



UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO

THREELAB - UMA FERRAMENTA DE ENSINO E CORREÇÃO
DE ATIVIDADES DE PROGRAMAÇÃO

WILIAM GOMES CONCEIÇÃO

Salvador - Bahia

28 DE DEZEMBRO DE 2020

THREELAB - UMA FERRAMENTA DE ENSINO E CORREÇÃO DE ATIVIDADES DE PROGRAMAÇÃO

WILIAM GOMES CONCEIÇÃO

Trabalho de Conclusão de curso apresentado
como requisito parcial para obtenção do título
de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Rodrigo Rocha Gomes
e Souza.

Salvador - Bahia

28 de dezembro de 2020

THREELAB - UMA FERRAMENTA DE ENSINO E CORREÇÃO DE ATIVIDADES DE PROGRAMAÇÃO

WILIAM GOMES CONCEIÇÃO

Trabalho de Conclusão de curso apresentado
como requisito parcial para obtenção do título
de Bacharel em Ciência da Computação.

Banca Examinadora:

Prof. Dr. Rodrigo Rocha Gomes e Souza (Orientador)
UFBA

Prof. Me. Diego Zabot
ITS Alessandro Volta Trieste

Prof. Dr. Rubisley de Paula Lemes
UFBA

À minha família

Agradecimentos

Primeiramente, agradeço a Deus, por ter me dado paciência e sabedoria para que eu chegasse até aqui. Segundamente à minha família, por ter me dado apoio e pela confiança. Aos meus amigos por estarem sempre presentes quando eu precisei e pelos bons momentos, aos meus professores, por terem tornado as aulas mais interessantes e construtivas, ao longo de toda a minha vida, e ao meu orientador, Rodrigo Rocha Gomes e Souza pela paciência e compreensão durante a construção desse projeto.

‘A Matemática não mente. Mente quem faz mau uso dela’.

Albert Einstein

Resumo

O Threelab é uma ferramenta de execução e correção de atividades de programação, criada para facilitar a criação de atividades por parte dos docentes e a absorção de conhecimento dos discentes, com explicações e casos de teste para suas matérias para que os alunos respondam a essas atividades num editor de código, este que é compilado ou interpretado e seu resultado exibido na página. A plataforma se utiliza de um editor de código, o CodeMirror, um banco de dados numa aplicação em PHP e MySQL para que seja possível o armazenamento das respostas e um dashboard do professor, além de páginas HTML para exibição do conteúdo e explicação das atividades. Com base em uma avaliação realizada com 13 estudantes de graduação, vemos sua efetividade principalmente devido ao fato de a correção ser automática.

Abstract

Threelab is an execution and correction tool of programming activities, created to enable professors and students to easily create activities with explanations and test cases for programming topics and answer these activities in a code editor which compiles or interprets this code and displays the output on the page. The platform uses a code editor, CodeMirror, a database in a PHP/MySQL application which makes possible the storage of the answers and a dashboard for the professor, and even HTML pages for content display and explanation of the activities. According to an evaluation made with 13 undergraduate students, we saw its effectiveness mainly due to the fact of the correction being automatic.

Sumário

1	Introdução	1
	Introdução	1
2	Trabalhos relacionados	3
2.1	Codecademy	3
2.2	Spoj	9
2.3	Autolab	14
2.4	Stepik	24
2.5	Programming Education Assistant	31
2.5.1	Jupyter	34
2.5.2	Go Toolchain	35
2.5.3	Bazel	35
2.5.4	Docker	36
2.5.5	Google Cloud SDK	36
3	Solução - Threelab	44
3.1	Solução existente	44
3.1.1	Arquitetura	48
3.2	Threelab	51
3.2.1	Novo interpretador	53
3.2.2	Saída e testes	53
3.2.3	Dashboard	55
3.3	Comparativos	60
4	Avaliação	63
4.1	Experimento piloto	63
4.2	Metodologia	63
4.3	Resultados	66
4.4	Discussão	70

5 Conclusão	72
Referências Bibliográficas	75
A Questionário	78

Lista de Figuras

2.1	Catálogo de cursos do Codecademy, relacionados à construção de carreira	4
2.2	Exemplo de curso de carreira do Codecademy	4
2.3	Skill Paths no Codecademy	5
2.4	Cursos de linguagens de programação do Codecademy	5
2.5	Página inicial do Codecademy - aba ‘Today’	6
2.6	Página inicial do Codecademy - projetos recomendados	6
2.7	Página inicial do Codecademy - questionários	7
2.8	Exemplo de página tripla para exercícios do Codecademy	8
2.9	Exibição de erro no Codecademy	8
2.10	Exibição de erro com dicas e link de revisão do Codecademy	9
2.11	Página de inscrição no SPOJ	10
2.12	Página inicial de problemas do SPOJ	11
2.13	Problemas no SPOJ resolvidos, exibidos na tabela	12
2.14	Página do exercício ‘Sistema Cipoviário’	13
2.15	Página de clusters - hardwares usados na execução das soluções dos problemas	13
2.16	Página de submissão de resposta para um problema no SPOJ	14
2.17	Página de submissão do usuário, na qual são apresentadas todas as tentativas de solucionar o problema ‘Rumo aos 9s’	15
2.18	Página de execução de código do ideone, na qual ele apresenta os problemas de compilação do código sumbetido	15
2.19	Página de melhores submissões do exercício ‘Rumo aos 9s’	16
2.20	Ranking do SPOJ no dia 01/05/2020’	17
2.21	Página de demonstração do Autolab	18
2.22	Página de cursos do Autolab, na página de demonstração	19
2.23	Página de um curso do Autolab; cada módulo é de um determinado tipo, e cada tipo possui um conjunto de atividades	20
2.24	Página do estudante	21
2.25	Página de gerenciamento de curso	22
2.26	Página de edição de nota	22

2.27	Página de edição de atividade - seção Basic	23
2.28	Página de edição de atividade - seção Handin	23
2.29	Página de edição de atividade - seção Penalties	24
2.30	Página de edição de atividade - seção Problems	25
2.31	Página de notas da atividade	25
2.32	Página de extensão de prazo de atividades	26
2.33	Página de visualização de submissões	26
2.34	Página de visualização de estatísticas	27
2.35	Planos da plataforma Stepik	28
2.36	Página inicial da plataforma Stepik	29
2.37	Página de criação de cursos da plataforma Stepik	30
2.38	Página principal de um curso no Stepik	31
2.39	Página principal de um curso no Stepik, na visão do estudante	32
2.40	Página de conteúdo do curso ‘JavaScript for Beginners’, no Stepik	32
2.41	Página de quiz do curso ‘JavaScript for Beginners’ no Stepik	33
2.42	Página de quiz do curso ‘JavaScript for Beginners’ no Stepik na qual o usuário respondeu a pergunta corretamente	33
2.43	Página de conteúdo do curso ‘JavaScript for Beginners’, no Stepik no qual o estudante respondeu a pergunta de forma incorreta	34
2.44	Página de histórico de respostas para uma pergunta no Stepik	35
2.45	Página de exercício de Match no Stepik	36
2.46	Página de conteúdo com código no curso ‘JavaScript for Beginners’ no Stepik	37
2.47	Página de exercício de código no curso ‘JavaScript for Beginners’ no Stepik	38
2.48	Página de exercício de código no curso ‘JavaScript for Beginners’ no Stepik na qual o estudante submeteu um código que não passou em todos os testes	38
2.49	Página inicial da plataforma Jupyter	39
2.50	Página de carregamento de exemplo do Programming Education Assistant no Jupyter	39
2.51	Página de instalação de componente por meio do Jupyter Notebook	40
2.52	Página de exibição de resposta incorreta no Jupyter Notebook, da atividade dataframe-pre1-master, dentre os exercícios do Programming Education Assistant	40
2.53	Página inicial do Go	41
2.54	Editor de código do Go, com saída	41
2.55	Página inicial do Bazel	42
2.56	Página inicial do guia de instalação do Docker	42
2.57	Página de projeto vazio no Google Cloud	43

3.1	Página inicial do repositório de aulas	45
3.2	Página do componente curricular Paradigmas de Linguagens de Programação	45
3.3	Página de Prolog do componente curricular Paradigmas de Linguagens de Programação, nesta temos apresentação de imagem e código que não é possível para o estudante modificar	46
3.4	Página de demonstração, na qual há um editor de código e podemos observar a saída no console	47
3.5	Código em Markdown de uma página na solução pré-existente, de um conteúdo de Paradigmas de Linguagens de Programação	47
3.6	Código no qual há uma sequência de crases triplas, que são códigos não modificáveis pelo aluno	48
3.7	Página carregada num servidor local para ser necessária a realização de uma avaliação, após o armazenamento de um valor	50
3.8	Exemplo de página criada no Threelab - layout Page	51
3.9	Exemplo de página criada no Jekyll - layout Triplepage	52
3.10	Código de página tripla em Markdown	53
3.11	Editor exibido em 100% do comprimento da tela no Jekyll	54
3.12	Exibição de página sem editor	55
3.13	Saída do valor de uma variável no Threelab	56
3.14	Impressão de saída para teste correto	57
3.15	Mensagem de erro no console do Threelab	57
3.16	Saída de valor com entrada no Threelab	58
3.17	Quantidade de testes passados exibida abaixo do campo de saída	58
3.18	Caso de teste no qual não houve sucesso, com saída esperada e saída obtida	59
3.19	Estrutura da div dos casos de teste	59
3.20	Página de apostilas no Threelab	60
3.21	Página de notas no Threelab	61
3.22	Página de uma questão do Threelab	61
4.1	Página de instruções do experimento	64
4.2	Página da primeira questão da primeira atividade no layout de rolamento .	64
4.3	Página da segunda questão da primeira atividade no layout de rolamento .	65
4.4	Página da primeira questão da segunda atividade no layout triplo	65
4.5	Página da segunda questão da segunda atividade no layout triplo	66
4.6	Gráfico de setores representando em porcentagem quantos estudantes participaram do grupo 1 e quantos participaram do grupo 2	67
4.7	Gráfico de barras no qual é apresentado o tempo de finalização da atividade 1	67
4.8	Gráfico de barras no qual é apresentado o tempo de finalização da atividade 2	68

4.9	Gráfico de setores representando as porcentagens das preferências dos estudantes	69
4.10	Gráfico de setores representando a porcentagem de estudantes com relação à influência do layout no tempo de execução	69

Capítulo 1

Introdução

A programação é cada vez mais importante devido à sua maior integração com outras áreas de atuação profissional. Por exemplo, devido ao crescimento da ciência de dados, há maior interesse em se usar programação em áreas como Jornalismo, Estatística e Economia, o que torna necessário o aprendizado de linguagens de programação para que os profissionais se mantenham atualizados. Isso culminou então no aumento da demanda de aprendizado, e dessa forma a importância do ensino de programação para outras áreas se tornou mais importante. Apesar de alguns cursos utilizarem plataformas de ensino, algumas funcionalidades destas podem não trazer a liberdade de criação que o professor deseja, como, por exemplo, criar exercícios e testes à sua maneira, o que limita o planejamento de aulas dos docentes e a aquisição de novos conhecimentos por parte dos discentes, pois estes não se sentem motivados caso não haja estímulos para que melhorem suas habilidades [31].

A limitação de tempo dos semestres e o nível de dificuldade do curso são fatores que tornam essencial o feedback rápido de respostas, para que a irresolução de atividades não prejudique a difusão de conhecimento, pois assim os educandos podem otimizar o preparo para avaliações futuras e os educadores terão mais tempo para elaborar aulas de forma a aperfeiçoá-las ou dedicar-se à elucidação de incertezas presentes durante a experiência de aprendizado dos componentes curriculares. Um fator que pode ser decisivo nesse rápido retorno de respostas é a correção automática de exercícios, uma vez que, também pela limitação de tempo, alguns professores não conseguem ou têm obstáculos para cumprir prazos como o de disponibilizar notas com 48 horas de antecedência de avaliações, presente na Universidade Federal da Bahia. Destarte, haveria menos limitações e seria prescindível a correção manual e prolongada.

Com o propósito de encontrar uma solução para este problema, desenvolvemos o Threelab, uma ferramenta de contribuição ao aprendizado de linguagens de programação por meio da criação de conteúdo didático, correção automática de atividades, atribuição

de nota e curta espera de retorno. A ferramenta é implementada como um conjunto de scripts (na linguagem JavaScript) que podem ser integrados de forma simples a páginas HTML. Devido à praticidade de integração da linguagem JavaScript com páginas HTML (já que é a linguagem presente nos navegadores), esta foi escolhida como primeira linguagem utilizada nos editores; entretanto, devido à necessidade de testar a ferramenta, foi adicionada a linguagem C++. O armazenamento e a visualização de notas e exercícios por parte dos professores são possíveis devido a uma aplicação feita em MySQL e PHP.

No próximo capítulo, veremos alguns trabalhos relacionados, suas semelhanças e diferenças com a plataforma, além de inspirações para sua criação; no capítulo seguinte, o funcionamento do Threelab é apresentado com mais detalhes, sendo também exemplificado. A seguir, aplicações da plataforma no ambiente do componente curricular MATA37 (Introdução à Lógica de Programação) da Universidade Federal da Bahia (UFBA) serão apresentadas com o roteiro de experimento e os resultados obtidos. Por último, considerações finais e projetos futuros.

Capítulo 2

Trabalhos relacionados

A plataforma deste projeto, por abranger diferentes funções, não somente a de execução de atividades, se assemelha a diferentes projetos já existentes. O Codecademy, mais similar, foi o projeto que mais inspirou a criação desta plataforma. O SPOJ, pela semelhança no quesito de possuir correção automática de testes, também foi um inspirador. Além deles, há também outras plataformas que serão apresentadas a seguir, com o objetivo de que seja demonstrada a relevância e a necessidade de criação deste projeto para uso em instituições de ensino de programação - Autolab, Stepik e Programming Education Assistant.

2.1 Codecademy

O Codecademy [5] é uma plataforma de ensino de linguagens de programação (como JavaScript, Python e Ruby) e de marcação (como HTML e CSS). Fundada por Ryan Bubinski e Zach Sims em 2011, tinha como objetivo construir uma plataforma para aprendizado e ensino para todos os funcionários e também para pessoas que quisessem aprender linguagens de programação [6]. A plataforma possui um plano grátis, no qual o usuário tem acesso à lição inicial de cada assunto, que corresponde à parte teórica, e um plano que pode ser anual, pagando 15.99 dólares por mês ou mensal de 19.99 dólares, que permite ao usuário ter acesso a atividades sobre problemas reais que envolvem o assunto e a linguagem especificados. Veremos a seguir como é estruturada a plataforma.

Para efetuar login ou entrar no Codecademy, é necessário que o usuário forneça e-mail e senha ou use uma conta do Facebook, Gmail, LinkedIn, Github ou Twitter.

No Codecademy há uma página que exibe o catálogo de cursos, divididos em cursos da carreira (Figura 2.1), que são cursos que selecionam linguagens e fundamentação teórica para ensino. As atividades podem ser assistir vídeos, ler artigos, fazer lições ou até mesmo questionários, como mostra a Figura 2.2.

The screenshot shows the Codecademy homepage with a navigation bar at the top. Below the navigation, there's a section titled "CAREER PATHS" with a sub-section "Code Foundations". This section includes a green icon with binary-like symbols, a brief description, and a list of topics: Computer Science History, Career Exploration, Applications, and one marked as "In-progress". Another section below it is "Computer Science", featuring a purple icon with a tree-like graph, a brief description, and a list of topics: Python, Data Structures, Command Line, and Git.

Figura 2.1: Catálogo de cursos do Codecademy, relacionados à construção de carreira

This screenshot shows two specific courses from the catalog. Course 1, "Code Foundations: Introduction", is listed under the "Code Foundations" path. It shows 1 video and 1 lesson, with a "View Detail" button. Course 2, "Learn How to Code", is listed under the "Computer Science" path. It shows 1 article, 7 lessons, and 7 quizzes, with a "View Detail" button. Both courses have a "Start" button in the top right corner.

Figura 2.2: Exemplo de curso de carreira do Codecademy

No catálogo também há cursos de habilidade (Skill Paths), que podem ser de análise de dados ou até mesmo criação de jogos; entretanto, todos são para usuários premium (Figura 2.3).

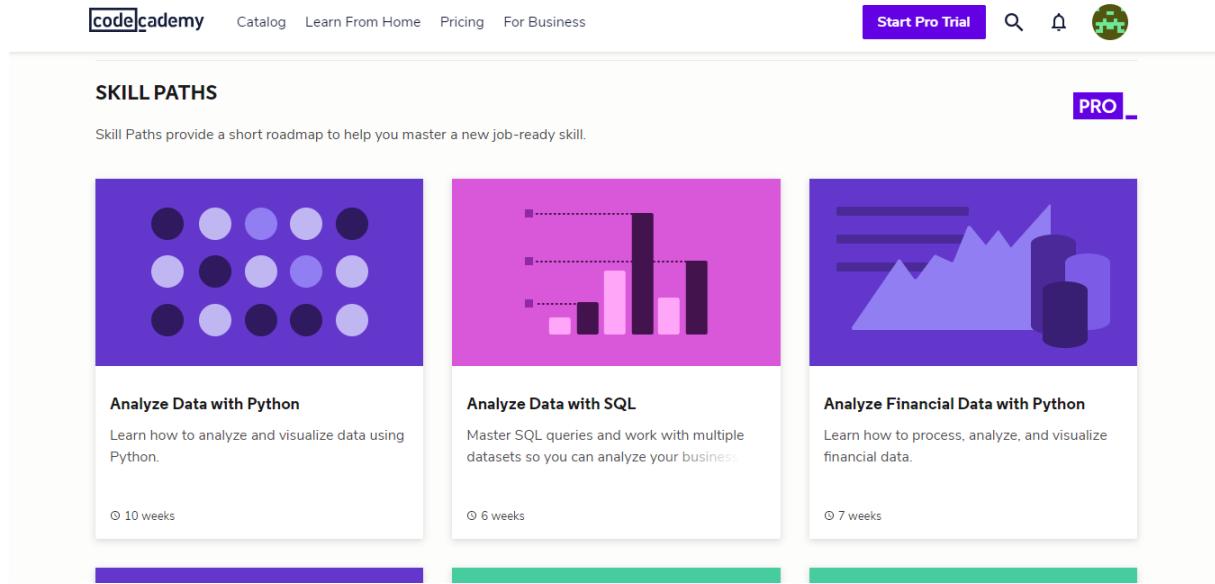


Figura 2.3: Skill Paths no Codecademy

Finalmente, há os cursos das linguagens de programação em si, que podem ser divididos em vários módulos, a depender da complexidade de aprendizado da linguagem, como apresentado na Figura 2.4.

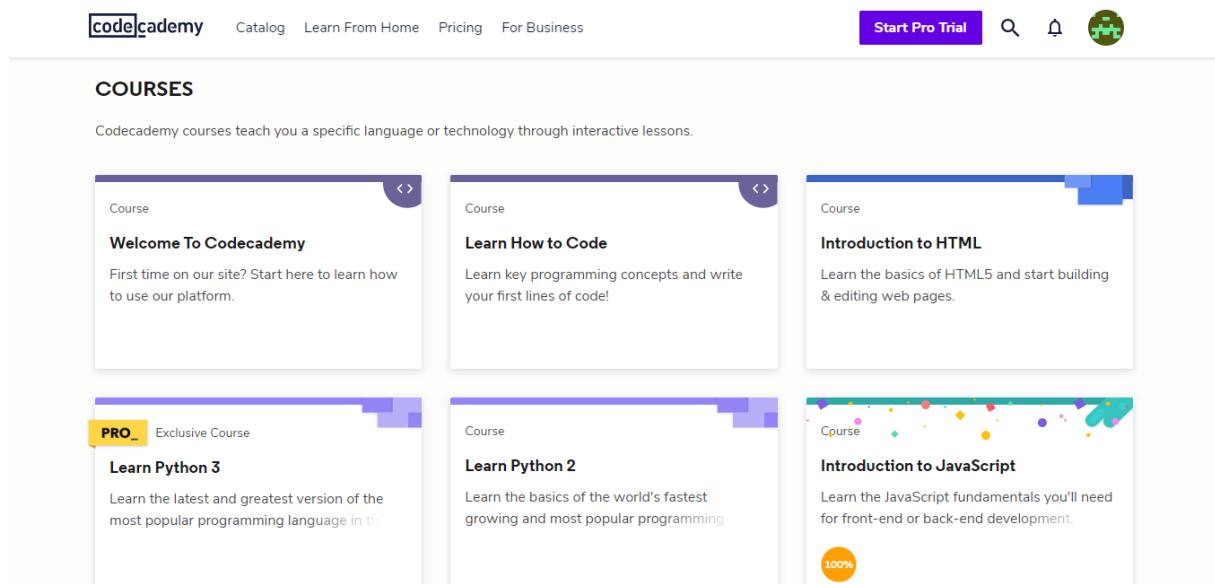


Figura 2.4: Cursos de linguagens de programação do Codecademy

A página inicial do Codecademy é dividida em três partes: a aba ‘Today’ (Figura 2.5), que exibe uma recomendação da plataforma com base na atividade recente

do usuário, a aba ‘My Courses’, que mostra os cursos que o usuário está fazendo ou os cursos concluídos, e a aba ‘Practice’, que contém desafios e questionários para revisão dos assuntos (Figura 2.6 e Figura 2.7).

The screenshot shows the Codecademy homepage with the 'Today' tab selected. At the top, there's a navigation bar with links for Catalog, Learn From Home, Pricing, and For Business, along with a 'Start Pro Trial' button and a search icon. Below the navigation is a horizontal menu with 'Today', 'My Courses', and 'Practice'. A yellow 'PRO-' button with the text 'Try Codecademy Pro' and a subtext 'Enjoy unlimited learning, practice materials and other premium features. TRY IT FOR FREE' is prominently displayed. The main content area is divided into three sections: 'Learn', 'Practice', and 'Apply'. The 'Learn' section shows a 'Learn CSS' module at 100% completion, currently on 'The Box Model'. It lists three items: 'Lesson' (The Box Model), 'Lesson' (Changing the Box Model), and 'Project' (The Box Model: Davie's Burgers). The 'Practice' section shows an 'Advanced CSS Grid' challenge with a 10-minute duration. The 'Apply' section indicates 'Nothing to build yet'.

Figura 2.5: Página inicial do Codecademy - aba ‘Today’

The screenshot shows the Codecademy homepage with the 'Practice' tab selected. At the top, there's a navigation bar with links for Catalog, Learn From Home, Pricing, and For Business, along with a 'Start Pro Trial' button and a search icon. Below the navigation is a horizontal menu with 'Today', 'My Courses', and 'Practice'. A yellow 'PRO-' button with the text 'Challenge Projects' is visible. The main content area displays three recommended projects: 'Build Your Own Cheat Sheet', 'Number Guesser', and 'Build a Website Design System'. Each project has a thumbnail image, a title, a brief description, and a list of tags at the bottom.

Figura 2.6: Página inicial do Codecademy - projetos recomendados

Na Figura 2.8, é possível perceber que a página é dividida em 3 partes: a primeira, que corresponde ao material de estudo, que pode conter partes de código e explicações, bem como imagens, para que possa ser passado o conteúdo. Logo abaixo desse quadro de explicações, há um menu com todas as lições do módulo, para que possa ser fácil a navegação entre elas. Na coluna do meio, há o editor de código, com abas de navegação

The screenshot shows the Codecademy homepage with a navigation bar at the top. Below the navigation, there is a section titled "Recommended Quizzes PRO" featuring nine quiz cards arranged in a 3x3 grid. Each card includes the quiz title, a brief description, and tags.

Quiz Title	Description	Tags
Introduction to HTML	Learn the basics of HTML, including how to structure web pages.	html, web development
CSS Setup and Selectors	Learn how to style web pages using CSS.	css, web development
Python Syntax	Learn the basic syntax of Python, including variables, loops, and functions.	python, programming
HTML Tables	Learn how to create tables in HTML.	html, web development
Introduction to JavaScript	Learn the basics of JavaScript, including how to add interactivity to web pages.	javascript, web development
Conditional Statements Quiz	Test your knowledge of conditional statements in programming.	javascript, web development
Functions Quiz	Test your knowledge of functions in programming.	javascript, web development
Strings and Console Output	Learn how to work with strings and console output in programming.	python, programming
Hello World	Test your knowledge of the Hello World program in Java.	java, programming

Figura 2.7: Página inicial do Codecademy - questionários

entre arquivos, que serão necessários para a execução e construção tanto do problema quanto da atividade - estes podem estar do lado de um símbolo de pasta, que possui todos os arquivos que serão usados, no módulo inteiro ou em uma lição. Além disso, há um botão para expandir o editor, para que este fique em 100% do comprimento da página, para melhor visualização do código. Logo abaixo do editor, há 3 botões: ‘Run’, que executa o código do editor, um botão para que o código volte ao estado original, sem alterações feitas por quem está aprendendo, e um botão para copiar o código do editor. Na parte inferior há um botão que permite voltar para a lição anterior, um indicador da lição atual e quantas lições há no módulo, e um botão de avançar, que pode estar desativado, enquanto a lição não for concluída, ou permite que o usuário avance para a próxima lição ou para o próximo módulo. Na última coluna é exibida a saída, seja um texto ou uma página, neste caso, que assim como o editor, possui um botão de expansão para melhor visualização da saída. Logo ao lado do nome do curso, há um ícone de um botão para que, caso queira, o usuário defina um tempo máximo para execução do módulo ou da atividade, e em seguida fazer uma pausa (focus timer).

