

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Hard Skills: Desafios técnicos partilháveis por recruiters

Bruno Filipe Rocha Coelho



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado em Engenharia de Software

Supervisor: Gil Manuel Magalhães de Andrade Gonçalves

30 de Julho de 2019

Hard Skills: Desafios técnicos partilháveis por recruiters

Bruno Filipe Rocha Coelho

Mestrado em Engenharia de Software

Abstract

Technological developments have increased the demand for jobs in existing companies, so IT companies waste a lot of time on candidate analysis, this often leads to incorrect analysis, this hasty analysis can lead to that the candidate to be chosen is not the candidate with the best skills for the position he will hold.

The objective of this dissertation is combat what was mentioned previously, to develop a platform that is capable of generating questionnaires that will be according to the candidates abilities, this will be achieved with the evaluation by difficulty of the questions to be placed, which will cause that when the questionnaire is generated the questions that will be placed will be according to the level of experience of the candidate. This will allow the candidate to have a recruitment taking into account their level of experience, which may lead to it not being discarded due to having less experience.

The implementation was done in three parts: database, platform, web crawler. Initially it was given the implementation of the database and all its structures to be able to store the information that would be generated by the platform; Next, the platform that generates the questionnaires was developed; finally the Web Crawler was implemented so that it was possible to search new questions automatically.

With this dissertation intends to make the recruitment phase of companies more efficient, so that there are no candidates who are more qualified for a position that are discarded without having an assessment of their capabilities.

Resumo

A evolução tecnológica tem aumentado a procura de emprego nas empresas existentes, por esta razão as empresas de IT desperdiçam um longo tempo na análise de candidatos, isto leva a que muitas das vezes essa análise seja feita de maneira incorreta, esta análise precipitada pode levar a que o candidato escolhido não seja o que tem as melhores capacidades para o cargo que irá desempenhar.

O objetivo desta dissertação é combater o que foi referido anteriormente, passa por desenvolver uma plataforma que seja capaz de gerar questionários que vão estar de acordo com as capacidades do candidato, isto vai ser alcançado com a avaliação por dificuldade das questões a serem colocadas, o que irá fazer com que quando o questionário for gerado as questões que irão ser colocadas vão ser de acordo com o nível de experiência do candidato. Isto vai permitir que o candidato tenha um recrutamento tendo em conta o seu nível, com isto o candidato poderá não ser descartado devido à sua experiência profissional.

A implementação foi feita em três partes: base de dados, plataforma, web crawler. Inicialmente deu-se a implementação da base de dados e todas as suas estruturas para ser possível armazenar a informação que iria ser gerada pela plataforma,; De seguida procedeu-se ao desenvolvimento da plataforma que irá gerar os questionários; por fim implementou-se o Web Crawler para assim ser possível pesquisar novas questões de forma automática.

Com esta dissertação pretende-se tornar a fase de recrutamento das empresas mais eficiente, para não haver candidatos que sejam mais qualificados para determinado cargo que sejam descartados sem terem uma avaliação das suas capacidades.

Agradecimentos

Mais uma etapa que chega ao fim, neste pequeno espaço irei agradecer a todas a pessoas que diretamente ou indiretamente me ajudaram a finalizar mais um etapa da minha vida.

Inicialmente começo por agradecer aos meus pais e irmã por toda a paciência e apoio que me deram nesta jornada, por me ajudarem sempre a crescer como pessoa e por tudo o resto, porque sem eles nada disto era possível.

Agradecer a toda a minha família pelos conselhos e por todo o apoio dado.

Aos meus amigos, gostaria de agradecer a todos que de uma forma ou de outra me apoiaram durante esta jornada.

Ao professor Gil Manuel Gonçalves por me ter guiado e por ter dado todos os seus conselhos para a elaboração desta dissertação.

Por fim, agradeço à equipa da Winprovit pela ajuda prestada e pela facilidade e disponibilidade de me receberem.

Bruno Coelho

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	4
1.3	Problema	4
1.4	Objetivos	5
1.5	Contribuições	5
1.6	Conclusão	6
2	Estado da Arte	7
2.1	Introdução	7
2.2	Plataformas existentes para avaliação das Hard Skills	7
2.2.1	CoderPad	8
2.2.2	HackerRank	10
2.2.3	Codility	13
2.3	Pesquisa, Avaliação e Organização das Questões	14
2.4	Web Crawling	15
3	Caraterização do Problema e Planeamento	17
3.1	Caraterização dos Problemas	17
3.1.1	Avaliação Hard Skills	17
3.1.2	Nível de Dificuldade das Questões	18
3.2	Planeamento Plataforma	18
3.3	Requisitos	19
3.3.1	Validação da Solução	20
3.4	Metodologias e Ferramentas Utilizadas	22
3.4.1	Metodologia	22
3.4.2	Ferramentas Utilizadas	23
4	Implementação	25
4.1	Base Dados	25
4.1.1	Estrutura das tabelas Users e Roles	26
4.1.2	Estrutura das tabelas das Questões	26
4.1.3	Estrutura das tabelas dos Candidatos	27
4.1.4	Estrutura das tabelas dos Questionários	28
4.2	Plataforma WEB	29
4.2.1	Ações Comuns	30
4.2.2	Ações do Administrador	30
4.2.3	Ações do Publisher	32

4.2.4	Ações do Recruiter	35
4.3	Implementação Web Crawler	39
5	Validação	41
5.1	User Stories	41
5.2	Validação	42
5.2.1	Validação das User Stories do Administrador	42
5.2.2	Validação das User Stories do Publisher	43
5.2.3	Validação dos User Stories do Recruiter	46
5.3	Conclusão	50
6	Conclusões e trabalho futuro	51
6.1	Satisfação dos objetivos	51
6.2	Trabalho Futuro	51
6.2.1	Algoritmo Avaliação Automática das Questões	52
6.2.2	Escrita de Código	52
6.2.3	Melhoramento Web Crawler	52
	Referências	53

Lista de Figuras

1.1	Tipos de Métodos usados no início do recrutamento	2
1.2	Fases do Processo Recrutamento	3
2.1	CoderPad - Possibilidade de chamada de vídeo	8
2.2	CoderPad - Gravação da Entrevista	9
2.3	CoderPad - Execução de Código	9
2.4	CoderPad - Seleção da Linguagem	10
2.5	CoderPad - Criação/Seleção de Questões	10
2.6	HackerRank - Criação de teste	11
2.7	HackerRank - Seleção da Linguagem	12
2.8	HackerRank - Seleção das Questões	13
2.9	HackerRank - Painel de Execução do Teste	13
2.10	Plataforma do Codility	14
2.11	Organização das Questões	15
2.12	Funcionamento Web Crawler	16
3.1	Ciclo da Metodologia Scrum	22
3.2	Diagrama do MVC	24
4.1	Diagrama da Base Dados	25
4.2	Diagrama das tabelas Users e Roles	26
4.3	Diagrama da estrutura das Questões	27
4.4	Diagrama da estrutura dos Candidatos	28
4.5	Diagrama das tabelas da estrutura dos Questionários	29
4.6	GUI-Login da Plataforma HardSurvey	30
4.7	GUI-Registo de Novo Publisher na Plataforma	31
4.8	GUI-Registo de Novo Recruiter na Plataforma	32
4.9	GUI-Registo de Nova Questão	33
4.10	Implementação da verificação e registo de nova questão na base de dados	34
4.11	GUI-Pop-up adicionar nova skill	35
4.12	Implementação de verificação e armazenamento de nova skill na Base Dados	35
4.13	GUI-Questões Registadas	35
4.14	GUI-Lista dos Candidatos Registrado	36
4.15	GUI-Perfil do Candidato	36
4.16	GUI-Informação necessária para gerar questionário	37
4.17	GUI-Questionário Gerado	38
4.18	Método criado para obter questões	38
4.19	Função AJAX	39
4.20	GUI-Questionário Corrigido	39

4.21	Classe QuestionSearch	40
5.1	GUI-Menu de Opções do Administrador	42
5.2	GUI-Registo de Novo Recruiter	43
5.3	GUI-Menu de Opções do Publisher	44
5.4	GUI-Preenchimento Nova questão	45
5.5	GUI-Preenchimento Nova Skill	46
5.6	GUI-Menu de Opções do Recruiter	47
5.7	GUI-Registo de Novo Candidato	48
5.8	GUI-Perfil Candidato Criado	48
5.9	GUI-Adicionar nova experiência	49
5.10	GUI-Informação do questionário	49
5.11	GUI-Resultados das questões	50

Lista de Tabelas

3.1	Requisitos do projeto	19
3.2	Endpoints das ações do Admin	21
3.3	Endpoints das ações do Publisher	21
3.4	Endpoints das ações do Recruiter	21
5.1	User Stories	41

Acrónimos

BD	Base Dados
GUI	Interface gráfica do utilizador
WWW	World Wide Web
URL	Uniform Resource Locator

Capítulo 1

Introdução

Neste capítulo será expresso o contexto, motivação e os objetivos pretendidos para o desenvolvimento desta dissertação.

1.1 Enquadramento

Esta dissertação irá ser desenvolvida num ambiente empresarial, sendo que o participante externo é a Winprovit e o objetivo é o desenvolvimento de uma aplicação web em que seja possível gerar questionários sobre hard skills selecionadas com base em alguns fatores como a experiência, stack de desenvolvimento e skills.

A Winprovit é uma empresa de tecnologias de informação que tem no seu core de serviços a reengenharia de processos e consultoria na área de arquitetura dos sistemas de informação.

Atualmente o número de candidatos qualificados com as skills adequadas não está a ter uma evolução tão rápida como a oferta que existe no mercado, o que faz com que a concorrência durante o recrutamento seja elevada, o que torna os trabalhadores de IT os segundo grupo mais difícil de recrutar e cerca de 40% empregadores relata uma grande escassez de talento. Este processo de recrutamento faz com que seja desperdiçado bastante tempo seja por analisar candidatos não qualificados, seja como os que poderão ser qualificados. Durante o recrutamento podem haver falhas, e assim perdermos talento simplesmente porque um candidato não consegue apresentar todas as suas habilidades no seu currículo, por isso a melhor maneira de combater isto é avaliar as capacidades técnicas dos candidatos no início do processo de recrutamento, o que vai levar a que não se percam talentos, embora seguir este método se torne caro.[22]

Esta avaliação das capacidades é utilizada para avaliar diversas capacidades embora a que seja mais normal avaliar no início da fase de recrutamento sejam os testes de habilidades, como mostra a figura 1.1 .

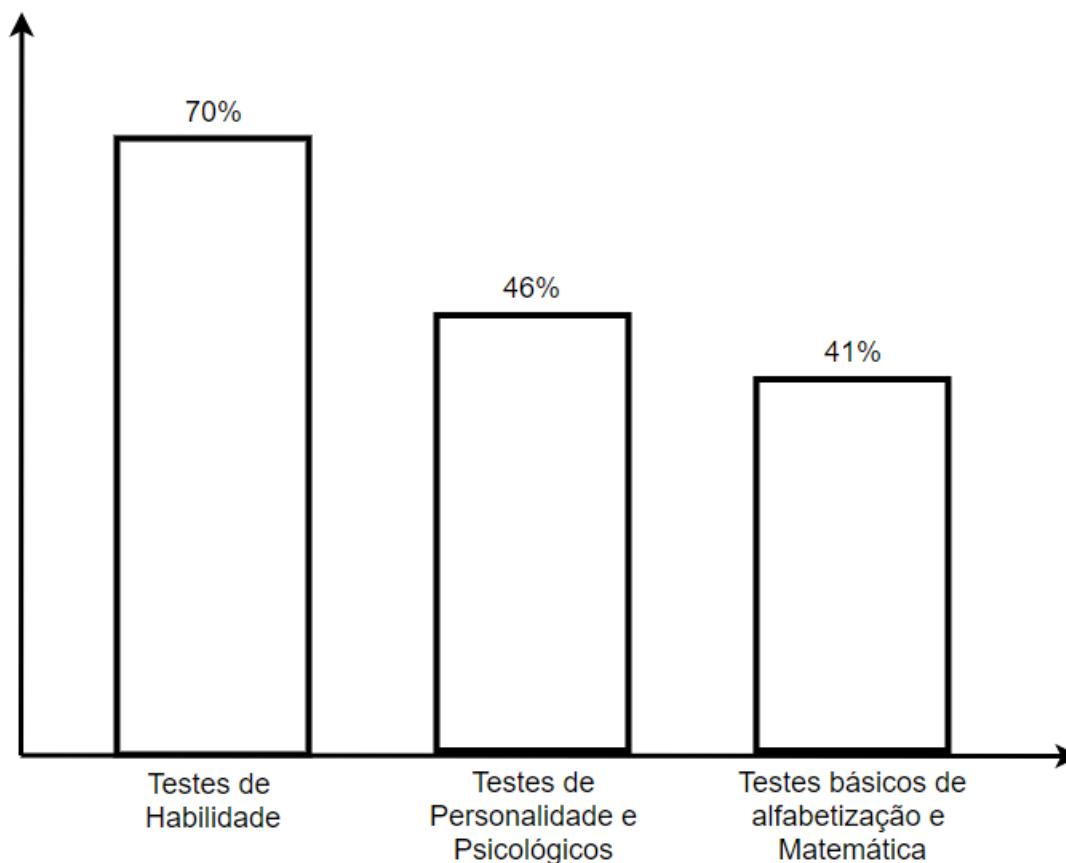


Figura 1.1: Tipos de Métodos usados no início do recrutamento

Devido à dificuldade de recrutamento e escassez de talento todas as empresas pretendem contratar os melhores profissionais da área da tecnologia, para isso é necessário que o recrutamento seja feito da forma mais rápida e efetiva possível, isto é necessário porque o talento tecnológico atualmente é contratado rapidamente logo as empresas pretendem ter um processo de recrutamento rápido e efetivo para fazerem ofertas antes dos seus concorrentes diretos. Para isso devem seguir um processo de recrutamento que tem 5 fases, como é mostrado na seguinte figura 1.2. Primeiro é feita a identificação de candidatos que poderão ser valiosos para o pretendido pela empresa, aqui têm de ser identificados vários candidatos não só os que se candidataram ao cargo, mas também existir uma pesquisa que possa identificar possíveis candidatos que por várias razões podem não se ter candidatado. Na segunda fase vão ser selecionados os candidatos que irão conseguir fazer um bom trabalho isto é feito através da avaliação das skills que o candidato poderá ter. Na terceira fase são convidados candidatos que serão entrevistados e onde irãp serão avaliadas as suas qualidades técnicas embora a forma como isso é feito por vezes é ineficiente. No fim é feita a oferta

de empresa e possivelmente a contratação, embora segundo a MRINetwork [8] mais de 50% das ofertas são recusadas isto acontece devido à competição existente entre empresas, o que leva a que por vezes seja necessário contratar um candidato que não era a primeira escolha.[22]

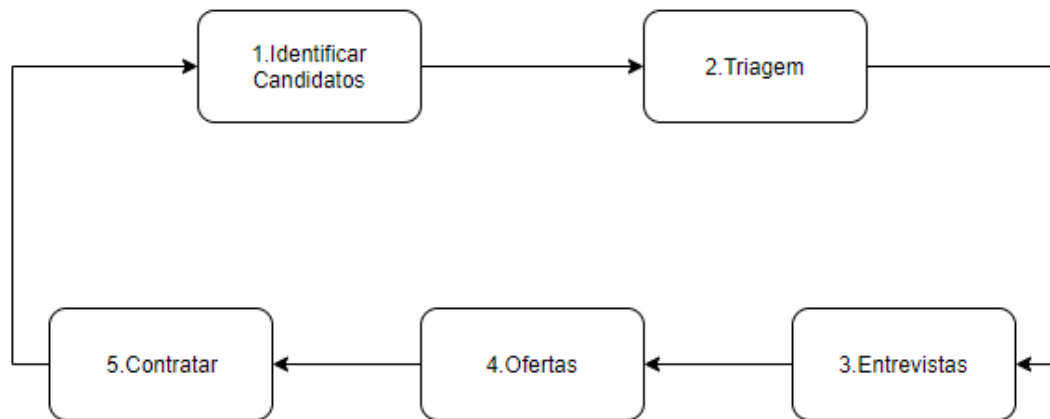


Figura 1.2: Fases do Processo Recrutamento

Focando na plataforma a ser desenvolvida o objetivo passa por ser uma plataforma que irá ser usada numa fase inicial do recrutamento para que assim não sejam perdidos talentos, que não consigam exprimir as suas habilidade nos seus currículos, sendo que depois poderá ser usada novamente na terceira fase, as entrevistas, para que assim seja possível aprofundar e avaliar de uma maneira mais abrangente as capacidades dos candidatos, isso irá ser feito através de um questionário de questões técnicas que irá ser gerado através da plataforma esse questionário vai ter em consideração tanto a experiência bem como as skills que o candidato possui. Com esta plataforma irá ser possível obter de forma mais eficiente as capacidades técnicas de cada candidato ao cargo oferecido. As questões que irão estar nesses questionário serão questões técnicas que estarão guardadas numa base de dados que vai sendo atualizada com novas questões.

