



IView – Aplicação de Entrevistas e Currículos

Rui Pedro Franco
Diogo Mendes Aires

Orientadores

Paula Graça
Frederico Ferreira

Relatório final realizado no âmbito de Projeto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2017/2018

Julho de 2018

Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

IView – Aplicação de Entrevistas e Currículos

40278 Diogo Mendes Aires

36210 Rui Pedro Gama Franco

Orientadores: Paula Graça, ISEL
Frederico Ferreira, DoItLean

Relatório final realizado no âmbito de Projeto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2017/2018

Julho de 2018

Resumo

O projeto IView, tem como objetivo principal desenvolver uma plataforma de contratações para oferta de empregos na área de tecnologias de informação, para ser usado tanto pelas empresas contratadoras como por quem procura emprego. Um dos objetivos deste projeto é dar às empresas um processo de contratação mais interativo e informativo entre Empresa-Candidato.

Para as empresas, a aplicação disponibiliza uma administração simples dos aspetos associados a uma nova contratação (e.g. entrevistas, candidatos, projetos, cliente e vagas). Um dos pontos fortes desta plataforma é também o facto de apresentar nas ofertas de emprego, quais as aptidões que mais se adequam à posição que procuram, facilitando assim o processo de seleção.

Por outro lado, a plataforma pretende proporcionar aos candidatos, uma nova forma (atualizada segundo os modelos das tecnologias recentes), de administrar as suas informações e oferecer candidaturas a vagas. Oferece também a possibilidade de os clientes disponibilizarem todas as informações que acharem relevantes, como por exemplo, a disponibilidade de horários para entrevistas, entre outras.

A plataforma é composta por duas vertentes: *Mobile* e *Web*, sendo que a vertente *Mobile* vai disponibilizar notificações ao candidato sobre o estado dos vários processos, nos quais está envolvido, bem como a disponibilização de informações relevantes sobre os mesmos; a vertente *Web*, além de disponibilizar as mesmas funções da aplicação *Mobile*, disponibiliza também as ofertas de emprego e gestão de candidaturas.

Durante o processo de desenvolvimento deste projeto, foram colocados em prática os nossos conhecimentos da arquitetura escolhida para a sua implementação, a plataforma *OutSystems*, cujo o conhecimento é cada vez mais requisitado no mercado informático e tecnológico de hoje, tanto nacional como estrangeiro, o que torna qualquer experiência sobre a mesma uma mais valia para o futuro.

Palavras-chave: Candidatos; Candidaturas; Currículos; Emprego; Entrevistas; *WebApp*; *MobileApp*; *OutSystems*.

Índice

RESUMO	V
LISTA DE FIGURAS	IX
LISTA DE CÓDIGOS	XIII
1. INTRODUÇÃO	1
1.1. ENQUADRAMENTO	1
1.2. OBJETIVOS	1
1.3. ORGANIZAÇÃO DO DOCUMENTO	2
2. FORMULAÇÃO DO PROBLEMA	3
2.1. ESTADO DA ARTE	3
2.2. DESCRIÇÃO DO PROJETO	4
2.2.1. <i>Aplicação web</i>	5
2.2.2. <i>Aplicação mobile</i>	10
2.3. DESCRIÇÃO DA PLATAFORMA, MODELOS DE DESENVOLVIMENTOS	11
2.3.1. <i>Servidores Fundamentais</i>	11
2.3.2. <i>Sincronismo</i>	12
2.4. <i>Segurança de Informação</i>	12
3. SOLUÇÃO PROPOSTA	15
3.1 ARQUITETURA DO PROJETO	15
3.2 MODELO ENTIDADE-ASSOCIAÇÃO DA BASE DE DADOS	16
3.2.1 <i>Utilizadores</i>	16
3.2.2 <i>Vagas</i>	18
3.2.3 <i>Eventos</i>	19
3.3 WIREFRAMES DO PROJETO	20
3.4 DESENVOLVIMENTO WEB	31
3.4.1 <i>Listagem</i>	31
3.4.3 <i>Adicionar e Remover do Profile</i>	33
3.4.4 <i>Introduzir novo candidato</i>	33
3.4.5 <i>Verificar Eventos</i>	34
3.4.6 <i>Adicionar Evento</i>	37
3.4.7 <i>Adicionar Oferta</i>	38
3.4.8 <i>Verificar percentagens de vaga</i>	39
3.4.9 <i>Pesquisa de ofertas</i>	40
3.4.10 <i>Continuação da candidatura</i>	41
3.4.11 <i>Incluir disponibilidade</i>	43