Na Figura 2.9, é exibido o exemplo de erro na terceira parte, que é a saída. Assim como num compilador ou interpretador, é exibida a inconsistência, a linha e o conjunto de bibliotecas violadas com esta.

A cada lição, há uma ou mais etapas que devem ser feitas de forma correta, para que haja progressão no módulo. Caso o usuário não faça a atividade de forma correta, o progresso não será feito, a próxima etapa será bloqueada e na tela será exibida uma dica de como resolver a etapa em questão, além de um link para uma página contendo uma revisão de comandos da linguagem a ser aprendida. Caso haja erros sucessivos, o usuário

The screenshot shows a triple-page layout for a CSS exercise. The left panel displays a 'Review' section about the box model. The middle panel shows a code editor with CSS code for a navigation bar, with a 'Run' button highlighted. The right panel features a terminal window with a banner for 'Conservation Efforts at Lake Tahoe'. Navigation buttons 'Back', 'Up Next', and 'Get Help' are visible at the bottom.

Figura 2.8: Exemplo de página tripla para exercícios do Codecademy

The screenshot shows a triple-page layout for a JavaScript exercise. The left panel discusses 'Else If Statements'. The middle panel shows a code editor with a syntax error in a file named 'main.js'. The right panel displays an error stack trace. Navigation buttons 'Back', 'Next', and 'Get Help' are visible at the bottom.

```

/home/ccuser/workspace/learn-javascript-conditionals-else-if-statements/main.js:6
      else{
      ^^^^
SyntaxError: Unexpected token else
    at createScript (vm.js:53:10)
    at Object.runInThisContext (vm.js:95:10)
    at Module._compile (module.js:543:28)
    at Object.Module._extensions..js
  (module.js:580:10)
    at Module.load (module.js:488:32)
    at tryModuleLoad (module.js:447:12)
    at Function.Module._load (module.js:439:3)
    at Module.runMain (module.js:605:10)
    at run (bootstrap_node.js:427:7)
    at startup (bootstrap_node.js:151:9)

```

Figura 2.9: Exibição de erro no Codecademy

tem a opção de obter a resposta, mas esta se aplica à lição por completo, não somente da etapa que o usuário não obteve sucesso em passar. Na Figura 2.10 é mostrado como é exibida a divisão de etapas, numa resposta incorreta do usuário, além da dica oferecida e o botão de link para a revisão, que é situado na mesma linha dos botões de executar, copiar e reiniciar exercício.

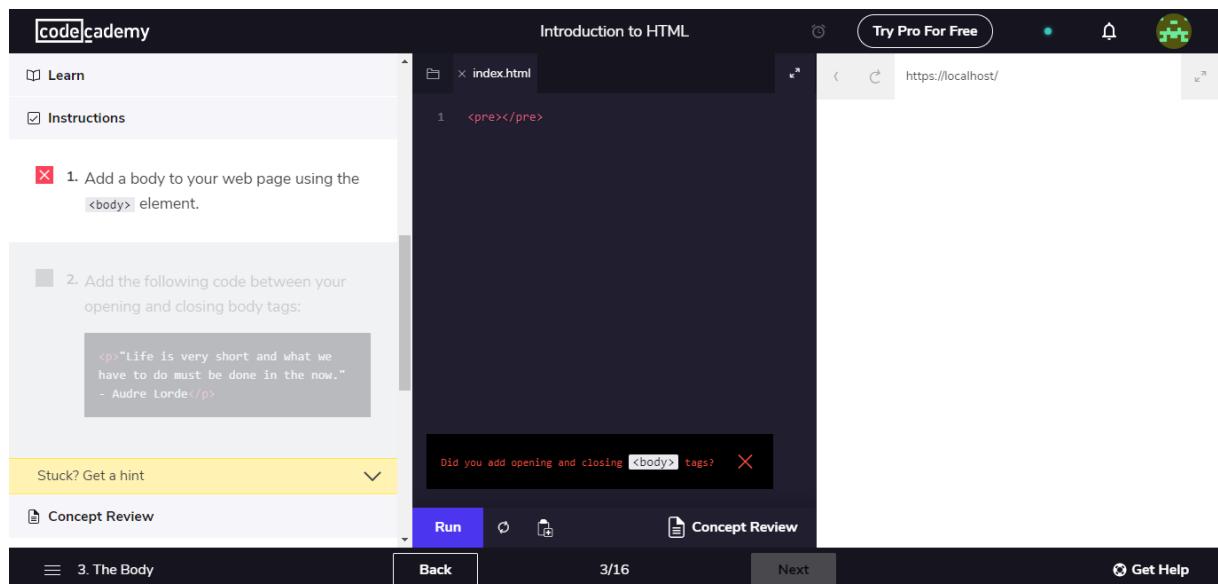


Figura 2.10: Exibição de erro com dicas e link de revisão do Codecademy

O Codecademy tem muitas funções valiosas para ensino, como visto anteriormente; entretanto, não há como professores fazerem exercícios personalizados, adaptando para o nível das turmas e para o conteúdo de planejamento das matérias, o que torna menos viável para uso na Universidade. Mesmo assim, algumas das ideias desse site serviram de inspiração para o Threelab: o uso de páginas divididas em três partes (explicação, editor de código e console), a divisão de exercícios em módulos e a possibilidade de exibição de resultado em tela, para que os alunos tenham acesso à resposta de sua solução de forma mais rápida.

2.2 Spoj

Sphere Online Judge (SPOJ) [34] é um repositório de problemas de regionais, seletivas e olimpíadas de programação [3], para que pessoas de quaisquer lugares do mundo possam ter acesso a esses problemas, tentar resolvê-los e testá-los na plataforma. No caso do SPOJ Brasil, por exemplo, há problemas de provas passadas brasileiras para que estudantes de computação que desejam participar da Maratona de Programação [10] ou da Olimpíada Brasileira de Informática [11] possam treinar, para conseguir resultados melhores. Para conseguir submeter trabalhos no SPOJ, é necessário enviar um e-mail para

os administradores da plataforma contendo o enunciado da prova, arquivos de entrada, arquivos de saída e soluções dos juízes [3]. Portanto, para professores que desejam aplicar exercícios personalizados, não é possível a menos que estes sejam problemas passados de regionais, seletivas ou olimpíadas de programação. Veremos a seguir como ocorrem o funcionamento da plataforma e o processo de envio de respostas e correção do Spoj.

The screenshot shows the 'Register' page of the SPOJ website. At the top, it says 'Create an account and start to learn programming, compete with other users and solve challenging problems!'. Below that are four input fields: 'Username' (placeholder: 'Select your username'), 'Email' (placeholder: 'Enter your e-mail'), 'Password', and 'Confirm Password'. Underneath these fields is a checkbox with the text: 'I have read and accept the [Terms of Service](#). Sphere Research Labs Sp. z o.o. processes my personal data in accordance with the [Privacy Policy](#)'. Below the checkbox is a reCAPTCHA interface with a checkbox labeled 'I'm not a robot' and a CAPTCHA image. At the bottom of the form is a blue 'Create account' button.

Figura 2.11: Página de inscrição no SPOJ

Para se inscrever no SPOJ, é necessário que o usuário possua apenas um e-mail e crie um nome de usuário, além da senha (Figura 2.11). O login do SPOJ também funciona no SPOJ Brasil, então o usuário que criar uma conta no SPOJ pode acessar ambos os sites com o mesmo login.

Na Figura 2.12 vemos que a plataforma do SPOJ possui uma tabela com todos os problemas submetidos. Cada problema possui uma ID (caso o usuário tenha tentado resolver o problema, a ID do problema fica em campo azul, com um link para a submissão do usuário, mostrado na Figura 2.13), um nome por extenso, um código, e sua descrição. Nesta tabela, ao se clicar em um dos valores do cabeçalho, a tabela é organizada a partir deste valor. O campo ‘S’ indica se o usuário já resolveu o problema (quando este é resolvido há um visto, como na Figura 2.13), o campo usuário indica a quantidade de usuários que tentaram e o campo ‘ACC %’ indica a porcentagem de todas as soluções aceitas do problema, em porcentagem.

Na página do exercício, o usuário pode ler a descrição (Figura 2.14) do exercício (além da descrição da entrada e da saída), assim como um ou mais exemplos de entrada e saída. O usuário pode enviar o seu arquivo de código em qualquer linguagem, exceto as linguagens ditas inválidas. Há também especificação de limites de tempo, memória usada e tamanho do arquivo, sendo que é determinado o modelo de hardware que executará o

The screenshot shows the homepage of the Sphere online judge (SPOJ) website. At the top, there's a navigation bar with links for 'SPOJ Brasil' and 'PROFILE'. On the left, there's a sidebar with links for 'News', 'Problems' (containing categories like 'contest_nocturno', 'mineira', 'obi', 'regionais', 'seletivas', 'seletiva_joi', and 'sulamericana'), 'Status', 'Ranking', 'Forum', 'info', 'rules', and 'links'. Below the sidebar is an advertisement for Adobe Stock. The main content area has tabs for 'All problems', 'TODO problems', and 'Unsolved problems'. A large list of problems is displayed, with columns for ID, S (status), NAME, CÓDIGO, usuários, and ACC %. The first few problems listed are:

ID	S	NAME	CÓDIGO	usuários	ACC %
1734		Quem vai ser reprovado	PLACAR	2694	18.65
1735		Não é Mais um Joguinho Canadense	CONTAGEM	841	25.46
1736		Cardápio da Sra Montagny	CARDAPIO	238	12.60
1737		Mesa da Sra Montagny!	MESA	996	23.59
1745		Recuperação	RECUPERA	1623	25.90
1746		Sistema Cipóviário	CIPO	557	20.09
1747		Números de Dinostratus	DINOSTRA	212	16.91
1749		Final Mundial de 2008	MINIMO	426	24.03
1750		Construtores de Totens	QUADRADO	117	9.67
1751		A lei vai a Cavalo!	CAVALOS	398	30.81
1752		Fatorial	FATORIAL	781	10.39
1753		Demonstração de Honestidade!	HONESTID	36	9.79
1757		Brainfuck	BRAIN	197	17.89

Figura 2.12: Página inicial de problemas do SPOJ

código, com detalhes de informações (Figura 2.15). No fim da página o usuário pode fazer um comentário sobre o exercício, mas não é permitido que sejam comentados códigos de solução. No topo da página, é possível ir para as páginas de submissão, respostas do usuário, todas as respostas para o problema da página, as melhores submissões e uma opção de download das especificações em PDF, além de um link para voltar á página anterior.

Na página de submissão, é possível escolher entre digitar um código para submissão e enviar um arquivo com o código na linguagem desejada. A linguagem escolhida pelo usuário também deve ser determinada logo abaixo, ao lado do botão de submissão. A Figura 2.16 ilustra a página de submissão no caso de seleção de C como linguagem para a resposta.

Na página de submissões do utilizador, é possível visualizar tentativas anteriores do usuário, que estão apresentadas em uma tabela (Figura 2.17). As submissões aceitas são apresentadas em linha verde, enquanto que as recusadas são apresentadas em laranja; os erros são apresentados também no campo 'RESULT' (estes podem ser, dentre outros resultados, 'erro de compilação' - quando o usuário comete um erro de sintaxe ou de semântica -, 'tempo limite excedido' - quando a resposta ultrapassa o limite estabelecido na questão -, 'resposta errada' - quando se é esperado um resultado e outro é encontrado em um dos casos de teste - e 'erro em tempo de execução' - quando, durante a execução de problema, ocorre um erro, que é especificado de acordo com o código que acompanha este erro. As submissões possuem ID, data de submissão, nome do usuário que submeteu, tempo de execução (TIME), consumo de memória (MEM) e linguagem escolhida pelo usuário (LING). É possível editar a resposta em 'edit', para tentar resolver o problema



The screenshot shows a table of solved problems from SPOJ. The columns are labeled: ID, Problem Name, Status, Language, Submissions, and Acceptances. A 'Feedback' button is visible on the right.

ID	Problem Name	Status	Language	Submissions	Acceptances
1760	ET phone home		ET	50	21.70
1761	Back to the future		FUTURE	85	24.95
1762	Final do ICPC		ICPC	95	25.44
1763	Sudoku		SUDOIME	1356	29.49
1764	Engenharia de Software		ENGENHAR	24	5.88
1765	Lisp é melhor que Java, C e CPP		LISP	18	14.09
1818	✓ A Viagem de Teobaldo		TEOBALDO	223	22.82
1819	Festa de Forró		FORRO	52	14.13
1820	O Relógio Decadário		DECADARI	536	33.23
1821	O Bolo de Apostas		BAPOSTAS	1150	26.07
1822	Natureza		NATUREZA	431	22.07
1823	Problemas SQF		PSQF	60	11.85
1824	Monitorando a Amazônia		MONAMAZ	186	36.14
1831	f91		F91	4187	36.90
1832	Posto de Bombeiros		PBOMBEIR	92	23.70
1849	Futebol		FUTEBOL	162	15.40
1850	Conte os Fatores		CFATORES	1095	37.27
1851	Comércio de Camelos		COMCAMEL	339	32.94
1856	Pré, Em e Pós		PREEMPOS	384	38.89
1884	Caixero Viajante		CAIXVIAJ	107	50.00
1885	Posições Livres		POSLIVRE	529	42.15
1886	Tráfego		TRAFEGO	35	26.63
2279	Espiral		ESPIRAL	151	30.23
2280	Encontre o telefone		ENCOTEL	2553	36.61
2281	✓ Rumo aos 9s		RUMO9S	926	35.94
2282	Sete mares		SETEMARE	68	30.80
2284	Palavras primas		PAPRIMAS	1994	29.83
2285	Krakóvia		KRAKOVIA	451	20.76

Figura 2.13: Problemas no SPOJ resolvidos, exibidos na tabela

apresentado ao se executar cada uma das tentativas de resposta do usuário, ou executar a tentativa em ‘run’, sendo que nesse caso o código é enviado para um site externo (Ideone [26]), que executa o código e exibe os erros apresentados (Figura 2.18).

Assim como a página de submissões do usuário, há a página de submissões de todos os usuários, que exibe as submissões da data mais próxima da atual para a data mais distante. As submissões são divididas em páginas. Há também a página das melhores submissões (Figura 2.19), na qual são exibidas as estatísticas do problema: quantidade de submissões aceitas, quantidade de submissões, porcentagem de submissões aceitas e quantidade por tipo de erro (resposta errada, erro de compilação, erro em tempo de execução e tempo limite excedido). Ademais, é exibida uma tabela com todas as submissões, ordenadas em tempo de execução (crescente) e data (também crescente).

É possível também, no perfil do usuário, ver todas as submissões de todos os problemas, além de um histórico de atividades em um intervalo de 1 ano, na aba ‘Status’. Na aba ‘Ranking’ é possível ver quais usuários possuem maior pontuação geral na plataforma (Figura 2.20).

O SPOJ é uma plataforma direcionada para competições, e apesar de ter funcionalidades importantes para aprendizado, como opções de submissão e correção automática, não há opções de exercícios personalizados do professor, além de ser uma plataforma de correção binária sem possibilidade de visualizar casos de teste, ou seja, o aluno receberia a nota inteira ou zero e teria mais dificuldades para resolver as incorreções de suas soluções, caso haja alguma, o que não é adequado para um ambiente de aprendizado. No entanto, ainda é aplicável para universidades para alunos que desejam participar de seletivas, regionais ou olimpíadas de programação.

Sphere online judge

SPOJ Brasil | PROFILE ▾

[Submeter](#) [Minhas submissões](#) [Todas submissões](#) [Melhores](#) [PDF](#) [Voltar](#)

News
Problems
[contest_noturno](#)
[mineira](#)
[obi](#)
[regionais](#)
[seletivas](#)
[seletiva_joi](#)
[sulamericana](#)

Status
Ranking

Forum
[info](#)
[rules](#)
[links](#)



CIPO - Sistema Cipoviário

Os pesquisadores do departamento de pesquisa operacional da Universidade da Columbia Britânica foram contratados para uma estranha tarefa. Vários países da África resolveram se unir a utilizar oficialmente o meio de transporte que ficou mundialmente conhecido nos filmes do Tarzan: o cipó. Há milhões de cipós na África e é surpreendente com que velocidade e eficiência uma pessoa pode se deslocar na selva utilizando esse meio de transporte. Só surgiu um pequeno problema: Os cipós são dominados por três grandes tribos: os makelelés, os malouhdás e os abedis. As tribos exigem ser pagas por cipó usado no sistema de transporte. Como eles ainda não sabem o significado das palavras como cartel, cada uma fez o seu preço, e divergiram bastante. Enquanto os makelelés exigem 1235 bongós por cipó usado, os malouhdás exigem 8977 e os abedis 10923 (a Jane ainda está viva, e ajudou a intermediar a negociação para esta tribo).

Os pesquisadores foram contratados para escolher os cipós que comporão o primeiro sistema cipoviário do mundo. Os contratantes construirão milhões de "Pontos de cipó" pela selva africana e desejam que os cipós sejam escolhidos de tal forma que seja possível ir de qualquer ponto a qualquer outro usando os cipós contratados (você pode ter de trocar de cipó algumas vezes, como fazia o Tarzan). Você deve dizer qual o custo de um sistema que atenda estes requisitos e seja o mais barato possível.

Você pode supor que existam cipós suficientes na selva para que sempre exista um sistema cipoviário que atenda os requisitos.

Entrada

A entrada é composta de diversas instâncias. A primeira linha de cada instância contém dois inteiros n ($1 \leq n \leq 1000$) e m ($1 \leq m \leq 2000000$), onde n é o número de "Pontos de cipó" e m é o número de cipós. Cada uma das m linhas seguintes contêm três inteiros u , v e c indicando que existe um cipó que vai do ponto u e até o ponto v com custo c , onde $1 \leq u, v \leq n$ e $c = 1235$ ou 8977 ou 10923 .

Figura 2.14: Página do exercício ‘Sistema Cipoviário’

Sphere online judge

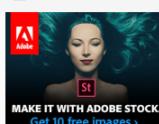
SPOJ Brasil | PROFILE ▾

[Submeter](#) [Minhas submissões](#) [Todas submissões](#) [Melhores](#) [PDF](#) [Voltar](#)

News
Problems
[contest_noturno](#)
[mineira](#)
[obi](#)
[regionais](#)
[seletivas](#)
[seletiva_joi](#)
[sulamericana](#)

Status
Ranking

Forum
[info](#)
[rules](#)
[links](#)



Clusters

Your submissions are judged on two clusters called Pyramid and Cube. It is up to problem setter (author of the problem) to choose which cluster should be used to judge submissions to the problem.

- Pyramid [deprecated]** (*Intel Pentium III 733 MHz*) - consists of old solid Pentium III machines that serve SPOJ since its very beginning in 2004. Thanks to the fact that these machines are slow, judges created by SPOJ problem setters can more easily test the complexity of the algorithms used in your solution without creating enormous datasets (we can even tell the difference between $O(n)$ and $O(n\log n)$ if needed). On this cluster memory limit for submissions is 256 MB.
- Cube** (*Intel Pentium G860 3GHz*) - this new cluster consists of modern and fast Intel Pentium G860 CPUs. On Cube your submissions will run from 30 to 50 times faster than on Pyramid so you can expect that if you test your solution at home then it will have similar execution time on SPOJ. On this cluster memory limit for submissions is 1536 MB.

Figura 2.15: Página de clusters - hardwares usados na execução das soluções dos problemas

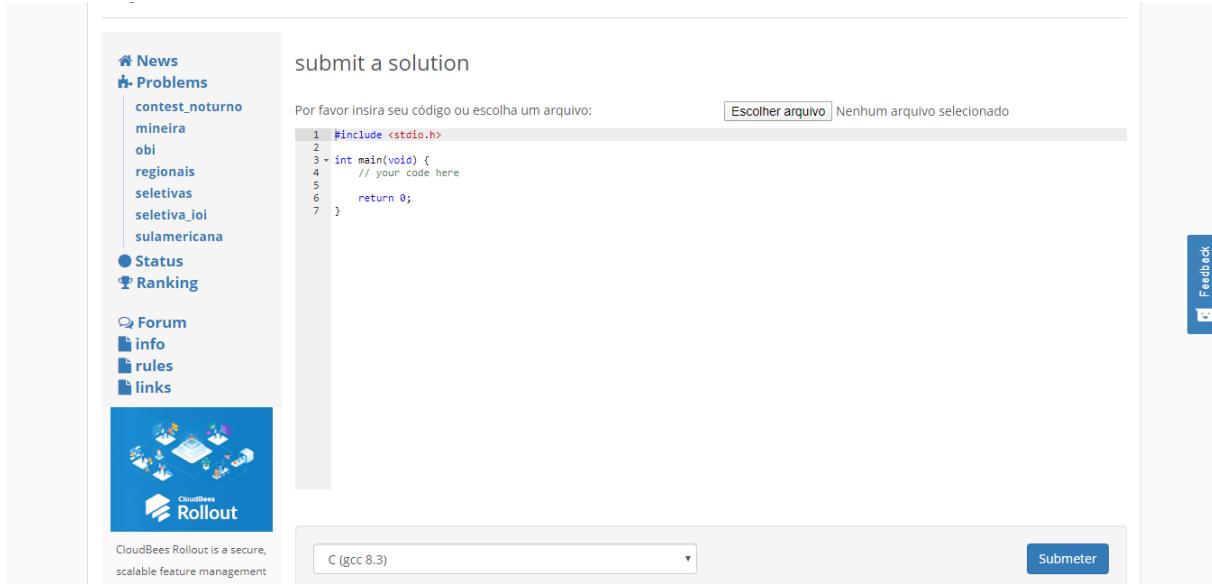


Figura 2.16: Página de submissão de resposta para um problema no SPOJ

2.3 Autolab

“Autolab [38] é um serviço de gerenciamento de cursos, criado por um grupo de estudantes da Universidade Carnegie Mellon, que permite que instrutores disponibilizem atividades de programação com autocorreção para seus estudantes na internet [39].” É uma ferramenta open-source desde 2014, para que diversos colégios e universidades em todo o mundo possam utilizá-la. A licença da plataforma é a Apache 2.0 [30]. A seguir, veremos seu funcionamento e funcionalidades.

A ferramenta possui um site de demonstração, mas há a possibilidade de se criar um site próprio para a universidade. Para se cadastrar, é necessário primeiro e último nomes, senha (com confirmação) e e-mail. O login pede apenas e-mail e senha, mas no site de demonstração, pode-se entrar também pela página ‘Developer Login’, na qual apenas é necessário o preenchimento do campo ‘e-mail’ (neste, pode-se ter acesso ao modo de desenvolvedor com o e-mail admin@foo.bar). A Figura 2.21 mostra a página inicial de demonstração do Autolab.

Para usar o Autolab, é necessário ter Ruby instalado, além do SQLite e MySQL (SQLite apenas caso o usuário queira fazer alguns testes), Rails e Tango, o serviço de autocorreção da plataforma, feito em Python. Este serviço se utiliza de execução em máquinas virtuais, utilizando uma API de alto nível de sistema de gerenciamento de memória virtual, e suporta trabalhos na plataforma Docker [12] (que é o mais recomendado pelos desenvolvedores, pois segundo eles, é mais rápida e fácil), ou Amazon EC2 [2]. Por padrão, a linguagem da plataforma é C.

Na página inicial, o Autolab exibe os cursos, dividindo-os em completos e em an-

The screenshot shows the Sphere online judge interface. On the left, there's a sidebar with links for News, Problems (including contest_noturno, mineira, obi, regionais, seletivas, seletiva_joi, sulamericana), Status, Ranking, Forum, info, rules, and links. The main area displays a table of submissions for 'Wiliam Gomes: Submissões' for the problem 'Rumo aos 9s'. The table has columns for ID, DATA, PROBLEM, RESULT, TIME, MEM, and LING. The first submission was accepted, while others show errors like compilation errors or time limits exceeded. Below the table are buttons for Invert, Selected submissions dropdown, and Execute.