Essa base de dados vai estar constantemente a ser atualizada com novas questões, essas questões irão ser pesquisadas de forma automática através do Web Crawling. O Web Crawling é um bot que procura em páginas WEB informação que possa interessar e guarda essa informação, o que permite manter as base de dados atualizadas.[14] Neste caso o web crawling vai ser utilizado para pesquisar novas questões para que essas possam ser analisadas e depois serem inseridas na base de dados, isto vai permitir que a base de dados esteja sempre em atualização e assim aumentar as questões disponíveis.

1.2 Motivação

Nesta dissertação existem vários objetivos que têm de ser cumpridos, atualmente já existem várias ferramentas que permitem fazer a avaliação técnicas dos candidatos, e assim conseguirmos avaliar as suas Hard Skills, mas o problema nestas plataformas é que testam apenas a capacidade de desenvolver código, sendo que não têm forma de avaliar as capacidades técnico-teóricas dos candidatos.

Com esta nova plataforma que esta a ser desenvolvida nesta dissertação vai ser possível avaliar os candidatos com questões de nível técnico-teórico, o que vai permitir avaliar de uma forma mais abrangente os candidatos, e assim levar a uma melhor análise das suas capacidades, para além disto como referido anteriormente se esta plataforma for utilizada numa fase inicial do recrutamento pode tornar a avaliação muito mais pormenorizada o que leva a que candidatos que poderão ser excluídos por diversas razões não o sejam pois vai ser submetido a um teste que demonstre um bom nível técnico que poderia não ter sido avaliado. Outra das características que vai distinguir esta plataforma é o facto de que irá existir uma ligação entre a experiência do candidato e a dificuldade das questões que irão ser colocadas o que vai permitir que os testes gerados estejam de acordo com a experiência do candidato em questão, esta questão é importante porque o que poderia ocorrer se isto não existisse era fazer questões que poderiam ser demasiado complicadas a um candidato que tem pouca experiência de trabalho, sendo que é normal que um candidato com uma grande experiência num ambiente de trabalho tenha mais conhecimentos que um recém-licenciado. Além de ser criada uma nova plataforma de vai ajudar no recrutamento de novos talentos, vai existir uma biblioteca de questões teóricas de nível técnico que vai estar em constante atualização devido ao Web Crawling.

1.3 Problema

Este projeto surge pelo facto de ainda existirem diversos problemas que necessitam de ser resolvidos como por exemplo muitas empresas durante a fase de recrutamento apenas fazem a avaliação do candidato tendo como base a experiência passada e as soft skills que o candidato possui, muitas vezes as hard skills que focam mais nas questões técnicas e testam as capacidades do candidato acabam por ser ignoradas, isto pode levar a que tendo entrevistado vários candidatos se possa não ter optado pelo candidato mais capaz a nível técnico e assim perder-se um bom talento, esta questão pode ser resolvida através da criação desta plataforma, em que o objetivo é que exista uma avaliação das capacidades técnicas dos candidatos.

Outro dos problemas existentes é apesar de essas capacidades técnicas serem avaliadas, as questões colocadas podem não estar de acordo com o nível de experiência do candidato o que pode

levar a que a sua avaliação técnica seja incorreta, isto acontece porque o nível de dificuldade das questões colocadas não está alinhado com o nível de competência e experiência do candidato, este problema irá ser resolvido através de um estudo das diversas questões, sendo que essas questões irão ser avaliadas por nível de dificuldade sendo que durante a criação do questionário ao ser especificada a experiência do candidato vai ser possível obter as melhores questões para o seu nível, para além da associação com as skills correspondente do candidato.

1.4 Objetivos

No fim desta dissertação pretende-se obter uma plataforma que seja usada no recrutamento de novos candidatos por parte das empresas, vai ser uma plataforma de geração de questionários, em que as suas questões irão ser questões teóricas de nível técnico, com estas questões vai ser possível avaliar o candidato de outras formas, para além das suas capacidades a nível de escrita de código.

Outro dos objetivos propostos é que exista uma avaliação de dificuldade das questões de forma correta, isto vai resolver o problema da associação entre a dificuldade das questões e a experiência do candidato a quem estão a ser colocadas essas questões.

A criação de um motor de busca automática, o Web Crawling, para que assim seja possível encontrar questões que poderão ser adicionadas na base de dados depois de serem analisadas para lhes ser atribuído um nível de dificuldade, este motor vai permitir que a base de dados criada esteja em constante evolução e atualização.

1.5 Contribuições

Relativamente às contribuições que esta dissertação irá trazer serão uma nova plataforma de avaliação das hard skills sendo essa plataforma capaz de basear os seus questionários tendo em conta a experiência, skills e as tecnologias que pretendem ser testadas. Vai ser criado um algoritmo de apoio à decisão, sendo que neste caso o objetivo deste algoritmo é selecionar as melhores questões tendo em conta a experiência do candidato e as linguagens pretendidas. Por fim irá ser criado um novo "banco de perguntas" onde irão estar todas as questões técnicas que irão ser colocadas nos questionários, e que podem ser utilizadas para outros fins.

1.6 Conclusão

Esta secção serviu para inicialmente dar a introdução da dissertação que irá ser realizada, explicou-se qual a motivação para o desenvolvimento do projeto. De seguida abordamos alguns dos problemas que se pretendem solucionar com o desenvolvimento deste projeto, bem como os objetivos que temos definidos. Por fim apresentamos algumas das contribuições que esta dissertação poderá vir a fornecer.

Capítulo 2

Estado da Arte

2.1 Introdução

As empresas estão constantemente à procura de novos talentos no mundo tecnológico para efetuarem o recrutamento para as suas equipas. Para avaliar e testar os diversos candidatos essas empresas necessitam de ferramentas de avaliação.

Atualmente existem diversas plataformas de avaliações técnicas de candidatos, embora sejam quase todas focadas em escrita de código, o que abandona a avaliação técnico-teórica dos candidatos.

De seguida serão apresentadas e analisadas diversas plataformas que podem ser usadas nas avaliações técnicas dos candidatos, embora todas foquem nas capacidades de escrita de código e na sua avaliação, e a comparação com a plataforma que está a ser desenvolvida nesta dissertação.

Sendo que a maior existente entre a nossa plataforma e as já existentes, são a diferença das questões, sendo que na maioria das plataformas existentes sejam questões que necessitam de resposta em código, enquanto na nossa plataforma são questões teórico-técnicas de escolha múltipla, a outra grande diferença existente é o facto das questões que serão colocadas nos questionários criados pela nossa plataforma terem um nível de dificuldade associado.

2.2 Plataformas existentes para avaliação das Hard Skills

Aqui irão ser apresentadas algumas das plataformas existentes que efetuam a avaliação das hard skills. Os pontos que vão ser mais analisadas é o facto de como é feita essa avaliação, o tipo de questões e se as questões que irão ser colocadas são diferenciadas por nível de dificuldade.

Depois de efetuar a pesquisa e análise chegou-se à conclusão que a maioria das plataformas disponíveis faz a sua avaliação através de escrita de código e as suas questões não estão avaliadas por dificuldade.

Seguidamente irão ser apresentadas algumas dessas ferramentas.

2.2.1 CoderPad

O CoderPad é uma plataforma paga que permite efetuar entrevistas e efetuar uma avaliação das capacidades técnicas dos candidatos.

É uma ferramenta mais focada para entrevistas que sejam feitas através de chamadas online, esta plataforma possui ferramentas que fazem com que seja possível fazer uma chamada de vídeo durante a entrevista, figura 2.1 o que leva a que não seja necessário usar uma plataforma externa, depois da entrevista estar concluída é possível rever toda a conversa que foi feita durante a entrevista, figura 2.2.

Abordando agora as ferramentas que possui para avaliação, o CoderPad avalia os candidatos através da execução do código implementado pelo candidato, este código pode ser executado na plataforma, como mostra na figura 2.3, para além disso suporta questões nas mais variadas linguagens de programação, como é apresentado na figura 2.4. Nesta plataforma é possível criar novas questões, que depois podem ser escolhidas durante a criação do teste, figura 2.5. O que difere para o projeto que está a ser desenvolvido é o facto de as questões no CoderPad serem apenas questões baseadas na escrita e resolução de problemas através de código, enquanto nesta dissertação as questões serão mais teóricas e de escolha múltipla o que pode levar a que abranjam outros aspetos, outro ponto em falta no CoderPad é o facto de as questões não terem um nível de dificuldade associado e ser necessário criar e escolher as questões que pretendemos para o teste.[3]