3.5	DESENVOLVIMENTO MOBILE	44
3.5.1	<i>Notificações</i>	44
3.5.2	<i>Sincronismo</i>	45
4	AVALIAÇÃO EXPERIMENTAL	47
4.1	CANDIDATURA ESPONTÂNEA – UTILIZADOR NÃO REGISTRADO	47
4.2	ESTABELECEER / RECUSAR NOVOS CANDIDATOS – COLABORADOR	48
4.3	ESTABELECEER NOVO CANDIDATO – UTILIZADOR NÃO REGISTRADO.....	49
4.4	AUTENTIFICAÇÃO DE UTILIZADOR – UTILIZADOR	50
4.5	ATUALIZAR CURRÍCULO – CANDIDATO.....	51
4.6	ATUALIZAR PROFILE – CANDIDATO	53
4.7	ATUALIZAR DISPONIBILIDADE – CANDIDATO	55
4.8	ESTABELECEER EMPRESA – ADMINISTRADOR.....	56
4.9	INTRODUZIR PROJETO – COLABORADOR	57
4.10	INTRODUZIR FORM – COLABORADOR	59
4.11	INTRODUZIR OFERTA – COLABORADOR.....	60
4.12	PESQUISAR/APLICAR A OFERTA – CANDIDATO.....	63
4.13	CONTINUAR/ACABAR CANDIDATURA – COLABORADOR	65
4.14	MARCAR ENTREVISTA – COLABORADOR	68
5.	CONCLUSÕES E TRABALHO FUTURO.....	71
	REFERÊNCIAS	73
A.1	MODELOS DE DADOS	75

Lista de Figuras

Figura 1 – Visão geral do IView.....	5
Figura 2 - Casos de utilização, Utilizadores Não Registados.	6
Figura 3 - Casos de utilização, Colaborador 1.....	7
Figura 4 - Casos de utilização, Colaborador 2.....	8
Figura 5 - Casos de utilização, Candidato.	9
Figura 6 - Casos de utilização, Gerente.	10
Figura 7 - Casos de utilização, Candidato Mobile.....	10
Figura 8 - Arquitetura OutSystems [5].	11
Figura 9 - Estrutura 4 Layer Canvas.....	15
Figura 10 - Modelo EA, Candidatos.....	17
Figura 11 - Modelo EA, Vagas.....	18
Figura 12 - Modelo EA, Eventos.....	20
Figura 13 - Frames, Home Page	21
Figura 14 - Frames, Menu de Colaboradores.	21
Figura 15 - Frames, Menu de Candidatos.....	22
Figura 16 - Frame, MoreInfo Values Tab.....	22
Figura 17 - Frame, MoreInfo Patnerships Tab.	23
Figura 18 - My Curriculum, Show.....	23
Figura 19 - Frame, MyCurriculum, Edit.....	24
Figura 20 - Frames, MyProfile.	24
Figura 21 - Frames, MyProfile Pop-up.....	25
Figura 22 - Frames, Forms.....	25
Figura 23 - Frames, Applications.	26
Figura 24 - Frames, Vacancies.	27
Figura 25 - Frames, Events.....	27
Figura 26 - Frames, Candidate.....	28
Figura 27 - Frames, Vacancy General Information Tab.	28
Figura 28 - Frames, Vacancy Steps Tab.	29
Figura 29 - Frames, Vacancy Tools and Languages Tab.....	29
Figura 30 - Frames, FormEditAdd.....	30
Figura 31 - Frames, VacancyAdd Form Chosse Pop-up.	30
Figura 32 - Candidatura Espontânea.....	47
Figura 33 - Introdução de SpontaneousCurriculum e de SpontaneousCurriculumFile	48
Figura 34 - Candidaturas Espontâneas.....	48
Figura 35 - Email de submissão da candidatura espontânea.....	48