Figura 2.17: Página de submissão do usuário, na qual são apresentadas todas as tentativas de solucionar o problema ‘Rumo aos 9s’

The screenshot shows the Ideone execution page. At the top, it says 'Compilation error #stdin compilation error #stdout 0s 0KB' and 'comments (0)'. Below that, there are sections for 'Stdin' (standard input is empty) and 'Stdout' (standard output is empty). The main area contains the compilation log:

```

Compilation error #stdin compilation error #stdout 0s 0KB
comments (0)

Stdin
Standard input is empty
copy

Stdout
Standard output is empty
copy

Stderr
Standard error is empty
copy

Feedback
Feedback icon

compilation info
prog.c: In function 'degrau2':
prog.c:14:11: warning: implicit declaration of function 'degrau'; did you mean 'degrau2'? [-Wimplicit-function-declaration]
    return (1+degrau(sint(n)));
    ^
degrau2
prog.c:14:18: warning: implicit declaration of function 'sint' [-Wimplicit-function-declaration]
    return (1+degrau(sint(n)));
    ^
prog.c: At top level:
prog.c:49:1: error: expected identifier or '(' before '-' token
  - See more at: http://ideone.com/NKU1w6#sthash.M4cONONj.dpuf
  ^
prog.c: In function 'main':
prog.c:35:1: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
    scanf("%s", num);
    ^
prog.c:45:1: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result [-Wunused-result]
    scanf("%s", num);
    ^

```

Figura 2.18: Página de execução de código do ideone, na qual ele apresenta os problemas de compilação do código sumbeto

Rumo aos 9s estatísticas & melhores submissões

Submissões aceitas	Submissões	ACC %	Resposta errada	Erro de compilação	Erro em tempo de execução	Tempo limite excedido
926	4566	1641	1454	490	396	581

Anterior 1 2 3 4 5 6 7 8 9 10 11 Próximo >

RANK	DATA	Usuário:	RESULT	TIME	MEM	LING
1	2008-01-05 21:50:42	Carlos Gomes [IME - RJ]	accepted	0.00	3.4M	CPP
2	2008-01-06 06:01:24	Ricardo Bittencourt	accepted	0.00	2.5M	CPP
3	2008-01-06 14:14:18	Andre Hahn Pereira	accepted	0.00	2.5M	CPP
4	2008-01-08 01:46:09	Thiago Sonego Goulart	accepted	0.00	1.6M	C
5	2008-01-11 04:17:54	joel	accepted	0.00	2.5M	CPP
6	2008-01-15 02:55:10	Carlos Eduardo Rodrigues Alves [USJT]	accepted	0.00	2.5M	CPP
7	2008-01-19 15:17:07	Marcio Oshiro	accepted	0.00	1.6M	C
8	2008-01-26 13:17:41	Rodrigo Alves Lima [IME-USP]	accepted	0.00	3.2M	CPP
9	2008-01-26 14:29:40	azuge	accepted	0.00	2.5M	CPP
10	2008-03-19 14:20:46	Alantel Freire Marins	accepted	0.00	1.6M	C
11	2008-04-07 17:25:13	Vladimir Araujo Tavares[UFC]	accepted	0.00	2.5M	CPP
12	2008-04-08 05:18:48	reschoene	accepted	0.00	2.6M	CPP
13	2008-04-16 19:12:55	Lucas Hermann Negri [UDESC]	accepted	0.00	1.5M	C
14	2008-05-02 15:28:51	Gisele Moraes Simas	accepted	0.00	1.5M	C
15	2008-05-09 20:52:53	Paulo Roberto Santos de Sousa	accepted	0.00	1.5M	C
16	2008-05-16 21:37:08	Lucas Pereira Madeira [UCDB]	accepted	0.00	1.5M	C
17	2008-05-21 19:56:55	Erick Pereira de Oliveira [UCDB]	accepted	0.00	3.2M	CPP
18	2008-05-23 03:17:44	Regis	accepted	0.00	1.6M	C
19	2008-05-26 04:28:37	Marlon Fernandes de Alcantara [IC-UNICAMP]	accepted	0.00	2.2M	C

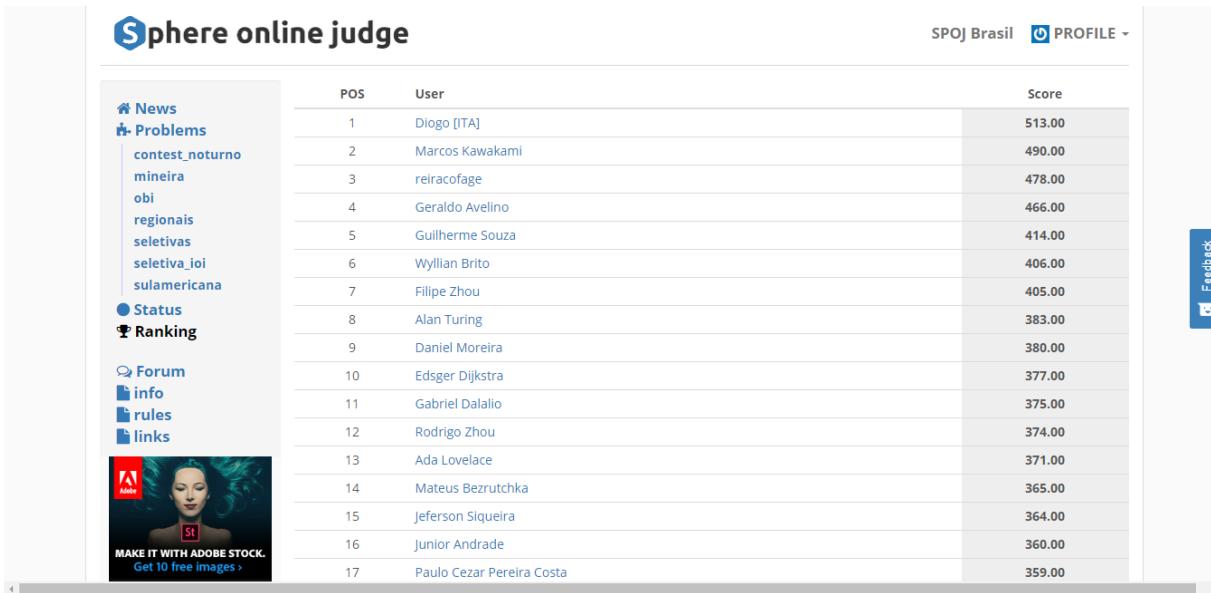
Figura 2.19: Página de melhores submissões do exercício ‘Rumo aos 9s’

damento, ordenados em ordem alfabética (Figura 2.22). Nestes, há conjuntos de módulos, que possuem conjuntos de atividades, estas podendo ser de submissão de código ou questionários (as categorias também podem ser criadas pelo desenvolvedor). Além disso, no cabeçalho da pagina há a seção ‘Manage Autolab’, na qual o usuário pode criar um curso novo, gerenciar usuários e as aplicações e enviar um e-mail para todos os instrutores.

Na página do curso, são exibidos todos os módulos em ordem alfabética (Figura 2.23) e no cabeçalho da página é possível ver uma lista com notas (Figura 2.22), uma página de trabalhos de autocorreção dos professores e seu desempenho, além de uma página de gerenciamento do curso (Figura 2.25). Todas as páginas do curso possuem uma janela de navegação para que seja fácil voltar para páginas já visitadas ou mais abrangentes.

As linhas da tabela correspondem aos alunos do curso, e as colunas são e-mail (que possui um hiperlink para a página do determinado aluno (Figura 2.24)), seu nome e sobrenome, e suas notas de acordo com o módulo - as notas de cada uma das atividades, e a média do módulo. Logo após, há a média de notas geral de cada aluno, seguida pelo seu email. É possível realizar uma pesquisa no campo de pesquisa acima da tabela, além de ser possível baixar os dados da tabela em um arquivo SVG. No botão ‘Statistics’ é possível ver as médias de notas de cada módulo, além de outros dados estatísticos, que são também exibidos na página de gerenciamento de curso (Figura 2.25) e atividade, e o botão ao lado atualiza os dados da tabela.

Na página do estudante, é possível ver diversas tabelas, cada uma delas correspondendo a um módulo. Há também a exibição de notas, com cor - caso a nota seja zero, ela passa a ter a tonalidade vermelha, e ao posicionar o mouse em cima da nota,



The screenshot shows the 'Sphere online judge' website with the URL 'SPOJ Brasil'. The main content is a table ranking users by score. The columns are 'POS', 'User', and 'Score'. The table lists 17 users from 1st to 17th place. A sidebar on the left contains links for News, Problems (including contest_noturno, mineira, obi, regionais, seletivas, seletiva_joi, sulamericana), Status, Ranking, Forum, info, rules, and links. An advertisement for Adobe Stock is visible at the bottom left.

POS	User	Score
1	Diogo [ITA]	513.00
2	Marcos Kawakami	490.00
3	reiracofage	478.00
4	Geraldo Avelino	466.00
5	Guilherme Souza	414.00
6	Wyllian Brito	406.00
7	Filipe Zhou	405.00
8	Alan Turing	383.00
9	Daniel Moreira	380.00
10	Edsger Dijkstra	377.00
11	Gabriel Dalailo	375.00
12	Rodrigo Zhou	374.00
13	Ada Lovelace	371.00
14	Mateus Bezrutchka	365.00
15	Jeferson Siqueira	364.00
16	Junior Andrade	360.00
17	Paulo Cezar Pereira Costa	359.00

Figura 2.20: Ranking do SPOJ no dia 01/05/2020⁷

é exibida uma mensagem de que não houve submissão. Caso haja submissão, a nota é exibida em tonalidade azul, e ambas as notas são exibidas ao lado da pontuação total da atividade, e na última linha é exibida a média das notas. Há também duas colunas: GDU (Grace Days Used), que corresponde ao número de dias daquela atividade nos quais não houve penalidade, e PLD (Penalty Late Days), que corresponde ao número de dias nos quais foi aplicada penalidade para aquela atividade. Ao clicar na nota, o usuário é redirecionado para uma página que exibe o histórico de submissão do aluno para aquela atividade (Figura 2.24).

A página de gerenciamento exibe, para cada atividade de cada módulo, respectivamente, média, mediana, desvio-padrão, valor mínimo e valor máximo, para que os professores possam ter uma base do desempenho e da diferença de nível de aprendizado entre os estudantes.

Na página de gerenciamento também é possível ser redirecionado para outras páginas, como a página de configurações de curso; nela, é possível alterar ou preencher um pequeno formulário que possui como campos Semestre, nome de exibição, late slack (número de segundos que o servidor aceitaria uma submissão depois do prazo máximo de entrega), grace days (quantidade de dias nos quais os alunos podem enviar uma atividade sem sofrer penalidade), late penalty (porcentagem ou quantidade de pontos retirada por dia de atraso de submissão do aluno), version threshold (se uma submissão tem versão maior que a estabelecida, há penalidade - quantidade de submissões permitidas), version penalty (penalidade aplicada para cada quantidade a mais de submissões que a quantidade permitida, em pontos ou porcentagem), datas de início e fim de submissões duas caixas de seleção - a primeira deve ser selecionada se o instrutor quiser impedir submissões para

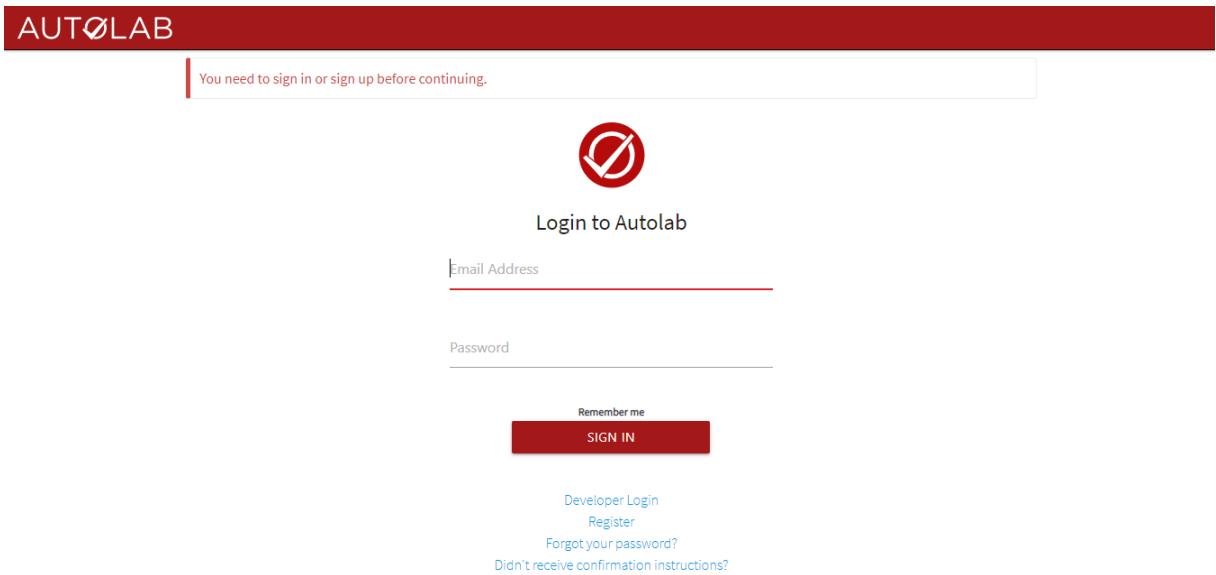


Figura 2.21: Página de demonstração do Autolab

determinada atividade e a segunda deve ser selecionada se for uma prova, e os estudantes não poderão ver submissões anteriores. Por fim, o professor pode escrever uma mensagem que será exibida na tela na página de notas, para, por exemplo, explicar os critérios.

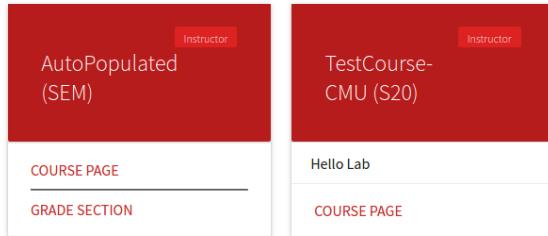
Há a página de gerenciamento de estudantes, que exibe, numa tabela, a quantidade, os e-mails, primeiro e último nome, tipo de usuário (instrutor ou estudante), colégio, ano, apelido e uma coluna que indica se o estudante desistiu ou não do curso. É possível pesquisar estudantes na tabela, assim como adicionar novos estudantes e fazer download da tabela de estudantes nessa página.

Na página de avaliação, é possível ver, fazer download e upload dos métodos de correção dos instrutores. Na página de e-mail, é possível enviar e-mails para ou o instrutor do curso, ou todos os estudantes, de forma a divulgar comunicados, por exemplo. Na página de gerenciamento de anexos, é possível adicionar ou remover anexos de atividades do curso. A página de gerenciamento de anúncios possibilita que o instrutor divulgue ou anuncie eventos importantes para o curso, indicando título, data de início e fim, conteúdo, e mais duas caixas de seleção, para que o anúncio fique sempre fixado nas páginas do curso (primeira caixa) ou permaneça fixado em todas as páginas da plataforma (segunda caixa).

Há a página de programação, na qual o instrutor pode determinar atividades para os estudantes, com intervalos. A página de visualização do estudante permite que as páginas do curso sejam exibidas na visão do estudante e a página de verificação de cópias permite que seja detectado plágio entre os trabalhos dos estudantes, escolhendo o curso, número máximo de linha, linguagem ou arquivo base, além de ser possível enviar outros arquivos para que estes sejam comparados. Por fim, há a zona de perigo, que possui duas funcionalidades: liberar todas as notas para visualização e recarregar o arquivo de

Courses

Current



Completed



Figura 2.22: Página de cursos do Autolab, na página de demonstração

configuração do curso.

Na página de exibição de histórico do estudante, é apresentado todo o histórico de atividades, assim como elas, com opção de download ou visualização das submissões, data de submissão, notas para cada um dos problemas, quantidade de dias após o prazo nos quais a submissão ocorreu, nota total (soma-se as notas de cada uma das notas dos problemas) e, como administrador, é possível modificar a nota, apagar a submissão ou editá-la (Figura 2.26).

Na página de edição, é necessário adicionar uma nova nota; porém, além disso, é possível atribuir uma penalidade ao estudante.

Cada atividade possui sua página, e nesta página há diversos menus: um menu de exibição para administrador, instrutor ou usuário comum (estudante). No menu de administrador, é possível editar atividade, ver notas, baixar atividade, configurar o corretor automático, extensões e submissões, ver estatísticas, importar ou exportar notas de um arquivo CSV (comma-separated values, arquivo de tabela no qual as páginas são separadas por vírgulas [33]); na zona de perigo, pode-se retirar todas as notas, escondê-las da visualização dos estudantes e recarregar o arquivo de configuração da atividade. No menu de instrutor, é possível dar notas para a própria submissão ou todas as atividades, além de baixar uma tabela com todas as notas; na zona de perigo, pode-se liberar as notas para visualização dos estudantes e recarregar o arquivo de configuração da atividade. Já o menu de exibição para o usuário comum permite que este veja suas submissões, notas e opinião de professores ou instrutores, descrição da atividade e download de materiais e código inicial da atividade.

The screenshot shows the Autolab interface. At the top, there's a red header bar with the Autolab logo and navigation links: Section Instructor Gradebook, Gradebook, Jobs, Manage Course, Manage Autolab, Autolab Administrator. Below the header, a breadcrumb trail shows 'AutoPopulated (SEM)'. A message box at the top says 'Scheduled Website and Database Refresh' and 'This is a demonstration website. It refreshes at 0,6,12,18 Hours (UTC) daily. Do not use this site to store important information.' with 'EDIT' and 'DELETE' buttons. The main content area has sections for 'Instructor Actions' (with a 'INSTALL ASSESSMENT' button) and 'Assignments'. Under 'Assignments', there are three categories: 'CategoryAutograde' (Lab Template, Homework 5), 'Homework' (Homework 0, Homework 1), and 'Lab' (Lab 0, Lab 1).

Figura 2.23: Página de um curso do Autolab; cada módulo é de um determinado tipo, e cada tipo possui um conjunto de atividades

Na página de edição da atividade no menu de administrador, há 4 menus: Basic, Handin, Penalties e Problems.

Na seção Basic (Figura 2.27), é possível dar um nome (em alfanuméricos) para a atividade, um nome de exibição, categoria (que pode ser uma já criada ou uma nova categoria), tamanho de grupo (um numeral, que define se a atividade é conjunta ou única), descrição, materiais de apoio e instruções. Além disso, há possibilidade de tornar a atividade parte de um questionário, adicionar autocorreção (estabelecendo imagem, tempo máximo de execução, uma caixa de seleção autorizando ou não a exibição de nota após execução e submissão do estudante, e dois botões para upload de arquivos makefile e TAR (arquivo de compressão ou de armazenamento/extracção de arquivos [15]) do arquivo de correção) ou placar de submissões (com texto de banner em html ou especificação de coluna).

Na seção Handin (Figura 2.28), deve-se estabelecer data de início, fim (com e sem penalização), data final de exibição de notas, duas caixas de seleção (a primeira, para tirar datas de início e fim da atividade e a segunda para permitir submissões de usuários não cadastrados), nome da pasta que os estudantes baixarão com os dados da atividade, diretório remoto dos dados (diretório para onde os dados serão baixados), nome do arquivo de dados e tamanho máximo do arquivo em Megabytes.

Na seção Penalties (Figura 2.29), é possível editar número máximo de submissões (se o valor for ‘-1’ o número é ilimitado), número máximo de dias que o estudante deve levar para concluir a atividade, a penalidade por dia de atraso do estudante (em pontos ou porcentagem), a quantidade máxima de submissões permitidas (se for deixada em branco é ilimitada) e a penalidade, em pontos ou porcentagem, para cada submissão a mais.

CATEGORY	AUTOGRADE	GDU	PLD	Final Score
Lab Template		0	0	0.0/100.0
Homework 5		0	0	0.0/200.0
Category Average				0.0

HOMEWORK	GDU	PLD	Final Score
Homework 0	0	0	79.0/300.0
Homework 1	0	0	167.0/300.0
Homework 2	0	5	98.0/300.0
Homework 3	0	0	0.0/200.0
Category Average			86.0

LAB	GDU	PLD	Final Score
Lab 0	0	3	107.0/300.0
Lab 1	0	0	101.0/300.0

QUIZ	GDU	PLD	Final Score
Quiz 0	0	0	172.0/300.0
Quiz 1	0	0	101.0/300.0

Figura 2.24: Página do estudante

Na seção Problems (Figura 2.30), é possível adicionar, editar ou remover problemas, que serão exibidos, e cada problema terá nome, nota máxima e indicação se ele é opcional ou obrigatório.

Na opção de dar nota a submissões (Figura 2.31), as mesmas são exibidas em uma tabela que indica e-mail e nome, notas para cada uma das questões, penalidades descontadas, se o arquivo foi plagiado ou é um plágio e o valor total da nota.

A opção de extensão (Figura 2.32) permite que as atividades sejam prorrogadas, em dias ou até um dia específico, para determinado aluno cujo nome ou e-mail deve ser determinado (a extensão pode ser infinita). A lista de estudantes com atividades prorrogadas é exibida do lado esquerdo.

Na página de visualização de submissões (Figura 2.33), é possível ver as submissões numa tabela, com autor, versão (quantidade de submissões daquele autor), data de submissão, arquivo, endereço IP e três botões, de executar novamente o corretor automático naquele determinado arquivo, editar as propriedades de nota da submissão e deletar a submissão; além disso, há uma coluna que indica se a submissão é a última possível ou não. Pode-se também adicionar submissões, fazer download de todas, ou apenas das finais, e ver estudantes que não fizeram submissões, além de executar novamente a correção automática para todas as submissões finais dos estudantes.

Na página de estatísticas (Figura 2.34), há uma exibição mais elaborada destas, de forma agrupada em critérios como por seção, por instrutor, por escola e por ano, para ser possível medir a efetividade de cada uma das atividades e dos problemas. Há um menu ao lado indicando cada um dos critérios e com um link para exibição direta das tabelas dos mesmos.

Course Statistics

	Mean	Median	Std. Dev.	Min	Max
labtemplate	0.0	0.0	0.0	0.0	0.0
lab0	120.5	119.0	37.5	48.0	184.0
quiz0	165.7	159.5	40.4	80.0	227.0
lab1	135.0	138.0	50.0	49.0	224.0
homework0	129.6	130.5	33.1	79.0	197.0
quiz1	141.5	128.5	56.3	63.0	255.0
lab2	135.9	121.0	56.0	67.0	271.0
homework1	152.4	153.0	53.4	68.0	246.0
quiz2	144.0	148.5	43.6	72.0	207.0
homework2	127.5	128.0	30.8	76.0	177.0
hello	0.0	0.0	0.0	0.0	0.0
homework5	0.0	0.0	0.0	0.0	0.0
CategoryAutograde Average	0.0	0.0	0.0	0.0	0.0
Homework Average	136.5	130.3	21.4	113.7	178.3

Figura 2.25: Página de gerenciamento de curso

Edit admin@foo.bar's Submission #2 for Lab 0

User:	admin@foo.bar
Assessment:	Lab 0
Version:	2
File:	admin@foo.bar_1_handin.c
Notes:	Notes on submission
Tweak:	Number of points or percentage points

UPDATE SUBMISSION

Figura 2.26: Página de edição de nota

Name
lab0

The unique (for this course) name of the assessment. Must be lowercase alphanumeric (no punctuation). E.g., **malloclab**

Display name
Lab 0

Name that will be displayed on the course home page. E.g., 'Malloc Lab'

Category name
Lab

or

New Category

Select an existing category or enter a new category name (*you can always change it later*)

Group Size
1

Set the maximum size of groups for this assessment. If group size is 1, the assessment is solo. If the size is decreased, groups that are too large will not be broken up. If the size is set to 1, groups will be saved, but the assessment will be solo.

Figura 2.27: Página de edição de atividade - seção Basic

Start at
February 26, 2020 7:01 PM

The time this assessment is released to students.

Due at
March 2, 2020 7:01 PM

Students can submit before this time without being penalized or using grace days.

End at
March 7, 2020 7:01 PM

Last possible time that students can submit (except those granted extensions.)

Grading deadline
March 8, 2020 6:01 PM

Time after which final scores are included in the gradebook

Disable handins

Figura 2.28: Página de edição de atividade - seção Handin

Max submissions
-1
The maximum number of times a student can submit the assessment. Set this to -1 to allow unlimited submissions.

Max grace days
0
Maximum number of grace days that a student can spend on this assessment. E.g., 2. If left blank, all of the remaining available course grace days can be spent on this assessment.