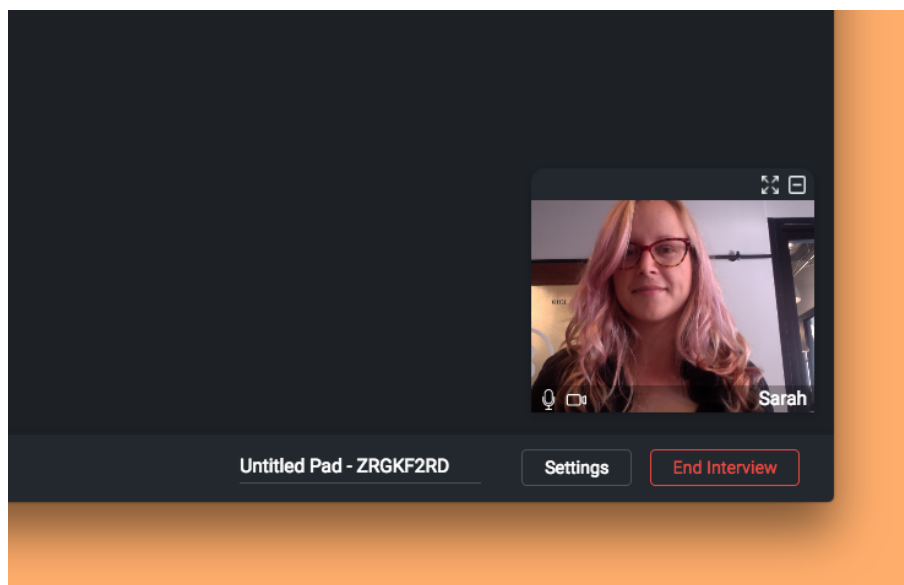


Figura 2.1: CoderPad - Possibilidade de chamada de vídeo

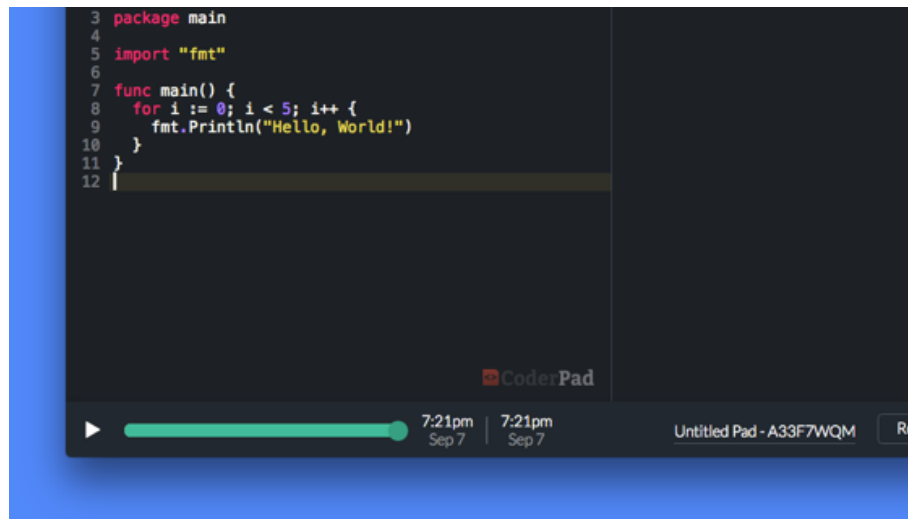


Figura 2.2: CoderPad - Gravação da Entrevista

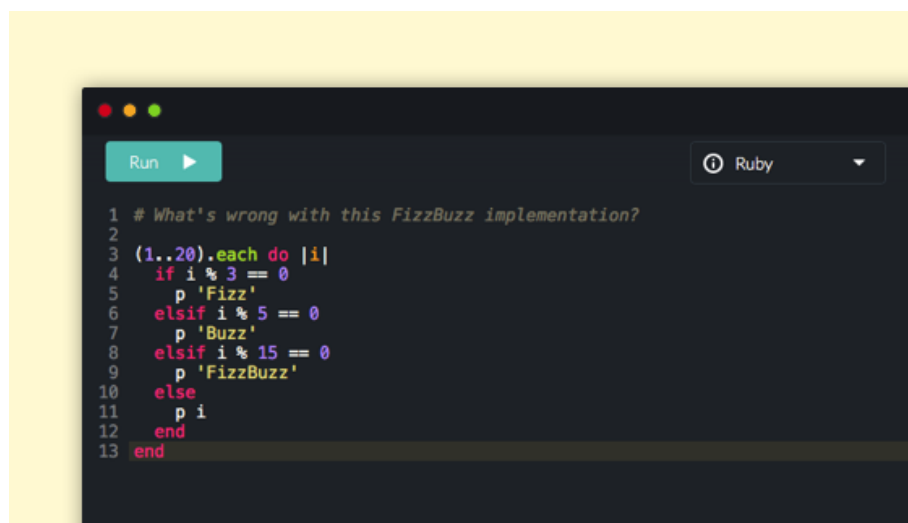


Figura 2.3: CoderPad - Execução de Código

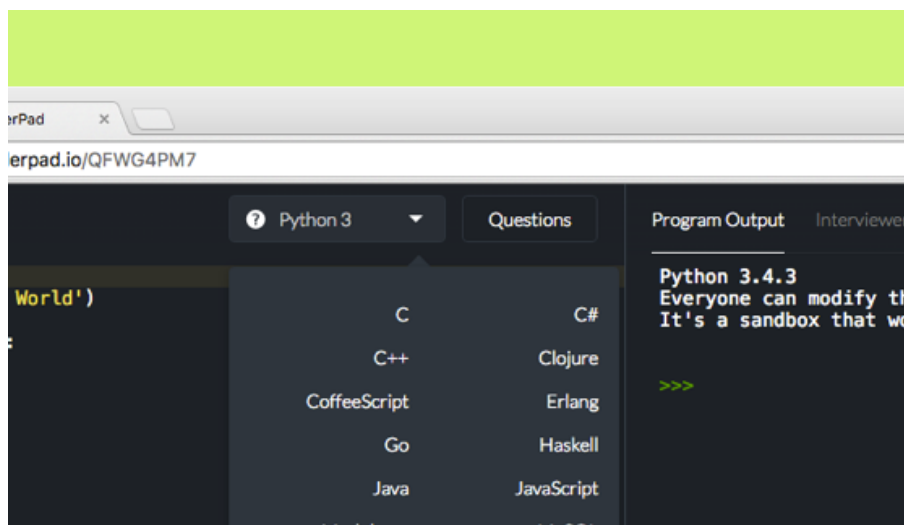


Figura 2.4: CoderPad - Seleção da Linguagem

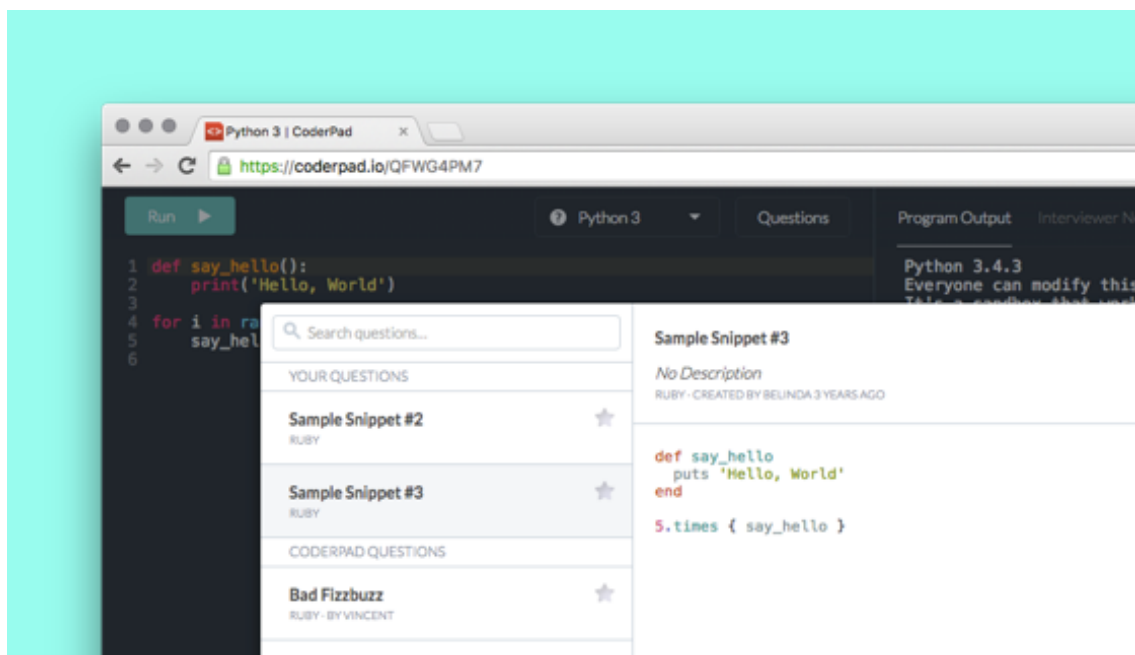


Figura 2.5: CoderPad - Criação/Seleção de Questões

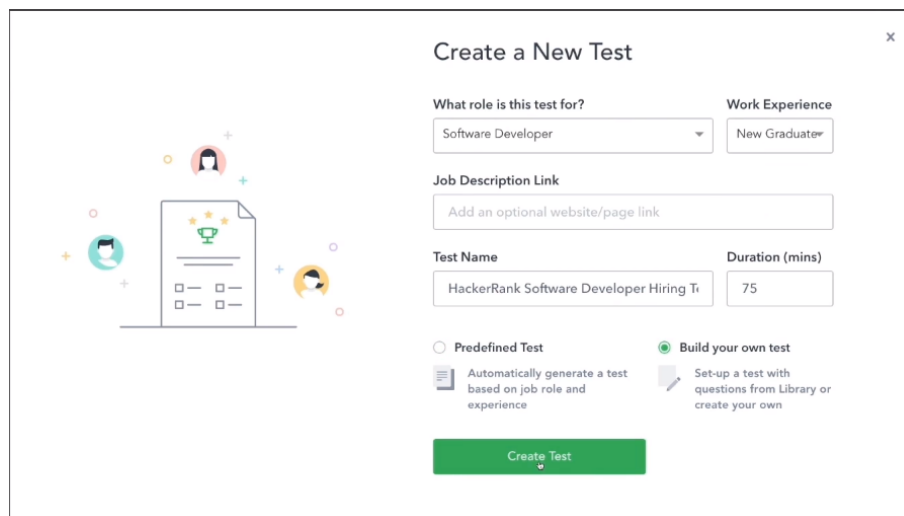
2.2.2 HackerRank

O HackerRank é uma plataforma que para além de ser possível criar testes para fazer a avaliação de candidatos durante entrevistas, também é uma plataforma em que qualquer utilizador poderá registar e praticar as suas skills de código.[6]

Quanto às ferramentas que o HackerRank possui, através desta plataforma é possível criar novos testes ou então usar testes anteriormente criados, figura 2.6, na criação desse teste é possível definir o cargo para que o teste vai ser feito, a experiência do candidato que vai ser testado, e a

duração total para a realização do teste. Durante a criação do teste é possível escolher as linguagens que pretendem ser testadas, como mostra na figura 2.7, relativamente à escolha das questões para o teste essas questões indicam a linguagem a que se referem e além disso indicam o nível de dificuldade dessa questão, figura 2.8. Quando o teste está a ser realizado tanto o entrevistado como o entrevistador estão a visualizar a questão a ser respondida, no qual contêm um chat, e onde é possível como no CoderPad 2.2.1 correr o código na plataforma, figura 2.9.[6]

Comparando com o projeto a ser desenvolvido esta é a plataforma que contém os aspetos mais parecidos sendo a grande diferença no tipo de questões colocadas, sendo que no HackerRank as questões são todas baseadas em resposta de código enquanto na plataforma a ser desenvolvida são questões de teor teórico respondidas em escolha múltipla.



The image shows a 'Create a New Test' form with the following fields and options:

- What role is this test for?:** Software Developer (dropdown)
- Work Experience:** New Graduate (dropdown)
- Job Description Link:** Add an optional website/page link (text input)
- Test Name:** HackerRank Software Developer Hiring Ti (text input)
- Duration (mins):** 75 (text input)
- Predefined Test:** Automatically generate a test based on job role and experience (radio button, unselected)
- Build your own test:** Set-up a test with questions from Library or create your own (radio button, selected)
- Create Test:** Green button at the bottom

Figura 2.6: HackerRank - Criação de teste

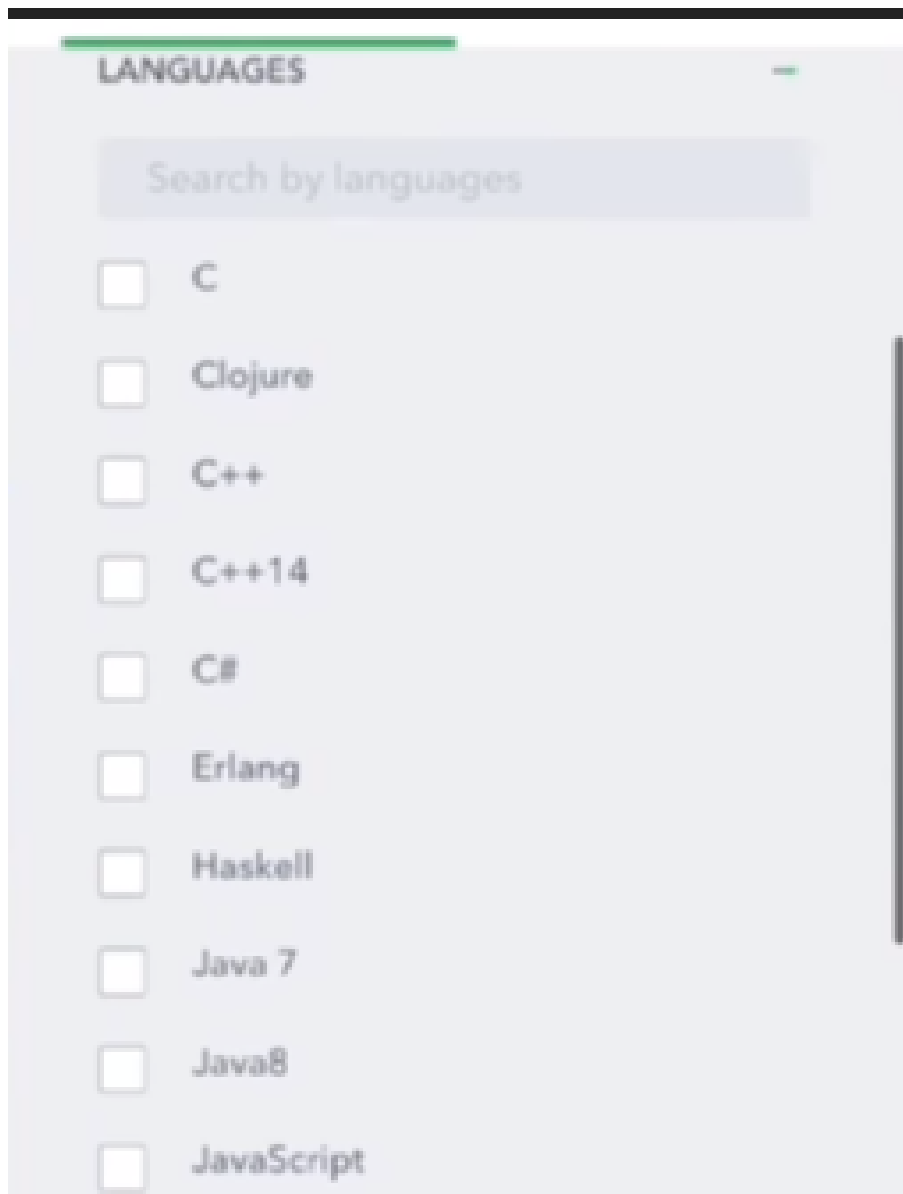


Figura 2.7: HackerRank - Seleção da Linguagem

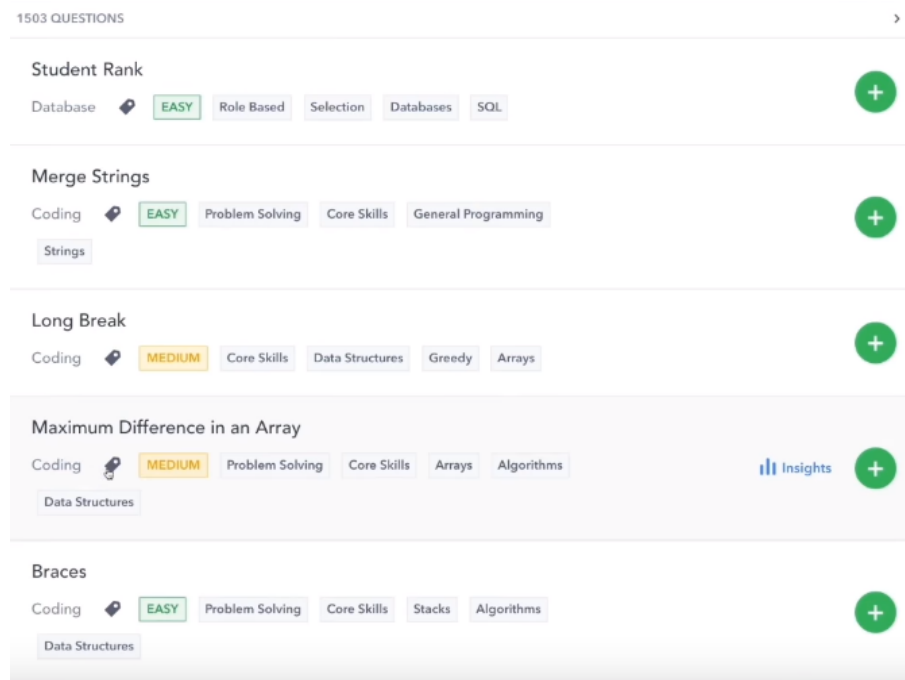


Figura 2.8: HackerRank - Seleção das Questões

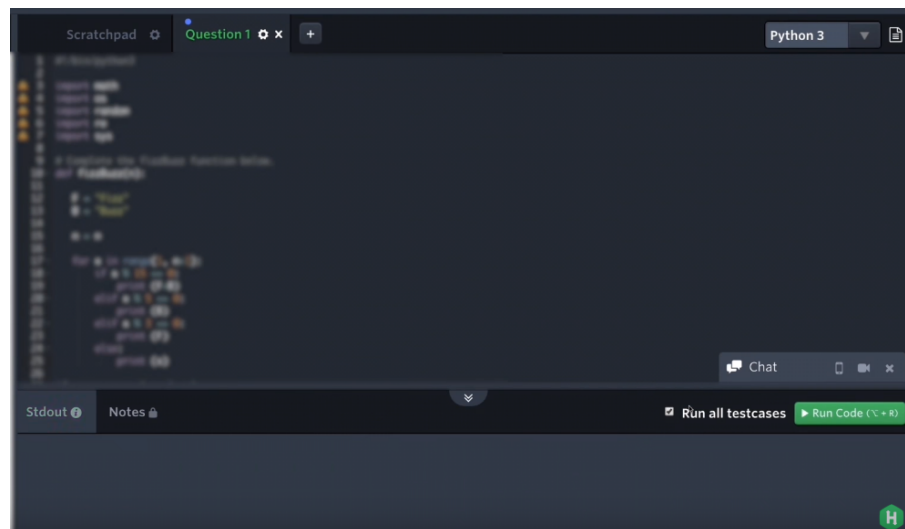


Figura 2.9: HackerRank - Painel de Execução do Teste

2.2.3 Codility

O Codility é uma plataforma bastante semelhante ao CoderPad 2.2.1, possui as mesmas ferramentas, como a entrevista é feita através de uma chamada de vídeo, ambos, entrevistador como entrevistado estão no ambiente em que as questões estão a ser respondida, e possui a capacidade de conseguir correr o código na sua própria plataforma, todas estas ferramentas estão apresentadas na figura 2.10. [4]

Apesar disto o Codility não possui a avaliação de dificuldade das suas questões, e todas as suas questões são baseadas em resposta de código. [4]

The screenshot displays the Codility platform interface. On the left, a task description is shown: 'This is a demo task. Write a function: int solution(vector<int> &A); that, given an array A of N integers, returns the smallest positive integer (greater than 0) that does not occur in A. For example, given A = [1, 3, 6, 4, 1, 2], the function should return 5. Given A = [1, 2, 3], the function should return 4. Given A = [-1, -3], the function should return 1. Write an efficient algorithm for the following assumptions: N is an integer within the range [1..100,000]; each element of array A is an integer within the range [-1,000,000..1,000,000]. Copyright 2009–2018 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.' The main area shows a C++ solution for the 'Missing Integer' problem, which uses a bitset to track the presence of numbers. The code is as follows:

```

17 vector<int> B;
18 int __tmp_arr_i;
19 int __tmp_arr_len;
20 int __tmp_arr_val_int;
21 vector<int> __tmp_array_int;
22 int i;
23 int n;
24 n = __py_len(A);
25 __tmp_arr_len = (n + 2);
26 __tmp_arr_val_int = 0;
27 B.resize(__tmp_arr_len);
28 for (__tmp_arr_i = 0; __tmp_arr_i < __tmp_arr_len; ++__tmp_arr_i)
29     B[__tmp_arr_i] = __tmp_arr_val_int;
30 }
31 for (i = 0; i < n; ++i) {
32     if (((A[i] <= n) && (A[i] > 0))) {
33         B[A[i]] = 1;
34     }
35 }
36 for (i = 0; i < (n + 1); ++i) {
37     if (B[i + 1] == 0) {
38         return (i + 1);
39     }
40 }
41 return (-1);
42 }

```

Below the code, the 'Test Output' section shows two example tests: [1, 2, 3] returning OK and [-1, -3] returning OK. A 'Run Tests' button is visible. At the bottom, a message states: 'Your code is syntactically correct and works properly on the example test. Note that the example tests are not part of your score. On submission at least 8 test cases not shown here will assess your solution.'

Figura 2.10: Plataforma do Codility

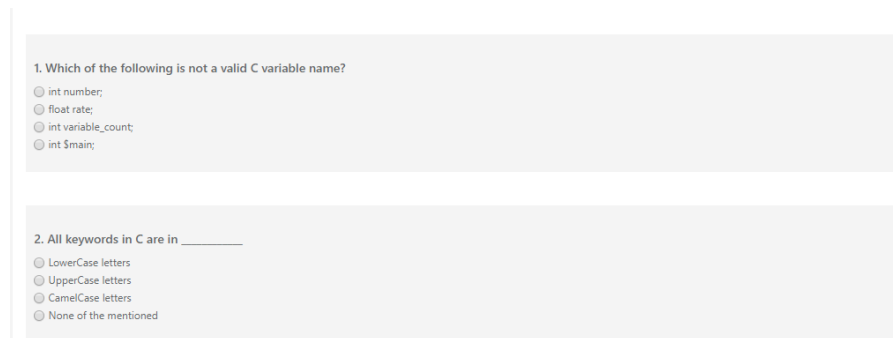
2.3 Pesquisa, Avaliação e Organização das Questões

Nesta dissertação um dos pontos fulcrais são as questões que irão ser colocadas nos questionários de avaliação das hard skills, relativamente à pesquisa dessas questões, inicialmente essas questões irão ser pesquisadas em base de dados já existentes e assim adicionadas na base de dados da nossa plataforma, após o Web Crawling estar desenvolvido essas questões irão ser pesquisadas automaticamente sendo que depois de serem encontradas e retiradas pelo Web Crawling vão ser guardadas num documento que irá conter todas as questões encontradas, essas questões não poderão ser adicionadas automaticamente na base de dados, devido à necessidade de essas questões serem avaliadas por dificuldade.

