

Codeflex – Aplicação web de Programação Competitiva

Miguel Brito
Instituto Politécnico da Guarda

Celestino Gonçalves
Instituto Politécnico da Guarda

Resumo - O meio académico é sem dúvida uma das principais vias de introdução na área da programação, no entanto, neste meio onde a prática é crucial para o desenvolvimento de bases sólidas, a quantidade de exercícios realizados fica bastante aquém da pretendida, em especial pelo processo trabalhoso que é a avaliação de código. Numa tentativa de colmatar esta falha, foi implementada uma aplicação web que servirá como um repositório de problemas (típicos da área da programação) para prática e aprendizagem, mas que também contará com uma parte competitiva, e essencialmente, terá de ser capaz de avaliar as submissões dos utilizadores no momento. A solução para a avaliação das submissões passa pela utilização de compiladores associados à respetiva linguagem de programação e uma execução paralela para melhorias na eficiência com que os recursos do módulo de avaliação são utilizados.

Abstract—The academic environment is undoubtedly one of the main means of introduction to the field of programming, however, in this environment where practice is crucial for the development of solid foundations, the number of given assignments falls short from ideal, especially because of the time it takes to evaluate. To fill this gap, a web application has been implemented to serve as a repository of problems (typical to programming) for practice and learning but will also contain a competitive section. Above all things, the app should be capable of evaluating user's submissions at the time. The solution for evaluating submissions involves the use of compilers associated with their programming language and a parallel execution to bring improvements in resource usage efficiency.

Keywords—Web Application, web services, Java EE, Spring, Competitive Programming, React

I. MOTIVAÇÃO

Nos dias que correm existe uma grande procura no que diz respeito à área das Tecnologias da Informação, sendo a programação um dos focos do mercado. Nesta área, a prática e o estudo de conceitos como algoritmos e estruturas de dados é um dos fatores nucleares para o crescimento enquanto programador. Sendo o meio académico uma das principais vias onde é feita a introdução à programação e onde são desenvolvidas bases fundamentais, é essencial que haja um aumento na quantidade de exercícios resolvidos por parte dos alunos, que se encontra, no entanto, longe do ideal devido ao trabalho e complexidade envolvido na avaliação dos mesmos [1]. Neste contexto, as aplicações web de programação competitiva têm uma grande importância como um método de preparação, estudo e avaliação, e como tal, existe a necessidade de uma aplicação que possa reunir problemas tipo para facilitar uma maior imersão dos utilizadores com a resolução de problemas recorrendo à programação.

II. OBJETIVOS

Este projeto tem como objetivo a implementação de uma aplicação web que permitirá a prática de programação competitiva. Esta prática consiste na criação de soluções para um determinado problema, proposto por outros utilizadores, recorrendo a código que é posteriormente compilado e executado relativamente a inputs predefinidos. Caso o código submetido pelo utilizador gere os outputs pretendidos considera-se a resposta como válida. Tendo em vista o proposto, a aplicação web deverá cumprir os seguintes objetivos.

- Sistema de autenticação de utilizadores
- Desafios para prática sem limitações de tempo, organizados por categorias.
- Torneios limitados por tempo
- Classificação de utilizadores utilizando o *rating* Elo
- Editor de texto embutido
- Compilação e avaliação das soluções submetidas
- Criação de torneios privados por parte do utilizador final

III. ESTADO DA ARTE

A realização do estudo e de uma análise crítica às plataformas existentes na área é um passo essencial para identificar e procurar colmatar as falhas existentes no modelo de aplicações atual.

A. Aplicações existentes

Este processo é essencial para identificar lacunas em aplicações semelhantes. Como tal, foi realizada uma análise a um conjunto de plataformas escolhidas de acordo com os seguintes parâmetros: Características inovadoras, Semelhança com os objetivos definidos e Conhecimento e experiência anterior de algumas das plataformas na qual resultou a informação constante na Tabela 1 1.