Late penalty
10
points
The penalty applied to late submissions after a student runs out of grace days. It represents the number of points or a percentage of the total score removed per day, and must be a non-negative number. If left blank, the course default is used.

Version threshold
Leave blank to use course default.
The number of unpenalized submissions allowed. After this threshold, each additional submission is penalized according to the version penalty. If set to -1, no submissions are penalized. If this is left blank, the course default is used.

Figura 2.29: Página de edição de atividade - seção Penalties

Na seção de submissões próprias nas opções normais, é possível, na seção de histórico, visualizar submissões anteriores e notas, mas apenas o instrutor pode recorrer, retirar ou editar nota.

O Autolab é uma ferramenta extremamente robusta e completa e diversas universidades a utilizam, para facilitar o ensino de linguagens de programação, com as funcionalidades apresentadas. A ferramenta dá liberdade aos professores e o frontend é a camada mais utilizada pelo usuário, apesar de ter muitas categorias e, dessa forma, possivelmente não ser de fácil aprendizado. É extremamente válido o uso desta plataforma para ensino, dados os seus aspectos positivos; entretanto, não é muito simples integrar a páginas existentes, todo o conteúdo deve ser adaptado ao estilo da plataforma e, assim, o layout e a estrutura também serão diferentes - quanto a isso, não haverá muita liberdade para os professores, no sentido de construir atividades com possibilidade de utilizar mais de um layout e com estrutura diferente, como ocorre no Threelab.

2.4 Stepik

Stepik [36] é uma plataforma em nuvem designada para criar e distribuir conteúdos de educação interativa, assim como provê vários tipos de atividades corrigidas automaticamente com resposta em tempo real. [35] Além da plataforma online em site, é possível acessar num aplicativo em Android ou iOS. A plataforma possui três planos (Figura 2.35), o Basic (gratuito, com editor online, publicação de cursos sem prazos e datas finais, comunicação com estudantes por comentários e avisos e conhecimento à base de auto-serviço,

Scheduled Website and Database Refresh
This is a demonstration website. It refreshes at 0,6,12,18 Hours (UTC) daily. Do not use this site to store important information.

EDIT **DELETE**

BASIC HANDIN PENALTIES PROBLEMS

Problem	Max Score	Optional?	Options
problem0	100.0	no	
problem1	100.0	no	
problem2	100.0	no	

ADD PROBLEM

Autolab Project · Contact · GitHub · Facebook · Logout v2.3.0

Figura 2.30: Página de edição de atividade - seção Problems

#	EMAIL & NAME	LEC	SEC	PROBLEM0 (100.0)	PROBLEM1 (100.0)	PROBLEM2 (100.0)	LATE PENALTY	TWEAK	TOTAL
1	Administrator, Autolab admin@foo.bar	1	Instructor	93.0	84.0	28.0	0.0	-	205.0
2	0, User user0_AutoPopulated@foo.bar	1	None	67.0	28.0	27.0	15.0	-	107.0
3	1, User user1_AutoPopulated@foo.bar	1	None	48.0	42.0	94.0	0.0	-	184.0
4	2, User user2_AutoPopulated@foo.bar	1	None	69.0	78.0	19.0	0.0	-	166.0
5	3, User user3_AutoPopulated@foo.bar	1	None	66.0	51.0	26.0	0.0	-	143.0
6	4, User user4_AutoPopulated@foo.bar	1	None	96.0	7.0	15.0	15.0	-	103.0
7	5, User user5_AutoPopulated@foo.bar	1	None	58.0	0.0	15.0	25.0	-	48.0
8	6, User user6_AutoPopulated@foo.bar	1	None	39.0	81.0	35.0	20.0	-	135.0
9	7, User user7_AutoPopulated@foo.bar	1	None	34.0	35.0	37.0	10.0	-	96.0

Figura 2.31: Página de notas da atividade

Current Extensions (0)		Create New Extension	
Email	Length	Student Name/Email	Start typing student name or email
Select a new due date (Currently due at: 2020-03-02 19:01:09)			
Due at <input type="text" value="March 2, 2020"/>			
-OR- Specify number of days to extend:			
Days <input type="text"/>			
-OR- Infinite Extension?			
<input type="checkbox"/> Infinite			
CREATE			

Figura 2.32: Página de extensão de prazo de atividades

[Create New Submission](#) | [Download All Submissions](#) | [Download Final Submissions](#) | [Missing Submissions](#)

SUBMITTED BY	VERSION	SUBMISSION DATE (YYYY-MM-DD)	FILE	IP ADDRESS	ACTIONS	ISLATEST
5, User user5_AutoPopulated@foo.bar	1	2020-03-07 05:30:30	user5_AutoPopulated@foo.bar_0_handin.c			true
6, User user6_AutoPopulated@foo.bar	2	2020-03-07 01:51:42	user6_AutoPopulated@foo.bar_1_handin.c			false
3, User user3_AutoPopulated@foo.bar	2	2020-03-06 19:48:53	user3_AutoPopulated@foo.bar_1_handin.c			false
6, User user6_AutoPopulated@foo.bar	1	2020-03-06 15:20:53	user6_AutoPopulated@foo.bar_0_handin.c			true
8, User user8_AutoPopulated@foo.bar	3	2020-03-06 13:55:27	user8_AutoPopulated@foo.bar_2_handin.c			false
5, User user5_AutoPopulated@foo.bar	2	2020-03-06 02:15:17	user5_AutoPopulated@foo.bar_1_handin.c			false
n ilcar		2020-02-06				

Figura 2.33: Página de visualização de submissões

All Grades

Problem	Mean	Median	StdDev	Max	Min
problem0	60.7	62.0	19.8	96.0	34.0
problem1	39.0	37.0	24.9	81.0	0.0
problem2	30.8	26.5	22.6	94.0	12.0
Total	120.5	119.0	37.5	184.0	48.0

Statistics by Lecture

1 [10 Students]

Problem	Mean	Median	StdDev	Max	Min
problem0	60.7	62.0	19.8	96.0	34.0
problem1	39.0	37.0	24.9	81.0	0.0
problem2	30.8	26.5	22.6	94.0	12.0

Figura 2.34: Página de visualização de estatísticas

com suporte técnico padrão), o plano Pro (com valor de 99,99 dólares por mês, mas que pode ser testado por 14 dias; possui todos os recursos do plano gratuito, além de: lista de notas e relatório, controle de acesso ao curso, permissão de criação de cursos privados, certificados de curso e suporte técnico avançado) e um plano a combinar, no qual uma empresa deve entrar em contato com os criadores da plataforma para que um preço seja estabelecido (possui todos os recursos dos planos gratuito e pro, além de: exames com inspeção, atribuidor de notas externo, desafios de Linux e de dados, penalização por atraso - e integração às atividades - e suporte técnico de alta prioridade). Para realizar cadastro, é necessário ter mais de 16 anos e prover nome completo e e-mail válidos, além de uma senha, de acordo com os termos de serviço [37].

Na página inicial, há um catálogo (Figura 2.36) com cursos divididos em matérias: Matemática, Estatística, Ciência da Computação, Ciências Naturais, Ciências Sociais e Humanidades; ou categorias, que podem ser Escolha do Editor ou Populares. É possível também procurar cursos por instituição (há uma barra logo abaixo da barra de pesquisa que permite que o usuário selecione uma das instituições através do logotipo. Ao lado da barra de pesquisa, há um botão para criação de um curso.

Na página de criação de curso (Figura 2.37), vemos que é possível criar uma classe num curso já existente, criar um curso pago, acertando pagamento diretamente pelo Stepik e criar competições online. É exibida também a quantidade de recursos apresentados e os tipos de planos para os quais estes estão sendo disponibilizados, e a princípio, nesta página, deve-se primeiramente fornecer um nome para o curso.

Para criar uma classe de curso, é necessário que o usuário escolha um entre três

PLAN	Basic	Pro	Enterprise
PRICING	\$0 Free	\$99,99 1 course a month	Contact us for pricing
TOP FEATURES	Course online editor Open and free courses publication without deadlines and closing dates Communication with students through comments and newsletters Self-served knowledge base, standard technical support	All Basic features, plus: Gradebook and reports Course access control Private courses allowed Course certificates Advanced technical support	All Basic and PRO features, plus: Exams with proctoring External grader, LTI consumer, Linux and Data challenges LTI integration High priority technical support
	Sign up for free	Start 14 Days trial	Contact us

Figura 2.35: Planos da plataforma Stepik

planos oferecidos pela plataforma: um plano grátis, no qual o usuário pode ter quantos alunos e classes quiser nos cursos por ele criados, além de 500 alunos no total em cursos de outros usuários e número de classes ilimitado; um plano grátis para professores, que permite que os usuários tenham uma quantidade ilimitada de alunos para outros cursos (500) caso sejam professores; e um plano Pro, no qual os usuários têm acesso a mais de 500 estudantes no total, além de prioridade elevada no suporte ao consumidor - este plano é de 980 dólares a cada 3 meses. Nos cursos pagos, o instrutor ganha 88% do valor, enquanto a plataforma ganha 12%.

Ao se criar um curso, no plano básico, há opção de alterar a visibilidade somente para os estudantes do curso ou para todos os usuários (apenas com o plano Pro é possível deixar o curso visível para convites), no botão Publish, que se posiciona na página principal (Figura 2.38). Esta página se divide em 4 subseções, Information, Review, Syllabus e News. É possível também alterar foto do curso e, num menu ao lado do botão Publish, editar informações e programação do curso, ver estatísticas e painel de instrutor, para usuários de planos grátis e alterar anexos, permissões, certificados, ver notas e relatórios, além das funcionalidades já citadas, para usuários de plano Pro. Na subseção Information, é possível ver a descrição do curso, resumo, linguagem e se é ou não possível receber certificado com ele (é possível também ver como a visualização para o estudante é feita). Na subseção Reviews, é possível ver revisões de estudantes do curso criado; na subseção Syllabus é possível ver e editar a programação do curso; já na subseção News, é possível adicionar ou gerenciar as notícias do curso.

Na página de curso, na visão do estudante (Figura 2.39), é exibida toda a informação editada na visão do instrutor: título, descrição, tempo médio de conclusão,

The screenshot shows the Stepik platform homepage. At the top, there is a navigation bar with links for 'Catalog', 'My courses', 'Teach', 'Search', 'English', 'WG William', and a 'Create a course' button. Below the navigation bar, there is a search bar labeled 'Search catalog' and a 'Search' button. To the right of the search bar is a '+ Create a course' button. The main content area features a banner with three course cards under the heading 'Editors' choice'. The first card is titled 'LaunchCode's Discovery' by 'LaunchCode Education', featuring a rocket icon. The second card is titled 'Data Structures' by 'Niema Moshiri, Liz Izhikevich', featuring a diagram of nodes A, D, T, and A. The third card is titled 'Bioinformatics Algorithms' by 'Phillip Compeau', featuring a grid of letters T, C, G, A. To the right of these cards is a sidebar titled 'Subjects' with categories: Mathematics, Statistics, Computer science, Natural science, Social science, Humanities, Languages, and 'Русский'. At the bottom of the page, there are several logos for partners like Computer Science Center, BIOINFORMATICS INSTITUTE, Академия Яндекса, SAMSUNG, and БФУ им. Ильинского.

Figura 2.36: Página inicial da plataforma Stepik

indicação de que o curso emite certificado, caso ele emita, perfil do instrutor, conteúdo, exibido em módulos, e as revisões dos estudantes. À direita, é exibida a taxa de inscrição (caso seja um curso pago), um botão de inscrição no curso e tipo de material incluso (por exemplo, quantidade de questionários, lições e aulas interativas). Logo abaixo, pode-se ver o último dia de atualização do conteúdo do curso.

O conteúdo do curso, quando um estudante está inscrito, é apresentado em módulos de acordo com a criação do instrutor (Figura 2.40). Ao lado esquerdo, é exibido todo o conteúdo a partir de módulos e seções; há também dois botões, um mais à esquerda, para que o estudante possa ver o conteúdo em tela cheia, e um mais à direita, de opções, no qual o estudante pode imprimir o conteúdo do material ou se inscrever para receber notificações de comentários naquela seção. Há também uma barra de progresso do curso, exibida no canto superior esquerdo, de forma que o usuário possa acompanhar seu progresso no curso. No canto superior também é possível ver o progresso da seção, no qual cada quadrado corresponde a uma subseção que foi ou não estudada. Os comentários são exibidos logo abaixo, e há um botão para avançar para a página seguinte ('Next step') logo abaixo do conteúdo. É possível ainda avaliar o conteúdo do módulo, em dois botões, um para positivo e outro para negativo (Figura 2.41).

A plataforma pode conter exercícios de perguntas e respostas, nos quais o estudante deve escolher uma alternativa com base no conteúdo já apresentado (Figura 2.41). Caso a resposta esteja correta, é indicado na tela e o estudante pode continuar o conteúdo (o botão de avançar torna-se disponível); além disso, é exibida a quantidade de pontos obtidos pelo estudante e a quantidade de pontos máxima que o estudante pode obter (Figura 2.42). Caso a resposta esteja errada, é indicado que houve erro (Figura 2.43).

The screenshot shows the Stepik Teach interface. At the top, there's a navigation bar with 'stepik' logo, 'Catalog', 'My courses', 'Teach' (highlighted in green), 'Search', 'English', 'WG William', and a user profile icon. Below the navigation, there are four circular icons with corresponding text descriptions:

- Teach your students using content from other courses on Stepik.** Create a [classroom](#) in any course from the catalog and get access to your students' submissions and progress.
- Create your own open or private course** by choosing one of our [plans](#).
- Monetize your knowledge.** Create a [paid course](#) and accept payments directly through Stepik.
- Arrange online competitions,** for example see [Bioinformatics Contest](#).

Below these sections is a heading 'Share Knowledge with Stepik' followed by two columns of features:

<ul style="list-style-type: none"> – An online course editor and a place for your course in our catalog – 20+ types of assignments with autograding – Communication with students through comments and newsletters 	<ul style="list-style-type: none"> – Gradebook and reports PRO – Instructor review PRO – Exams and deadlines PRO – Course access control PRO 	<ul style="list-style-type: none"> – Course certificates PRO – Exams with proctoring Enterprise – External grader Enterprise – LTI integration Enterprise
---	--	---

At the bottom, there's a 'Course title' input field and a 'Create course' button. A small note below says: 'You'll create a draft of the Basic course. You can always upgrade to PRO for more handy features.'

Figura 2.37: Página de criação de cursos da plataforma Stepik

Para mais de uma resposta para a mesma questão, o histórico de respostas fica disponível para que o estudante visualize as respostas já enviadas (Figura 2.44), e a pergunta pode ser respondida novamente mesmo sendo respondida de forma correta anteriormente. Em exercícios de múltipla escolha, caso haja respostas incorretas em meio às corretas, não há indicação de quais delas estão corretas ou incorretas, ou seja, o estudante deve continuar tentando resolver a questão múltiplas vezes, se necessário.

Na página de histórico (Figura 2.44) são apresentadas todas as tentativas do usuário, assim como data de submissão, pontuação e código de submissão. Além disso, são mostrados seção, subseção e curso correspondentes às submissões, assim como links para visualização das páginas dos mesmos.

Os exercícios também podem ser de combinação de colunas correspondentes (Figura 2.45), de forma que o usuário deve colocar as colunas de forma que duas colunas de uma mesma linha sejam correspondentes (e para que a resposta seja válida, todas devem ser correspondentes).

Para ensino de linguagens de programação, o texto do código pode receber um design diferente para destaque, assim como em editores de texto. Eles também podem ser implementados entre o conteúdo do material, assim como no Threelab (Figura 2.46).

Em exercícios de programação, há um editor de código simples no qual são exibidos erros de sintaxe e semântica, para que o estudante possa corrigi-los antes de executar o código - que não é executado para testes, apenas submetido. São apresentadas também as limitações denotadas pelo instrutor, como limite de tempo e memória (Figura 2.47).

Caso haja erros no código, o usuário pode vê-los em um texto semelhante ao terminal do Linux ou Prompt de Comando do Windows, que indica o tipo de erro,

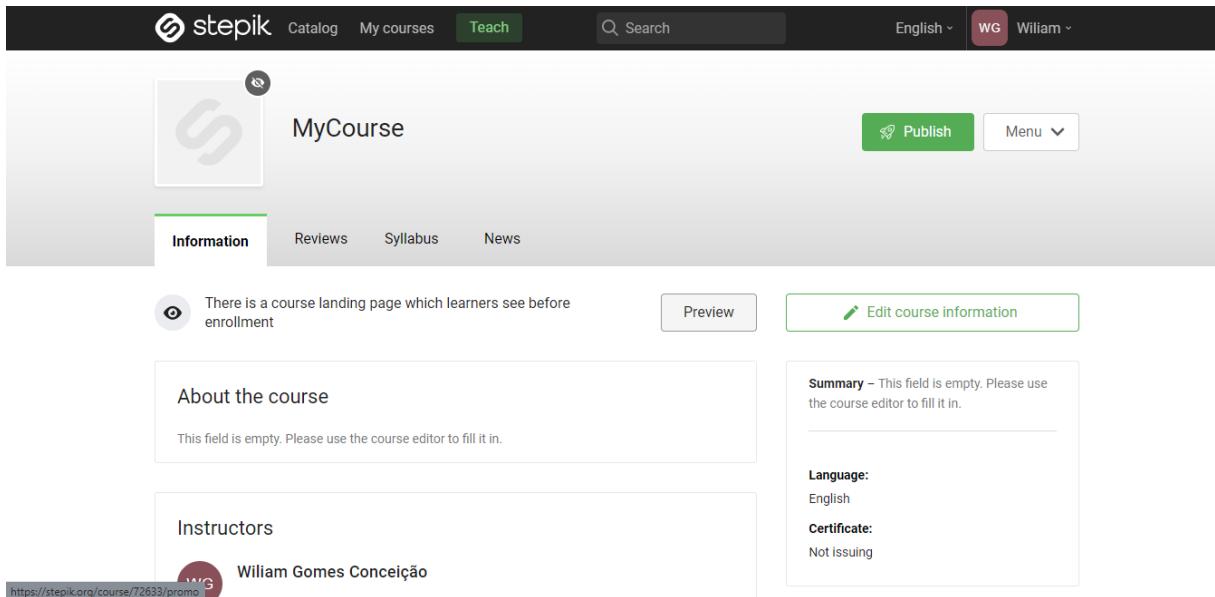


Figura 2.38: Página principal de um curso no Stepik

bibliotecas violadas e linha na qual o erro está. Torna-se disponível também o botão ‘Start Over (Reset)’ que, caso seja pressionado, o código volta ao estado original, escrito pelo instrutor (Figura 2.48). Além disso, é exibida a quantidade de testes nos quais o código passou; entretanto, a nota pode ser ou total ou zero, já que os testes com falha tornam a questão incorreta. Se o código passar em todos os testes, é considerado correto e o estudante pode continuar o aprendizado.

Por fim, pode-se concluir que, apesar de possuir recursos favoráveis, como autocorreção e possibilidade de planos grátis, os recursos principais do Stepik estão no plano pago, o que traria ainda mais gastos à universidade. Em comparação a plataformas como o Threelab e a Autolab, é pouco viável a ela, já que alguns de seus recursos pagos são oferecidos por ambas as plataformas de forma gratuita.

2.5 Programming Education Assistant

“O Programming Education Assistant [29] é um projeto com um conjunto de ferramentas para adicionar a capacidade de autocorreção a cursos de programação em Python que utilizam Jupyter [25] ou Colab [20]. O público alvo principal são professores que desenvolvem cursos de programação em Jupyter [25], uma plataforma de desenvolvimento de software e serviços para computação. Nessa plataforma, ao se integrar com o Programming Education Assistant, é possível autocorrigir atividades dos estudantes, extrair as versões de código dos mesmos nas atividades e envio fácil das notas para a nuvem. A plataforma possui a licença da Apache 2.0 também [30], e o foco principal são universidades japonesas, sendo que os exercícios de demonstração estão disponíveis em japonês

The screenshot shows the Stepik platform interface. At the top, there's a navigation bar with the Stepik logo, 'Catalog', 'My courses', 'Teach', a search bar ('Search'), language selection ('English'), and user profile ('WG Wiliam'). Below the header, the course title 'JavaScript for Beginners' is displayed in large bold letters. A brief course description follows: 'This course covers the basics of programming in JavaScript, as well as some of the tools and data models that are necessary for the practical use of JavaScript.' Below the description are several metrics: a clock icon with '5 hours', a certificate icon with 'Certificate Stepik', a star rating of '4.8' with '58 reviews', and '66,623 learners'. The main content area starts with a section titled 'About this course' which includes a short description of the course purpose and a green 'Join this course' button. To the right, it says 'Free'. Below this, there's a section titled 'Who is this course for' with a note about no special requirements. To the right, a box lists 'This course includes' with '17 lessons' and '28 quizzes'. Further down, there's a section titled 'Meet the Instructors'.

Figura 2.39: Página principal de um curso no Stepik, na visão do estudante

This screenshot shows a specific lesson page from the 'JavaScript for Beginners' course. The left sidebar shows the course structure with sections like '1 Introduction to JavaScript' and '2 Standard objects'. The current lesson, '1.1 Few words about JavaScript', is selected and expanded. The main content area displays the following text:

JavaScript is a programming language that is executed on the user's side using a browser. It allows you to control the elements of a web page - to make them change their properties and location, move, react to events such as moving a mouse or pressing a keyboard, and create many other interesting effects.

JavaScript is often abbreviated to **JS**, which is often used in the names of different frameworks (`Node.js`, `Ember.js`) and also in the extension of files with **JavaScript** code.

JavaScript as a name is a registered trademark and belongs to the company **ORACLE**.

JavaScript was born on December 4, 1995, at least on that day the language got its name. Previously, it was called **LiveScript** and was developed jointly by **Netscape Communications** and **Sun Microsystems** as a language for managing elements and ensuring the interaction of constituent parts of web resources - images, plug-ins, Java applets and other elements used for creating web pages.

JavaScript has a syntax similar to the **C** language, but it has a number of significant differences:

- Ability to work with objects, including determining the type and structure of the object during program execution.
- Ability to transfer and return functions as parameters, and also assign them to a variable.
- Presence of the automatic type matching mechanism.
- Automatic garbage collection.
- Using anonymous functions.

These possibilities will be considered in the following lessons.

Figura 2.40: Página de conteúdo do curso 'JavaScript for Beginners', no Stepik

JavaScript for Beginners
Course progress: 0/66

1.1 Few words about JavaScript. 1 out of 5 steps passed 0 out of 3 points received

In which year did JavaScript get its name?

Select one option from the list

1995
 1987
 2004
 2001

1 point per submission.

Submit

Passed: 27822 Correct submissions: 87%

73 likes 4 dislikes Step 2

Next step >

5 Comments

Most liked

Figura 2.41: Página de quiz do curso ‘JavaScript for Beginners’ no Stepik

JavaScript for Beginners
Course progress: 1/66

1.1 Few words about JavaScript. 2 out of 5 steps passed 1 out of 3 points received

In which year did JavaScript get its name?

Select one option from the list

Right.
 1995
 1987
 2004
 2001

Next step Solve again

Your submissions You got: 1 point out of 1

Passed: 27823 Correct submissions: 87%

73 likes 4 dislikes Step 2

Next step >

Figura 2.42: Página de quiz do curso ‘JavaScript for Beginners’ no Stepik na qual o usuário respondeu a pergunta corretamente

The screenshot shows a course page for 'JavaScript for Beginners' on the Stepik platform. The sidebar on the left lists course sections and steps. The main area displays a step titled '1.1 Few words about JavaScript' with the question 'In which year did JavaScript get its name?'. Below the question, it says 'Select one option from the list'. A red 'X' icon indicates an incorrect answer. The options are: 1987 (selected), 2001, 1995, and 2004. A green 'Try again' button is visible. At the bottom, it shows 'Your submissions' and 'You got: 1 point out of 1'. The overall theme is dark with green highlights.

Figura 2.43: Página de conteúdo do curso ‘JavaScript for Beginners’, no Stepik no qual o estudante respondeu a pergunta de forma incorreta

e inglês. O projeto não é oficialmente da Google, mas foi criado por funcionários da mesma, que especificamente nega qualquer garantia quanto a qualidade, comercialização e adaptação para propósito particular. A seguir, veremos todas as plataformas e softwares para desenvolvimento de backend do autocorretor. A linguagem utilizada na plataforma, assim como de alguns dos componentes, é Python”. A seguir, teremos detalhes sobre as ferramentas utilizadas pela plataforma.

2.5.1 Jupyter

Jupyter [25] é uma plataforma de desenvolvimento de software e serviços de computação (Figura 2.49). Possui uma interface online ou pode ser executada através de instalação no ambiente Linux (Jupyter Notebook) e desenvolvido abertamente no GitHub. No Programming Education Assistant, é usado como a plataforma de exibição dos códigos e apostilas, e cada bloco de texto é lido com o botão ‘Run’, sendo que, caso este seja um bloco de código, é executado e seu resultado exibido na própria apostila (Figura 2.50).

