

Inspetor Online

Francisca Leisia M. Nascimento, Jonathan B. O. Silva, Leonardo Lucena

Diretoria Acadêmica de Gestão e Tecnologia da Informação – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

{jonathan.batista, leisia.medeiros}@academico.ifrn.edu.br,
leonardo.lucena@ifrn.edu.br

Abstract. *The Inspetor Online is a web system that assists teachers in the correction of tests and exercises of computer programming, avoiding manual operations and reducing the deadlines for making results available to students. The tool is also critical for students as they will have instant feedback on the questions answered. The students can answer the questions initially offered by the teacher, and have the possibility to make new attempts at the resolution of the questions whose result was not satisfactory.*

Resumo. *O Inspetor Online é um sistema web que auxilia os professores na correção de provas e exercícios de programação de computadores, evitando operações manuais e reduzindo os prazos para disponibilização de resultados aos alunos. A ferramenta também é fundamental para os alunos, pois eles terão um feedback instantâneo das questões respondidas. Os alunos podem responder as questões disponibilizadas inicialmente pelo professor, e tendo também a possibilidade de realizar novas tentativas na resolução das questões cujo resultado não foi satisfatório.*

1. Introdução

Buscamos o tempo inteiro ferramentas que nos auxiliem nas atividades do dia-dia. A intenção cada vez mais é substituir o trabalho manual pelo uso de ferramentas que possam nos ajudar nesse contexto. Atualmente, existem ferramentas online de correção de código, como juízes online, muito utilizadas na preparação dos alunos para olimpíadas de programação. E no contexto educacional? Se trabalhamos com disciplinas que possuem essas questões, como podemos usar uma ferramenta como essa?

A prática e a repetição regular de exercícios são essenciais para a aprendizagem de programação [Paes et al. 2013, apud Galvão et al. 2016, p. 141]. Errar faz parte do processo de aprendizagem. Por isso, um rápido feedback é essencial para que os alunos possam perceber os erros e assim evitar que fiquem frustrados na tentativa de aprender os princípios da programação de computadores [Pelz et al. 2012, apud Galvão et al. 2016, p. 141].

Nesse contexto, desenvolvemos uma ferramenta que auxilie na correção das atividades atribuídas pelos professores de disciplinas de programação. A ferramenta automatiza suas atividades, traz mais agilidade ao processo e, conseqüentemente, proporciona ao aluno uma maior interatividade na resolução dos problemas, podendo acompanhar seus resultados em tempo real. Essa ferramenta, chamada Inspetor Online, é um sistema de avaliação automática de questões de programação, seguindo a mesma abordagem dos Juízes Online.

Um Juiz Online, também chamado de sistema de avaliação automática, contém um conjunto de problemas de programação que devem ser resolvidos pelos usuários. O sistema corrige automaticamente aos códigos-fonte enviados através do uso de casos de teste. A correção é realizada comparando a saída gerada pela execução do programa com a saída esperada. [Kurnia et al. 2001, apud Francisco et al. 2016, p. 12].

2. Contexto de uso e público alvo

As plataformas de juízes online são geralmente utilizadas na preparação dos usuários para competições, como por exemplo, maratonas de programação. A proposta do inspetor é o foco em utilização dentro do ambiente educacional, ou seja, em sala de aula. Ele visa sempre facilitar o trabalho do professor e proporcionar imersão do aluno no ambiente sala de aula, e não somente preparação para competições externas. Isso traz a sensação de ambiente exclusivo, e que pode ser aplicado de acordo com a metodologia do professor.

O inspetor online surgiu da necessidade de auxiliar os professores nas correções de provas e exercícios das disciplinas de programação com auxílio de uma ferramenta que poderá realizar esse trabalho de forma instantânea.

"[...] em uma turma heterogênea, o planejamento e correção de problemas gera uma sobrecarga no professor, dificultando o feedback aos alunos, o que pode causar frustrações no processo de ensino-aprendizagem". [Vier et al. 2015 apud Francisco et al. 2016, p. 11].

Buscamos um processo onde possamos diminuir a preocupação com prazos para disponibilizar notas e gabaritos, aumentar a produtividade do professor, e motivar o aluno no aprendizado de programação com auxílio da ferramenta. Após a submissão da questão - realizada pelo aluno - o Inspetor Online já disponibiliza o resultado que poderá ser visualizado e ficará disponível no painel do aluno.