Plataforma	Codeflex	LeetCode [1]	HackerRank [2]	CodeChef [3]	HackerEarth [4]	Codeforces [5]	TopCoder [6]
Características							
Torneios	✓	✓	✓	✓	✓	✓	✓
Prática de Desafios	✓	✓	✓	✓	✓	✓	✓
Simulação de Torneios		✓					
Conquistas			✓	✓			✓
Classificação	✓	✓	✓	✓	✓	✓	✓
Procura de Emprego			✓		✓		✓
Networking		✓	✓	✓	✓	✓	✓

Mock interviews		✓					
Torneios Privados	✓						
Classificação Alexa		2878	3844	7145	7381	7524	27773

Tabela 1 - Análise de outras plataformas

B. Análise Crítica

Das soluções analisadas mais detalhadamente, *LeetCode* e *HackerRank*, podem-se retirar pontos fortes e fracos. Para a *LeetCode* o seu ponto forte passa pela preparação para entrevistas de emprego mas a diversidade de torneios é inferior à *HackerRank*. A segunda solução é uma das aplicações mais completas das analisadas, que contém desde um bom repositório para aprendizagem como boa oferta de torneios. Esta aplicação oferece também a possibilidade de criação de torneios, que no entanto ficam disponíveis para qualquer utilizador. A aplicação que se propõe tem como um dos objetivos completar esta lacuna que ajudará em situações de avaliação de alunos ou realização de *Hackathons*.

IV. METODOLOGIA

Uma metodologia de *software* é um procedimento que tem um papel fundamental na estruturação, planeamento e controlo do processo de desenvolvimento de software. É o seguimento de uma metodologia que permite que o desenvolvimento seja realizado de forma focada para com as exigências do cliente e com um maior grau de organização e qualidade. Para o desenvolvimento deste projeto optou-se por o uso de uma *framework* Scrum [7], uma metodologia Ágil [8], em especial por não haver clareza da totalidade da aplicação na sua fase mais inicial e pela introdução de todo um conjunto de tecnologias nunca utilizadas anteriormente, entre as quais se destacam Java EE, ReactJs e o serviço *cloud* Amazon Web Services.

V. ANÁLISE DE REQUISITOS

Os requisitos de um sistema são um conjunto de informação sobre o que se espera que a aplicação faça e permita fazer. Foi feita uma análise tendo em conta os objetivos inicialmente traçados com suporte num conjunto de diagramas recorrendo à linguagem UML, com a qual foi planeada a implementação de casos de uso. Desta análise resultaram os seguintes requisitos:

- Consultar categorias de problemas práticos
- Consultar lista de problemas de uma categoria
- Submeter solução(código) para um problema
- Consultar classificação global
- Consultar classificação de torneio
- Consultar submissões para determinado problema
- Consultar torneios
- Participar em torneios
- Adicionar, editar, eliminar torneio privado
- Adicionar, editar, eliminar problemas de um torneio
- Adicionar, editar, eliminar casos de teste de um problema
- Gerir categorias
- Gerir problemas públicos
- Gerir casos de teste associados a problemas públicos

VI. ARQUITETURA DO SISTEMA

A arquitetura de um sistema é o modelo conceptual como a estrutura, o comportamento e todas as interações entre os diversos componentes ocorrem [2]. A definição da arquitetura de um sistema deve ser um passo tomado com consciência de todos os elementos envolventes no projeto e em possíveis elementos a serem adicionados futuramente.

No presente projeto optou-se pelo uso de uma arquitetura de *web services*, que é uma arquitetura bastante flexível e modular visto que cria um nível de abstração entre o que se pretende expor (dados, lógica de negócio) e o que se pretende consumir. Utilizando uma arquitetura *web services* é necessário ter em consideração a segurança do mesmo, caso contrário qualquer utilizador pode fazer pedidos e obter informação à qual não deveria ter acesso. Para tal, adicionou-se um sistema de *tokens*, recorrendo ao JWT (JSON Web Tokens) [3] para gerir quem tem acesso ao quê. Este sistema, permite transmitir informação num *token*, acompanhada de uma chave de verificação. Sempre que um pedido é feito ao sistema e o *token* não condiz com os *tokens* gerados, o acesso é negado.

A arquitetura do sistema implementado segue-se ilustrada na Figura 1.