O carregamento do Jupyter no sistema também permite instalação de aplicações, atualizações e componentes na máquina do estudante, sendo que é possível adicionar os comandos de instalação, passo a passo, na apostila, para que estes sejam executados (Figura 2.51). Assim como respostas corretas, a plataforma também exibe erros de interpretação (já que Python é uma linguagem interpretada) de respostas incorretas (Figura 2.52).

Submissions of the [Few words about JavaScript](#) lesson from the [JavaScript for Beginners](#) course

← Step 2: Multiple Choice Quiz →

User	Submissions	Date	Score
Wiliam Gomes Conceição	#216226249	5 minutes ago	0
Wiliam Gomes Conceição	#216224011	10 minutes ago	1

© 2013 – 2020. Stepik
[Terms of service](#) and [privacy policy](#)

Pricing
[Press](#)
[Team](#)

Help Center
[About Stepik](#)
[Contacts](#)

216,039,121 submissions

Download on the [App Store](#) GET IT ON Google Play

Public user contributions licensed under [CC BY-SA 4.0](#) license with attribution required

Figura 2.44: Página de histórico de respostas para uma pergunta no Stepik

2.5.2 Go Toolchain

Go [17] é uma linguagem de execução de código da Google que tem como função facilitar a criação de software, na qual estão implementados tanto o servidor quanto o autocorretor do Programming Education Assistant (Figura 2.53). Seus termos de serviço, portanto, são os mesmos da Google [18]. É possível fazer testes na plataforma, na seção ‘Play’, na qual o usuário executa um código pré-determinado na linguagem Go numa lista apresentada pela plataforma, e sua saída é apresentada abaixo do editor, assim como no Threelab (Figura 2.54). Dessa forma, permite que a saída também seja exibida na execução de cada passo no Jupyter Notebook.

2.5.3 Bazel

Bazel [16] é uma ferramenta de configuração e testes de linguagens de programação, desenvolvida pela Google, que facilita a configuração e testes com rede de execução avançada, distribuição de cache, memória de armazenamento de dados que serão utilizados em breve pela CPU, de alta velocidade, análise otimizada de dependências e execução paralela (Figura 2.55). Com esta ferramenta, é possível executar o código do Programming Education Assistant de forma mais rápida, para que o tempo de resposta na tela seja menor. Os termos de serviço da ferramenta também são os da Google, por ter sido criada por esta empresa também [18].

Figura 2.45: Página de exericício de Match no Stepik

2.5.4 Docker

Docker [12] é uma plataforma de desenvolvimento, envio e execução de aplicações que permite a divisão de aplicações por meio da infraestrutura, de forma a entregar software de forma mais rápida, reduzindo o tempo entre escrever e executar testes de código (Figura 2.56). Está sob licença Apache 2.0 [30], e para a plataforma Programming Education Assistant, é usada para dividir a infraestrutura da aplicação e também para execução de testes locais no backend da mesma.

2.5.5 Google Cloud SDK

Google Cloud SDK [19] são bibliotecas e ferramentas para interagir com os produtos e serviços de computação em Nuvem da Google, que permitem armazenamento de dados em Nuvem; podem ser executados tanto em Linux, quanto Windows e macOS, e possuem planos pagos ou grátis, e também está sob a licença de serviços da Google [18]. No plano pago, o usuário deve se comunicar com a equipe da Google Cloud SDK para definir preço com base nas necessidades da empresa ou do usuário. No plano grátis, o usuário tem limite de 12 projetos. Para o Programming Education Assistant, a Google Cloud é o servidor de respostas dos estudantes, ou seja, os instrutores devem criar um projeto nesta plataforma (Figura 2.57) para que as respostas e a correção dessas sejam enviadas e armazenadas.

A ferramenta Programming Education Assistant é extremamente útil, visto que as respostas podem ser armazenadas na Nuvem e enviadas em tempo real, para cada um dos estudantes, e por ser grátis, pode ser implementada em universidades também. Entretanto, a limitação por ser apenas em ambiente Jupyter ou Colab, além de ser em

The screenshot shows a computer interface for the Stepik learning platform. On the left, there's a sidebar with a dark background containing a navigation tree for a 'JavaScript for Beginners' course. The tree includes sections like '1 Introduction to JavaScript' (with '1.2 Code implementation, ...' highlighted in green), '2 Standard objects', and various sub-sections such as '1.3 Variables, data types.', '1.4 Simple operations.', etc. At the top right, there's a user profile for 'William' with a dropdown arrow. The main content area has a light gray background. It starts with a heading 'Let's consider both ways:' followed by text about placing code inside an HTML file. Below this is a code editor window showing a snippet of JavaScript code:

```
<script>
  ...
</script>
```

Then, under the heading 'Let's see what the code looks like on this page:', there's another code editor window showing the full HTML code for a sample page:

```
<html>
  <head>
    <title>Code sample page with JavaScript</title>
    <script>
      alert("Hello World!");
    </script>
  </head>

  <body>
    This is the main page text
  </body>
</html>
```

Figura 2.46: Página de conteúdo com código no curso ‘JavaScript for Beginners’ no Stepik

Python, torna a ferramenta menos viável, por haver uso de mais de uma linguagem de programação nas universidades, além de tornar o ambiente Jupyter obrigatório. Semelhantemente, não é possível adaptar a ferramenta a páginas que não sejam no estilo da plataforma, com uso de Go, o que dificulta ainda mais a adaptação.

No capítulo 3.12 veremos uma comparação entre todas as plataformas. A seguir, veremos a solução para os problemas citados - facilitação para criação de conteúdo do professor e a ausência de feedback rápido para os estudantes, a ferramenta criada neste projeto final, o Threelab.

In this task, you have some working code that is commented out. You need to remove all multiline comments so that the code works and click the "Send" button. Warning! You must leave single-line comments to prevent of dangerous code! :)

Write a program, test using stdin → stdout

Time limit: 5 seconds
Memory limit: 256 MB

```

1 /* function testComment(a, b) { */
2     var x;
3 //    a = b / z_dead;
4     x = a * b;
5
6 //    return x;
7 }
```

Maximum 1 point per submission.

Submit

Passed: 6184 Correct submissions: 31%

Figura 2.47: Página de exercício de código no curso ‘JavaScript for Beginners’ no Stepik

Write a program, test using stdin → stdout

✗ Incorrect, sorry.

Now you have access to the [Forum of Solutions](#) where you can discuss your solution with others.

Failed test #1 of 10. Runtime error
`/sandbox/main.js:9
 a = b / z_dead;`

`^`

ReferenceError: z_dead is not defined
`at testComment (/sandbox/main.js:9:15)
 at Socket.<anonymous> (/sandbox/main.js:5:17)
 at Socket.emit (events.js:182:13)
 at addChunk (_stream_readable.js:283:12)
 at readableAddChunk (_stream_readable.js:264:11)
 at Socket.Readable.push (_stream_readable.js:219:10)
 at Pipe.onStreamRead [as onread] (internal/stream_base_commons.js:94:17)`

Time limit: 5 seconds
Memory limit: 256 MB

```

1 function testComment(a, b) {
2     var x;
3     a = b / z_dead;
4     x = a * b;
5
6     return x;
7 }
```

Submit **Start over (reset)**

Figura 2.48: Página de exercício de código no curso ‘JavaScript for Beginners’ no Stepik na qual o estudante submeteu um código que não passou em todos os testes

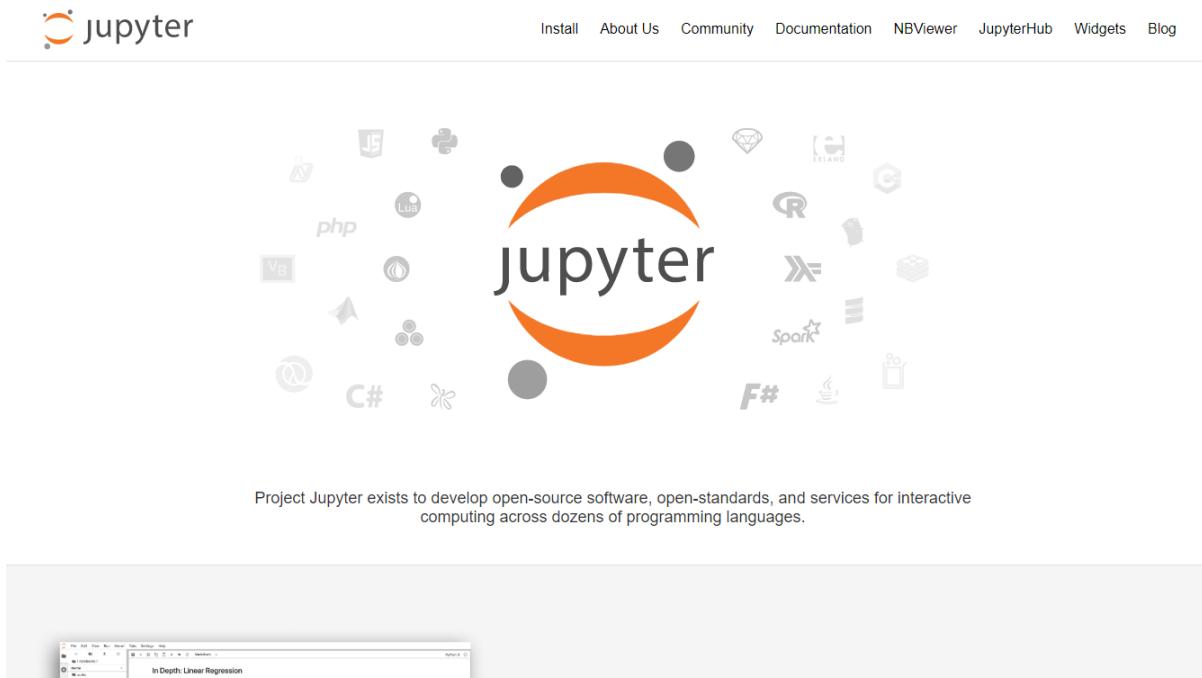


Figura 2.49: Página inicial da plataforma Jupyter

This screenshot shows a Jupyter notebook interface. The top navigation bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Help', 'Logout', and a 'Python 2' kernel selector. The main area displays a code cell with the following content:

```
In [3]: d = Date(2019, 4, 7)
print('{}/{}/{}'.format(d.year, d.month, d.day))
2019/4/7
```

Below it is another code cell:

```
In [4]: class Date(object):
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    # This is another special function that allows us to call print(...) on a
    # Date object.
    def __str__(self):
        return '{}/{}/{}'.format(self.year, self.month, self.day)
```

And a third code cell:

```
In [5]: d = Date(2019, 4, 7)
print(d)
2019/4/7
```

At the bottom, there's a section titled 'Exercise: Using the Date class and its properties'.

Figura 2.50: Página de carregamento de exemplo do Programming Education Assistant no Jupyter

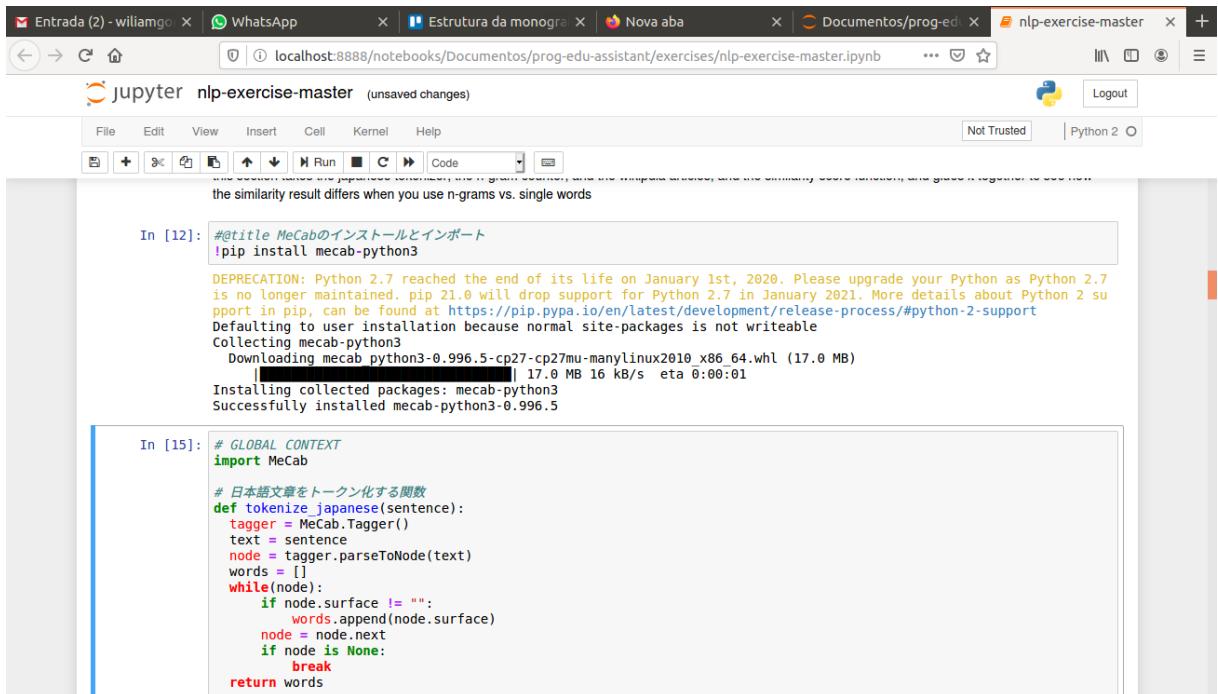


Figura 2.51: Página de instalação de componente por meio do Jupyter Notebook



Figura 2.52: Página de exibição de resposta incorreta no Jupyter Notebook, da atividade dataframe-pre1-master, dentre os exercícios do Programming Education Assistant

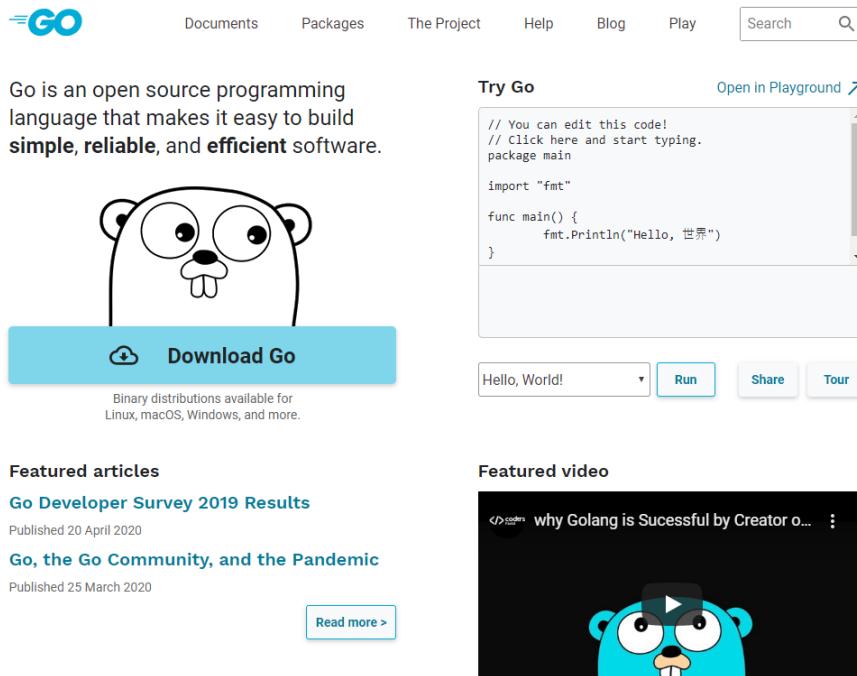


Figura 2.53: Página inicial do Go

The screenshot shows the Go Playground at <https://play.golang.org/>. The top bar includes buttons for Run, Format, Imports, Share, Clear, and About. The main area is a code editor with the following Go code:

```

1 package main
2
3 import (
4     "fmt"
5     "strings"
6     "time"
7 )
8
9 func main() {
10     const col = 30
11     // Clear the screen by printing \x0c.
12     bar := fmt.Sprintf("\x0c[%-%vs]", col)
13     for i := 0; i < col; i++ {
14         fmt.Printf(bar, strings.Repeat("=", i)+">")
15         time.Sleep(100 * time.Millisecond)
16     }
17     fmt.Printf(bar+" Done!", strings.Repeat("=", col))
18 }
19
20
21
22
23
24
25
26
27
28
29
30
```
[=====] Done!
Program exited.

```

Below the code editor is a terminal window showing the output of the program: '[=====] Done! Program exited.'

Figura 2.54: Editor de código do Go, com saída

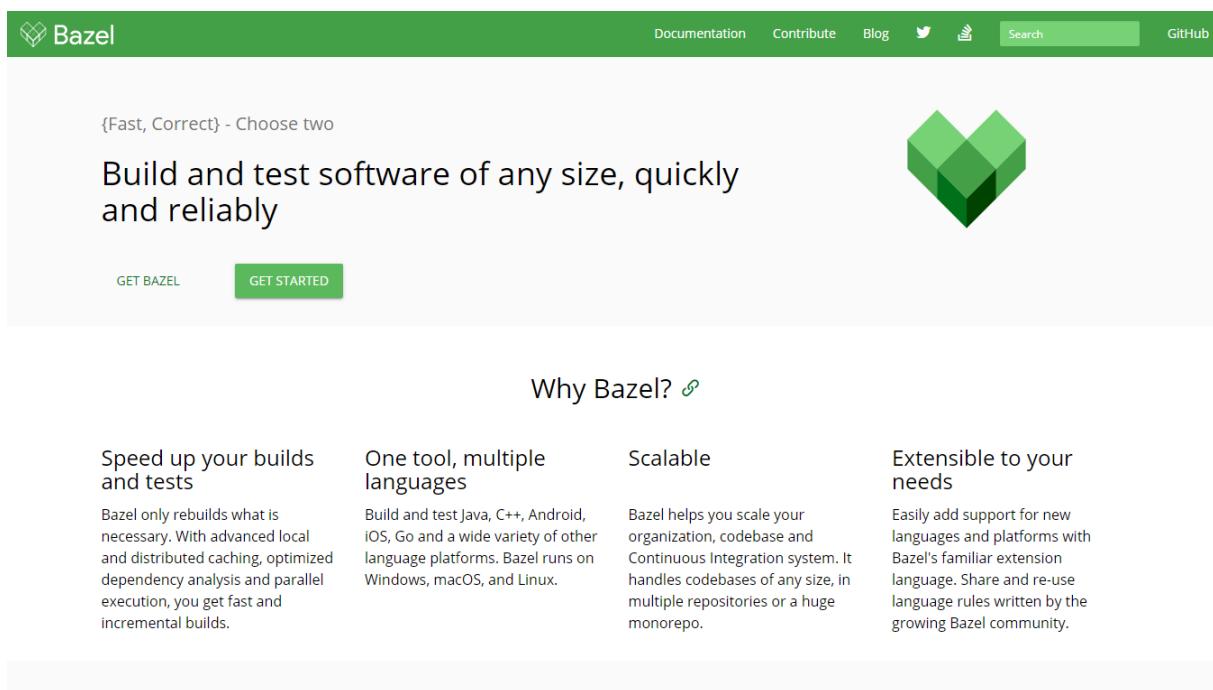


Figura 2.55: Página inicial do Bazel

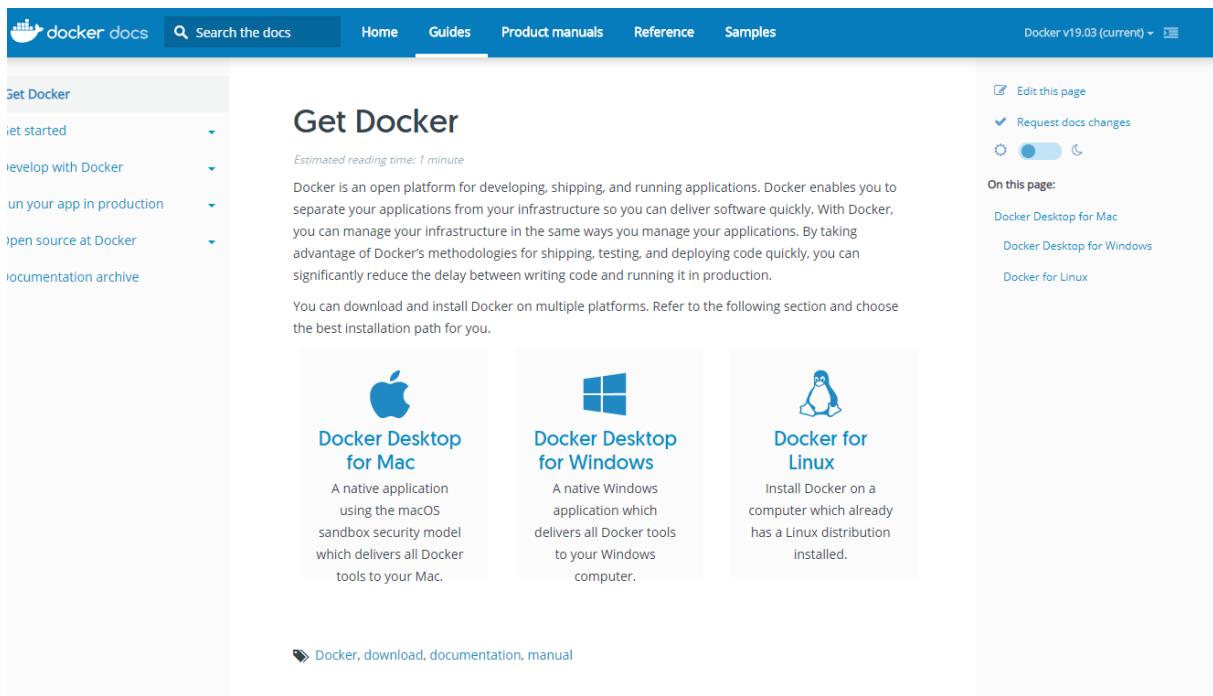


Figura 2.56: Página inicial do guia de instalação do Docker

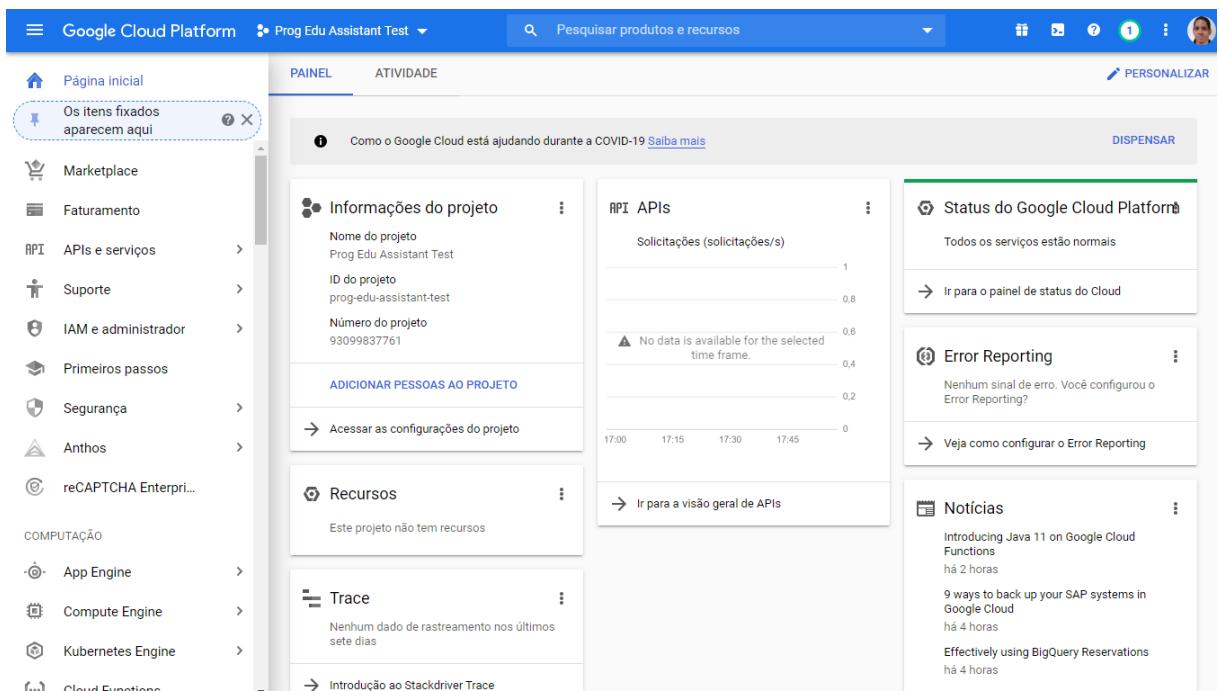


Figura 2.57: Página de projeto vazio no Google Cloud

# Capítulo 3

## Solução - Threelab

O Threelab é uma ferramenta criada com o objetivo de auxiliar o corpo docente no ensino e na correção de atividades de programação e o corpo discente no aprendizado de linguagens de programação, através do feedback rápido e a facilidade de criação de páginas por parte do professor, esta em PHP e MySQL. Além disso, esta utiliza uma aplicação para armazenamento de notas e atividades dos estudantes, acessível para o professor. É uma ferramenta simples, tanto de aprendizado quanto de integração, é possível customizar, devido à liberdade de criação, com ótimo desempenho, já que a construção da página é feita de forma rápida e simples, utilizando apenas elementos HTML, e possível de ser utilizada offline.

A ferramenta foi criada a partir do repositório de aulas do professor Dr. Rodrigo Rocha Gomes e Souza [14], que foi criado a partir do gerador de sites estáticos Jekyll [32], para apresentação de conteúdo em aulas na Universidade Federal da Bahia. A partir desta ferramenta, foi possível criar páginas com novo estilo de layout, no qual a página é apresentada com divisão em conteúdo, editor e entrada/saída de código. Além disso, foram criadas páginas para informação de notas das atividades e das respostas dos estudantes para as mesmas, que também serão apresentadas a seguir, na seção ‘Dashboard’. Com isso, é possível também que os alunos façam login, para que suas respostas sejam armazenadas.

Na próxima seção, será apresentada a solução já existente antes de o Threelab ser criado.

### 3.1 Solução existente

No repositório de aulas, o conteúdo é apresentado com páginas de rolamento, com conteúdos de determinadas matérias da Universidade Federal da Bahia, ministradas pelo professor Dr. Rodrigo Rocha Gomes e Souza. A seguir, veremos como é estruturada a exibição do conteúdo do repositório.

A página inicial é um conjunto de hiperlinks com todo o conteúdo das matérias lecionadas, divididas em graduação e pós graduação (Figura 3.1). As páginas das matérias contêm seu conteúdo dividido em tópicos e podem conter algumas informações extras, como informações de contato, bibliografia e informações sobre avaliação, como mostra a Figura 3.2.

Disciplinas (UFBA)

## Início

Disciplinas ministradas pelo professor Rodrigo Rocha na UFBA.

### Disciplinas

Graduação:

- MATA37: Introdução à Lógica de Programação
- MATA56: Paradigmas de Linguagens de Programação
- MATA55: Programação Orientada a Objetos
- MATA62: Engenharia de Software I
- MATB25: Tópicos em Engenharia de Software - controle de versão e testes
- MATC89: Aplicações para Dispositivos Móveis
- MATA60: Banco de Dados
- MATA40: Estruturas de Dados e Algoritmos
- MAT220: Empreendedores em Informática (duas turmas)
- MATA76: Linguagens para Aplicação Comercial

Pós-graduação:

- MATE94: Engenharia de Software Experimental

Ver também: [Ensino](#).

Figura 3.1: Página inicial do repositório de aulas

Disciplinas (UFBA)

## MATA56 - Paradigmas de linguagens de programação - 2020.1

### Objetivo

Compreender e aplicar conceitos e paradigmas de linguagens de programação, de forma a facilitar o aprendizado de novas linguagens de programação e fornecer elementos para uma comparação crítica entre linguagens.

### Informações básicas

- Professor: Rodrigo Rocha [rodrigorgs@ufba.br](mailto:rodrigorgs@ufba.br)
- Entre no [Google Classroom](#) e inscreva-se nesta turma clicando no botão + e digitando o código `7aaxcjax`.

### Avaliações

- Prova prática de Prolog [peso: 3, individual]
- Caracterização de uma linguagem de programação [peso: 3, em grupo de até 5 pessoas]
- Prova prática de programação funcional e concorrente com JavaScript [peso: 4, individual]

### Ferramentas

Figura 3.2: Página do componente curricular Paradigmas de Linguagens de Programação

As páginas do conteúdo podem conter texto, código de exemplo, o qual o estudante não pode modificar, imagens ou editor de código na linguagem especificada, sendo que eram suportadas Scheme e JavaScript. Na Figura 3.3, pode-se ver a estrutura de uma

página com imagem e código, e na Figura 3.4 é possível ver um código demonstração na linguagem JavaScript, sendo a saída exibida no console do navegador (geralmente, apertando-se F12).

Temos então como limitações que, para ver a saída é necessário comprimir a página (ao abrir a aba de console do navegador) e caso seja necessário testar entradas, isso não é possível, visto que não há como digitar conteúdo na entrada padrão.

## Prolog: recursão

### Revisão

Resolução de exercícios selecionados da aula passada: 12, 17 e outros conforme solicitação.

### Recursão

