

## DESENVOLVIMENTO DE UM SISTEMA AUTOMATIZADO PARA GERAÇÃO E COMPARAÇÃO DE RESULTADOS NA EXECUÇÃO DE PROGRAMAS

André Coelho Xavier, Raul Coelho Viana Faria, Victor Hugo Siqueira Alencar, Wagner dos Santos Clementino de Jesus.

Fundação Valeparaibana de Ensino/Colégio Técnico Unidade Centro, Avenida Paraibuna, 75, Centro - 12245-020 - São José dos Campos-SP, Brasil, [andreexav@gmail.com](mailto:andreexav@gmail.com), [raulcvf.official@gmail.com](mailto:raulcvf.official@gmail.com), [vhsa2007@gmail.com](mailto:vhsa2007@gmail.com), [wagner@univap.br](mailto:wagner@univap.br).

### Resumo

Discutir a forte demanda por aplicações que facilitem o trabalho dos docentes nas escolas, onde observa-se a necessidade da implementação de um sistema automatizado para correção de atividades de programação, utilizando tecnologias avançadas para melhorar a eficiência e precisão no processo de avaliação. A solução envolve o uso de um sistema computacional que analisa resultados dos códigos de programação enviados pelos estudantes, identificando erros de lógica e eficiência, auxiliando os alunos na correção de seus erros e no aprimoramento de suas habilidades de programação. Para o desenvolvimento da aplicação, foram empregadas técnicas de análise e comparação de código, garantindo que o corretor possa lidar com uma ampla variedade de problemas e linguagens de programação. Os resultados demonstram que o sistema é capaz de identificar erros com precisão, no intuito de conduzir a tecnologia como agente facilitador, corroborando as hipóteses que motivaram a criação do projeto e mostrando seu potencial para revolucionar a forma como a avaliação em cursos de programação é realizada.

**Palavras-chave:** Corretor. Programação. Automatização.

**Curso:** Ensino Médio / Técnico em Informática.

### Introdução

Atualmente, o computador representa, para a maioria das pessoas, um simples instrumento capaz de solucionar problemas desde os mais simples até os mais complexos. Também é capaz de sistematizar, coletar, manipular e fornecer os resultados em forma de informação para os usuários (Cunha, et.al, 2017). Para o desenvolvimento de um sistema necessita-se do conhecimento de programação, nesse quesito os sistemas de algoritmo são uma linguagem intermediária entre a linguagem humana e as linguagens de programação sendo amplamente utilizado para representar a solução de um problema. Assim descrevendo instruções a serem executadas pelos computadores (Vasconcellos, 2019).

A correção automática de programas escritos pelos estudantes é foco de frequentes pesquisas visando atender melhor ao aprendiz de programação fornecendo feedback automático de suas atividades. As técnicas mais utilizadas para correção automática de programas são a análise dinâmica e análise estática. Na análise dinâmica o algoritmo é executado com entradas pré-definidas e suas saídas são comparadas com as saídas esperadas. (ALA-MUTKA.,2005).