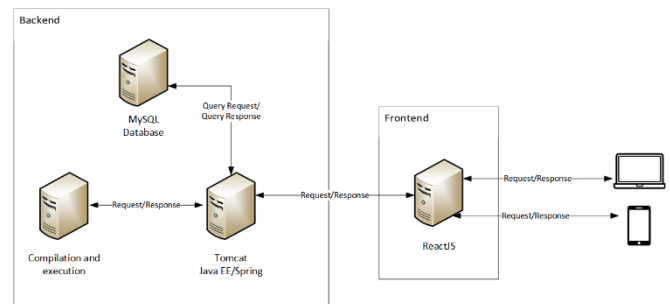


Figura 1- Arquitetura do Sistema

VII. COMPILAÇÃO, EXECUÇÃO E AVALIAÇÃO DE SUBMISSÕES

Um dos objetivos do presente projeto passa por avaliar o código submetido pelos utilizadores para determinado problema. Para que isso seja possível este módulo deverá possuir as seguintes capacidades: compilação de submissões, execução de casos de teste relativos a uma submissão, avaliação parcial das submissões e limitações para cada execução. De uma forma muito simples, o sistema criado, passa pela construção de vários comandos que são executados paralelamente numa máquina Linux relativamente a um *input*, através de SSH(*Secure Shell*). O *output* obtido é depois validado de acordo com o *output* esperado.

A. Segurança

Possibilitar ao utilizador que execute o código que pretender nas máquinas que correm a aplicação é de facto o sonho de qualquer atacante informático. Como tal, foram tomadas um conjunto de medidas com o objetivo de lidar com esta situação.

A mais significativa passou pelo uso de um ambiente *sandbox* para execução das submissões do utilizador. Este tipo de ambientes evita que o código malicioso se propague para a restante máquina.

B. Performance

Tendo em conta que a compilação e execução de programas é um processo que exigirá dos recursos de processamento da máquina, em especial para casos de teste com vários milhares de *inputs*, é importante ter em consideração o tempo de processamento das submissões uma vez que se pretende que haja um *feedback* em tempo real. Para tal, foi feita uma implementação recorrendo a programação multitarefa que permitirá que o servidor continue a expor todos os métodos para o *frontend* enquanto realiza a avaliação das submissões uma vez que os métodos não são assíncronos. A execução dos casos de teste no servidor será feita paralelamente.

Para verificar se há de facto vantagens em fazer uma execução paralela dos casos de teste, foram realizados testes de comparação recorrendo a um CPU i5 8250u, de 4 núcleos de processamento e 3.4GHz de frequência máxima. Os resultados do primeiro teste, o método de execução sequencial são apresentados na Tabela 2.

Nº Submissões	Nº de casos de teste totais	Duração (segundos)
5	50	26
10	100	53
20	200	68
45	450	135

Tabela 2 - Testes de performance – Método de execução sequencial

De seguida o mesmo teste foi realizado para uma execução em paralelo. Os resultados para o mesmo estão reunidos na Tabela 3.

Nº Submissões	Nº de casos de teste totais	Duração (segundos)
5	50	15
10	100	32
20	200	45
45	450	82

Tabela 3 - Testes de performance - Método de execução paralela

O método de execução em paralelo apresenta melhorias de tempo de execução em média de 39% (42, 40, 34 e 49, para valores do número de submissões de 5, 10, 20 e 45 respetivamente), relativamente ao método de execução sequencial para os testes realizados (Figura 2). Sendo esta execução feita num processador com 4 núcleos, esperava-se que o método em paralelo obtivesse tempos de execução perto de 4 vezes inferiores, no entanto, o tempo de envio do código e dos casos de teste para o servidor de compilação é significativo visto que todas as execuções se realizam de forma relativamente rápida. Por outro lado, para execuções em que ocorre o pior caso possível¹ o tempo utilizado a transferir o código e os casos de teste para o servidor será

menos significativo relativamente ao tempo de execução. Nesse caso, as melhorias no tempo de execução serão mais visíveis e aproximar-se-ão do esperado.