Figura 36 - Mensagem de erro e de sucesso da candidatura.....	48
Figura 37 - Alteração sobre SponatnousCurriculum	49
Figura 38 - Email de estabelecimento do candidato	49
Figura 39 - Email de recusa da candidatura espontânea	49
Figura 40 - Criação do novo utilizador	49
Figura 41 - Estabelecimento de User.....	50
Figura 42 - Estabelecimento de CandidateCurriculum	50
Figura 43 - Email com novo utilizador	50
Figura 44 - Mensagens de erro	50
Figura 45 - Autentificação Web.....	50
Figura 46 - Autentificação Mobile.....	51
Figura 47 - Alterar Currículo, Web	51
Figura 48 - Alterar Currículo, Mobile	52
Figura 49 – Alterações a CandidateCurriculum.....	52
Figura 50 – Currículo depois da primeira alteração.....	52
Figura 51 - Currículo depois da segunda alteração.....	53
Figura 52 - Introduzir Tecnologia.....	53
Figura 53 - Introduzir Projeto	54
Figura 54 - Alteração a CandidateTechnology	54
Figura 55 - Alteração a CandidateProject.....	54
Figura 56 - Tecnologias do Candidato.....	54
Figura 57 - Projetos do Candidato	55
Figura 58 - Mensagens de Erro de Tecnologias.....	55
Figura 59 - Mensagens de Erro de Projetos	55
Figura 60 - Inserir Disponibilidades	55
Figura 61 - Disponibilidades depois da introdução	56
Figura 62 - Alteração a CandidateAvailability	56
Figura 63 - Mensagem de erro ao inserir disponibilidade	56
Figura 64 - Introduzir Empresa.....	57
Figura 65 – Nova instância de Company.....	57
Figura 66 - Empresa introduzida, Informação geral	57
Figura 67 - Empresa Introduzida, Projectos	57
Figura 68 - Empresa Introduzida, Localidades	57
Figura 69 - Introduzir Projeto, informação geral	58
Figura 70 - Introduzir Projeto, responsáveis.....	58
Figura 71 - Adicionar Projeto, escolher empresa de responsável	58
Figura 72 – Nova Instância de Project.....	59

Figura 73 – Novas Instâncias de ProjectCompany	59
Figura 74 – Novas Instâncias de ProjectResponsible	59
Figura 75 – Projeto na lista da empresa	59
Figura 76 - Criar formulário	60
Figura 77 - Nova instância de Form	60
Figura 78 Novas instâncias de FormQuestions.....	60
Figura 79 - Estabelecer Oferta, informação geral	61
Figura 80 - Estabelecer Oferta, escolha de projeto	61
Figura 81 - Estabelecer Oferta, tecnologias e linguagens.....	61
Figura 82 - Estabelecer Oferta, escolher formulário.....	62
Figura 83 - Estabelecer Oferta, tecnologias e linguagens.....	62
Figura 84 - Nova instância de Vacancy	62
Figura 85 - Novas instâncias de VacancySteps	62
Figura 86 - Novas instâncias de VacancyTools e VacancyLanguages	63
Figura 87 - Pesquisa por posição	63
Figura 88 - Pesquisa por posição e linguagem	63
Figura 89 - Pesquisa por posição e tecnologia.....	64
Figura 90 - Nova instância de Candidacy.....	64
Figura 91 - Nova instância de CandidacyStep.....	64
Figura 92 - Candidatura, web	64
Figura 93 - Candidatura mobile	64
Figura 94 - Passos de uma oferta.....	65
Figura 95 - Passo de uma oferta.....	65
Figura 96 - Candidatura	66
Figura 97 - Finalização de uma candidatura, email	66
Figura 98 - Finalização de uma candidatura, notificação	66
Figura 99 - Novas instâncias de VacancyInterview.....	67
Figura 100 - Alteração sobre VacancyStep	67
Figura 101 - Candidatura depois.....	67
Figura 102 - Continuação da candidatura, email	67
Figura 103 - Continuação da candidatura, notificação	67
Figura 104 – Eventos	68
Figura 105 - Eventos com disponibilidade	68
Figura 106 - Pop-up do novo evento	69
Figura 107 - Pop-up para escolher localidades	69
Figura 108 - Nova instância de Event e Interview.....	69
Figura 109 - Nova entrevista, informação geral	69

