernos UniFOA, n. 27, p. 5-15, 2015.
- WAINER, Jacques et al. **Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação**. Atualização em informática, v. 1, p. 221-262, 2007.