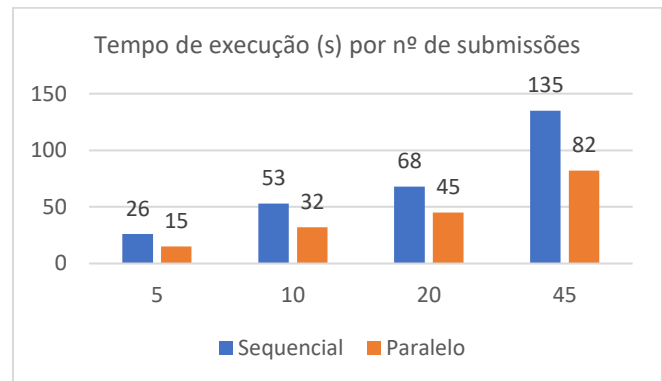


Figura 2 - Gráfico de comparação de tempo de execução

VIII. CONCLUSÕES

A aplicação *web* desenvolvida cumpre com todos os objetivos iniciais estabelecidos, e como tal encontra-se preparada para acolher utilizadores dispostos a aprender ou participar e realizar os seus torneios. Tanto no desenvolvimento do *backend* como do *frontend* foi fulcral recorrer a *frameworks* para que o processo se tornasse mais eficiente.

O desenvolvimento do módulo de avaliação foi sem dúvida das partes mais trabalhosas do projeto uma vez que não existem muitas soluções do género, e como tal, pouca informação sobre o tópico. Optou-se por ir mais além no desenvolvimento deste módulo e procurar melhorar o seu desempenho e torná-lo mais seguro, algo que foi atingido, mas fica uma grande margem para melhoria e aperfeiçoamento. O desenvolvimento deste módulo permitiu obter novos conhecimentos sobre tecnologias nunca antes utilizadas (ex: *sandboxes*, execução de processos em paralelo) e adquirir uma melhor perceção sobre o sistema operativo *Linux*.

O uso de um serviço *cloud* como o da *Amazon* foi também algo de novo para com as tecnologias conhecidas. Fica para recordar que o *deploy* é bastante mais trabalhoso do que aparente, e nesse processo, surgem um conjunto de outros problemas não existentes até à data.

Como trabalho futuro, e de forma a tornar a aplicação mais completa existem um conjunto de funcionalidades que poderão ser consideradas. Começando pelo módulo de avaliação das submissões, implementar o uso de *control groups* nas *sandboxes* utilizadas para garantir que o processamento é consistente para todas as submissões. Para tornar a aplicação mais dinâmica em contexto académico seria interessante introduzir um sistema de deteção de plágio e um maior controlo e visão por parte do organizador dos torneios, quer pela consulta individual das submissões de cada utilizador, quer pela gestão dos participantes no torneio.

¹ A execução prolonga-se pelo tempo máximo atribuído, sendo este um valor arbitrário para cada uma das linguagens.

IX. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] “LeetCode Landing Page,” [Online]. Available: <https://leetcode.com/>. [Acedido em 12 Junho 2018].
- [2] “HackerRank Landing Page,” [Online]. Available: <https://www.hackerrank.com/>. [Acedido em 12 Junho 2018].
- [3] “Codechef,” [Online]. Available: <https://www.codechef.com/>. [Acedido em 17 7 2018].
- [4] “HackerEarth,” [Online]. Available: <https://www.hackerearth.com/>. [Acedido em 17 7 2017].
- [5] “Codeforces,” [Online]. Available: <https://codeforces.com/>. [Acedido em 17 7 2017].
- [6] “TopCoder,” [Online]. Available: <https://www.topcoder.com/>. [Acedido em 17 7 2017].
- [7] “Scrum Portugal,” [Online]. Available: <http://www.scrumportugal.pt/scrum/>. [Acedido em 14 Junho 2018].
- [8] “Agile Manifesto,” [Online]. Available: <http://agilemanifesto.org/>. [Acedido em 18 Junho 2018].
- [9] Oonc, B. Cheanga, A. Kurniaa, A. Limb e Wee-Chong, “Avaliação automatizada de exercícios de programação numa instituição académica,” *On automated grading of programming assignments in an academic institution*, Setembro 2003.
- [10] Thalheim, H. Jaakkola e Bernhard, “Architecture-driven modelling methodologies,” em *Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII.*, 2011.
- [11] “JSON Web Tokens,” [Online]. Available: <https://jwt.io/>. [Acedido em 15 7 2018].