Relativamente à avaliação por dificuldade, essas questões irão ter de ser revistas e analisadas, e com base em estudos feitos, ver a percentagem de acerto que essas questões tiveram quando foram respondidas, e tendo como base essa percentagem irá ser possível avaliar a dificuldade dessa questão, sendo que isso irá ser essencial para existir a associação da dificuldade da questão com as características do candidato, esta avaliação futuramente poderá ser alterada em algumas questões, sendo que a base de dados dos resultados irá aumentar, e baseando-nos nas questões em que existir uma maior taxa de erro, poderão ser revistas e o seu nível de dificuldade baixar. [21]

Explorando agora a organização e estrutura de como as questões irão ser colocadas nos questionários irão ser questões de resposta em escolha múltipla. Os questionários irão ser elaborados

baseando-se na estrutura da biblioteca SurveyJS, com esta estrutura irá ser possível obter um questionário bem organizado e elaborado, como exemplificado na figura 2.11, através disto irá facilitar o trabalho de correção e armazenamento das questões respondidas na base de dados, como já foi falado com esta capacidade de analisar os resultados iremos conseguir avaliar o nível de dificuldade das questões sendo que se as questões estiverem com uma taxa de erro muito elevada deverão ser revistas. [11] [17]



The image shows a screenshot of a survey form with two questions. The first question is: "1. Which of the following is not a valid C variable name?" with four radio button options: "int number;", "float rate;", "int variable_count;", and "int \$main;". The second question is: "2. All keywords in C are in _____" with four radio button options: "LowerCase letters", "UpperCase letters", "CamelCase letters", and "None of the mentioned".

Figura 2.11: Organização das Questões

2.4 Web Crawling

Neste projeto vai ser utilizado um Web Crawling de forma a conseguirmos obter mais questões que depois de serem analisadas e avaliadas por dificuldade irão ser colocadas na base de dados. Explicando do que se trata o Web Crawling, é um bot, muitas vezes chamado de "Crawler", este bot pode ser utilizado para diversas tarefas, um exemplo disso é o que se passa no google em que todas as páginas visitadas vão ficar guardadas numa cópia o que vai permitir que uma futura pesquisa se torne mais rápida, podem ser utilizadas para tarefas de manutenção, e o que pretendemos para este projeto e que pode ser executado pelo "Crawler" é o facto de conseguir retirar informação de páginas web fornecidas pelo utilizador[15]. Na seguinte figura 2.12 é exemplificado o funcionamento do Web Crawler . [14] [12]

Com esta ferramenta como foi dito anteriormente iremos ser capazes de retirar questões da web o que vai aumentar o numero de questões que poderão ser colocadas nos questionários que serão gerados.

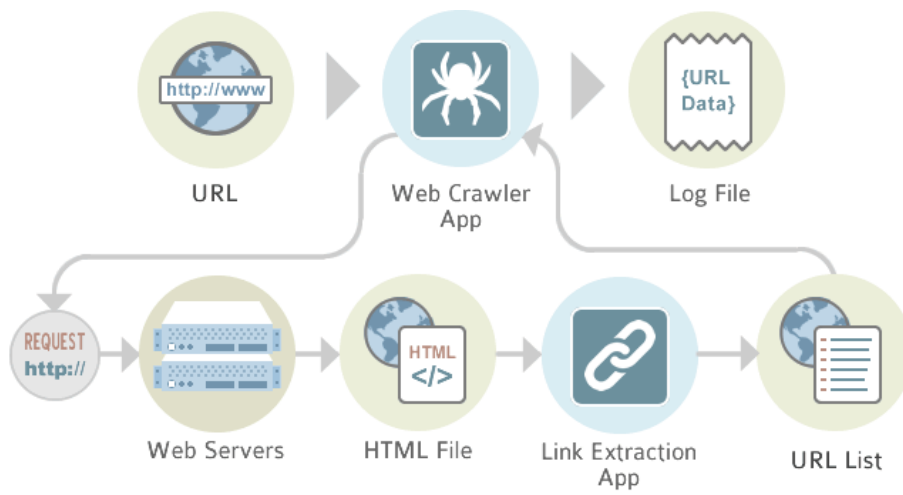


Figura 2.12: Funcionamento Web Crawler

Capítulo 3

Caraterização do Problema e Planeamento

Neste capítulo serão abordados em mais detalhe os problemas detetados, detalhando assim o que será necessário para resolver esses problemas, bem como o planeamento feito para dar início à implementação do projeto com a devida resolução dos problemas identificados.

Este capítulo começará com a exposição dos problemas identificados, são eles o facto de não existirem aplicações focadas na avaliação teórica, e também a questão de as questões não serem avaliadas por nível de dificuldade, o que leva a que exista o problema de muitas vezes não ser feita uma correspondência do nível das questões com a experiência do candidato.

A segunda parte do capítulo vai focar essencialmente no planeamento do projeto, identificando os requisitos, validação do planeamento.

A terceira parte do capítulo irá ser apresentada a metodologia que vai ser usada e irão expostas as tecnologias que iram ser utilizadas.

3.1 Caraterização dos Problemas

Como referido anteriormente nesta secção vão ser discutidos os problemas existente e a sua resolução.

3.1.1 Avaliação Hard Skills

Um dos problemas identificados é o facto de atualmente muitas das empresas existentes no mercado não terem forma de avaliação das Hard Skills, o que por vezes leva a que a fase de recrutamento não seja feita da maneira mais correta, este problema pode levar a que as empresas quando identificam os candidatos pretendidos para determinado posto na sua empresa, podem não estar a identificar os melhores candidatos porque não conseguiram fazer da melhor maneira a avaliação das skills mais preponderantes para determinado cargo, o que leva muitas vezes candidatos com mais experiência no tipo de cargo que estão a concorrer a serem deixados de fora. Por esta razão este projeto visa elaborar uma forma de se conseguir obter os melhores candidatos para determinado cargo, fazendo com que todos os candidatos elaborem um questionário com questões teórico-práticas de escolha múltipla onde vão ser testadas as suas capacidades e assim ajudar as empresas no recrutamento mais acertado dos candidatos escolhidos. O que levaria a que fossem sempre encontrados os melhores candidatos para determinado cargo.

3.1.2 Nível de Dificuldade das Questões

Outro dos problemas identificados é o facto de as questões colocadas noutras plataformas não serem avaliadas por nível de dificuldade, este problema pode levar a que as questões colocadas a determinado candidato não sejam questões que correspondam ao seu nível de experiência. Logo foram identificados estes dois problemas, o facto das questões não serem avaliadas em nível de dificuldade, e a ausência de correspondência entre as questões colocadas e o nível de experiência do candidato. De forma a resolver este problema nesta plataforma pretende-se que seja possível que as questões tenham um determinado nível de dificuldade que será dividido em nível (Junior, Medium, Senior), bem como a skill de cada questão, isto fará com que seja possível gerar questionários adaptados para cada candidato, em que se pode testar as qualificações de determinado candidato tendo em conta o seu nível de experiência, o que leva a que se possa avaliar da melhor maneira possível o nível do candidato que se encontra em fase de recrutamento.

3.2 Planeamento Plataforma

Neste ponto irá ser discutido o planeamento da plataforma a ser desenvolvida.

Tendo em conta que irá ser necessário ter uma plataforma capaz de gerar questionários para determinadas características de um candidato, irá ser necessário elaborar uma base de dados onde serão guardadas todas as informações da plataforma.

Abordando o planeamento e desenvolvimento da plataforma irá ser necessário existir roles com permissões para o que poderá ser feito por cada uma delas, sendo assim serão criadas as roles: Admin, Publisher e Recruiter, cada uma destas roles com funções diferentes. O Admin irá ficar

responsável pelo registo de Publisher's e Recruiter's. Os utilizadores com a role de Publisher serão capazes e ficaram responsáveis por ações de criação e procura de questões que serão adicionadas na base de dados. Enquanto os Recruiter's ficarão responsáveis pelos aspetos de recrutamento, com estes aspetos refiro a recolha das informações do candidato que irá responder ao questionário, esse questionário que irá ser gerado e verificado pelo Recruiter.

Para além do referido e para facilitar o trabalho dos Publisher's irá ser desenvolvido um Web Crawler, que será capaz através de um link e outros aspetos fornecidos recolher de forma automática questões que serão guardadas na base de dados, e que depois poderão ser adicionadas pelos Publisher's na base de dados. [16]

3.3 Requisitos

Tendo como base o que foi descrito anteriormente na secção 3.2 foram definidos no início do projeto alguns requisitos para assim ser possível validar o projeto, contudo alguns deste requisitos ao longo do tempo foram sofrendo alterações.

Tabela 3.1: Requisitos do projeto

Nº	Requisitos
1	<i>Registo de Publishers e Recruiters</i>
2	<i>Efetuar Login</i>
3	<i>Efetuar Logout</i>
4	<i>Adicionar Skills</i>
5	<i>Adicionar Questões</i>
6	<i>Visualizar Lista de Questões</i>
7	<i>Criar Candidato</i>
8	<i>Adicionar Experiência</i>
9	<i>Visualizar Lista de Candidatos</i>
10	<i>Gerar Questionários</i>
11	<i>Ver Lista de Questionários</i>
12	<i>Ver Resultado Survey</i>
13	<i>Pesquisa de Questões</i>

O primeiro requisitos diz respeito ao processo de registo dos trabalhadores, estes trabalhadores sejam eles recruiters ou publishers irão ser registados por um admin, estes trabalhadores dependendo do seu tipo, caso seja um Publisher irá ser responsável pelos requisitos 4,5,6,11 . Enquanto o trabalhador que irá ter o trabalho de Recruiter será capaz de utilizar os requisitos 7,8,9,10.

Relativamente aos requisitos 2 e 3, com a existência do requisito 2, Efetuar Login, onde é necessário existir um login para que quem aceder à plataforma consiga fazer as suas respetivas ações caso seja um Admin, Publisher ou Recruiter, conseqüentemente existindo um Login é necessário existir o requisitos 3, Efetuar Logout, para ser possível aos utilizadores efetuarem o logout da sua conta.

Relativamente ao requisitos 4,5,6 são requisitos que serão utilizados pelos Publisher's, abordando o requisito 4, Adicionar Skills, este requisitos deve-se ao facto de ser necessário adicionar as skills que serão utilizadas para definir qual a skill respetiva de cada pergunta que será apresentada nos questionários quer seja para serem adicionadas na experiência do candidato. O requisito 5, Adicionar Questões, este requisito é necessário para ser possível adicionar as questões que são pretendidas na base de dados existente sendo que esta questões terão uma skill associada e nível de dificuldade. O requisito 6, Visualizar Lista de Questões, existe para assim ser possível ao Publisher ver quais as questões adicionadas e se assim pretender eliminar algumas delas.

Os requisitos 7,8,9,10,11,12 são requisitos que vão ser maioritariamente utilizados pelos Recruiters, através destes requisitos vai ser possível criar os questionários tendo como base o candidato e o seu nível de experiência. Abordando o requisito 7, Criar Candidate, com este requisito irá ser possível ao utilizador, neste caso com o cargo de Recruiter, criar um novo candidato que irá responder aos questionários propostos. Esses questionários irão ser gerados através da utilização do requisito 10, Gerar Questionários, este questionários vão ser gerados de acordo com o nível do candidato e de acordo com a experiência(requisito 8, Adicionar Experiência) dele mesmo. A existência do requisito 9, Visualizar Lista de Candidatos, existe para que o Recruiter seja capaz de ver a lista de candidatos registados e assim conseguir identificar o candidato que pretende, sendo depois encaminhado para o perfil do candidato onde poderá adicionar experiência desse mesmo candidato, gerar um questionário com base nas suas características, bem como visualizar os surveys feitos(requisito 11, Ver Lista de Questionários), bem como os resultados obtidos nesse mesmo survey(requisito 12, Ver Resultado Survey).

Relativamente ao ultimo requisito existe devido ao facto de ser necessário existir uma forma de recolha de questões automaticamente para não ser necessário ao publisher procurar questões de forma manual, para isso vai ser implementado um web crawler, que será capaz através de um link fornecido e dos parâmetros necessários retirar automaticamente as questões desse mesmo website.

3.3.1 Validação da Solução

Para que os requisitos sejam cumpridos foi pensado um conjunto de endpoints, onde foram divididos em tabelas conforme as roles apresentadas anteriormente e existentes na plataforma, com isto conseguiremos criar uma validação do planeamento feito.

Tabela 3.2: Endpoints das ações do Admin

Método	URL	Descrição
POST	/Admin/RegistrarPublisher	Registrar novo Publisher
POST	/Admin/RegistrarRecruiter	Registrar novo Recruiter

Relativamente aos endpoints apresentados na tabela 3.2, com estes endpoints vai ser possível validar os objetivos que foram propostas para o administrador da plataforma que com isto poderá criar novos Publisher's e Recruiter's.

Tabela 3.3: Endpoints das ações do Publisher

Método	URL	Descrição
POST	/Publisher/AddSkill	Adiciona Nova Skill
POST	/Publisher/AddQuestion	Adiciona Nova Questão
GET	/Publisher/VerQuestoesRegistadas	Possibilidade de ver a lista de questões adicionadas
DELETE	/Publisher/ApagarQuestão	Apaga Questão pretendida

Na tabela 3.3 foram elaborados os endpoints para que exista a possibilidade de validar todos os requisitos referentes aos Publisher's da plataforma, em que ele vai conseguir adicionar skills que posteriormente poderão ser usadas na criação de uma nova questão, ao ser adicionada a questão ira ser colocada a pergunta, as possíveis respostas, o seu nível de dificuldade e a skill correspondente. Para além disso conseguirá visualizar todas as questões adicionadas e se assim o pretender eliminar a questão.

Tabela 3.4: Endpoints das ações do Recruiter

Método	URL	Descrição
POST	/Recruiter/CreateCandidate	Registrar novo candidato
POST	/Recruiter/AddExperience	Adiciona nova experiência ao candidato
GET	/Recruiter/CandidateExperience	Experiência do candidato
GET	/Recruiter/VerListaCandidatos	Possibilidade de ver lista de candidatos
GET	/Recruiter/VerPerfilCandidato	Possibilidade de ver perfil do candidato
POST	/Recruiter/GerarQuestionário	Criado Questionário
GET	/Recruiter/CriarQuestionario	Apresentado questionário
GET	/Recruiter/VerSurveyResults	Possibilidade de ver resultados do questionário

Na tabela 3.4 os endpoints de validação criados vão permitir aos Recruiter's criar novos candidatos, adicionar experiência a esses mesmo candidatos, vai ter possibilidade de ver a lista de

candidatos vem como o seu perfil onde estará toda a sua informação apresentada, experiência, questionários feitos e resultados. Para além disto conseguirá gerar questionários para determinado candidato.

3.4 Metodologias e Ferramentas Utilizadas

Nesta secção iremos apresentar a metodologia que irá ser utilizada durante o desenvolvimento do projeto, seguidamente iremos abordar as tecnologias que serão utilizadas.

3.4.1 Metodologia

Relativamente à metodologia que irá ser utilizada irá ser utilizada o SCRUM em conformidade com a empresa, esta é uma metodologia de desenvolvimento de software ágil como mostra é demonstrado na figura 3.1. Um dos principais objetivos utilizando esta metodologia é obter um produto rapidamente com a maior qualidade possível.



Figura 3.1: Ciclo da Metodologia Scrum

Foi decidido adotar esta metodologia visto que tornaria mais fácil a implementação, visto que assim o projeto iria ser dividido em ciclos com uma certa duração, chamados Sprints. Com isto

foram definidas as tarefas a dividir por cada ciclo o que tornou mais fácil a sua validação no fim de cada Sprint.[10]

3.4.2 Ferramentas Utilizadas

Abordando agora as ferramentas utilizadas para a construção da base de dados foi utilizada o Microsoft SQL Server 3.4.2.1, enquanto na implementação da plataforma foi utilizada a framework web de ASP.NET MVC 3.4.2.2.

3.4.2.1 Microsoft SQL Server

O Microsoft SQL Server é um sistema gerenciador de base de dados relacional, o que significa que a informação criada e gerada na plataforma vai ser armazenada em campos ou tabelas criadas, essa informação depois pode ser solicitada de várias maneiras, o que vai manter uma ligação entre a base de dados e a plataforma.[9]

3.4.2.2 ASP.NET MVC

O ASP.NET MVC é uma framework de aplicações WEB, que utiliza o padrão model-view-controller(MVC)[18].