O professor cadastrará suas listas de questões – contendo casos de teste e gabarito para o aluno visualizar posteriormente – que serão disponibilizadas para que os alunos cadastrados no sistema possam responder. As respostas dos alunos são comparadas com os casos de teste, inserido pelo professor anteriormente, e dessa forma o aluno obtém o resultado instantaneamente da questão que foi respondida, podendo realizar outras tentativas caso não obtenha sucesso.

O aluno no seu painel, visualizará as listas contendo as questões cadastradas pelos professores, e poderá selecionar a questão que desejará responder, e irá enviar sua resposta na linguagem de programação previamente selecionada por ele. Assim, o inspetor fará seu trabalho de compilar o código do aluno na linguagem escolhida, e comparar os resultados com os casos de teste cadastrados pelo professor referente aquela questão.

Nesse mesmo sentido, Ihantola et al. (2015) afirma que *“a avaliação contínua durante um curso de programação garante que os estudantes pratiquem bastante, bem como obtenham feedback sobre a qualidade de suas soluções”*. (apud Galvão et al. 2016, p. 141). A intenção é auxiliar tanto o aluno – que poderá receber feedback sobre sua pontuação de forma rápida e precisa – quanto o professor, que não precisará passar horas fazendo correções manuais de provas e trabalhos de programação.

3. Desenvolvimento

O Inspetor Online foi desenvolvido utilizando a linguagem de programação Scala [Odersky, Martin, Lex Spoon e Bill Venner, 2016], em conjunto a framework Play [Hilton, Peter, Erik Bakker e Francisco Canedo, 2013] e o banco de dados PostgreSQL.

Primeiramente foram desenhadas as telas da aplicação, que seguem o design material da Google. Foi estabelecido em reunião o uso da framework Materialize (uma framework baseada no design material) na implementação do front-end. Todas as telas receberam uma versão estática para que em seguida a lógica fosse implementada no back-end.

No back-end, o desenvolvimento deu-se início na conexão ao banco de dados, para facilitar o acesso aos dados e a criação de consultas de forma funcional (um dos paradigmas da linguagem Scala), foi integrado a aplicação a biblioteca Slick, que permite utilizar o Scala ao invés do SQL para realizar consultas, permitindo trabalhar com o banco, como se estivesse trabalhando com coleções da linguagem.

Para a autenticação dos usuários ao sistema, foi integrado o Silhouette, uma biblioteca para framework Play que suporta vários métodos de autenticação, entre eles a autenticação básica utilizada pelo sistema. O usuário autenticado é identificado entre 3 papéis: administrador, aluno e professor, isso é fundamental para o sistema identificar qual módulo será utilizado.

As bases dos módulos de aluno e professor, foram desenvolvidos paralelamente. No módulo aluno, foi implementada a lógica de envio das respostas primeiro, tendo em mente um editor online que suportasse as linguagens interpretadas pelo sistema. Utilizou-se o ACE editor, um editor embutido em Javascript que suporta por padrão as linguagens Ruby, Python e Scala.

Ao desenvolver o módulo professor, foi pensado como os testes seriam esquematizados pelo usuário. Seguindo o padrão de juízes online, implementou-se uma rotina de teste que executa um código enviado pelo aluno, com as entradas definidas pelo professor gerando saídas, que são realizadas comparadas e categorizadas. Definiu-se o envio de arquivos de teste com uma sintaxe simples, onde o professor define a entrada e a saída no mesmo arquivo, com cada arquivo representando um caso de teste.

4. O Inspetor Online

O sistema possui dois módulos distintos, um para professores e outro para estudantes, que são executados de acordo com os privilégios da conta do usuário. Para acessar o sistema o usuário deve estar cadastrado no sistema da instituição de ensino. Esse sistema faz uso do Google Classroom (um serviço de gerenciamento de turmas virtuais), onde suas informações serão adquiridas, através da API disponibilizada pelo serviço.

No módulo do professor, a funcionalidade principal é o gerenciamento de listas de exercícios. As listas servem somente para agrupar questões e definir um assunto geral (Figura 1).