Figura 110 - Nova entrevista, responsável.....	70
Figura 111 - Nova entrevista, participante.....	70
Figura 112 - Convite da entrevista, email.....	70
Figura 113 - Convite da entrevista, notificação	70
Figura 114 - Erro na marcação	70

Lista de Códigos

Código 1 - Criação da lista	31
Código 2 - Paginação.....	31
Código 3 - Adicionar CandidateAvailability	32
Código 4 - Adicionar ao Profile.....	33
Código 5 - Remover do Profile.....	33
Código 6 - Começar a criação do Candidato	33
Código 7 - Estabelecimento do novo candidato	34
Código 8 - Formação dos Events para um mês	35
Código 9 - Alteração do calendário de mês para semana	36
Código 10 - Formação dos Events para uma semana	36
Código 11 - Início de criação de um Event, ou escolha de um Event existente	37
Código 12 - Adição de um novo Event.....	38
Código 13 - Adição de Vacancy	39
Código 14 - Geração de percentagens para o Pie Chart.....	40
Código 15 - Pesquisa de Vacancy.....	40
Código 16 - Verificar Vacancy em função das suas ferramentas	41
Código 17 - Continuação duma Candidacy	42
Código 18 - Adicionar disponibilidade.....	43
Código 19 - Filtrar Eventos	44
Código 20 - Registro de utilizador na OneSignal	45
Código 21 - Sincronismo Read-Only.....	46

1. Introdução

Neste capítulo introduzimos algumas ideias fundamentais sobre o projeto IView, incluindo algumas inspirações e enquadramento no mercado de informática atual, os objetos gerais e a organização deste relatório.

1.1. Enquadramento

Muitas empresas de sistemas de informação portuguesas, têm um processo de entrevistas bastante desatualizado e limitado, não tirando partido dos seus conhecimentos tecnológicos. Normalmente, os processos de contratação envolvem a transferência de ficheiros importantes, como por exemplo currículos e dossiês de competências¹ por correio (eletrónico e/ou físico). Por outro lado, a maioria das marcações de entrevistas ocorrem por comunicação telemóvel ou *email*, o que torna o candidato dependente do colaborador, em que qualquer falha de comunicação poderá resultar na não marcação da entrevista, ou que esta seja marcada incorretamente.

Uma das raras exceções que não se limita ao processo acima descrito, é a consultoria *Mind Source* [1], que fornece o “Portal de Emprego”. Contudo, este portal oferece pouco mais que a capacidade de guardar ficheiros e realizar candidaturas a cargos disponíveis. Outro exemplo é a aplicação “My Profile” desenvolvida pela empresa de recursos humanos *Randstad* [2], que apenas disponibiliza um formato simples e interativo para descrever o currículo e dossiê de capacidades.

Atendendo às limitações descritas, nasceu o projeto **IView** que tem como objetivo principal não só expandir as funcionalidades das duas aplicações referidas, como também incluir outras ideias úteis para as empresas que o vierem a utilizar.

O desenvolvimento do IView, utilizou como base uma empresa de tecnologia fictícia a que chamámos PS IT, servindo de enquadramento para outras possíveis empresas.

1.2. Objetivos

O projeto IView, foca-se no desenvolvimento de duas aplicações, uma *web* e uma *mobile*, que funcionam em conjunto para garantir a comunicação entre candidatos e colaboradores de uma forma simples e acessível durante um processo de candidatura, desde o momento em que uma nova oferta é aberta, até ao momento final da contratação.