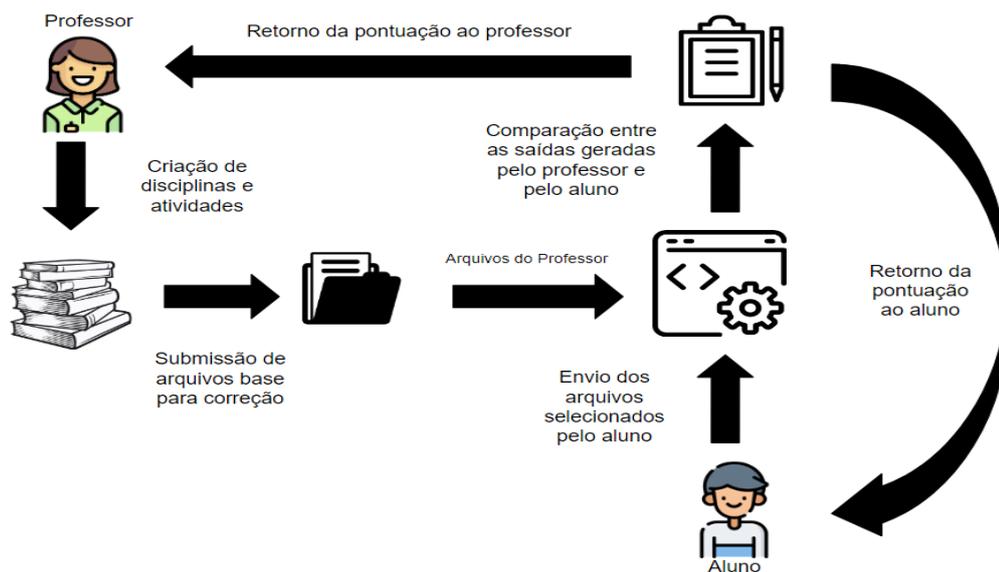
No contexto educacional, os professores possuem papel essencial no ensino dessas metodologias para que os estudantes consigam produzir um sistema funcional, no entanto há um consenso entre os docentes de disciplinas de informática em torno de que é necessário trabalhos práticos para que os alunos consigam assimilar o conhecimento transferido (Tomayko.,1987). As aulas práticas podem ajudar no desenvolvimento de conceitos científicos, além de permitir que os estudantes aprendam como abordar de maneira assertiva os problemas propostos na programação e como desenvolver soluções para problemas complexos de forma eficiente (Lunetta, 1991). Entretanto, os educadores constataam dificuldades para a correção das listas propostas em aula, pois existe uma grande demanda de atividades para serem corrigidas em um determinado período, gerando assim um acúmulo de tempo exacerbante para corrigir atividades postadas pelos estudantes das disciplinas de programação.

Diante disso, alguns dos problemas resultantes, são: a dificuldade do docente em compreender a lógica do aluno, visto que é complexo a mudança de um raciocínio lógico depois de construído acarretando em uma difícil tarefa para o professor de compreender a lógica específica de cada aluno, turmas onde existem diferentes níveis de conhecimento entre os alunos e ritmo de aprendizagem, pois alguns alunos possuem contato prévio com a programação antes de ingressar no curso enquanto outros não tiveram contato com algoritmos e programação (SIEBRA,2009). Sendo assim, a pesquisa tem como objetivo analisar os resultados de códigos implementados nas linguagens de programação C#, C++, Python e Java utilizando métodos de compilação de código com técnicas de processamento, já vistas em outras aplicações técnicas de destaque como as maratonas de programação (obi, icpc, etc) através do BOCA por exemplo. Com isso, será criado um sistema automatizado para geração e comparação de resultados e execução de programas visando a melhor produtividade e eficácia da correção das atividades.

### Metodologia

Com o propósito de efetivar a análise dos dados, foi construído um algoritmo utilizando a linguagem de programação Node.js com a finalidade de automatizar o processo de correção das atividades, através dos compiladores e interpretadores das linguagens C++ (g++), C# (gsc), Java (javac) e Python (python) os arquivos enviados pelos usuários da plataforma possuem baterias de testes predefinidas pelo professor administrador onde seus arquivos são testados e com isso são gerados arquivos texto com os resultados e comparados automaticamente com as respostas geradas de um algoritmo base produzido pelo professor para a correção dos exercícios. No intuito de dinamizar a aplicação foi utilizado o padrão RESTful API para manutenção das rotas de maneira organizada a relação entre professor e aluno dentro do sistema.