Nova lista de exercícios

Nome: _____

Assunto: _____

ENVIAR **VOLTAR**

Figura 1. Tela de cadastro de lista

Quando cadastrando uma questão (Figura 2), o professor deve providenciar arquivos de teste para aplicação. Cada arquivo gera uma rotina onde são comparadas as saídas geradas pelo código do aluno. Uma linguagem de marcação simples foi implementada para interpretar esses arquivos.

Incluir nova questão

Número: _____

Enunciado: _____

Gabarito
Coloque o código nesta região

Exemplo de entrada:
Coloque a entrada nesta região

Exemplo de saída:
Coloque a entrada nesta região

ARQUIVO Enviar um ou mais arquivos de teste

Figura 2. Tela de cadastro de questão

No módulo aluno, o usuário seleciona entre as listas cadastradas pelo professor para responde uma questão. Informações como exemplos de entrada e saída são informadas, permitindo o aluno testar o seu código antes de enviar uma resposta. O sistema providencia um editor online (Figura 3) que realiza o destaque de sintaxe de acordo com a linguagem. As linguagens de programação disponíveis para resolução das questões são: Ruby, Python, Scala.

Nova Resposta

Linguagem
 Ruby

```

21 def run
22   n = STDIN.gets.strip.to_i
23
24   rects = []
25   1.upto(n) do
26     x1, y1, x2, y2 = STDIN.gets.strip.split(' ').map &:to_i
27     rects << Rect.new(x1,y1,x2,y2)
28   end
29
30   events = {}
31   rects.each_with_index do |r, i|
32     events[r.x1] ||= []
33     events[r.x2] ||= []
34     events[r.x1] << Event.new(:in, i)
35     events[r.x2] << Event.new(:out, i)
36   end
37
38   area = 0
39   active = []
40   x = 0
41   events.keys.sort.each do |event_x|
42     dirty = false
43     x_now = event_x
44     unless active.empty?
45       area += get_area(active, x, x_now)

```

ENVIAR VOLTAR

Figura 3. Imagem ilustrando resolução de uma questão

Após enviar o código fonte para o sistema, uma rotina de testes é executada, onde uma nota e um estado é atribuído ao resultado. O estado desse envio é classificado entre 5 categorias que ajudam o aluno a identificar, caso necessário, a diferença entre a saída gerado pelo seu código e a saída esperada (veja Tabela 1), deixando transparente o que deve ser alterado para que uma nota maior seja alcançada. Ao contrário de outros juízes, o Inspetor Online, mantém um sistema de notas mais simples que espelha mais a realidade do aluno, onde retirada a porcentagem de testes passados com sucesso uma nota de 0 a 10 é atribuída a resposta. O resultado fica disponível ao aluno na sua tela principal (Figura 4).

Tabela 1. Estados atribuídos a uma resposta pelo sistema

Categoria	Descrição
OK	Código passou com sucesso por todos os testes.
C	A saída gerada pelo código comparada a saída de um teste possui diferenças entre letras maiúsculas e minúsculas
S	A saída gerada pelo código comparada a saída de um teste possui espaços brancos a mais.
CS	Combinação da categoria C e S.
E	A saída gerada pelo código comparada a saída de um teste está totalmente diferente.

Questoes Resolvidas

Lista	Questão resolvida	Estado	Nota	Sua resposta
Lista 1	1	OK	10.0	Ver
Lista 1	1	C	8.0	Ver
Lista 1	1	CS	8.0	Ver
Lista 1	1	S	8.0	Ver
Lista 1	1	E	4.0	Ver
Lista 1	1	E	0.0	Ver

RESOLVER NOVA QUESTÃO

Figura 4. Tela de questões resolvidas

5. Trabalhos Relacionados

O feeper consiste em uma ferramenta web para apoiar a aprendizagem de programação em sala de aula, assim como em extraclasse. Na ferramenta, são disponibilizados exercícios práticos que são organizados de forma a desafiar o aluno de forma gradual na resolução e também acompanhar a sua evolução desde o início do uso desta ferramenta. [Alves et al. 2013, p. 1080]