A aplicação *web*, aplicação central onde estão desenvolvidas grande parte das funcionalidades, garante acessos de leitura e escrita à base de dados onde serão guardados, entre outros dados:

¹ Ficheiro em que um candidato consegue demonstrar as suas competências em tecnologias, idiomas e onde, normalmente, pode incluir aplicações que desenvolveu

- Currículos, dossiês de competências e disponibilidades² de candidatos;
- Informações gerais dos colaboradores que participam na aplicação;
- Empresas colaboradoras e projetos com as mesmas;
- Ofertas a novos cargos na empresa - Cada oferta inclui um processo de entrevista;
- Candidaturas a vagas, incluindo a situação da mesma no processo de entrevista;
- Entrevistas e outros eventos marcados;
- Resultados de entrevistas que já ocorreram;
- Informações gerais da empresa.

A aplicação *mobile* serve como apoio ao candidato, disponibilizando-lhe notificações sobre o estado dos processos onde está envolvido, bem como a consulta de informações relevantes sobre os mesmos. Através da aplicação *mobile*, também é possível ao candidato, realizar alterações dos dados pessoais, tais como nome, morada, etc.

1.3. Organização do documento

O restante relatório tem a seguinte composição de capítulos:

- Formulação do Problema – contém informação sobre da aplicação IView, incluindo referências para algumas aplicações semelhantes, detalhe dos requisitos funcionais e plataforma tecnológica onde o projeto é desenvolvido;
- Solução Proposta – neste capítulo inclui-se a arquitetura do projeto, o modelo da base de dados, esboços das principais interfaces de utilização, e os detalhes de implementação mais relevantes;
- Avaliação Experimental – onde é verificada a execução das várias funcionalidades do projeto através de casos de teste;
- Conclusões – o último capítulo deste relatório é onde se estabelecem as conclusões que foram formadas com o desenvolvimento do projeto, bem como sugestões para melhoramentos futuros.

² Por disponibilidades, referimos a janelas de tempo associados a dias de semana que o candidato considerar fiáveis para participar numa entrevista

2. Formulação do Problema

Neste capítulo, na secção 2.1, é considerada e discutida a utilidade da aplicação proposta no presentes projeto, bem como exemplos de aplicações semelhantes, incluindo algumas das suas limitações e como o IView as pode complementar. A secção 2.2 inclui uma breve descrição do projeto, apresentando as várias funcionalidades deste projeto. Por último, na secção 2.3, é descrita a plataforma de desenvolvimento *OutSystems*, com a qual o projeto foi desenvolvido.

2.1. Estado da arte

Existe um considerável número de consultorias e outras empresas da área informática, que utilizam uma aplicação semelhante ao IView, mas com algumas limitações que pretendemos colmatar com a IView.

A limitação mais marcante é a ausência de interação entre o candidato e o colaborador da empresa. As aplicações que foram encontradas, servem normalmente para a empresa gerir candidatos, os seus currículos e a partilha dos mesmos com os clientes, não incluindo qualquer tipo de comunicação entre as duas partes.

Por outro lado, as aplicações acima referidas funcionam unicamente com dossiês de capacidades em formato PDF. Desta forma, não só é dificultada a atualização do dossiê, como não é incluído qualquer mecanismo para filtrar informação desnecessária ou repetida, nem é indicado quais as competências tecnológicas que a empresa procura ou que considera importantes.

No mesmo contexto, também não incluem qualquer tipo de funcionalidade de marcação de entrevistas, embora algumas utilizem o Google Calendar. Aquelas que não utilizam nenhum mecanismo semelhante correm o risco de, caso ocorra alguma falha de comunicação entre as partes interessadas, a entrevista não ocorrer. Aquelas que utilizam o Google Calendar, conseguem ultrapassar esta limitação, mas o candidato fica completamente dependente de receber um convite para a entrevista, o qual pode não chegar.

As pesquisas efetuadas para o desenvolvimento do IView, revelaram, contudo, duas aplicações que permitem a comunicação entre candidato e colaborador. A primeira, já referida anteriormente, o “Portal de Emprego” [1] da *Mind Source*, permite aos candidatos duas funcionalidades principais:

- Gerir ficheiros, currículos e dossiês, ultrapassando assim a necessidade de comunicar com um colaborador sempre que é necessário realizar uma atualização de ficheiros;
- Verificar as vagas disponíveis e candidatar-se às mesmas.