Com esta framework é possível construir aplicações WEB através de 3 camadas lógicas são elas Model(Camada Lógica), View(Camada de apresentação) e o Controller(Camada de controlo). Seguidamente estas camadas serão explicadas: A View é responsável por controlar toda a interface, todo o conteúdo que será apresentado ao utilizador da plataforma. O Model é responsável por todas as ações que serão executadas através da camada de controlo(controller), isto quer dizer que se preocupa com o armazenamento, manipulação e geração de dados. O Controller fazem a junção da View com o Model, pois interpreta as ações de entrada, essas ações são enviadas para o Model e View onde irão ser realizadas as operações necessárias. Um diagrama exemplificando a relação entre estas camadas é apresentado na figura 3.2 , onde as linhas sólidas representam interações diretas, e as linhas tracejadas representam as interações indiretas. [1]

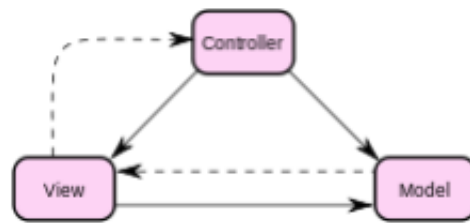


Figura 3.2: Diagrama do MVC

Capítulo 4

Implementação

Neste capítulo será apresentada a implementação usada para o desenvolvimento do projeto.

Inicialmente haverá a exposição da base de dados, abordando as tabelas criadas e a explicação dessas mesmas tabelas e as suas ligações.

A segunda parte é onde irá ser abordada a implementação da plataforma, demonstrando o trabalho feito e as ações disponíveis na plataforma criada.

Na ultima parte do capítulo irá ser discutido a forma como foi elaborado o Web Crawler e a explicação de como irá funcionar.

4.1 Base Dados

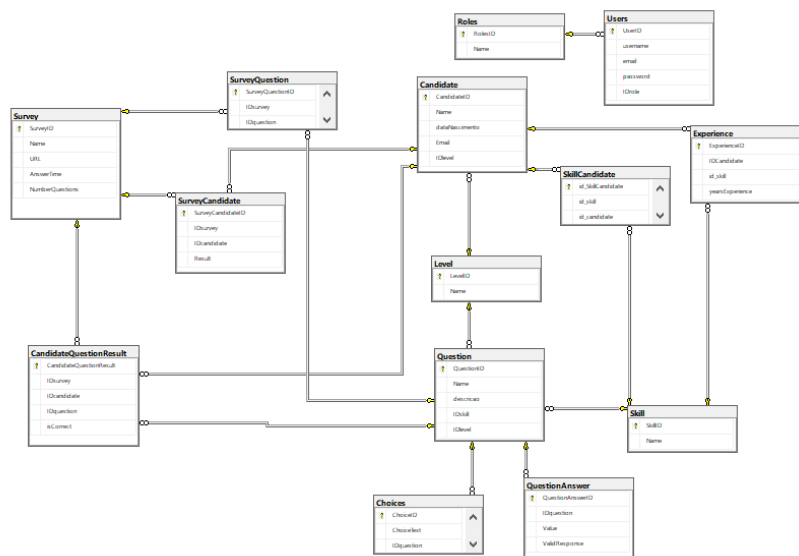


Figura 4.1: Diagrama da Base Dados

A base de dados foi implementada através do Microsoft SQL Server e abordada com a intenção de guardar toda a informação gerada pela plataforma WEB, para isso foi construída a base de dados apresentada na figura 4.1, todas as tabelas presentes nesta base de dados permitirão recolher e guardar toda a informação necessária para o funcionamento correto da plataforma, seguidamente iremos abordar a base de dados e clarificar as suas ligações, isso irá ser feito dividindo a base de dados e explicando por as ações presentes na plataforma.

4.1.1 Estrutura das tabelas Users e Roles

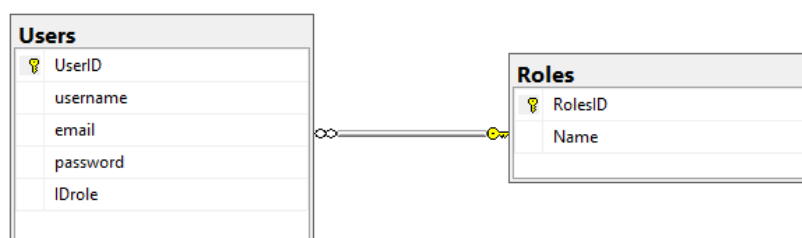


Figura 4.2: Diagrama das tabelas Users e Roles

Começaremos por abordar as tabelas Users e Roles presentes na figura 4.2, estas tabelas foram criadas de modo a armazenar todos os utilizadores criados, para cada um desses utilizadores será definida uma role que pode ser Admin, Publisher ou Recruiter. Inicialmente é necessário adicionar manualmente na tabela dos utilizadores um utilizador que irá ter a role de Admin, através dessa role conseguirá na plataforma registar novos Publisher's ou Recruiter's sendo que todos eles irão ficar armazenados na tabela Users com o devido atributo IDrole que permite fazer a ligação à tabela Roles de modo a identificar a Role pertencente ao utilizador. Dependendo da role do utilizador, este poderá fazer diferentes ações.

4.1.2 Estrutura das tabelas das Questões

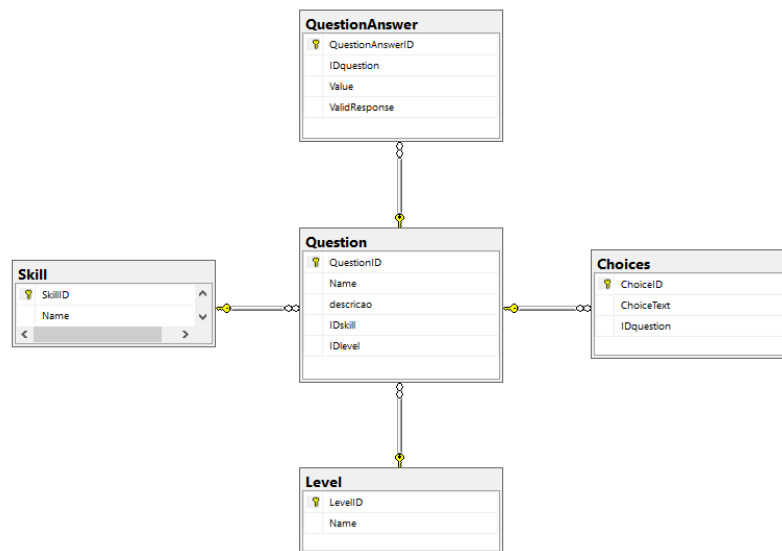


Figura 4.3: Diagrama da estrutura das Questões

Na figura 4.3, é apresentada a estrutura elaborada na base de dados para a criação da estrutura das questões que irão ser colocadas nos questionários gerados, como verificamos na figura 4.3, cada questão irá ter uma Skill e Level (Junior, Medium, Senior) associados, esta associação existe de modo a que quando o questionário for criado as questões que serão colocadas ao candidato estarão de acordo com as skills do candidato e o seu nível de experiência. As tabelas QuestionAnswer e Choices servem para a criação do questionário onde serão usadas para apresentar as escolhas possíveis para cada questão sendo que posteriormente irão ser validadas através do atributo ValidResponse presente na tabela QuestionAnswer sendo que se o valor for TRUE é porque a escolha está correta, se o valor escolhido for FALSE a escolha vai ser determinada como errada.

4.1.3 Estrutura das tabelas dos Candidatos

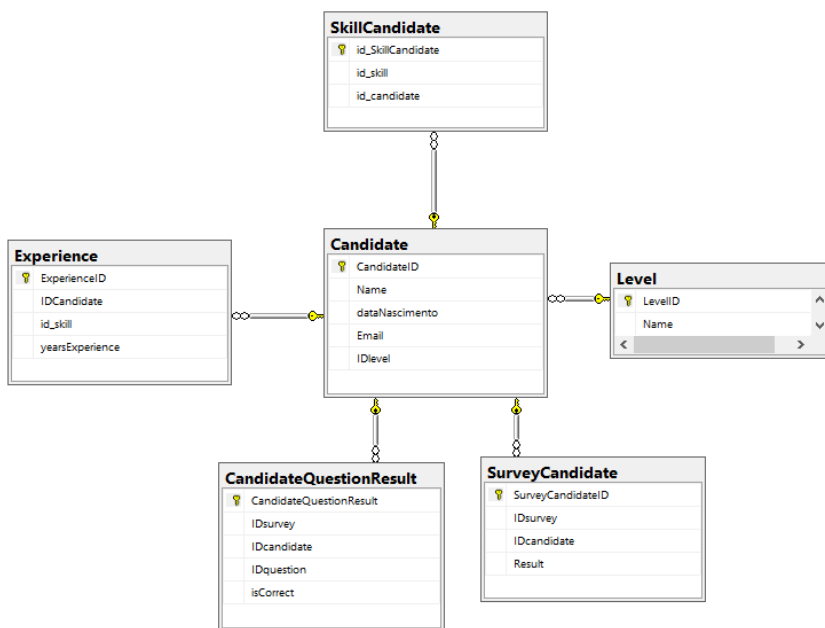


Figura 4.4: Diagrama da estrutura dos Candidatos

Através da figura 4.4 poderemos ver a estrutura elaborada para a criação do candidato e todas as tabelas necessárias. Analisando a estrutura do candidato verificamos pela tabela Candidate que é necessário usar novamente a tabela Level para associarmos um Level a cada candidato esse Level pode ser Junior, Medium ou Senior, isto permite fazer a associação com as questões que irão ser colocadas quando forem gerados os questionários, para além deste ponto verificamos através das tabelas SkillCandidate e Experience verificamos que é feita a associação à experiência do candidato tendo em conta as suas skills. A tabela SurveyCandidate serve para armazenar todos os questionários feitos pelo candidato, e o resultado obtido nesse mesmo questionário, daí surgiu a necessidade da criação da tabela CandidateQuestionResult para desta forma ser possível obter e guardar o resultado da escolha feita pelo candidato para cada questão respondida. Através desta estrutura é possível obter e guardar todas as informações relacionadas com os candidatos, como criar o questionário tendo em conta as suas skills e nível de experiência, como visualizar todos os questionários realizados.

4.1.4 Estrutura das tabelas dos Questionários

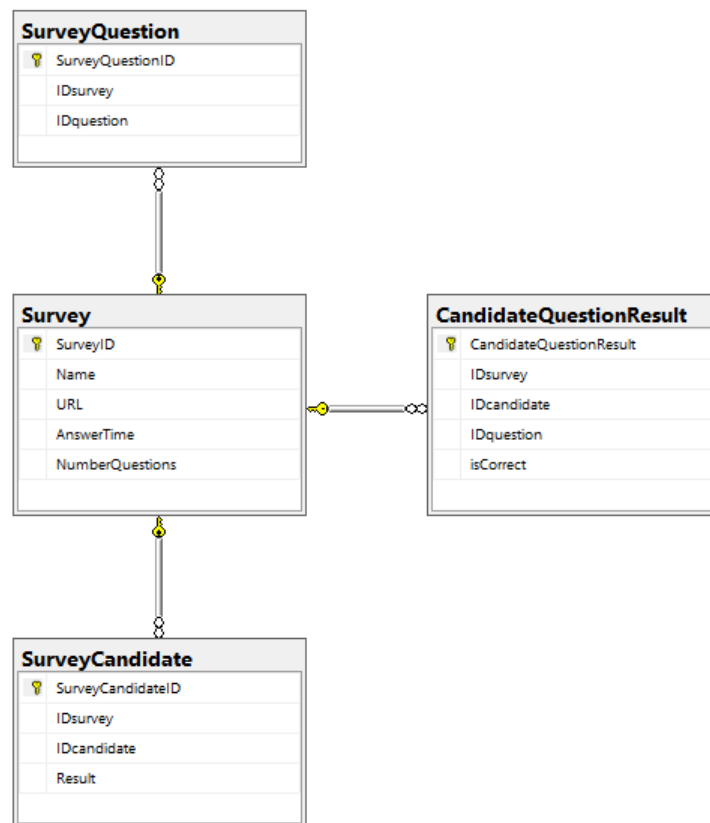


Figura 4.5: Diagrama das tabelas da estrutura dos Questionários

Analisando a figura 4.5 podemos ver a estrutura organizada para a geração dos questionários, vemos através da tabela SurveyQuestion quais as questões pertencentes ao questionário esta informação é gerada e guardada no momento em que o questionário é criado para um determinado candidato, essa informação irá ser guardada na tabela SurveyCandidate, através desta tabela ficará guardada a informação de que o questionário criado foi respondido pelo candidato escolhido, sendo que o objetivo é gerar questionários para um determinado candidato sendo que as questões colocadas são associadas às características do candidato, para além disso utilizamos a tabela CandidateQuestionResult para verificar o resultado de todas as questões colocadas ao candidato.

4.2 Plataforma WEB

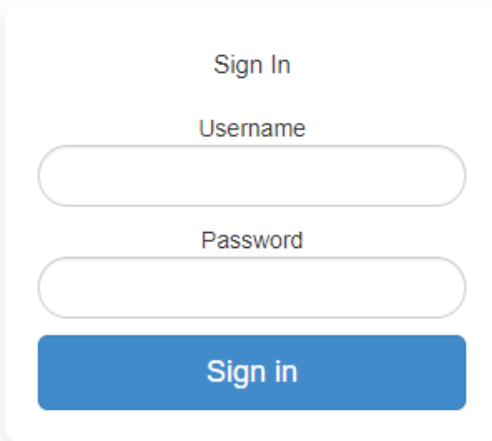
Nesta secção será apresentada a plataforma criada, esta plataforma cumpre todos os objetivos propostos no planeamento do projeto. O nome escolhido para a plataforma foi HardSurvey.

A análise e exposição da plataforma irá ser dividida em partes, devido à existência de diferentes roles, assim irão ser expostas as ações disponíveis para cada role existente na plataforma.

A plataforma foi implementada usando a framework ASP.NET MVC como foi referido em 3.4.2.2, sendo que para isso foram criados os diferentes controllers, cada controller está associado a um role.[2]

4.2.1 Ações Comuns

Começando pelas ações comuns existentes todas as roles conseguirão efetuar o login, como é apresentado na figura 4.6 após o utilizador fazer login, dependendo do role em que estiver inserido irá ter diferentes ações como será demonstrado.

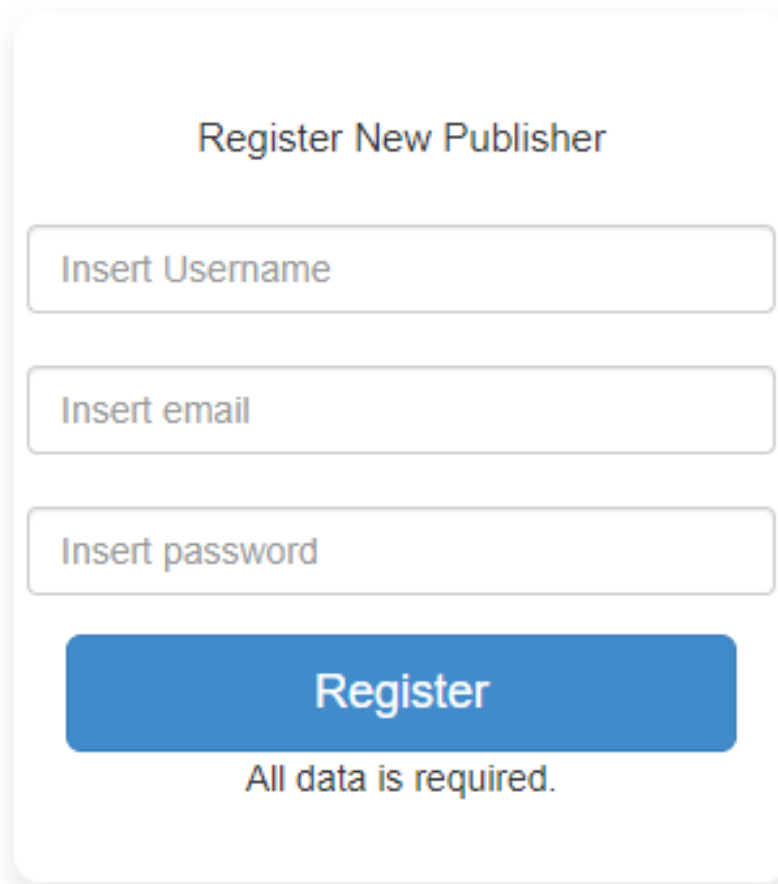


The image shows a login form with a white background and rounded corners. At the top, the text "Sign In" is centered. Below it are two input fields: "Username" and "Password", each with a light gray border and rounded ends. At the bottom of the form is a solid blue button with the text "Sign in" in white, centered.