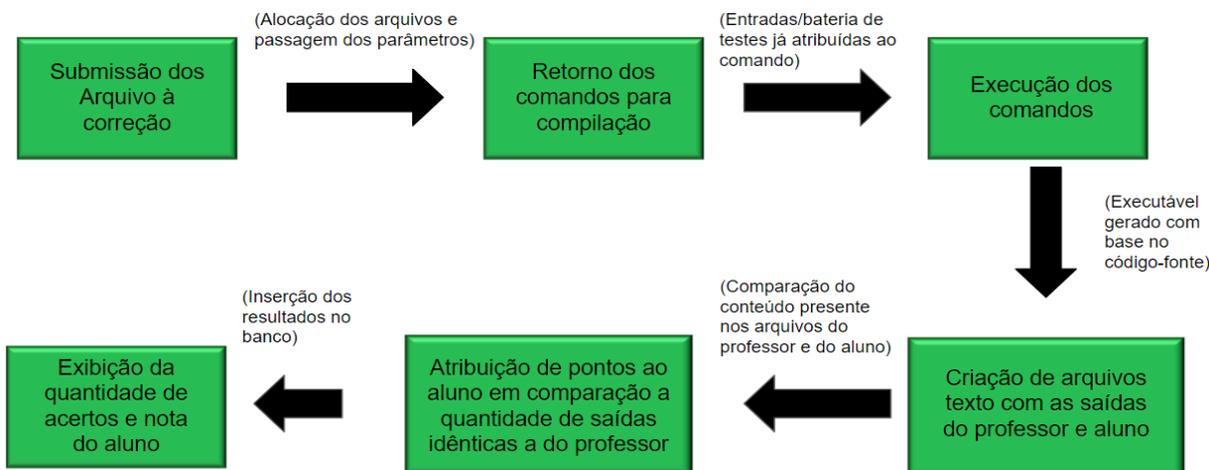
Figura 1 - Diagrama de funcionamento geral do sistema.



Fonte: Autor (2024).

A figura 2 apresenta a ordem sequencial na utilização do comparador durante o processo de avaliação de uma atividade para o aluno, os arquivos para correção são submetidos às seguintes fases: Submissão dos arquivos para correção, compilação dos arquivos, execução dos comandos predefinidos pelo professor, criação de arquivos texto com as saídas, comparação dos arquivos texto com as saídas esperadas, para que assim haja a atribuição e exibição da nota dos alunos.

Figura 2 - Diagrama da sequência de processos de funcionamento do Software.



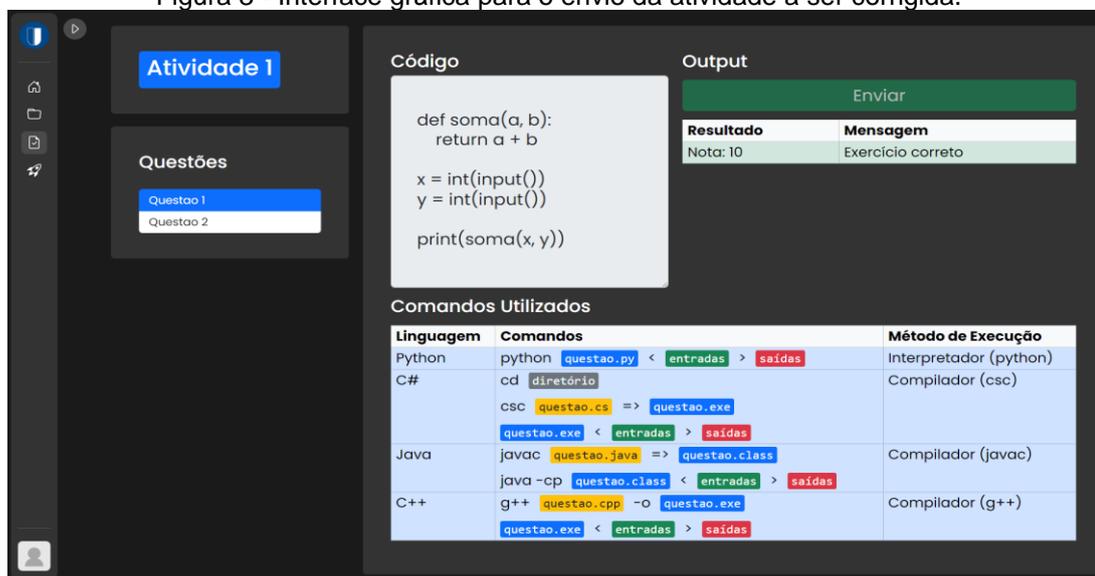
Fonte: Autor (2024).