Contudo, estas funcionalidades têm limitações importantes, como por exemplo: não são indicados quais os ficheiros que o candidato deve incluir, nem a sua estrutura. Já para as candidaturas, a informação fornecida é bastante limitada, sendo só indicadas quais as candidaturas que estão a ocorrer.

Outro exemplo é uma aplicação de gestão de perfil produzida pela *Randstad* [2], a qual não inclui qualquer capacidade de verificação de vagas e quais as candidaturas às mesmas, estando limitada unicamente à construção e gestão do perfil dos candidatos. Esta aplicação permite, contudo, o desenvolvimento e gestão do currículo e dossiê de capacidades de uma forma bastante fácil. De todas as formas, a *Randstad* não é uma empresa na área das tecnologias, incluindo por isso no perfil do candidato, um extenso raio de informações que torna a sua formação um pouco mais complexa que o necessário. Apesar das várias limitações enumeradas, esta a aplicação da *Randstad*, proporcionou parte da inspiração gráfica para o IView.

Existem ainda duas outras aplicações que estão, de alguma forma relacionadas com o IView, mas ao contrário das referidas anteriormente, não são limitadas a uma só empresa e são mais adequadas para a partilha de informação sobre perfis e vagas na área em questão. A primeira referência, trata-se da rede social *LinkedIn* [3], que permite gerir um perfil analogamente à aplicação da *Randstad*, mas de uma forma menos interativa. Esta aplicação, também não inclui qualquer tipo de informação sobre candidaturas ou entrevistas, mas é em si uma aplicação que permite a partilha fácil de informação de perfis adequados a empresas variadas. Quase numa vertente completamente oposta, a aplicação *ITJobs* [4] fornece uma extensa fonte de informação sobre empregos, formações e eventos na área de tecnologias e informática, mas não inclui qualquer possibilidade de gestão de perfis.

Tal como referido no início desta secção, as aplicações semelhantes ao IView, mesmo que limitadas, são bastante comuns em empresas na área dos sistemas de informação. Tal facto é indicativo que o IView pode ser bastante útil para esta área, procurando ir mais além na gestão da informação dos candidatos, bem como promovendo a facilidade de interação destes com as empresas. Por exemplo, utilizando de novo a experiência pessoal de um dos autores do projeto, em algumas entrevistas na área, este constatou que um extenso número de empresas nesta área, já dispõem de aplicações do tipo da IView, mas não suportam a possibilidade de os candidatos poderem interagir com as empresas através dessas aplicações. É portanto, um requisito que não foi ainda considerado nas aplicações existentes no mercado, mas que seria desejável - normalmente os entrevistadores referem que, no futuro, a empresa para a qual trabalham deverá iniciar um projeto interno para criar tal funcionalidade.

2.2. Descrição do Projeto

O projeto IView está dividido em duas aplicações, tendo cada aplicação um conjunto individual de requisitos funcionais, de acordo com o seu objetivo específico.

Na Figura 1 está representado um caso de utilização que descreve de forma geral as funcionalidades das aplicações para os vários perfis de utilizador. Nesta figura é possível verificar a existência de quatro perfis de utilizador:

- Utilizador Não Registrado – o nível mais baixo que existe, com funcionalidades limitadas;
- Candidato – representa o candidato às ofertas disponibilizadas pelas empresas, sendo os seus acessos limitados a leitura e escrita sobre a informação a si associada, com a exceção dos anúncios das vagas, que são públicos, tendo um acesso unicamente de leitura a estas informações;
- Colaborador – representa um colaborador na empresa, possuindo um maior conjunto de acesso, não limitado a instâncias associadas ao próprio;
- Administrador – este nível inclui todas as funcionalidades do colaborador, mais alguns extras que possibilitam a administração de utilizadores.