Figura 4.6: GUI-Login da Plataforma HardSurvey

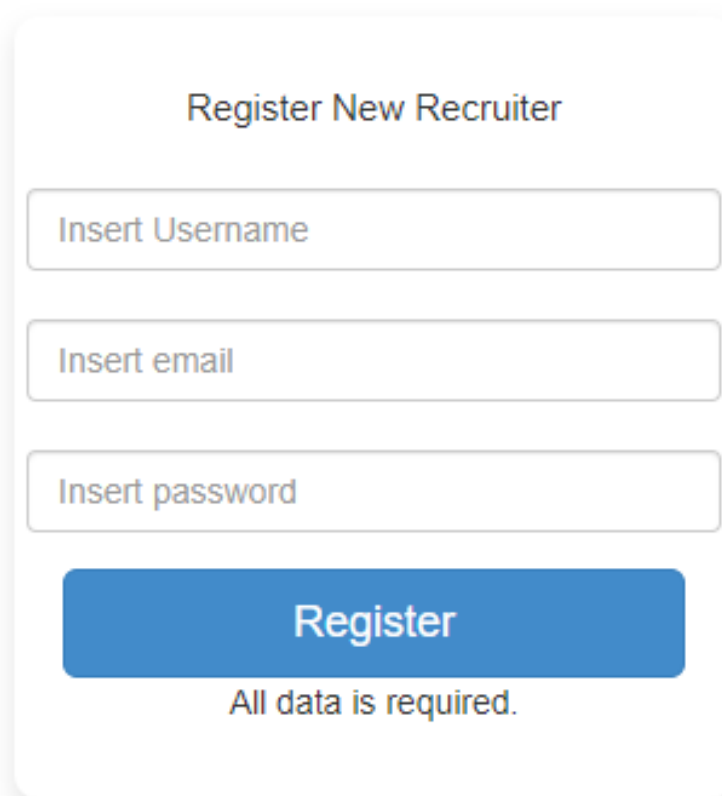
4.2.2 Ações do Administrador

Inicialmente irá ser necessário adicionar manualmente um utilizador com a role de Admin na Base de Dados, para assim este ganhar as permissões e poder efetuar as ações iniciais para o funcionamento da plataforma. Como dito anteriormente caso o utilizador tenha a role de Admin, irá ser capaz de executar as ações de registar publisher's ou recruiter's como é apresentado nas figuras 4.7 e 4.8 , a informação de registo de novos utilizadores irá ser guardada na base de dados. A única forma de registar novos utilizadores apenas é possível através de um utilizador com a role de Admin.



The image shows a registration form titled "Register New Publisher". It consists of three input fields: "Insert Username", "Insert email", and "Insert password". Below these fields is a blue "Register" button. At the bottom of the form, there is a message: "All data is required." The form is presented in a clean, white, rounded rectangular container.

Figura 4.7: GUI-Registro de Novo Publisher na Plataforma



Register New Recruiter

Insert Username

Insert email

Insert password

Register

All data is required.

Figura 4.8: GUI-Registo de Novo Recruiter na Plataforma

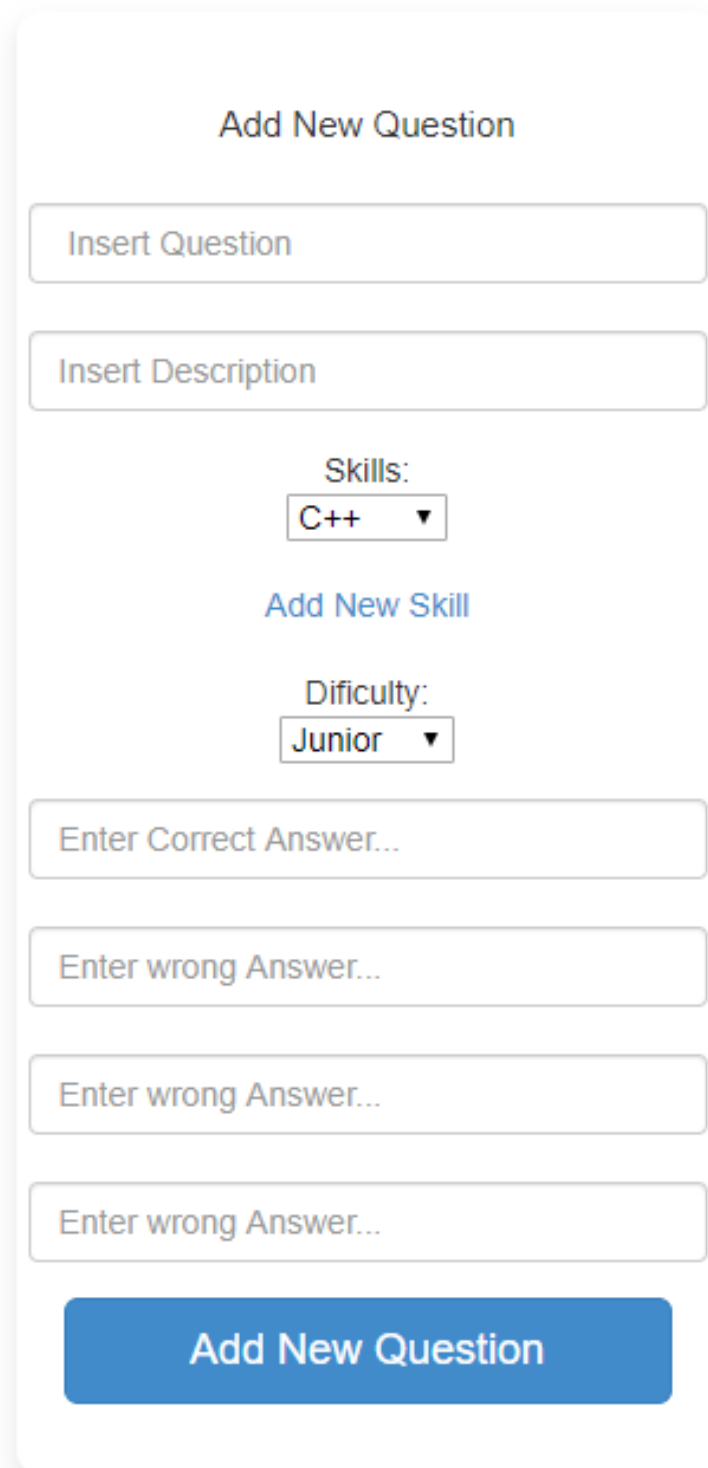
4.2.3 Ações do Publisher

Aqui irão ser apresentadas as ações disponíveis a serem usadas pelos utilizadores com a role de Publisher's, bem como o seu funcionamento.

O publisher é o utilizador que está responsável por adicionar novas questões e skills, sendo responsável também por rever as questões e se achar necessário fazer a sua eliminação. Inicialmente iremos abordar a forma como as questões são adicionadas, sempre que um Publisher pretender adicionar uma nova questão necessita obrigatoriamente de preencher qual a questão, as opções que essa questão irá ter, a skill que corresponde à questão a ser inserida, e o nível de dificuldade em que a questão está avaliada, sendo depois possível adicionar uma descrição da questão, sendo esta não obrigatória, como exemplifica a figura 4.9 .

Toda a informação adicionada é verificada antes de ser guardada na base de dados, visto que podem haver campos que não foram preenchidos ou existir já armazenada uma questão igual,

poderemos verificar a forma como a verificação e armazenamento é feito na figura , para além de ser criada a questão todas as suas opções são guardadas nas tabelas QuestionAnswer e Choices, que como foi referido em 4.1 pertencem à estrutura da questão.



The image shows a web form titled "Add New Question". It contains several input fields and dropdown menus. The fields are: "Insert Question", "Insert Description", "Skills:" with a dropdown menu showing "C++", "Add New Skill", "Difficulty:" with a dropdown menu showing "Junior", "Enter Correct Answer...", "Enter wrong Answer...", "Enter wrong Answer...", and "Enter wrong Answer...". At the bottom, there is a blue button labeled "Add New Question".

Figura 4.9: GUI-Registo de Nova Questão

```
if (String.IsNullOrEmpty(model.getName))
    ModelState.AddModelError("getName", "Required field");
else if (db.Users.Where(x => x.username == model.getName).Count() > 0)
{
    ModelState.AddModelError("getName", "Questions already exists.");
}

if (String.IsNullOrEmpty(model.getName))
    ModelState.AddModelError("correct", "Required field");

if (String.IsNullOrEmpty(model.getName))
    ModelState.AddModelError("wrong1", "Required field");

if (String.IsNullOrEmpty(model.getName))
    ModelState.AddModelError("wrong2", "Required field");

if (String.IsNullOrEmpty(model.getName))
    ModelState.AddModelError("wrong3", "Required field");

if (ModelState.IsValid)
{
    Question novo = new Question();
    novo.Name = model.getName;
    novo.descricao = model.getDescricao;
    novo.IDskill = model.getIDskill;
    novo.IDlevel = model.getIDlevel;
    db.Questions.InsertOnSubmit(novo);
    db.SubmitChanges();
}
```

Figura 4.10: Implementação da verificação e registo de nova questão na base de dados

O publisher além de adicionar as questões, poderá adicionar novas skills na base de dados, para fazer esta ação existem duas maneiras, uma é adicionar através da GUI criada para acrescentar novas skills, outra forma é utilizando uma janela pop-up que pode ser aberta quando o Publisher esta a adicionar uma nova questão, figura 4.11, para assim não ser necessário fazer a navegação para a página de adicionar uma nova skill. A skill inserida na GUI é verificada de modo a prevenir duplicações de skills, figura 4.12.

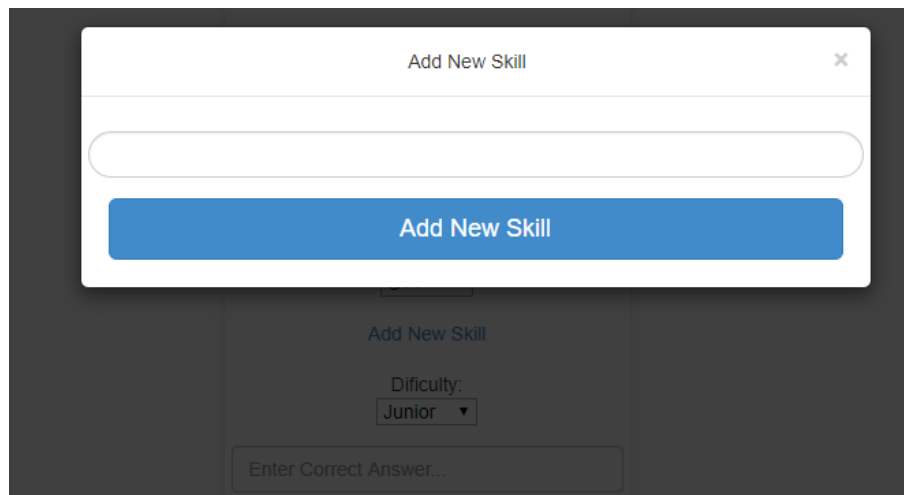


Figura 4.11: GUI-Pop-up adicionar nova skill

```

if (String.IsNullOrEmpty(skill))
    ModelState.AddModelError("skill", "Required field");
else if (db.Skills.Where(x => x.Name == skill).Count() > 0)
{
    ModelState.AddModelError("skill", "skill already exists.");
}

if (ModelState.IsValid)
{
    Skill novo = new Skill();
    novo.Name = skill;
    db.Skills.InsertOnSubmit(novo);
    db.SubmitChanges();
}

```

Figura 4.12: Implementação de verificação e armazenamento de nova skill na Base Dados

Para além do já referido, o Publisher poderá ver a lista de questões registadas e removê-las como mostra na figura 4.13.

Registered Questions			
Question	Description	Skill	Difficulty
Constant variables can be created in CPP by using _____.		C++	Junior Delete Question
_____ are used for generic programming		C++	Junior Delete Question
Which of the following converts a type to a signed byte type in C#?		C#	Medium Delete Question
Which of the following operator creates a pointer to a variable in C#?		C#	Junior Delete Question
What is the output of the following?	print("Hello {name1} and {name2}" format(name1='foo', name2='bin'))	Python	Junior Delete Question

Figura 4.13: GUI-Questões Registadas

4.2.4 Ações do Recruiter

Abordando as ações que poderão ser efetuadas pelo recruiter, o recruiter é o utilizador que ficará responsável por adicionar todos candidatos e todas as suas características que se encontram

na fase de recrutamento e gerar os questionários que serão feitos por cada um deles, esses questionários são gerados com base nas características de cada candidato.

Poderemos verificar através da figura 4.14, que o Recruiter tem acesso a uma lista de candidatos, sendo que pode aceder ao perfil de cada um dos candidatos registados, onde consegue verificar a experiência de cada candidato, os resultados obtidos em questionários feitos anteriormente, e gerar um novo questionário para esse mesmo candidato, um exemplo é mostrado na figura 4.15. Além de ter toda a informação do candidato consegue ver os detalhes de cada questionário, conseguindo assim verificar quais as questões que foram respondidas corretamente e as incorretas.

Ver Lista Candidatos

Name	Email	
Bruno Coelho	bruno@gmail.com	Ver Perfil
Nuno Ferreira Costa	nuno_costa@gmail.com	Ver Perfil

Figura 4.14: GUI-Lista dos Candidatos Registrado

Candidate Profile	
Name:	Bruno Coelho
Email:	brunocoelho@gmail.com
Birth Date:	31/03/1996 00:00:00

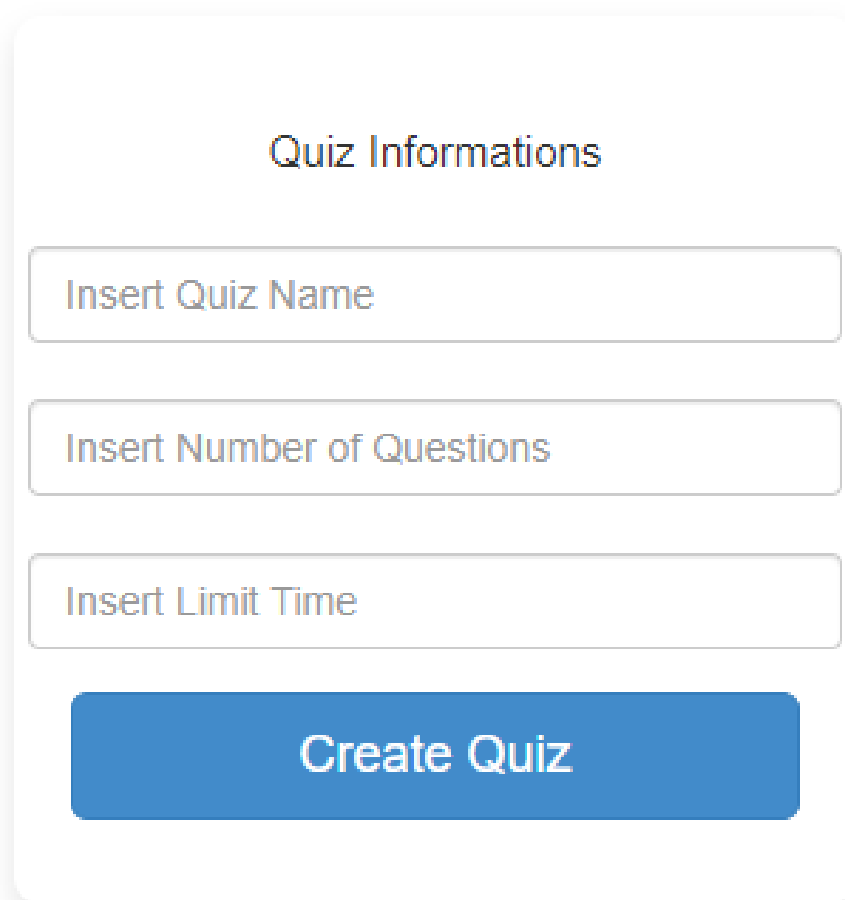
Skill	Years Experience
C#	1
C++	1
Python	1

Generate Quiz

Results	
Survey Name	Result

Figura 4.15: GUI-Perfil do Candidato

Relativamente à parte de gerar o questionário depois de o utilizador gerar questionário é deslocado para uma página, como mostra a figura 4.16, onde irá colocar os dados do questionário a ser criado, depois da informação ser preenchida é gerado o questionário, como exemplificado na figura 4.17. Para esta geração do questionário ser feita de acordo com as características de cada candidato foi necessário criar um método capaz de escolher apenas as questões que estivessem de acordo com o nível de experiência do candidato e as skills por ele praticadas, esse método é apresentado na figura 4.18.



The image shows a web form titled "Quiz Informations" with three input fields and a submit button. The first field is labeled "Insert Quiz Name", the second "Insert Number of Questions", and the third "Insert Limit Time". Below these fields is a blue button with the text "Create Quiz".