Outra iniciativa que também utiliza Juízes Online é a de Santos e Ribeiro, que propõe o JOnline [Santos e Ribeiro, 2011, apud Chaves et al. 2014, p. 241] um sistema que possui um Juiz Online próprio no qual é possível acessar problemas para serem resolvidos e submeter códigos-fonte para a visualização do resultado. Além da programação colaborativa, o sistema ainda adiciona funcionalidades didáticas ao juiz auxiliando o aluno no processo de aprendizagem. [Chaves et al. 2014, p. 241]

Temos também o MOJO, que é uma ferramenta que integra o ambiente Moodle com os Juízes online. O MOJO é uma ferramenta que se propõe a obter questões, de programação, de terceiros (Juízes Online) para resolução pelo aluno [Chaves et al. 2014, p. 247]. Como o foco é no professor, o MOJO recebe a resposta do Juiz Online e apresenta para o professor, para que este então dê um feedback para os seus alunos. [Alves et al. 2013, p. 1079].

É importante citar que existem trabalhos importantes nesse campo, e que estamos buscando formas de facilitar o ensino de programação e proporcionar um ambiente moderno por meio de ferramentas tecnológicas que cada vez mais estão surgindo para auxiliar no ambiente educacional.

O Inspetor Online, ao contrário dos trabalhos mencionados, segue o modelo código aberto e propõe interface de simples utilização e apoio no resultado da resolução das questões com resultados que podem facilmente ser identificados pelos alunos, além de permitir o cadastramento de listas com quantas questões o professor desejar. Outro diferencial é que ao final do período de submissão das questões, os alunos poderão ter acesso ao gabarito anteriormente cadastrado pelo professor, assim o professor pode

estimular o pensamento crítico do aluno acerca do problema com discursões em sala de aula.

6. Resultados Esperados

Buscamos melhorar cada vez mais a ferramenta e dessa forma implementar as funcionalidades que consideramos de suma importância para aplicação. Atualmente não foi implementada a funcionalidade de formação de turmas, o desempenho dos alunos é medido de forma individual, e o professor acompanha o desenvolvimento deles diretamente. Sendo disponibilizada essa funcionalidade, os alunos seriam inseridos em um grupo – que chamamos de turma – onde o professor teria acesso as práticas de um aluno da turma a qualquer momento, e também poderia medir o desempenho da turma como um todo.

Dessa forma, os alunos terão possibilidade de interagir melhor na turma, fazer comentários sobre questões, e podendo até ajudar aqueles colegas do grupo que estão sentindo dificuldades em uma questão específica. Isso é importante para interação e acompanhamento da turma, vai além da sala de aula, e é uma das funcionalidades que podem ser implementadas ao longo do desenvolvimento.

Espera-se que essa ferramenta de fato contribua para auxiliar os alunos e professores nas disciplinas de programação, proporcionando resultados de forma rápida, e incentivando o aluno a buscar melhorias no desenvolvimento de suas respostas, que ficarão visíveis a ele e ao professor. No primeiro semestre de 2017 o Inspecor Online será usado em turmas regulares em uma instituição de ensino. O código fonte está disponível em https://github.com/leisiamedeiros/Inspecor_Online e uma demonstração está disponível em <https://youtu.be/mwarNMXhbKg>.

7. Referências

- Francisco, R., Júnior, C.P. and Ambrósio, A.P., 2016, November. Juiz Online no ensino de Programação Introdutória - Uma Revisão Sistemática da Literatura. In Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE) (Vol. 27, No. 1, p. 11-12).
- Galvão, L., Fernandes, D. and Gadelha, B., 2016, November. Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. In Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE) (Vol. 27, No. 1, p. 141).
- Odersky, Martin, Lex Spoon, and Bill Venner. "Programming in Scala." (2016).
- Hilton, Peter, Erik Bakker, and Francisco Canedo. Play for Scala: Covers Play 2. Manning Publications Co., 2013.
- Alves, F.P. and Jaques, P., 2014. Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação. In Anais dos Workshops do Congresso Brasileiro de Informática na Educação (Vol. 3, No. 1, p. 1080).
- Chaves, J.O.M., Castro, A.F., Lima, R.W., Lima, M.V.A. and Ferreira, K., 2014. Mojo: uma ferramenta para integrar juízes online ao moodle no apoio ao ensino e aprendizagem de programação. HOLOS, 5, pp.241-247.