Após os procedimentos principais do software são armazenadas as notas acopladas aos arquivos da atividade enviados pelos alunos de uma determinada disciplina, o professor pode acessar estes arquivos na área das disciplinas e gerar planilhas com as informações de um ou vários alunos com sua respectiva nota em cada atividade.

## Resultados

Para o pleno funcionamento da aplicação, é obrigatório que o aluno submeta o arquivo de programação a ser corrigido, assim, gerando um chamado para o código de compilação e comparação, onde serão testadas as devidas entradas, definidas previamente pelo professor, posteriormente capturando as respectivas saídas do código do aluno que por fim serão comparados com as do professor e assim retornando a nota referente a questão. As telas seguem um padrão visual característico, como o proposto pela figura 3.

Figura 3 - Interface gráfica para o envio da atividade a ser corrigida.



Fonte: Autor (2024).

Após a conclusão da comparação dos arquivos da respectiva atividade é capturado o tempo de execução do código enviado pelo aluno, que sequencialmente é comparado a um tempo máximo de execução permitido, a fim de verificar a eficiência de sua lógica. A seguir, as informações referentes a esses processos, como a nota do aluno e o código enviado, são armazenadas e disponibilizadas aos usuários. Em consonância a esse procedimento, o professor administrador tem a opção de gerar planilhas e gráficos que retornem dinamicamente o resultado geral por aluno das atividades propostas.

## Discussão

O estudo apresentado para automatização da correção dos códigos de programação tem papel de grande importância educacional, assim os professores não precisam levar as atividades para corrigir em casa nem o aluno esperar pelo resultado, já que a plataforma os corrige instantaneamente. Isso também torna as aulas mais atrativas. Os estudantes entendem que os recursos tecnológicos não são apenas para as redes sociais e jogos, mas que é possível aprender, refletindo no rendimento positivo dos alunos (SEED, 2015).

Em uma nova fase o presente trabalho deve implementar uma ferramenta para que a limitação do tempo para a entrega das atividades seja automática de modo que os arquivos serão retirados do próprio Google Sala de Aula e já corrigidos na aplicação, para que haja melhor interação com diversos usuários e assim minimizando o envio de dados de forma manual bem como a transição entre sites para submissão dos arquivos.

## Conclusão

Em síntese, diante dos resultados apresentados, vale concluir que os principais objetivos do sistema foram materializados de forma eficaz, através do uso da aplicação para criar um ambiente mais atrativo para a integração entre professor e aluno. Ainda assim, obteve-se sucesso no uso da ferramenta para o desenvolvimento de uma forma dinâmica para o retorno de notas ao aluno de maneira automática, através do modelo de comparação de arquivos. Além dos destaques anteriores, vale ressaltar a otimização do tempo demandado pelos professores para a correção de atividades.

## Referências

ALA-MUTKA, K. M., **A Survey of Automated Assessment Approaches for Programming Assisments**. Computer Science Education, p. 83-102, 2005.]

Cunha, G. B.; Macedo, R. T. Silveira, R. S. **INFORMÁTICA BÁSICA**. Santa Maria: [s.n.], 2017. 1 v., p. 14.

LUNETTA, V. N. The school science laboratory: historical perspectives and contexts for contemporary teaching. International handbook of science education, p. 249-262, 1998.

Siebra, S. A. e Silva, D. R. D. (2009) "Prática de Ensino de algoritmos", Volume 2, **Universidade Federal Rural de Pernambuco**, <http://www.slideshare.net/aej400aej/prtica-de-ensino-de-algoritmo-volume-1-e-2>, outubro.

SEED: Secretaria de Estado da Educação. **Tecnologias transformam metodologia de ensino em escolas** estaduais. Disponível em: <http://www.aen.pr.gov.br/modules/noticias/article.php?storyid=99560&tit=>. Acesso em: 09/08/24.

Tomayko, J. E. Teaching a Project-Intensive Introduction to Software Engineering. **Carnegie Mellon University, Software Engineering Institute**, CMU/SEI-87-TR-20, August 1987.

Vasconcellos I. L. B. **Apostila de Lógica de Programação**. Rio de Janeiro: [s.n.], 2019.