Quiz Informations

Insert Quiz Name

Insert Number of Questions

Insert Limit Time

Create Quiz

Figura 4.16: GUI-Informação necessária para gerar questionário

Time Left: 03:59 minutes!

Questions :

Q1. Constant variables can be created in CPP by using _____ .

- enum
- #define
- All of these
- const

Q2. _____ are used for generic programming.

- Templates
- Virtual Functions
- None of these
- Inheritance

Q3. Which of the following operator creates a pointer to a variable in C#?

- &
- sizeof
- *
- typeof

Figura 4.17: GUI-Questionário Gerado

```

Candidate cand= db.Candidates.Single(x => x.CandidateID == model.CandidateID);
var idlevelcand = cand.IDLevel;

List<int> Candidateskills = db.Experiences.Where(x => x.IDCandidate == model.CandidateID).Select(x=>x.id_skill).ToList();
List<int> Candidatelevel = db.Candidates.Where(x => x.CandidateID == model.CandidateID).Select(x => x.IDLevel.Value).ToList();

var questions =db. Questions.Where(x => Candidateskills.Contains(x.IDskill.Value)).Where(x => Candidatelevel.Contains(x.IDlevel.Value)).ToList();
var questionsRandom = questions.OrderBy(x => Guid.NewGuid());
IQueryable<QuestionVM> questions = null;

questions = questionsRandom.Take(model.NumberQuestions).Select(q => new QuestionVM
{
    QuestionID = q.QuestionID,
    Name = q.Name,
    Descricao = q.descricao,
    Choices = q.Choices.Select(c => new ChoiceVM
    {
        ChoiceID = c.ChoiceID,
        ChoiceText = c.ChoiceText
    }).OrderBy(x=>Guid.NewGuid()).ToList()
}).AsQueryable();

return View(questions);

```

Figura 4.18: Método criado para obter questões

Para obter o resultado do questionário, foi criada uma função AJAX, figura 4.19, onde é feita a correção de cada uma das questões colocadas no questionário, que é executada após o questionário ser submetido o que permite ao candidato verificar quais as questões que errou e para cada uma delas qual a resposta correta, como é exemplificado na figura 4.20, para além disso essa informação é enviada para o controller (RecruiterController) de modo a que toda a informação seja armazenada na base de dados.

```

$.ajax({
  type: 'POST',
  url: [url.Action("OriaQuestionario", "Recrutista")],
  data: { resultQuiz, candidateID, surveyID },
  success: function (response) {
    if (response.result.length > 0) {
      for (var i = 0; i < response.result.length; i++) {
        if (response.result[i].isCorrect == true) {
          $('#MsgQ' + i).html('<div class="alert alert-success" role="alert"><span class="glyphicon glyphicon-thumbs-up" aria-hidden="true">/span> Correct answer</div>');
        } else {
          $('#MsgQ' + i).html('<div class="alert alert-danger" role="alert"><span class="glyphicon glyphicon-thumbs-down" aria-hidden="true">/span> Incorrect answer - The Good Answer is <br>= response.result[i].Answer = </div>');
        }
        i++;
      }
    } else {
      alert('Something Wrong');
    }
    //console.log(response.result.length);
  },
  error: function (response) {
  }
});

```

Figura 4.19: Função AJAX

Questions :

Q1. Constant variables can be created in CPP by using _____ .

const
 enum
 All of these
 #define

Incorrect answer - The Good Answer is All of these

Q2. _____ are used for generic programming.

Inheritance
 Templates
 Virtual Functions
 None of these

Correct answer

Submit Quiz

Figura 4.20: GUI-Questionário Corrigido

4.3 Implementação Web Crawler

Para implementação do web crawler foi implementado um novo projeto onde foi usado o HtmlAgilityPack [7], que permite a extração da informação pretendida dos websites. Relativamente ao funcionamento do crawler, inicialmente é aberta uma janela da linha de comandos a pedir a informação referente à página, como o URL do website de onde se pretende retirar a informação e os elementos onde as questões e respetivas respostas se encontram de modo a percorrer todo o website e retirar a informação pretendida [19]. No fim é gerado um documento com toda a

informação retirada de modo a que o utilizador, neste caso um Publisher ver todas as questões retiradas e assim decidir se pretende adicionar essa questão na plataforma. Para retirar a informação pretendida foi criada uma classe apresentada na figura 4.21.[13]

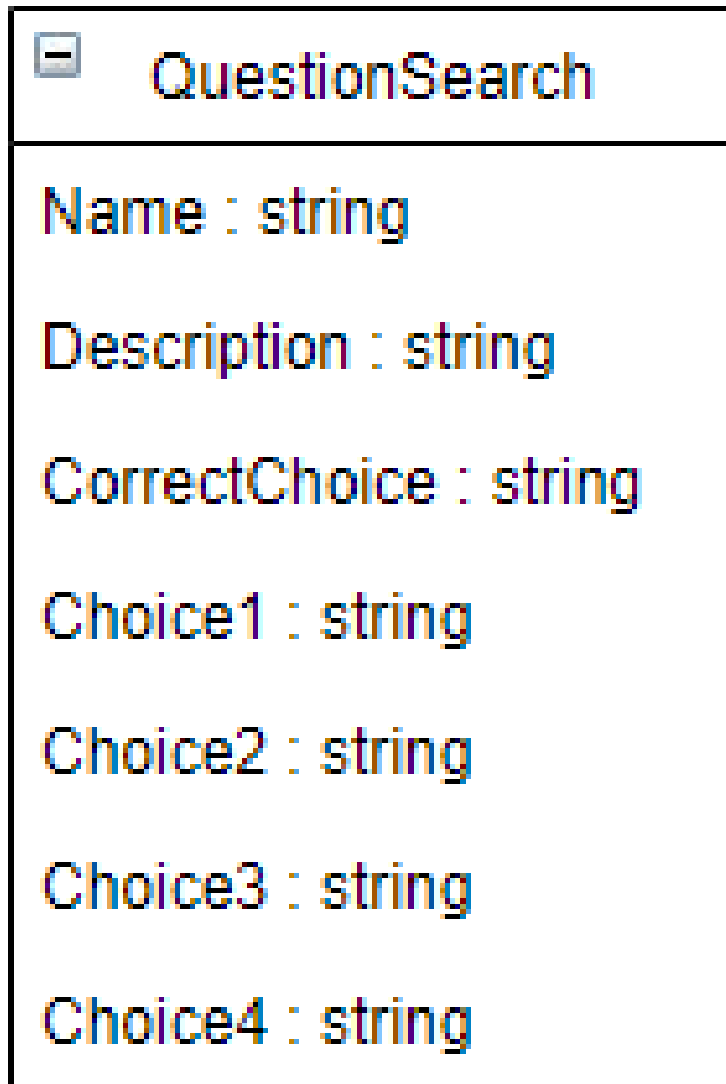


Figura 4.21: Classe QuestionSearch

Capítulo 5

Validação

A validação do projeto foi feita através do teste de vários cenários, para isso foram criados vários user stories de modo a ser possível validar as várias funcionalidades da plataforma.

5.1 User Stories

Para se realizar a validação da plataforma foram criadas as user stories, presentes na tabela 5.1. Seguidamente serão implementados os cenários de forma a validar todos os user stories.

Tabela 5.1: User Stories

#US	User Story
1	<i>Como Admin, eu quero registrar novos Publisher's/Recruiter's</i>
2	<i>Como Publisher, eu quero adicionar questões</i>
3	<i>Como Publisher, eu quero adicionar skills</i>
4	<i>Como Publisher, eu quero visualizar a lista de questões existentes</i>
5	<i>Como Recruiter, eu quero registrar novos candidatos</i>
6	<i>Como Recruiter, eu quero visualizar o perfil do candidato criado</i>
7	<i>Como Recruiter, eu quero adicionar uma nova experiência ao candidato</i>
8	<i>Como Recruiter, eu quero gerar um questionário para determinado candidato</i>
9	<i>Como Recruiter, eu quero verificar os resultados dos questionários</i>

5.2 Validação

Para efetuar a validação da plataforma foram criados vários cenários de modo a confirmar todos os user stories existentes na tabela 5.1. As user stories foram sendo validadas durante a implementação da plataforma, sendo que quando se obteve a plataforma final todas as user stories foram revistas e testadas novamente de modo a puderem ser validadas.[20]

5.2.1 Validação das User Stories do Administrador

Relativamente às user stories pertencentes ao administrador, a #US1, irá ser criado um cenário de registo de um novo Publisher ou Recruiter, apenas será efetuado um cenário visto que o registo para qualquer um deles é semelhante.

Inicialmente o Admin necessita de escolher que pretende registar como é mostrado na figura 5.1, depois de selecionar o registo que pretende irá ser apresentada a GUI, figura 4.8, com todos os dados necessários para efetuar o registo de um novo utilizador, são esses username, email e password. Para este cenário iremos registar um novo Recruiter, iremos utilizar o Username:Recruiter123, email:recruiter@outlook.com e a password:recruiter147, figura 5.2, seguidamente o Admin necessita de clicar no botão "Register" de modo a ser criado um novo Recruiter, que poderá efetuar o seu login e executar as ações possíveis.

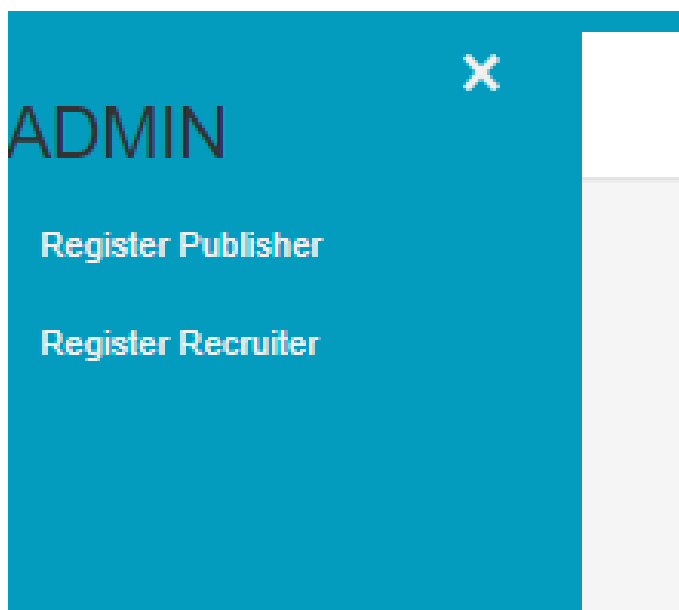
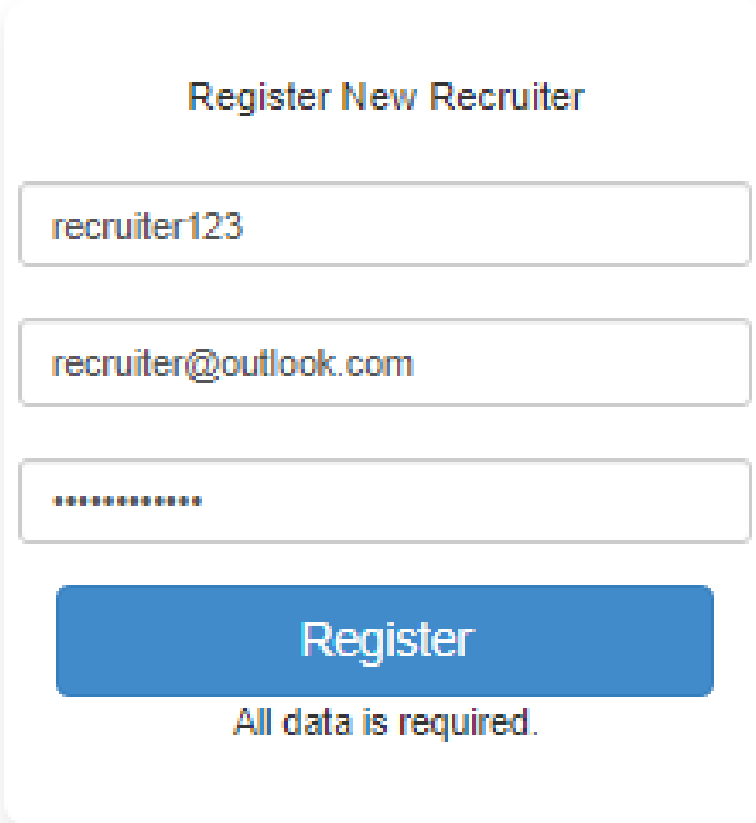


Figura 5.1: GUI-Menu de Opções do Administrador



Register New Recruiter

recruiter123

recruiter@outlook.com

.....

Register

All data is required.

Figura 5.2: GUI-Registro de Novo Recruiter

Através deste cenário foi possível validar a user storie referente ao administrador sendo esta baseada no registo de novos utilizadores com o role de recruiter ou publisher.

5.2.2 Validação das User Stories do Publisher

Para efetuar a validação das user stories referentes ao publisher foi criado um cenário que possibilita a validação de todos.

Neste cenário depois de ser efetuado o login pelo Publisher ele tem diversas ações que pode efetuar, figura 5.3, sendo que o utilizador seleciona "Add Question", será apresentada a GUI, figura 4.9, onde a informação pedida é a questão, descrição sendo que esta não é obrigatória, skill, nível de dificuldade, e as escolhas possíveis para a questão. Neste cenário iremos preencher os dados pedidos como é mostrado na figura 5.4, depois dos dados estarem todos preenchidos o Publisher clica no botão "Add New Question", o que leva a que a questão seja armazenada na base de dados e possa ser usada nos questionários. Depois é possível visualizar todas as questões registadas abrindo menu de ações, figura 5.3 e escolhendo a opção "See Registered Questions", onde será apresentada a GUI, figura 4.13 onde podem ser visualizadas todas as questões, e onde podem

ser apagadas bastando apenas clicar em "Delete Questions", através do cenário apresentado foi possível validar as user stories US2 e US4, referentes aos publisher's

Relativamente ao cenário de adicionar uma nova skill o Publisher no menu das suas opções escolhe a opção "Add Skill" ou insere uma nova questão através da GUI de adicionar nova questão onde surge uma pop-up para adicionar uma nova skill, o cenário aqui criado é através do menu de opções sendo que depois de ser seleccionado pelo utilizador(Publisher), é preenchido de acordo com a figura 5.5, depois é clicado o botão "Register" que permite que a skill seja armazenada, com este cenário é satisfeito o US3.

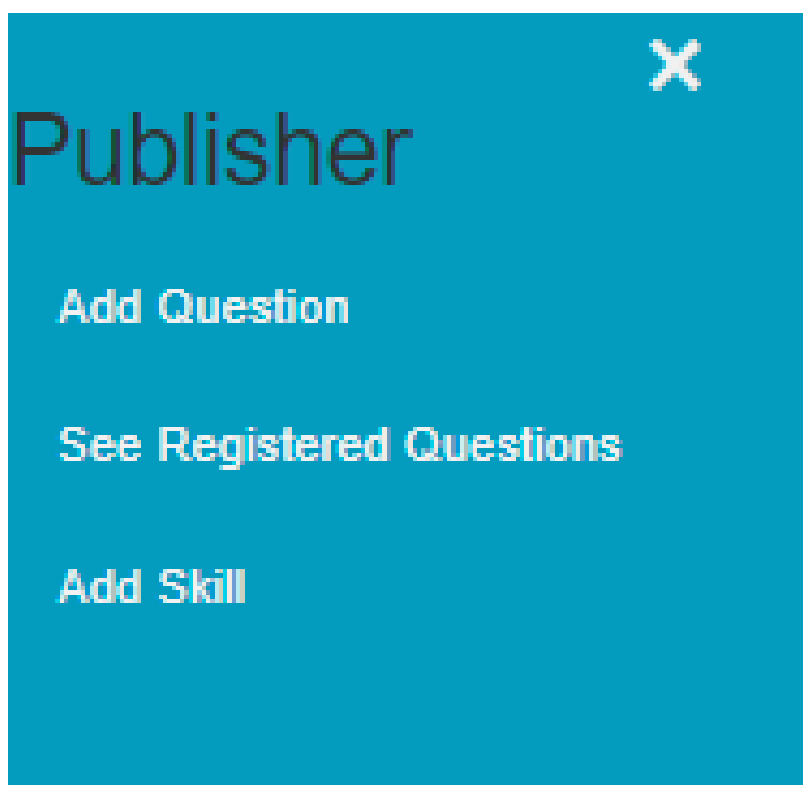


Figura 5.3: GUI-Menu de Opções do Publisher

Add New Question

What is correct syntax for main method of

Insert Description

Skills:

JAVA ▼

[Add New Skill](#)

Difficulty:

Junior ▼

public static void main(String[] args)

public static int main(String[] args)

public int main(String[] args)

None of these

Add New Question

Figura 5.4: GUI-Preenchimento Nova questão

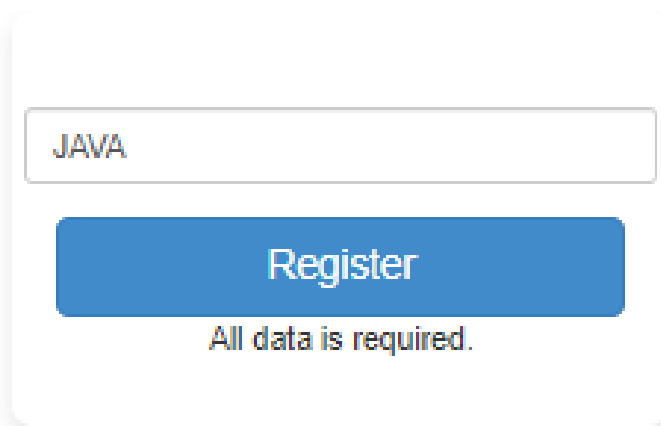


Figura 5.5: GUI-Preenchimento Nova Skill

Com os cenários foi possível validar todas as user stories referentes aos publishers.