```
progenitor(maria, jose).
progenitor(joao, jose).
progenitor(joao, ana).
progenitor(jose, julia).
progenitor(jose, iris).
progenitor(iris, jorge).
```

Figura 3.3: Página de Prolog do componente curricular Paradigmas de Linguagens de Programação, nesta temos apresentação de imagem e código que não é possível para o estudante modificar

O código das páginas é feito em Markdown, de forma a ser compatível com a arquitetura. A seguir veremos um exemplo de código, seguido de informações detalhadas sobre a arquitetura.

No início do código, deve ser definido o layout, que é page (o layout padrão) além do título da página, e podem ser definidos também data, uma categoria e alguma variável, caso necessário. O conteúdo é expresso em título, subtítulo, parágrafos, blocos de código e alguns outros itens, como imagens e hiperlinks. O subtítulo, de tag ‘`<h2>`’ representada em Markdown por ‘`##`’, determina a divisão do conteúdo. Os parágrafos são determinados por texto comum e editores de código são representados por blocos ‘`<textarea>`’ com variável ‘`class`’ equivalendo a ‘`code`’, nos quais o conteúdo de dentro da tag será o código, e a linguagem por padrão é JavaScript. Entretanto, caso seja necessário utilizar uma linguagem diferente, deve-se modificar o conteúdo da variável ‘`lang`’ (a outra linguagem que pode ser utilizada é Scheme, sendo que para que a linguagem seja modificada, `lang` deve ter o valor ‘`scheme`’). Ao se utilizar ‘`**`’ o texto entre os asteriscos duplos passam a ser em negrito.

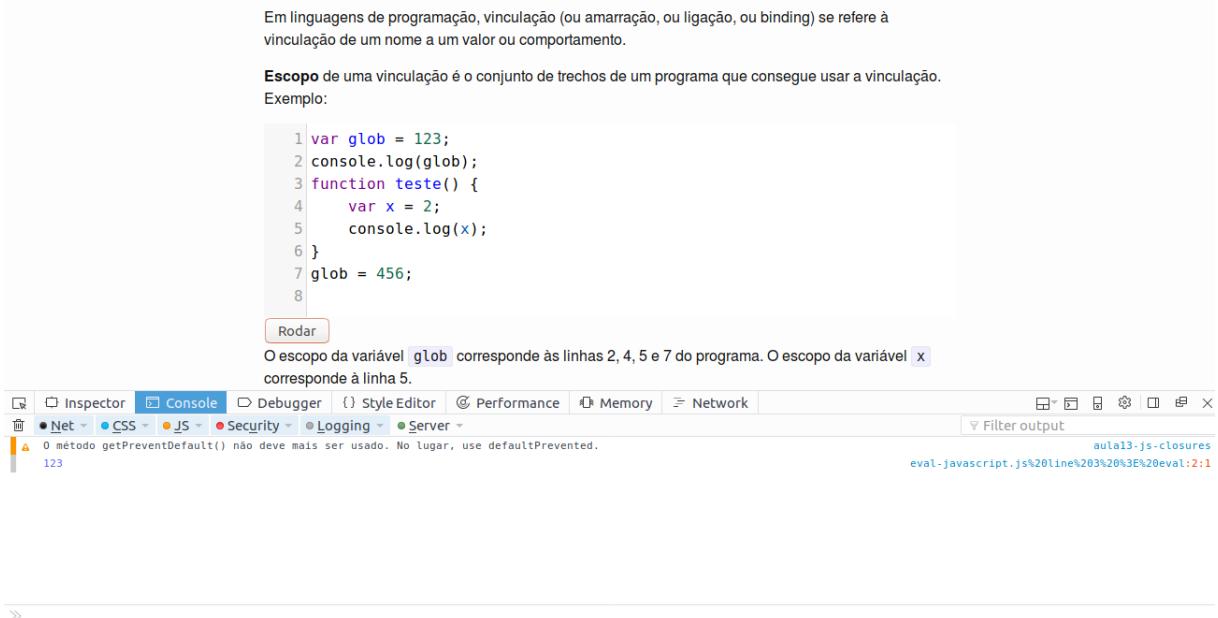


Figura 3.4: Página de demonstração, na qual há um editor de código e podemos observar a saída no console

```

<!---
layout: page
title: "JavaScript: Nomes, vinculação, escopo e closures"
date: 2017-03-08 16:40:00 -0300
categories: aula

Nomes, vinculação e escopo

Em programas de computador, **nomes** representam coisas, como valores, comportamentos etc. Os nomes são **vinculados** às coisas que representam em diversos momentos. Por exemplo:

* O nome <code>if é vinculado ao comportamento de estrutura de seleção em **tempo de projeto de linguagem**
* Em C, quando você escreve #define PI 3.14159, o nome PI é vinculado ao valor 3.14159 em **tempo de compilação**
* Em geral, variáveis são vinculadas aos seus valores em **tempo de execução**

Em linguagens de programação, vinculação (ou amarração, ou ligação, ou binding) se refere à vinculação de um nome a um valor ou comportamento.

Escopo de uma vinculação é o conjunto de trechos de um programa que consegue usar a vinculação. Exemplo:

<textarea class="code">
var glob = 123;
console.log(glob);
function teste() {
 var x = 2;
 console.log(x);
}
glob = 456;
</textarea>

O escopo da variável 'glob' corresponde às linhas 2, 4, 5 e 7 do programa. O escopo da variável 'x' corresponde à linha 5.

A variável 'glob' tem um escopo **global**, pois é acessível em todo o programa (essa é uma definição simplificada). A variável 'x' tem um escopo **local**, pois só é acessível dentro da função no qual ela é definida. A variável 'teste' também possui escopo global (lembre-se de que em Javascript funções também são valores!).

A maioria das linguagens de programação adota regras de **escopo léxico** (ou estático). Isso significa que é possível determinar o escopo de uma variável somente lendo o código-fonte do programa. Algumas linguagens de programação usam o **escopo dinâmico**: uma situação comum nesse caso é uma função ter acesso a variáveis definidas na função que a chamou. Pode-se argumentar que uma linguagem de programação que adota o escopo dinâmico tende a resultar em programmas mais difíceis de entender.

```

Figura 3.5: Código em Markdown de uma página na solução pré-existente, de um conteúdo de Paradigmas de Linguagens de Programação

A seção com três crases seguidas denota um bloco de código cuja linguagem pode ser definida pelo usuário ao colocar seu nome após as crases triplas, como por exemplo, no conteúdo do código da Figura 3.6 onde temos ‘html’, demonstrando que a linguagem é HTML.

```


```
layout: page
title: Demonstração
```

Para incluir um editor de código JavaScript, use

```html
<textarea class="code">
codigo inicial do editor
</textarea>
```

Resultado:

<textarea class="code">
codigo inicial do editor
</textarea>

Outras linguagens

Adicione ao `<textarea>` a classe `lang-NOMEDALINGUAGEM`, ex.: `lang-scheme`, `lang-ruby`.

Editor de código com resposta

Exemplo:

```html
<textarea class="code">
codigo inicial do editor
</textarea>

<textarea class="answer">
resposta
</textarea>
```

Resultado:


```

Markdown ▾ Largura da tabulação: 8 ▾ Lin 1, Col 1 ▾ INS

Figura 3.6: Código no qual há uma sequência de crases triplas, que são códigos não modificáveis pelo aluno

A seguir, vamos abordar a arquitetura da solução pré existente.

### 3.1.1 Arquitetura

Nas próximas subseções, serão apresentados todos os componentes da arquitetura, estes pré existentes, assim como suas funções e integrações.

#### Github Pages

O site é hospedado no Github Pages [24], uma ferramenta do Github que permite que seja construído um site a partir de um repositório do Github. O Github Pages tem compatibilidade com o Jekyll [32], a ferramenta que será apresentada a seguir.

#### Jekyll

Jekyll é uma ferramenta que permite a construção de sites a partir de código, desenvolvida em Ruby e criada pelo co-fundador do Github [23] (plataforma de hospedagem de código com controle de versão), Tom Preston-Werner. Com ela, é possível transformar

códigos em Markdown, Liquid, HTML e CSS em sites, desde que siga uma estrutura pré determinada. A ferramenta permite a construção de uma estrutura para um site em HTML a partir da estrutura definida pelo usuário. É possível também exibir o conteúdo do site, sendo que isso pode ser feito no navegador do usuário pelo próprio sistema. Apesar do layout do Jekyll padrão ser o ‘page’, é possível criar novos layouts ao adicionar um arquivo HTML com seu nome e uma estrutura pré-definida na pasta ‘\_layouts’.

## Codemirror

O editor utilizado, o CodeMirror [7], é um editor de texto implementado em JavaScript para navegador, que possui suporte para diversas linguagens [8], dentre elas, JavaScript e Scheme, disponíveis na solução existente. Com o CodeMirror, é possível adicionar o editor às páginas criadas com o Jekyll, assim como executar código nas páginas. Os blocos ‘<textarea>’ serão substituídos pelos editores do CodeMirror, e todo o seu conteúdo será código da linguagem estabelecida pela variável ‘lang’. O botão ‘rodar’ permite que o código seja executado.

## Biwascheme

BiwaScheme [22] é o interpretador de Scheme escrito em JavaScript utilizado na plataforma, que também utiliza licença MIT. Pode ser utilizado em navegadores ou em dispositivos móveis, além de Node.js, apenas carregando o arquivo ‘biwascheme.js’.

As funcionalidades disponíveis são praticamente todas as do R7RS small [1], incluindo continuação de primeira classe e otimização de chamada na cauda. As principais funcionalidades que faltam são regras de sintaxe, exceções e sistema de bibliotecas. Além disso, há duas limitações causadas devido à criação em JavaScript e que, segundo os criadores, não serão resolvidas: strings são imutáveis e ‘integers’ não são diferenciados de ‘float’.

## Eval

Eval [28] é uma função que se comporta como um interpretador de JavaScript. Como parâmetro, recebe uma string, que corresponde a um código feito na linguagem JavaScript, que caso seja nula, retorna ‘undefined’, e caso não seja, retorna o valor computado do código da expressão dado. Assim sendo, um código de JavaScript é interpretado pelo próprio JavaScript.

## Submissões

Para submissões no repositório, era necessária a instalação de PHP em um servidor (do qual todos os alunos teriam acesso), pelo professor, assim como alguns passos para início da avaliação e submissão de respostas: carregamento da página em Markdown da prova, criação de um banco de dados e login no mesmo, para então realizar a avaliação, e no final desta, o professor deveria armazenar os arquivos num pen drive para copiar para seu computador pessoal [13]. Havia páginas de salvamento e carregamento de respostas, login e logout de alunos e listagem de respostas para o professor. Na Figura 3.7 vemos o layout de uma página de submissão sendo carregada e armazenando o valor enviado ao ser pressionado o botão ‘Rodar’.

Com isto, vemos que o processo de armazenamento não possuía uma forma de visualização de notas (dashboard), para que os alunos pudessem ver seu progresso e porcentagem de notas, que deveria ser enviada a cada um deles (por email ou outra plataforma).

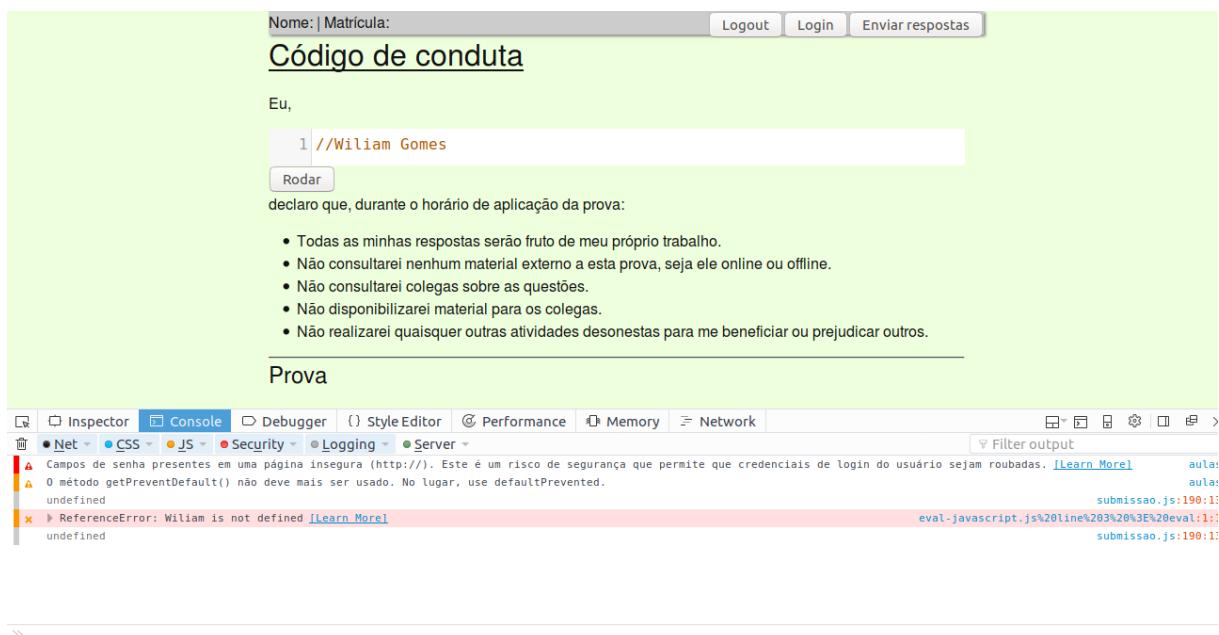


Figura 3.7: Página carregada num servidor local para ser necessária a realização de uma avaliação, após o armazenamento de um valor

Vemos também, na Figura acima, que o login era feito na barra superior da página, na qual o estudante informava login e senha para que fosse possível armazenar as respostas no banco de dados.

## Demonstração

Para incluir um editor de código JavaScript, use

```
<textarea class="code">
codigo inicial do editor
</textarea>
```

Resultado:



### Outras linguagens

Adicione ao `textarea` a classe `lang-NOMEDALINGUAGEM`, ex.: `lang-scheme`, `lang-ruby`, `lang-cpp`.

```
1 #include <iostream>
2
```

Figura 3.8: Exemplo de página criada no Threelab - layout Page

## 3.2 Threelab

Através de modificações na solução existente, foi criada a plataforma Threelab, que implementou um layout dividido em três partes, além de adicionar campos de entrada e saída e um dashboard (painéis de visualização), que veremos a seguir.

As ferramentas continuam sendo as mesmas utilizadas na solução existente, já que o layout foi a mudança mais significante da plataforma. Na Figura 3.8 é possível ver a exibição padrão do editor numa página, neste caso, da ferramenta Jekyll, agora com a presença de caixas para definição de entrada e saída dos casos.

O Jekyll foi utilizado para construção do layout das páginas, sejam elas páginas padrão, ou páginas específicas para programação. As páginas criadas são de layout ‘triple-page’, sendo que estas são divididas em três (Figura 3.9): a primeira parte, para explicação do conteúdo, dicas ou exemplos, feita pelo professor. A segunda parte, contendo o editor de código, e a terceira é dividida, horizontalmente, em duas: entrada, na parte de cima, e embaixo a exibição da saída do código. O professor pode criar páginas normais divididas em tags ‘`<h2>`’ e, caso queira que a página seja dividida, basta uma alteração no arquivo de página, que foi definida como o valor de uma variável ‘`triple`’ sendo ‘`true`’. Dessa forma, se ‘`triple`’ possuir ‘`true`’ como valor, a página será dividida em várias, horizontalmente em três; a tag ‘`<h2>`’ seria a tag que definiria o início de uma nova página e, consequentemente, todo o conteúdo que existir após o ‘`<h2>`’ pertencerá à página desta tag, e para que o conteúdo pertença a outra página, uma nova tag deve existir. Na Figura 3.10 é possível ver a estrutura da página tripla: as tags ‘`<h2>`’ da página são representadas por ‘`##`’ no início, pois estes representam esta tag na linguagem Markdown. Assim, os

alunos navegarão entre estas páginas criadas por meio de botões de avançar e retroceder.

```

1 vetor = [1, 2, 3, 4, 5, 6, 7, 8]
2 vetor.shift();
3 console.log(vetor);
4
5 teste(8, vetor.length);
6

```

**Entrada**

**Saída**

Rodar | Retroceder | Avançar | Expandir editor

Figura 3.9: Exemplo de página criada no Jekyll - layout Triplepage

Na página tripla, há também alguns botões (o botão ‘rodar’ foi mantido): O botão de expansão do editor, no qual o editor é exibido em 100% do comprimento da tela (Figura 3.11), e ambos o conteúdo de orientação e o console são ocultados da visualização do usuário. Dessa forma, a visualização do código torna-se mais legível, de forma que o editor de código ocupa toda a janela do navegador, permitindo melhor detecção de imprecisões. O botão de avançar permite que a próxima tela seja acessada, e este apresenta-se disponível para clique apenas se houver uma página seguinte. Da mesma forma, o botão de retroceder, que permite que a tela anterior seja acessada, torna-se indisponível caso esteja sendo exibida a primeira página do conjunto. As páginas, caso não possuam editor de código, exibem apenas o conteúdo (Figura 3.12).

Quando o botão de avançar ou retroceder é utilizado, o editor volta ao tamanho padrão e a página é avançada ou retrocede como desejado pelo usuário.

Na página sem editor, é possível ver que o botão de execução de código não existe e o editor não é apresentado, pois não há código para ser executado ou resultado a ser exibido na tela.

Na Figura 3.9 é exibido o editor no layout do Threelab, no qual ele é posicionado no meio da página, entre a página de orientação e o console.

Para que o editor seja exibido, assim como na solução existente, é necessário inserir o código dentro de tags ‘<textarea>’. No momento, apenas as linguagens C++, além de Scheme e JavaScript, já implementadas, estão disponíveis; logo, caso o instrutor deseje criar uma atividade em Scheme, ele deverá alterar a linguagem para ‘scheme’, atribuindo

```

layout: triple-page

title: Demonstração

Página com o triple-page

Use o layout 'triple-page'

Cada seção (`h2` em HTML, `##` em Markdown) inicia uma página.

Se dentro da seção houver um editor de código, ele aparece na coluna do meio, e o console aparece na coluna da esquerda.