5.2.3 Validação dos User Stories do Recruiter

Para validar as user stories do Recruiter, vai ser criado um cenário que consiste em adicionar um novo candidato, ver o perfil desse candidato, gerar um questionário para o candidato criado, e ver os resultados relativos a esse questionário.

Inicialmente mostrando o menu de opções do Recruiter, figura 5.6, sendo que o Recruiter seleciona a opção "Add Candidate", que será preenchida de acordo com a figura 5.7, depois de clicar no botão "Register" é automaticamente apresentado o perfil do candidato criado, figura 5.8, embora é possível acessar a esse perfil usando o menu de opções e a opção "See List of Candidates" onde são apresentados todos os candidatos registrados, figura 4.14. Depois de visualizar o perfil do candidato é necessário adicionar as experiências do candidato, clicando em "Add Experience" no perfil do candidato, onde será preenchida a informação pedida de acordo com a figura 5.9, depois de ter o perfil do candidato completo o Recruiter poderá gerar um questionário para o candidato escolhido carregando em "Generate Quiz", onde será ter de preencher todos os dados referentes ao questionário, figura 5.10. Depois de ser gerado e respondido pelo candidato, o resultado geral é apresentado no perfil do candidato, sendo que é possível verificar o resultado de cada questão pertencente ao questionário efetuado, como é mostrado na figura 5.11.

Através do cenário descrito todas as user stories referentes ao Recruiter foram validadas com sucesso.

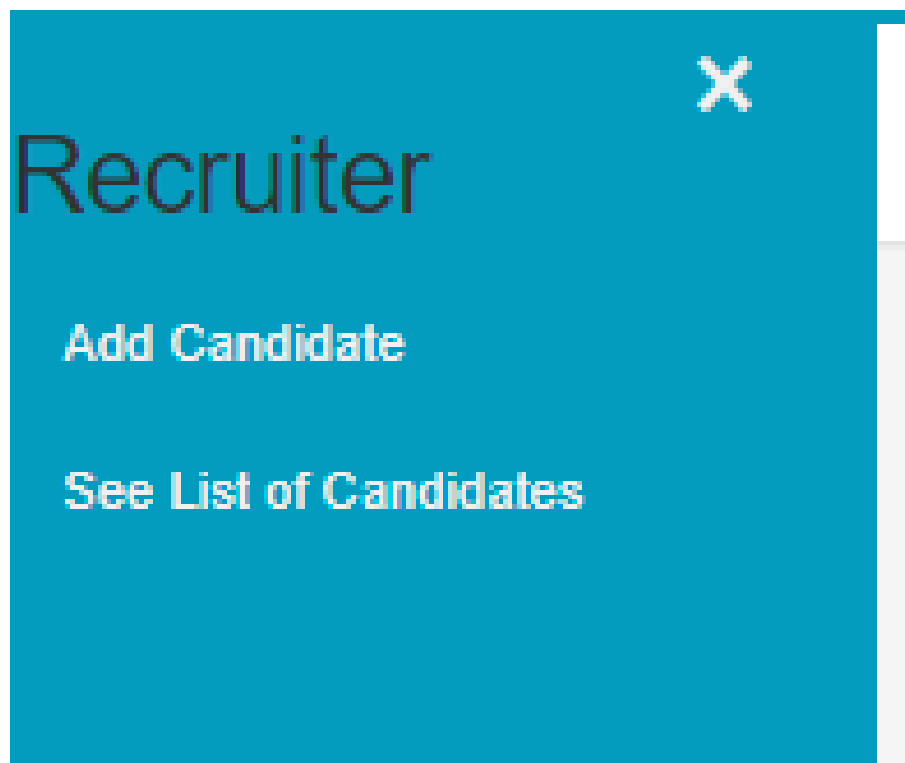


Figura 5.6: GUI-Menu de Opções do Recruiter

Registrar Novo Candidato

Nuno Ferreira Costa

nuno_costa@gmail.com

Data de Nascimento:

15/05/1990

Level:

Medium ▼

Register

All data is required.

Figura 5.7: GUI-Registo de Novo Candidato

Candidate Profile
Name: Nuno Ferreira Costa
Email: nuno_costa@gmail.com
Birth Date: 15/05/1990 00:00:00

Experience(Add Experience)

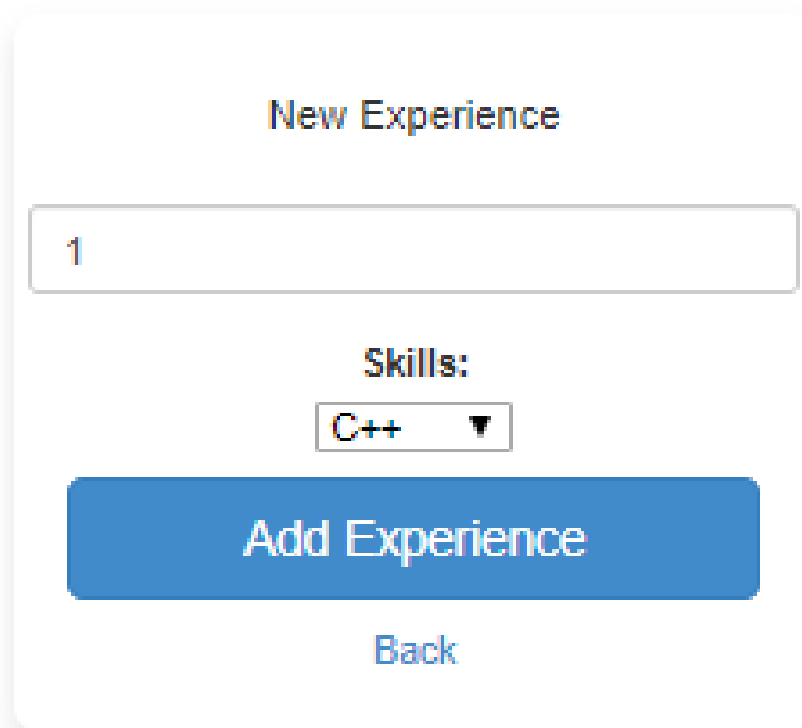
Skill	Years Experience

Generate Quiz

Results

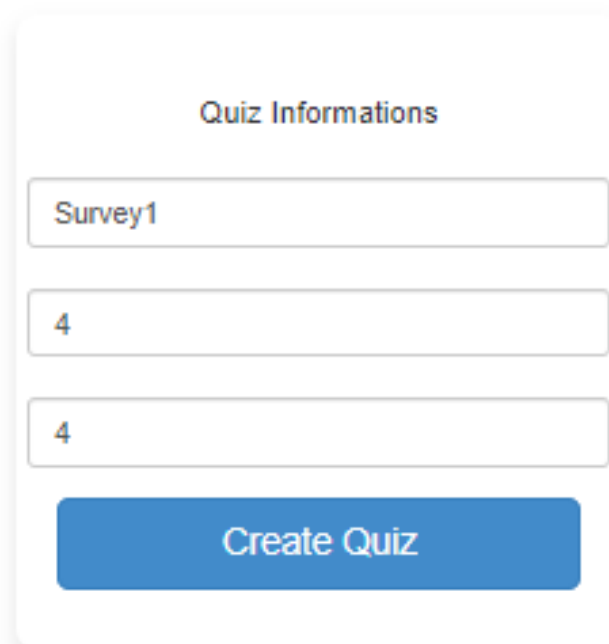
Survey Name	Result
-------------	--------

Figura 5.8: GUI-Perfil Candidato Criado




The screenshot shows a web form titled "New Experience". At the top, the title "New Experience" is displayed in a bold, black font. Below the title is a text input field containing the number "1". Underneath the input field is the label "Skills:" in a bold, black font. Below the label is a dropdown menu with "C++" selected and a downward-pointing arrow. At the bottom of the form is a large blue button with the text "Add Experience" in white. Below the button is a blue link labeled "Back".

Figura 5.9: GUI-Adicionar nova experiência



The screenshot shows a web form titled "Quiz Informations". At the top, the title "Quiz Informations" is displayed in a bold, black font. Below the title are three text input fields. The first input field contains the text "Survey1". The second and third input fields both contain the number "4". At the bottom of the form is a large blue button with the text "Create Quiz" in white.

Figura 5.10: GUI-Informação do questionário



Questions	Question Result
Constant variables can be created in CPP by using _____ .	Wrong
_____ are used for generic programming.	Wrong
Constant variables can be created in CPP by using _____ .	Wrong
_____ are used for generic programming.	Correct

Figura 5.11: GUI-Resultados das questões

5.3 Conclusão

Durante este capítulo foram encenados vários cenários de modo a ser possível fazer a validação de toda a plataforma, todos estes cenários foram verificados em conjunto com a empresa Winprovit, através deles podemos afirmar que a plataforma cumpre com todos os objetivos propostos inicialmente como é referido em [1.4](#). [5]

Capítulo 6

Conclusões e trabalho futuro

6.1 Satisfação dos objetivos

Relativamente aos objetivos propostos em 1.4 de modo a combater os problemas identificados, foi implementada uma solução através de uma plataforma que gera questionários de recrutamento, foi implementada para tornar possível a avaliação das capacidades dos candidatos para determinado cargo, os questionários gerados estarão sempre de acordo com o nível de experiência e as skills do candidato, esta associação foi obtida através da classificação por dificuldade das questões colocadas.

Outro dos objetivos propostos foi a implementação de um Web Crawler, este objetivo foi concluído através da implementação de uma solução capaz de retirar informação bastando apenas fornecer à aplicação do web crawler um link, e identificar os elementos em que a informação que se pretende retirar se encontram.

A validação da plataforma foi efetuada em conjunto com a empresa Winprovit que testou todas as funcionalidades, e aprovando todo o projeto, cumprindo os objetivos inicialmente propostos.

6.2 Trabalho Futuro

Este projeto contribui para o início de uma plataforma de recrutamento que poderá ser utilizada por várias empresas. Como trabalho futuro poderá ser implementado um algoritmo de avaliação automática das questões, adicionar questionários de escrita de código, e um melhoramento na implementação da solução do web crawler.

6.2.1 Algoritmo Avaliação Automática das Questões

Abordando este ponto que poderá ser trabalhado no futuro, com este algoritmo irá ser possível ajustar automaticamente a dificuldade das questões, para isto iria ser necessário a implementação de um algoritmo que reuni-se a percentagem de acerto das questões, através dessas percentagens iria ser feito um ajuste para um nível de dificuldade mais elevado ou mais pequeno. Como dito anteriormente este ajuste poderia ser feito através da percentagem de acerto, por exemplo caso uma questão tenha uma percentagem de acerto acima dos 90% e a questão estar avaliada com o nível de dificuldade elevado, tendo em conta o grande acerto essa questão poderia ser alterada para um nível mais baixo, o mesmo para o contrário do exemplo dado.

6.2.2 Escrita de Código

Outro ponto que poderá ser melhorada é adicionar questionários que além de conterem questões teórico-práticas como os que são apresentados neste projeto, adicionar além dessas, questões em que o candidato necessita-se de implementar código que iria ser corrigido utilizando um algoritmo implementado, de modo a ser possível avaliar mais capacidades relacionadas com o candidato.

6.2.3 Melhoramento Web Crawler

Para existir um melhoramento e facilidade de obter questões automaticamente poderia ser implementadas mais funcionalidades ao Web Crawler já implementado, algumas dessas soluções poderá ser o facto de tornar possível o fornecimento de vários links e os seus respetivos elementos, numa GUI mais completa e simples, outro ponto é tornar a execução do bot do web crawler automática sendo que já estariam definidos os elementos para cada URL adicionado.

Referências

- [1]
- [2] Asp.net mvc pattern. Disponível em <https://dotnet.microsoft.com/apps/aspnet/mvc>, Acedido a 15 de Fevereiro 2019.
- [3] Coderpad. Disponível em <https://coderpad.io/>, Acedido a 06 de Dezembro 2018.
- [4] Codility. Disponível em <https://www.codility.com/>, Acedido a 10 de Dezembro 2018.
- [5] Gui tests. Disponível em <https://www.qfs.de/en/qf-test/questions-before-gui-test-automation.html>, Acedido a 18 de Maio 2019.
- [6] Hackerrank. Disponível em <https://www.hackerrank.com/>, Acedido a 06 de Dezembro 2018.
- [7] Html agility pack. Disponível em <https://html-agility-pack.net/>, Acedido a 20 de Abril 2019.
- [8] Mrinetwork. Disponível em <http://www.mrinetwork.com/>, Acedido a 25 Novembro 2018.
- [9] Ms sql server. Disponível em https://www.tutorialspoint.com/ms_sql_server/ms_sql_server_overview.htm, Acedido a 10 de Janeiro 2019.
- [10] Scrum. Disponível em <https://www.desenvolvimentoagil.com.br/scrum/>, Acedido a 10 de Janeiro 2019.
- [11] Surveyjs. Disponível em <https://surveyjs.io/>, Acedido a 18 de Dezembro 2018.
- [12] Web crawler architecture. Disponível em <https://www.microsoft.com/en-us/research/publication/web-crawler-architecture/>, Acedido a 20 de Janeiro 2019.
- [13] Web crawling tutorial. Disponível em <https://medium.com/@thepen0411/web-crawling-tutorial-in-c-48d921ef956a>, Acedido a 18 de Abril 2019.
- [14] What is a web crawler? Disponível em <https://www.techopedia.com/definition/10008/web-crawler>, Acedido a 14 Dezembro 2018.
- [15] REDAÇÃO GLOBAL AD. O que é crawler? Disponível em <http://globalad.com.br/blog/o-que-e-crawler/>, Acedido a 26 de Dezembro 2018.

- [16] BetterEvaluation. Specify the key evaluation questions. Disponível em https://www.betterevaluation.org/en/rainbow_framework/frame/specify_key_evaluation_questions, Acedido a 30 de Dezembro 2018.
- [17] Floyd J. Fowler. Improving survey questions: Design and evaluation. Disponível em https://books.google.pt/books?hl=pt-PT&lr=&id=BZBB9NoqICYC&oi=fnd&pg=PA1&dq=IT+evaluation+questions&ots=W5aVOiCWki&sig=8IkUr78YNqp3aMN5UYi9514-sa0&redir_esc=y#v=onepage&q=IT%20evaluation%20questions&f=false, Acedido a 04 de Janeiro 2019.
- [18] Higor. Introdução ao padrão mvc. Disponível em <https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>, Acedido a 10 de Janeiro 2019.
- [19] Dr. G. P. Saradhi Varma b N. K. Kameswara Rao a. Architecture of a webcrawler. Disponível em https://www.researchgate.net/publication/313443438_Architecture_of_a_WebCrawler, Acedido a 25 de Janeiro 2019.
- [20] Lawal Abdulateef Olawale. Documentação ágil - user stories. Disponível em <https://medium.com/@tboschese/documenta%C3%A7%C3%A3o-%C3%Algil-user-stories-7c18f41a35e0>, Acedido a 18 de Maio 2019.
- [21] William Shadish. Some evaluation questions. Disponível em <https://pareonline.net/getvn.asp?v=6&n=3>, Acedido a 15 de Dezembro 2018.
- [22] Tom Winter. Hack the process of recruiting programmers. Disponível em <https://devskiller.com/doc/Hack-The-Process-Of-Recruiting-Programmers-Ebook.pdf>, Acedido a 25 Novembro 2018.