Execute o código para ver em ação

<textarea class="code">

function meuMap(funcao, lista) {

 let resultado = [];

 for (let i = 0; i < lista.length; i++) {

 resultado.push(funcao(lista[i]));

 }

 return resultado;

}

teste([2, 4, 6], meuMap(x => x * 2, [1, 2, 3]));
</textarea>
Outra página

Esta é uma página apenas explicativa, sem editor de código.


```

Markdown ▾ Largura da tabulação: 8 ▾ Lin 1, Col 1 ▾ INS

Figura 3.10: Código de página tripla em Markdown

este valor à variável ‘lang’. No caso de C++, devemos alterar a classe para “code = ‘lang-cpp’”

### 3.2.1 Novo interpretador

O Threelab suporta três linguagens de programação, como vimos anteriormente: além das linguagens já suportadas, ela suporta C++. Para isto, foi utilizado o interpretador JSCPP, que será apresentado a seguir.

JSCPP [21] é o interpretador de C++ utilizado na plataforma, feito na linguagem JavaScript. Seu pré requisito é apenas um navegador atualizado. Utiliza a licença MIT [27].

Em suma, as funcionalidades de C++ disponíveis no JSCPP são operadores, tipos primitivos, variáveis, vetores, ponteiros, condicionais, switchs, for, while, do-while, funções, escopos e diretivas de pré processador (como os includes). As funcionalidades ainda não disponíveis são go to, funções orientadas a objeto, namespaces e suporte a múltiplos arquivos.

As bibliotecas suportadas são iostream, cmath, ctype, cstring, cstdio e cstdlib, sendo estas duas últimas parcialmente.

### 3.2.2 Saída e testes

A saída do código será exibida no console, no campo de baixo da coluna à direita, se for o layout ‘triplepage’, e no campo de baixo à direita se for o layout ‘page’. Caso a



```

1 function meuMap(funcao, lista) {
2 let resultado = [];
3 for (let i = 0; i < lista.length; i++) {
4 resultado.push(funcao(lista[i]));
5 }
6 return resultado;
7 }
8
9 teste([2, 4, 6], meuMap(x => x * 2, [1, 2, 3]));
10

```

The screenshot shows a code editor window with the following code:

```

1 function meuMap(funcao, lista) {
2 let resultado = [];
3 for (let i = 0; i < lista.length; i++) {
4 resultado.push(funcao(lista[i]));
5 }
6 return resultado;
7 }
8
9 teste([2, 4, 6], meuMap(x => x * 2, [1, 2, 3]));
10

```

Below the code, there is a toolbar with the following buttons:

- Rodar (Run)
- Retroceder (Back)
- Avançar (Forward)
- Contrair editor (Collapse editor)

Figura 3.11: Editor exibido em 100% do comprimento da tela no Jekyll

saída seja o valor de alguma variável, esta será exibida no topo da coluna (Figura 3.13), e no caso de um teste, incorporado ao código, é mostrado ‘OK’ para cada um deles no console (Figura 3.14); novas execuções apagam o valor do console e o valor novo será exibido em seu lugar. Saídas muito grandes fazem com que uma barra de rolagem seja apresentada e o conteúdo final é sempre priorizado a ser exibido (caso haja barra de rolagem, esta ficará sempre para baixo). Em caso de incorreção, é exibida a mensagem de falha, com o conteúdo desta mensagem e a linha na qual ela foi detectada, a depender do compilador/interpretador utilizado (Figura 3.15).

Na hipótese de ser enviada a saída de uma variável para impressão no Threelab, seu conteúdo será exibido no console. Caso seja enviada uma entrada também, ela será processada e a saída será exibida de acordo com ela (Figura 3.16).

Caso haja erro de compilação, o erro é exibido onde há a porcentagem de testes, que são feitos pelo professor para que seja possível o estudante identificar mudanças necessárias para que a atividade seja concluída. Os erros de compilação são exibidos no campo de saída. mas caso o erro seja em algum teste, há um hiperlink em azul para que seja possível ver com mais detalhes o que era esperado e o que foi obtido.

No console, caso haja alguma inexatidão, a mensagem pode ser colocada em inglês no console (a depender do interpretador/compilador), além da linha na qual houve detecção. No caso acima, o usuário usou ‘>>’ em vez de usar ‘<<’, o que resultou em erro de compilação.

Os testes também são apresentados, caso o professor deseje, de forma a facilitar a resolução da atividade. É possível fazer qualquer quantidade de testes, e a porcentagem de sucessos é exibida na parte inferior direita da tela, embaixo do campo de saída (Figura



Figura 3.12: Exibição de página sem editor

3.17). Além disso, é possível visualizar em que caso de teste ocorreu a inexatidão, a saída esperada e a obtida (Figura 3.18, que exibe um caso de teste incorreto para o programa da figura anterior).

Para criar um caso de teste, o professor deve criar uma div, após a tag ‘`<textarea>`’, que deve ter ‘testcases’ como classe. Dessa forma, os testes dentro da div serão identificados como testes da questão em que eles estão. Assim, cada teste deve ser dividido da forma a seguir: casos de entrada separados por espaço, seguidos de um espaço, seguido por três fecha colchetes ‘]’ seguidos (caso não precise de entrada, apenas os três colchetes devem ser utilizados), e após mais um espaço, a saída esperada. Caso haja mais de um caso de teste, cada linha de teste deve ser seguida por uma linha contendo quatro ‘=’, e na linha após esta, deve-se colocar o teste seguinte (o último teste não precisa ser seguido por uma linha deste tipo). Dessa forma, temos que, como uma nova melhoria, os testes são automatizados. Na Figura 3.19 há um exemplo de casos de teste feitos para uma questão. Entretanto, os testes podem ser vistos pelos alunos, pois ficam no código-fonte da página HTML, sendo assim mais adequados para exercícios de fixação. Em caso de avaliação, o professor deve criar outros testes além dos já inseridos.

### 3.2.3 Dashboard

O dashboard, ou seja, a página de acesso às apostilas e às notas dos estudantes, é feito em páginas na linguagem PHP e com o uso de banco de dados MYSQL, devido à praticidade e facilidade de integração de ambos. As páginas são divididas em: Página de apostilas, na qual é exibida uma lista de todas as apostilas disponíveis; Página de notas, na qual são apresentadas as notas de todos os estudantes de uma determinada apostila;

The screenshot shows a Threelab interface. On the left, there's a sidebar with the title "Página com o triple-page". Below it, instructions say "Use o layout triple-page", "Cada subseção (h2 em HTML, ## em Markdown) inicia uma página", and "Se dentro da subseção houver um editor de código, ele aparece na coluna do meio, e o console aparece na coluna da esquerda". It also says "Execute o código para ver em ação". In the center, there's a code editor with the following JavaScript code:

```

1 function meuMap(funcao, lista) {
2 let resultado = [];
3 for (let i = 0; i < lista.length; i++) {
4 resultado.push(funcao(lista[i]));
5 }
6 return resultado;
7 }
8
9 console.log(meuMap(x => x * 2, [1, 2, 3]));
10

```

At the bottom of the code editor, there are buttons for "Rodar" (Run), "Retroceder" (Back), "Avançar" (Forward), and "Expandir editor" (Expand editor). To the right of the code editor is a vertical panel labeled "Entrada" at the top. Inside this panel, the output of the console.log statement is shown: "2,4,6".

Figura 3.13: Saída do valor de uma variável no Threelab

e a Página de atividade, cujo conteúdo é a resposta de um discente para uma questão de uma determinada apostila. Veremos a seguir o conteúdo de cada uma.

## Página de apostilas

Na página de apostilas há uma lista de todas as apostilas disponíveis, cada uma com um hiperlink para a sua página de notas específica. Cada apostila possui um nome, que é sua variável chave para busca no banco de dados, e é possível identificar a apostila através de uma string de até 20 caracteres que deve ser atribuída à variável global apostila (em JavaScript) nas páginas de listagem, de notas e de uma questão, as quais veremos a seguir. Ao clicar no nome de uma das apostilas, ordenadas em ordem alfabética, o usuário é redirecionado para sua página. Apostilas devem ter nomes diferentes. Na Figura 3.20, é possível ver um exemplo de página de apostilas.

## Página de notas

Na página de notas há uma tabela com todos os dados da apostila em questão; nesta, são ordenados por nome os estudantes que submeteram questões para a apostila e apresentados número de matrícula, nome completo e, para cada questão, a quantidade de testes nos quais o código do estudante passou (e nesta, um link para a página de resposta correspondente) para determinada questão e a porcentagem indicando a nota, sendo que esta pode estar expressa em tonalidades diferentes: vermelho, caso o estudante tenha uma nota inferior à média (por padrão, 5 pontos, pois esta é a média de nota da Universidade Federal da Bahia); azul, caso a nota corresponda a um valor maior ou igual a 5 e menor que 7 pontos, indicando situação de atenção; e verde, caso a nota seja igual a

**Página com o triple-page**

Use o layout `triple-page`

Cada subseção ( `h2` em HTML, `##` em Markdown) inicia uma página

Se dentro da subseção houver um editor de código, ele aparece na coluna do meio, e o console aparece na coluna da esquerda.

Execute o código para ver em ação

```

1 function meuMap(funcao, lista) {
2 let resultado = [];
3 for (let i = 0; i < lista.length; i++) {
4 resultado.push(funcao(lista[i]));
5 }
6 return resultado;
7 }
8
9 teste([2, 4, 6], meuMap(x => x * 2, [1, 2, 3]));
10

```

Rodar

Entrada

OK!

Retroceder Avançar Expandir editor

Figura 3.14: Impressão de saída para teste correto

**Bem-vindo ao curso!**

Joãozinho recebeu a tarefa de criar um programa para imprimir a mensagem "Bem-vindo ao curso!" na caixa de Saída . No entanto, ele cometeu alguns erros de programação. Corrija o programa.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6 COUT << "Bem-vindo ao curso!" >> endl;
7 return 0;
8 }

```

Rodar

Entrada

!!!!!
6:3 Variable COUT does not exist

Erro de compilação (ver saída)

Retroceder Avançar Expandir editor

Figura 3.15: Mensagem de erro no console do Threelab

**Calculadora de soma**

O programa a seguir deveria ler dois números inteiros e imprimir a sua soma, mas ele não funciona. Teste o programa digitando valores na caixa de Entrada. Corrija o programa para exibir o valor correto na caixa de Saída.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6 int a, b, resultado;
7
8 cin >> a >> b;
9
10 resultado = a + b;
11
12 cout << resultado << endl;
13
14 return 0;
15 }

```

Rodar

Entrada

Testes: 100%

Retroceder | Avançar | Expandir editor

Figura 3.16: Saída de valor com entrada no Threelab

**Calculadora de soma**

O programa a seguir deveria ler dois números inteiros e imprimir a sua soma, mas ele não funciona. Teste o programa digitando valores na caixa de Entrada. Corrija o programa para exibir o valor correto na caixa de Saída.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6 int a, b, resultado;
7
8 resultado = a * b;
9
10 cout << resultado << endl;
11 cin >> a >> b;
12
13 return 0;
14 }

```

Rodar

Entrada

Testes: 50%

Retroceder | Avançar | Expandir editor

Figura 3.17: Quantidade de testes passados exibida abaixo do campo de saída

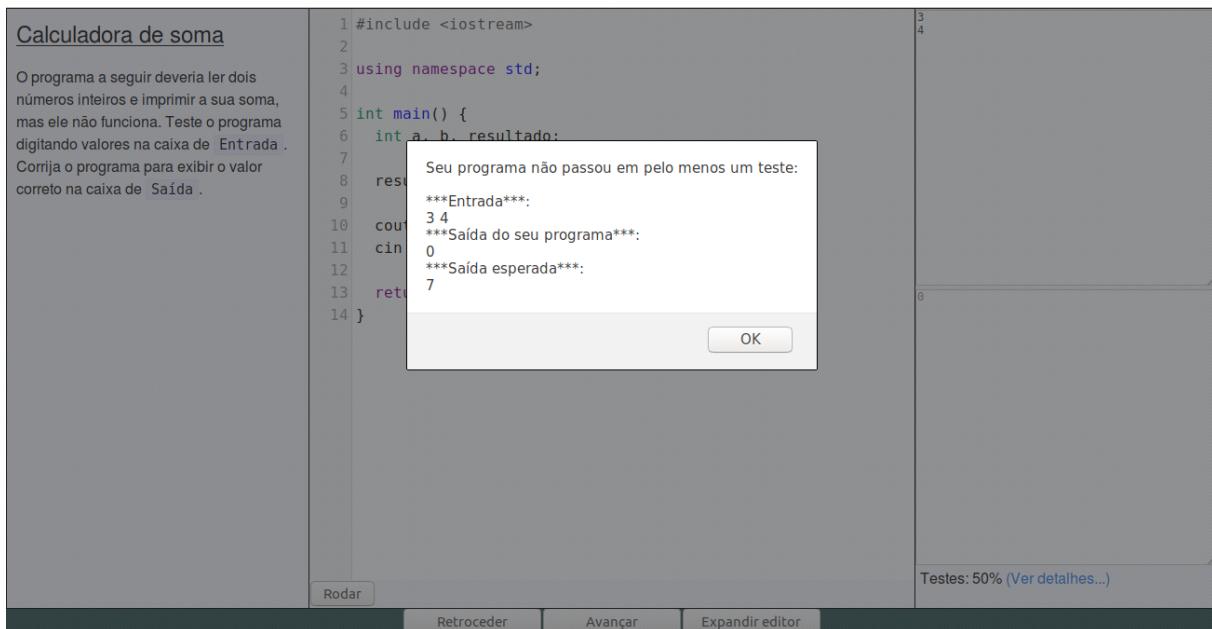


Figura 3.18: Caso de teste no qual não houve sucesso, com saída esperada e saída obtida

O programa a seguir deveria ler dois números inteiros e imprimir a sua soma, mas ele não funciona. Teste o programa digitando valores na caixa de 'Entrada'. Corrija o programa para exibir o valor correto na caixa de 'Saída'.

```

<textarea class="code lang-cpp">
#include <iostream>

using namespace std;

int main() {
 int a, b, resultado;
 resultado = a * b;
 cout << resultado << endl;
 cin >> a >> b;
 return 0;
}</textarea>

0 0]]] 0
=====
-1 1]]] 0
=====
3 4]]] 7
=====
-3 -4]]] -7
</div>
Fim da primeira lista
Você completou a primeira lista. Anote o horário atual. Calcule quantos minutos você demorou para resolver esta lista de exercícios e preencha no formulário do experimento.

<div>
<textarea></textarea>
</div>
Abra agora a [próxima lista de exercícios](ex2).


```

Figura 3.19: Estrutura da div dos casos de teste

# Respostas

atividade  
computacao  
lab  
programacao  
unica

Figura 3.20: Página de apostilas no Threelab

7 ou superior. Na Figura 3.21 é demonstrada uma situação de notas para uma apostila, sendo esta em uma tabela.

## Página de questão

Na página de questão, é apresentada a resposta do estudante para uma questão específica da página de notas. Para acesso, são enviados número da questão e matrícula do estudante, além do nome da apostila. Na Figura 3.22 é denotado um exemplo de exibição da página de uma questão.

## 3.3 Comparativos

Em resumo, temos a Tabela 3.1 com comparativos entre as plataformas e ferramentas. Como podemos ver, a maioria dos softwares estudados suportam linguagens variadas, exceto o Programming Education Assistant; a correção ainda é binária em quase todos eles, ou seja, as notas são 0 ou 10 (além de não haver possibilidade de acesso a casos de teste); apenas o SPOJ e o Threelab não possuem atividades de diversos tipos, sendo apenas de código e compilação ou interpretação; quatro deles não têm integração com páginas já existentes; três deles possuem funcionalidades e serviços ainda pagos e quase todos possuem um editor online, exceto o Autolab. O Threelab, por sua vez, apesar de possuir as mesmas vantagens do Autolab, exceto por não ter variedade de exercícios, possui um diferencial, a integração total das páginas e o fato de não sobrecarregar o servidor,

## Atividades

| Matricula  | Nome     | Atividade 1 | Atividade 2 | Atividade 3  | Atividade 4 |
|------------|----------|-------------|-------------|--------------|-------------|
| 0987654321 | Beltrano | 8 (80.00%)  | 1 (16.67%)  | 15 (100.00%) | 4 (40.00%)  |
| 1238634712 | Fulana   | 0 (0.00%)   | 0 (0.00%)   | 10 (66.67%)  | 7 (70.00%)  |
| 1234567890 | Fulano   | 3 (30.00%)  | 3 (50.00%)  | 0 (0.00%)    | 0 (0.00%)   |
| 1233214567 | Sicrana  | 5 (50.00%)  | 6 (100.00%) | 0 (0.00%)    | 0 (0.00%)   |
| 0988907654 | Sicrano  | 9 (90.00%)  | 0 (0.00%)   | 3 (20.00%)   | 0 (0.00%)   |

Figura 3.21: Página de notas no Threelab

```
; se quiser, pode definir funções auxiliares
(define (area medidas)
 'implementeAFuncao)

(teste 12 (area '(3 4)))
; Crie no mínimo 2 testes adicionais
;(teste 200 (area '(2 100)))
;(teste 200 (area '(25 8)))
;(teste 200 (area '(50 4)))
```

Figura 3.22: Página de uma questão do Threelab

| Nome       | Ling   | ANP | CE  | POE | IP   | G       | EO  |
|------------|--------|-----|-----|-----|------|---------|-----|
| Codecademy | Várias | Não | Não | Sim | Não  | Parcial | Sim |
| SPOJ       | Várias | Não | Não | Não | Não  | Total   | Sim |
| Autolab    | Várias | Sim | Sim | Sim | Não  | Parcial | Não |
| Stepik     | Várias | Não | Sim | Sim | Não  | Parcial | Sim |
| PEA        | Python | Não | Sim | Sim | Sim* | Total   | Sim |
| Threelab   | Várias | Sim | Sim | Não | Sim  | Total   | Sim |

\* Somente a páginas do Jupyter Notebook

**Legenda:** **Ling** - Linguagens; **ANP** - Admite Nota Parcial; **CE** - Permite criação de exercícios; **POE** - Possui outros tipos de exercício (por exemplo, quizzes); **IP** - Pode ser incorporado a páginas web; **G** - Indica se é gratuito; **EO** - Possui editor online; **SPOJ** - Sphere Online Judge; **PEA** - Programming Education Assistant

Tabela 3.1: Comparaçāo entre as plataformas e as ferramentas

já que a execuçāo do código se dá no cliente.

A seguir, será discutida a avaliação da plataforma, por meio da aplicação em um componente curricular da Universidade Federal da Bahia.

# Capítulo 4

## Avaliação

Um conjunto de atividades foi elaborado com o objetivo de analisar a efetividade do uso do layout dividido em três colunas do Threelab para exercícios de programação. Desta forma, foram elaboradas 4 questões, divididas em duas atividades: uma delas está com layout normal de página ('page') e a outra com layout de três colunas ('triple-page'). Cada uma das atividades possui duas questões, e foram criados dois grupos: no primeiro grupo, a primeira atividade possui layout 'triple-page', e a segunda possui layout 'page'; já no segundo grupo, os layouts são trocados: primeira atividade com layout 'page', e segunda com layout 'triple-page'. As atividades foram feitas na linguagem C++.

### 4.1 Experimento piloto

Foi realizado um experimento piloto, no qual um estudante de Mestrado na Universidade Federal da Bahia realizou as atividades de ambos os grupos, analisando a clareza das instruções e dos exercícios. Para realizar a atividade 1 do primeiro grupo, foi necessário um tempo de 2 minutos, mas na segunda atividade, o tempo de completude foi de 5 minutos. Ao realizar as atividades do segundo grupo, o tempo passou a ser de 2 minutos no total, pois ele já sabia as respostas. As correções das atividades ocorreram sem nenhum erro, e não houve melhorias recomendadas pelo experimentador. Segundo ele, o layout de três colunas seria melhor do que o layout de rolamento de página.

A seguir, veremos então a performance da ferramenta nos testes após o experimento piloto.

### 4.2 Metodologia

O experimento foi realizado no contexto da disciplina de Introdução à Logica de Programação (MATA37), da Universidade Federal da Bahia, com os alunos como parti-

pantes, no semestre suplementar de 2020. Os estudantes foram divididos em dois grupos, de acordo com seu número de matrícula (Figura 4.1). Assim, caso o número de matrícula de um aluno fosse ímpar, ele seria do grupo 1; caso fosse par, ele seria do grupo 2. Foi criado um questionário para que estes pudessem expor sua opinião sobre os layouts com base na execução das atividades. A seguir, veremos o conteúdo de cada uma das atividades.

Disciplinas (UFBA)

## Experimento - Orientações

### Experimento - orientações

O experimento consiste de duas listas de exercícios de programação, cada uma com dois exercícios. As duas listas de exercícios possuem *layouts* diferentes.

Antes de começar o experimento, preencha o [Termo de Consentimento](#).

Livre-se de qualquer distração antes de iniciar o experimento. Deve-se cronometrar o tempo gasto na resolução de cada lista de exercícios para preencher o [questionário](#).

Para iniciar o experimento, acesse a primeira lista de exercícios:

*Se seu número de matrícula é um número ímpar (isto é, termina com 1, 3, 5, 7 ou 9), você é do Grupo 1. [Clique aqui para iniciar](#)*

*Se seu número de matrícula é um número par (isto é, termina com 0, 2, 4, 6 ou 8), você é do Grupo 2. [Clique aqui para iniciar](#)*

Disciplinas (UFBA)

Figura 4.1: Página de instruções do experimento

Na primeira atividade, a primeira questão consistia em o estudante modificar um programa de impressão de mensagem na tela que possuía erros (Figura 4.2). Na segunda questão, havia uma calculadora de soma, com um código criado mas incorreto, e era necessário modificá-lo (Figura 4.3).

### Bem-vindo ao curso!

Joãozinho recebeu a tarefa de criar um programa para imprimir a mensagem "Bem-vindo ao curso!" na caixa de [Saída](#). No entanto, ele cometeu alguns erros de programação. Corrija o programa.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6 COUT << "Bem-vindo ao curso!" >> endl;
7 return 0;
8 }
```

Rodar

|         |       |
|---------|-------|
| Entrada | Saída |
|---------|-------|

Figura 4.2: Página da primeira questão da primeira atividade no layout de rolamento

Na segunda atividade, a primeira questão consistia em um detector de senha, este também não funcionando corretamente, e devendo ser modificado (Figura 4.4). A segunda

### Calculadora de soma

O programa a seguir deveria ler dois números inteiros e imprimir a sua soma, mas ele não funciona. Teste o programa digitando valores na caixa de Entrada. Corrija o programa para exibir o valor correto na caixa de Saída.

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6 int a, b, resultado;
7
8 resultado = a * b;
9
10 cout << resultado << endl;
11 cin >> a >> b;
12
13 return 0;
14 }
```

Rodar

|         |       |
|---------|-------|
| Entrada | Saída |
|---------|-------|

Figura 4.3: Página da segunda questão da primeira atividade no layout de rolamento

questão continha um programa que deveria dar boas vindas para o usuário e, a depender de sua idade, informada na entrada, deveria imprimir na tela se este estava no grupo de risco, ou não imprimir nada caso estivesse (Figura 4.5). Na seção a seguir, veremos quais foram os resultados desta experimentação.

|                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                            |                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <b>Digite a senha</b><br><br>Escreva um programa em que o usuário digita uma senha numérica e o programa informa Senha correta ou Senha incorreta , a depender do que foi digitado. A senha correta é 123 . O programa já está parcialmente escrito. | <pre> 1 #include &lt;iostream&gt; 2 3 using namespace std; 4 5 int main() { 6     int senha; 7 8     cin &gt;&gt; senha; 9 10    cout &lt;&lt; "Senha incorreta" &lt;&lt; endl; 11 12    return 0; 13 }</pre> <p>Rodar</p> | Entrada<br><br>Saida |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|

Retroceder | Avançar | Expandir editor

Figura 4.4: Página da primeira questão da segunda atividade no layout triplo

The screenshot shows a triple-column layout interface. The left column contains a text area with the title "Faixa de risco" and a descriptive paragraph about a program that asks for age and prints a welcome message if the age is between 60 and 80. The middle column is a code editor showing the following C++ code:

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6 int idade;
7
8 cin >> idade;
9
10 // Se idade estiver entre 60 e 80 anos,
11 // imprimir Faixa de risco
12
13 cout << "Bem-vindo(a)!" << endl;
14
15 return 0;
16 }

```

The right column has two sections: "Entrada" (Input) and "Saída" (Output), both currently empty. At the bottom, there are buttons for "Rodar" (Run), "Retroceder" (Back), "Avançar" (Forward), and "Expandir editor" (Expand editor).

Figura 4.5: Página da segunda questão da segunda atividade no layout triplo

## 4.3 Resultados

Dos 400 estudantes de MATA37 do semestre suplementar de 2020, apenas 13 responderam o questionário sobre os layouts. Dos 13, 4 fizeram a atividade do grupo 1, e 9 deles, do grupo 2 (Figura 4.6).

Em média, o tempo de completude da atividade 1 foi de aproximadamente 3.65 minutos. Já o tempo médio de completude da atividade 2, foi de aproximadamente 6.08 minutos. Da atividade 1, a moda foi de 4 minutos (4 estudantes demoraram este tempo para realizá-la), assim como a mediana, e houve um outlier (observação muito distanciada das demais [9]), já que um estudante demorou 10 minutos para completar. Com relação à atividade 2, a moda foi de 5 e 3 minutos (para 3 estudantes esta foi a duração da completude para ambos), a mediana foi 5, e houve dois outliers, já que para um estudante a etapa durou 12 minutos, e para outro, 20. As Figuras 4.7 e 4.8 demonstram um gráfico de barras contendo o tempo de completude das atividades 1 e 2, respectivamente, pelos estudantes, e a porcentagem.

Dividindo em grupos, temos que para o grupo 1, houve aproximadamente 3.33 de média, 3.5 de mediana e não houve moda para a atividade 1, com um outlier de 15; já na atividade 2, houve 8 de média, 5 de mediana, não houve moda, mas um outlier (20).

No grupo 2, temos como média, mediana e moda (4 alunos) 4 minutos na atividade 1, com um outlier (10 minutos), e na atividade 2, aproximadamente 5.2 de média, 5 ou 3 de moda (3 alunos cada) e 5 de mediana, com um outlier (12 minutos).

Com relação à preferência dos estudantes, 3 deles acharam o layout de três colunas

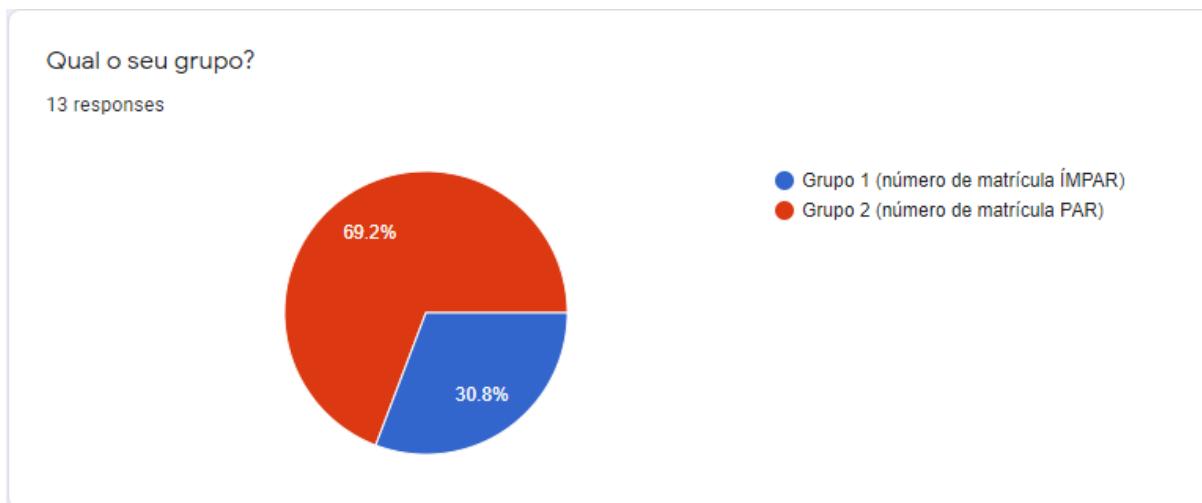


Figura 4.6: Gráfico de setores representando em porcentagem quantos estudantes participaram do grupo 1 e quantos participaram do grupo 2

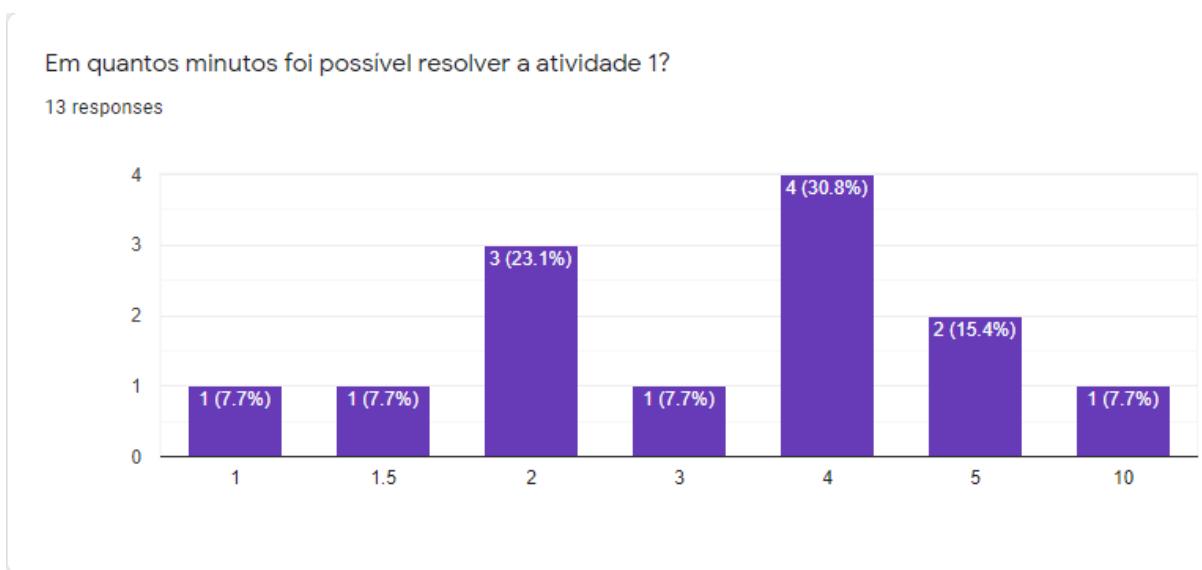


Figura 4.7: Gráfico de barras no qual é apresentado o tempo de finalização da atividade 1

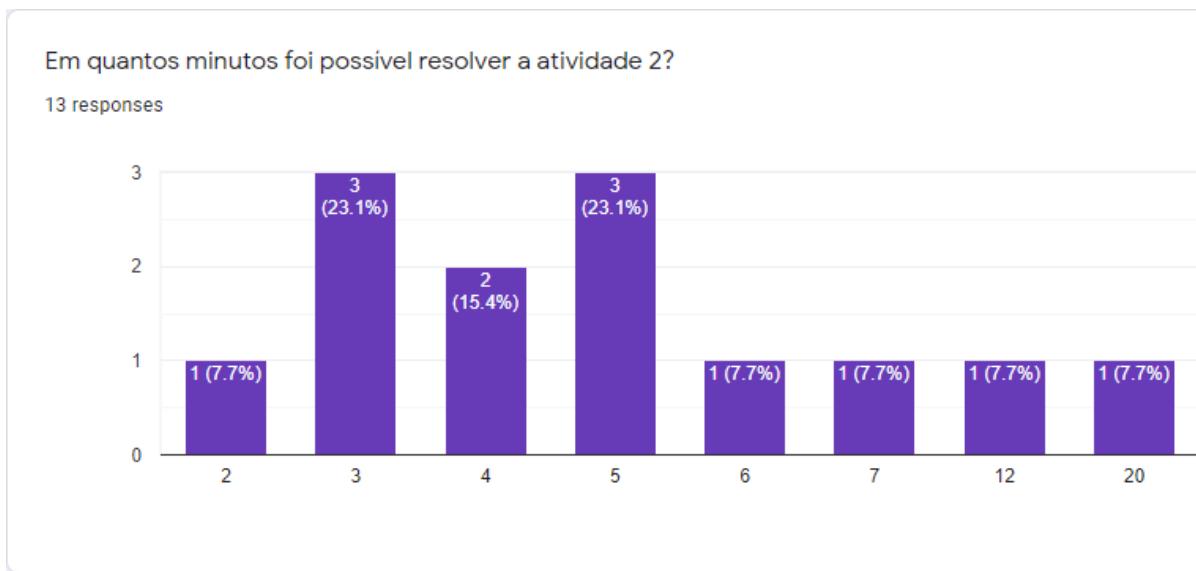


Figura 4.8: Gráfico de barras no qual é apresentado o tempo de finalização da atividade 2

um pouco melhor, 3 acharam o layout de três colunas muito melhor, 3 foram indiferentes, 1 achou o layout de rolamento um pouco melhor e 3 acharam o layout de rolamento muito melhor, ou seja, tirando os 7.7% (1 aluno) que achou o layout de rolamento um pouco melhor, 23.1% dos estudantes escolheram cada uma das outras alternativas (Figura 4.9). Com relação à preferência dos alunos para com os layouts no geral, aproximadamente 46.2% (6) deles preferiram o layout de três colunas, enquanto que aproximadamente 30.8% preferiu o layout de rolamento.

No grupo 1, 50% dos estudantes foi indiferente quanto ao layout e 50% preferiu a página de rolamento (muito melhor). No grupo 2, aproximadamente 44% dos estudantes preferiu ao layout de três colunas (muito melhor), assim como outros 44% achou um pouco melhor, e o restante foi indiferente quanto ao layout.

Com relação ao tempo de finalização, 9 dos 13 participantes não consideraram como influência o layout, ou seja, 69.2%, segundo a Figura 4.10. Os motivos da preferência, aspectos positivos e negativos serão discutidos na próxima seção. Não houve muitos comentários adicionais, e os que houveram foram um referente ao motivo de tempo excessivo para realização, e outro, um elogio a ambos os layouts.

No grupo 1, 75% dos estudantes não considerou o layout como fator de influência para o tempo de finalização da atividade. No grupo 2, esse número foi de 66.7%.

Com relação aos alunos que preferiram o layout de três colunas (tanto os que acharam o layout de três colunas um pouco melhor, quanto muito melhor), todos foram do grupo 2. Os principais motivos que os levaram a esta decisão foram: o foco em uma única questão, evitando desconcentrações, visualização, clareza e objetividade, pois há um

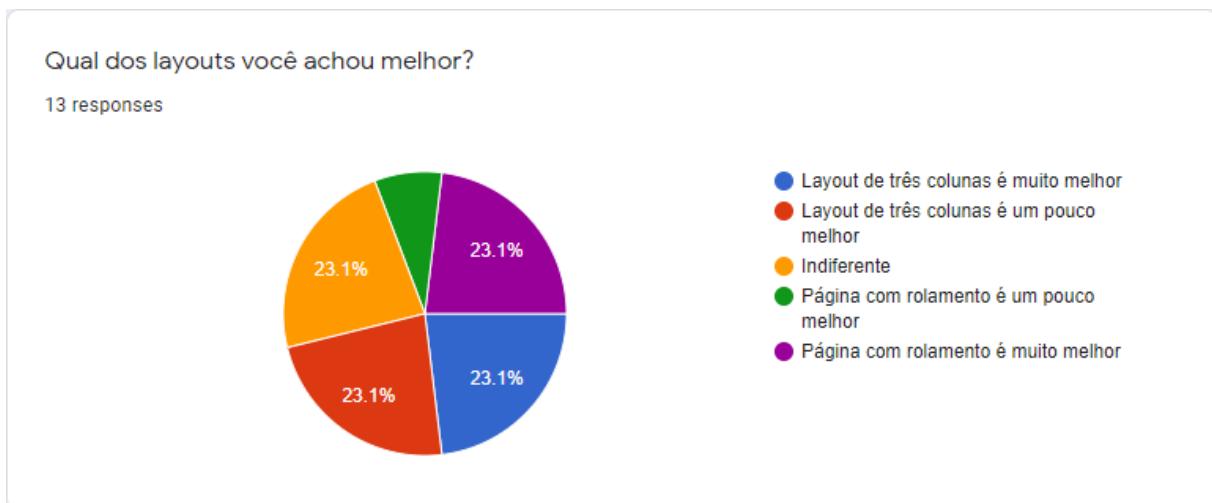


Figura 4.9: Gráfico de setores representando as porcentagens das preferências dos estudantes



Figura 4.10: Gráfico de setores representando a porcentagem de estudantes com relação à influência do layout no tempo de execução

problema que deve ser resolvido e este é o foco de quem está tentando resolvê-lo. Já os alunos que preferiram o layout de rolamento, excluindo apenas um, todos foram do grupo 1, e estes o preferiram por ser mais fácil o reuso de código, e por não haver necessidade de pular páginas.

Os pontos positivos mais destacados para o layout de três colunas foram a divisão dos problemas, deixando-os mais independentes, a visualização, por ser possível ver todos os aspectos do problema (entrada, saída, código e descrição) e, principalmente, a visualização mais clara do código. Para os pontos negativos, em suma, foram apresentados posição do botão de execução do código, visibilidade de entrada e saída (para que fosse mais claro identificar), cor do layout e o fato de entrada e saída não serem integrados.

## 4.4 Discussão

As opiniões dos participantes foram semelhantes com relação aos grupos nos quais eles estavam. Por parte das justificativas de escolha de layout, houve tanto positivas quanto negativas e neutras. A seguir, vamos analisar as opiniões a partir de cada seção apresentada anteriormente.

Com relação à amostragem, apesar de os grupos não serem muito equilibrados em número de participantes dos grupos, as opiniões são bastante equilibradas, visto que, mesmo que a maior porcentagem de alunos tenha preferência pelo layout de três colunas, mais da metade deles ainda é indiferente ou prefere o layout de rolamento. Em compensação, 6 dos 9 alunos que fizeram as atividades do grupo 2, e apenas alunos que fizeram atividades do grupo 2, preferiram o layout de três colunas, o que pode ter sido uma influência da ordem das atividades, já que neste caso (grupo 2) os estudantes fizeram as atividades no layout de três páginas por último, e apenas um deles preferiu a página de rolamento.

O tempo médio de realização das atividades foi próximo em ambos os grupos, mesmo com a presença de outliers, que não ocorreram em quantidade muito grande, já que de 13 alunos, apenas 3 criaram outliers no tempo de resposta dos exercícios. Com isto, de fato o layout não foi um fator de influência relevante para o tempo de término da atividade.

Os alunos que tiveram preferência pelo layout de rolamento o fizeram devido à preferência em fazer as atividades numa página só e para reuso de código. Os que favoreceram o layout de três colunas o fizeram por preferências muito importantes em realização de atividades: concentração, organização das atividades, clareza e objetividade, o que permite que os objetivos sejam alcançados com mais facilidade e o entendimento destes, mais simples.

Assim como os motivos de favorecimento do layout de três colunas, os pontos positivos também foram mais técnicos, sendo que apenas um foi indiferente com relação aos pontos positivos (e também aos negativos). A organização, o aproveitamento do espaço e o foco foram os principais motivos. Um dos alunos também favoreceu o sistema de correção, mas ambos os layouts o possuem.

Quanto aos pontos negativos, foram fatores apenas de visualização, ou seja, a plataforma não apresentou problemas técnicos. A maioria dos alunos consideraram o layout como confuso, devido à dificuldade de identificar os campos de entrada e saída, assim como o botão de executar o código. Houve também uma questão técnica, o reuso de código, mas este será integrado automaticamente com o salvamento de respostas, já que navegar entre as páginas não alteraria o conteúdo do código. Os pontos negativos podem ser modificados em trabalhos futuros, com soluções que serão apresentadas no capítulo a

seguir.

# Capítulo 5

## Conclusão

O Threelab é uma ferramenta de ensino e correção de código que consiste na divisão de um layout de atividade em três partes: explicação, código e entrada/saída. Dessa forma, com a presença de testes de código, os discentes podem conseguir identificar possíveis incorreções nos resultados de sua solução, assim como a completude da mesma. Além disso, os testes ajudam também o docente, já que pouparam tempo de correção manual das atividades, pois esta ocorre de forma automática.

A simplicidade de criação de páginas também é um fator importante, pois o professor passa menos do seu tempo elaborando as questões. Como o editor é opcional, ainda é possível utilizar a plataforma para passagem de conteúdo, tornando ainda mais fácil a integração da mesma com as aulas.

Como houve aprovação dos alunos, que não tiveram problemas de questão técnica com a plataforma, assim como as possíveis melhorias recomendadas foram em maioria visuais, o resultado do experimento foi bastante positivo. Mesmo que tenha sido uma amostragem pequena de discentes, os pontos positivos apresentados, por serem essenciais para facilitar o aprendizado e tornar a experiência de programar mais agradável e simples para quem não possui muita experiência na área tornam a plataforma ainda mais suscetível à implantação.

Os estudantes tiveram opiniões divididas sobre os layouts; logo, ficaria a cargo do professor decidir que layout seria mais efetivo para suas aulas, mas devido à possível presença de casos de teste e a possibilidade de visualizá-los, é extremamente recomendada a utilização do layout de três partes. A possibilidade de visualizar os casos de teste e alterar os programas a partir destes é um diferencial importante que seria atrativo também para os discentes, já que isto facilita a identificação do problema, bem como permite a garantia da completude das soluções.

Como trabalhos futuros, temos algumas implementações que podemos apresentar. Primeiro, a adição de um modo noturno, já que estudantes podem ter tempo livre para

atividades apenas à noite. Dessa forma, o aluno não cansaria facilmente de olhar para a tela, o que permitiria um desempenho e uma produtividade maiores.

A detecção de plágio, assim como no Autolab, é uma funcionalidade importante, para verificar possíveis programas repetidos (a depender da dificuldade do problema, já que alguns são simples o suficiente para que as soluções não sejam muito distintas).

A possibilidade de edição do texto de explicação na própria página também é uma funcionalidade interessante, já que tornaria mais fáceis alterações simples, ou até mesmo mais complexas, sem a necessidade de procurar o arquivo contendo o código-fonte da atividade.

A possibilidade de se obter notas parciais ou totais deve ser também adicionada para a plataforma, para que haja ainda mais liberdade para os professores. Dessa forma, além da possibilidade de criar testes ou não, seria possível que a nota fosse aplicada à quantidade de testes passados ou à totalidade, ou seja, caso todos eles sejam passados (correção binária).

Outra funcionalidade importante a ser adicionada seria a implementação de dicas, já que mesmo que o aluno saiba o caso de teste no qual seu código não obtém sucesso, este pode não conseguir identificar facilmente que bloco de código deve ser modificado, ou quando há erros simples, comuns e pouco perceptíveis, como a presença de um ‘;’ no fim de uma linha de código. Entretanto, estas seriam opcionais, e ficaria a cargo do professor decidir se haveria dicas nos exercícios ou não.

A implementação de editores de código com múltiplos arquivos, exibidos em abas, assim como no Codecademy seria uma funcionalidade muito importante também, visto que esta permitiria a facilitação de matérias mais avançadas, que utilizam de vários arquivos em conjunto para execução, assim como projetos um pouco maiores. Seguindo, temos também a integração com a biblioteca Three.js [4] de JavaScript, que é comumente usada para exercícios de Computação Gráfica. Assim, o uso da ferramenta na disciplina se tornaria totalmente possível, facilitando o aprendizado desta.

Assim como a integração com Computação Gráfica seria possível com trabalhos futuros, a integração com Teoria dos Grafos, Linguagens Formais e Autômatos e Teoria da Computação também seria com a criação de grafos e autômatos, que seriam visualizados e editados por código também por esta ferramenta, neste layout.

É importante destacar também como trabalho futuro a avaliação da plataforma com o professor, de forma a verificar possíveis melhorias nas páginas de questão, notas e apostilas, bem como na construção das atividades e testes. Assim, poderemos garantir que os objetivos desejadas na construção da mesma sejam cumpridos e fazer as alterações necessárias para isso.

Por último, um dos trabalhos mais importantes é o aumento do número de lingua-

gens de programação suportadas, já que, assim como as implementadas (JavaScript, C++, Scheme) há também outras muito utilizadas, como Python, C e Java. Dessa maneira, mais componentes curriculares estarão utilizando a ferramenta e tanto o corpo docente quanto o discente terão um ambiente mais focado no aprendizado de programação.

# Referências Bibliográficas

- [1] alexshinn. R7rs small-archive. <https://small.r7rs.org/>. Acesso: 28/10/2020.
- [2] Amazon. Amazon ec2. <https://aws.amazon.com/pt/ec2/>. Acesso: 21/05/2020.
- [3] Spoj Brasil. Spoj brasil. <https://br.spoj.com/embed/info>. Acesso: 01/05/2020.
- [4] Ricardo Cabello. three.js - javascript 3d library. <https://threejs.org/>. Acesso: 18/10/2020.
- [5] Codecademy. Codecademy. <https://www.codecademy.com/>. Acesso: 23/04/2020.
- [6] Codecademy. Codecademy. <https://www.codecademy.com/about>. Acesso: 24/04/2020.
- [7] CodeMirror. Codemirror. <https://codemirror.net/>. Acesso: 01/06/2020.
- [8] CodeMirror. Codemirror: Language modes. <https://codemirror.net/mode/index.html>. Acesso: 01/06/2020.
- [9] Paulo Roberto Medeiros de Azevedo. *Introdução à Estatística - 3<sup>a</sup> Edição*. edufrn, edufrn, Natal, 2016.
- [10] Sociedade Brasileira de Computação. Maratona de programação. <http://maratona.ime.usp.br/>. Acesso: 01/05/2020.
- [11] Sociedade Brasileira de Computação. Olimpíada brasileira de informática. <http://olimpiada.ic.unicamp.br/>. Acesso: 01/05/2020.
- [12] Inc. Docker. Docker. <https://www.docker.com/>. Acesso: 21/05/2020.
- [13] Dr. Rodrigo Rocha Gomes e Souza. Procedimento prova. [https://github.com/rodrigorgs/aulas/blob/master/\\_PROCEDIMENTO\\_PROVA.md](https://github.com/rodrigorgs/aulas/blob/master/_PROCEDIMENTO_PROVA.md). Acesso: 12/11/2020.
- [14] Rodrigo Rocha Gomes e Souza. Aulas na universidade federal da bahia. <https://github.com/rodrigorgs/aulas>. Acesso: 08/06/2020.

- [15] Inc. Free Software Foundation. Tar - gnu project - free software foundation. <https://www.gnu.org/software/tar/>. Acesso: 24/10/2020.
- [16] Google. Bazel - a fast, scalable, multi-language and extensible build system. <https://bazel.build/>. Acesso: 26/05/2020.
- [17] Google. The go programming language. <https://golang.org/>. Acesso: 26/05/2020.
- [18] Google. Google terms of service - privacy & terms - google. <https://policies.google.com/terms?hl=en>. Acesso: 26/05/2020.
- [19] Google. Sdk do cloud: Interface de linha de comando. <https://cloud.google.com/sdk>. Acesso: 26/05/2020.
- [20] Google. Welome to colaboratory - colaboratory. <https://colab.research.google.com/notebooks/intro.ipynb>. Acesso: 26/05/2020.
- [21] Felix Hao. Jscpp: A simple c++ interpreter written in javascript. <https://github.com/felixhao28/JSCPP>. Acesso: 28/10/2020.
- [22] Yutaka HARA. Biwascheme. <https://www.biwascheme.org/>. Acesso: 28/10/2020.
- [23] Github Inc. Github. <https://github.com/>. Acesso: 28/05/2020.
- [24] Github inc. Github pages. <https://pages.github.com/>. Acesso: 12/11/2020.
- [25] Project Jupyter. Project jupyter - home. <https://jupyter.org/>. Acesso: 26/05/2020.
- [26] Sphere Research Labs. Ideone. <https://ideone.com/>. Acesso: 01/05/2020.
- [27] MIT. Mit license. <https://mit-license.org/>. Acesso: 28/10/2020.
- [28] Mozilla. eval() - javascript — mdn. [https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/eval](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/eval). Acesso: 28/10/2020.
- [29] Google (não oficialmente). Programming education assistant. <https://github.com/google/prog-edu-assistant>. Acesso: 26/05/2020.
- [30] Opensource.org. Apache licence version 2.0. <https://opensource.org/licenses/Apache-2.0>. Acesso: 21/05/2020.
- [31] José Carlos Rocha Pereira Júnior and Clevi Elena Rapkiewicz. O processo de ensino-aprendizagem de fundamentos de programação: uma visão crítica da pesquisa no brasil. In *Anais do XII Workshop sobre Educação em Computação (SBC)*, volume 3, page 4, 2004.

- [32] Tom Preston-Werner. Jekyll - simple, blog-aware, static sites. <https://jekyllrb.com/>. Acesso: 28/05/2020.
- [33] Dominic John Repici. Csv comma separated value file format. <http://creativyst.com/Doc/Articles/CSV/CSV01.htm>. Acesso: 24/10/2020.
- [34] Spoj. Spoj. <https://www.spoj.com/>. Acesso: 01/05/2020.
- [35] Stepik.org. About stepik. <https://welcome.stepik.org/en/about>. Acesso: 22/05/2020.
- [36] Stepik.org. Stepik. <https://stepik.org/>. Acesso: 22/05/2020.
- [37] Stepik.org. Terms of service - stepik. <https://welcome.stepik.org/en/terms>. Acesso: 22/05/2020.
- [38] Carnegie Mellon University. Autolab. <http://www.autolabproject.com/>. Acesso: 21/05/2020.
- [39] Carnegie Mellon University. Autolab - sobre. <https://github.com/autolab/Autolab/blob/master/README.md>. Acesso: 21/05/2020.

# **Apêndice A**

## **Questionário**

# Questionário - Threelab (Experimento 2)

\* Required

1. Email address \*

---

2. Nome

---

3. Qual o seu grupo? \*

*Mark only one oval.*



Grupo 1 (número de matrícula ÍMPAR)



Grupo 2 (número de matrícula PAR)

4. Em quantos minutos foi possível resolver a atividade 1? \*

---

5. Em quantos minutos foi possível resolver a atividade 2? \*

---

## 6. Qual dos layouts você achou melhor? \*

*Mark only one oval.*

- Layout de três colunas é muito melhor
- Layout de três colunas é um pouco melhor
- Indiferente
- Página com rolamento é um pouco melhor
- Página com rolamento é muito melhor

## 7. Justifique sua resposta anterior \*

---

---

---

---

## 8. Para você, o layout influenciou no tempo de finalização da atividade? \*

*Mark only one oval.*

- Sim
- Não

## 9. De quê você gostou no layout de três colunas? \*

---

---

---

---

10. De quê você NÃO gostou no layout de três colunas? \*

---

---

---

---

11. Deseja fazer algum comentário adicional?

---

---

---

---

This content is neither created nor endorsed by Google.

Google Forms