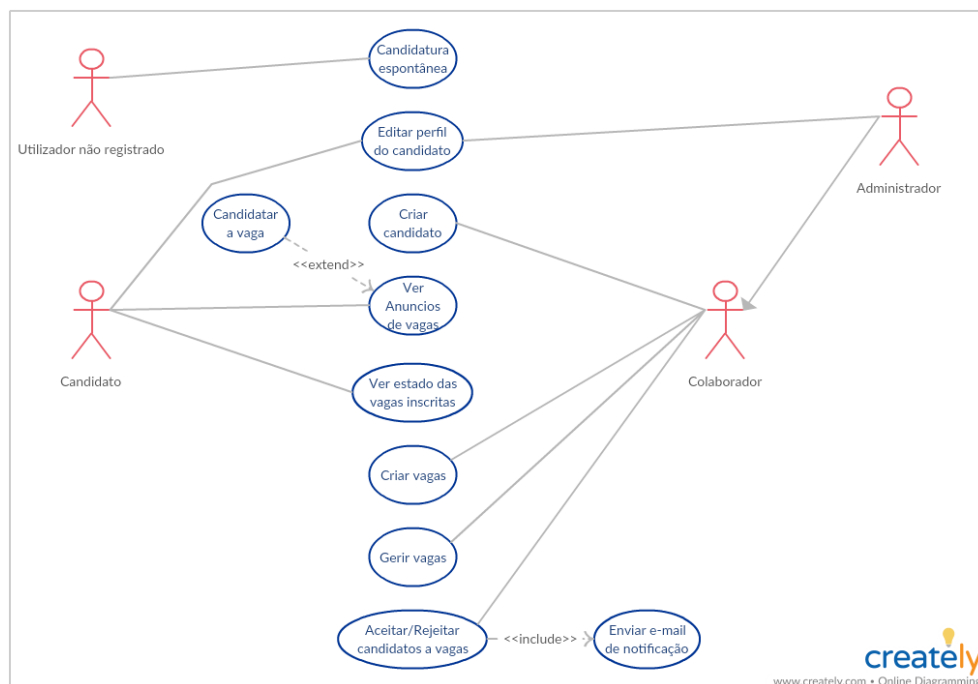


Figura 1 – Visão geral do IView.

Nas restantes 3 subsecções, são descritas as funcionalidades de cada aplicação (*web* e *mobile*), apresentando casos de utilização, seguidos de uma ideia geral dos mesmos. Também é incluindo algumas interações *back-end* que ocorrerão no decorrer das funcionalidades, referindo as entidades da base de dados, que podem ser verificadas na secção 3.2 onde apresenta o respetivo modelo entidade-associação.

2.2.1. Aplicação *web*

A vertente *web* do projeto, tem como foco a disponibilização de diferentes funcionalidades aos dois perfis principais participantes na aplicação: candidatos e colaboradores.

Também inclui, a publicação de informação geral da empresa, incluindo:

- Contactos;
- Missão e valores da empresa;

- Vagas para novas colocações;
- Informação geral sobre os colaboradores;
- Possibilitar a realização de candidaturas espontâneas.

Estas funcionalidades, as funcionalidades dos utilizadores não registrados, podem ser verificadas na Figura 2.



Figura 2 - Casos de utilização, Utilizadores Não Registrados.

Os colaboradores da empresa que participem no processo de entrevista são capazes de:

- Estabelecer novos candidatos em função de candidaturas espontâneas;
- Formar os processos de entrevista para novos cargos na empresa como também os formulários para as entrevistas;
- Gerir os vários passos que compõem um processo de candidatura e os respetivos estados de acordo com a sua progressão;
- Verificar candidaturas a novos cargos, permitindo promover um candidato para o próximo passo do processo, ou terminar o mesmo, com o candidato sendo informado automaticamente desta ação, por *email*;
- Marcar novas entrevistas e verificar entrevistas já marcadas;
- Guardar entrevistas, cuja informação poderá ser utilizada em futuras considerações.

A Figura 3 e a Figura 4 ilustram as funcionalidades acima enumeradas, de uma forma mais abrangente e detalhada. Na Figura 4 são demonstradas as funcionalidades mais centradas no processo de entrevista e outros eventos, sendo as restantes mostradas pela Figura 3.

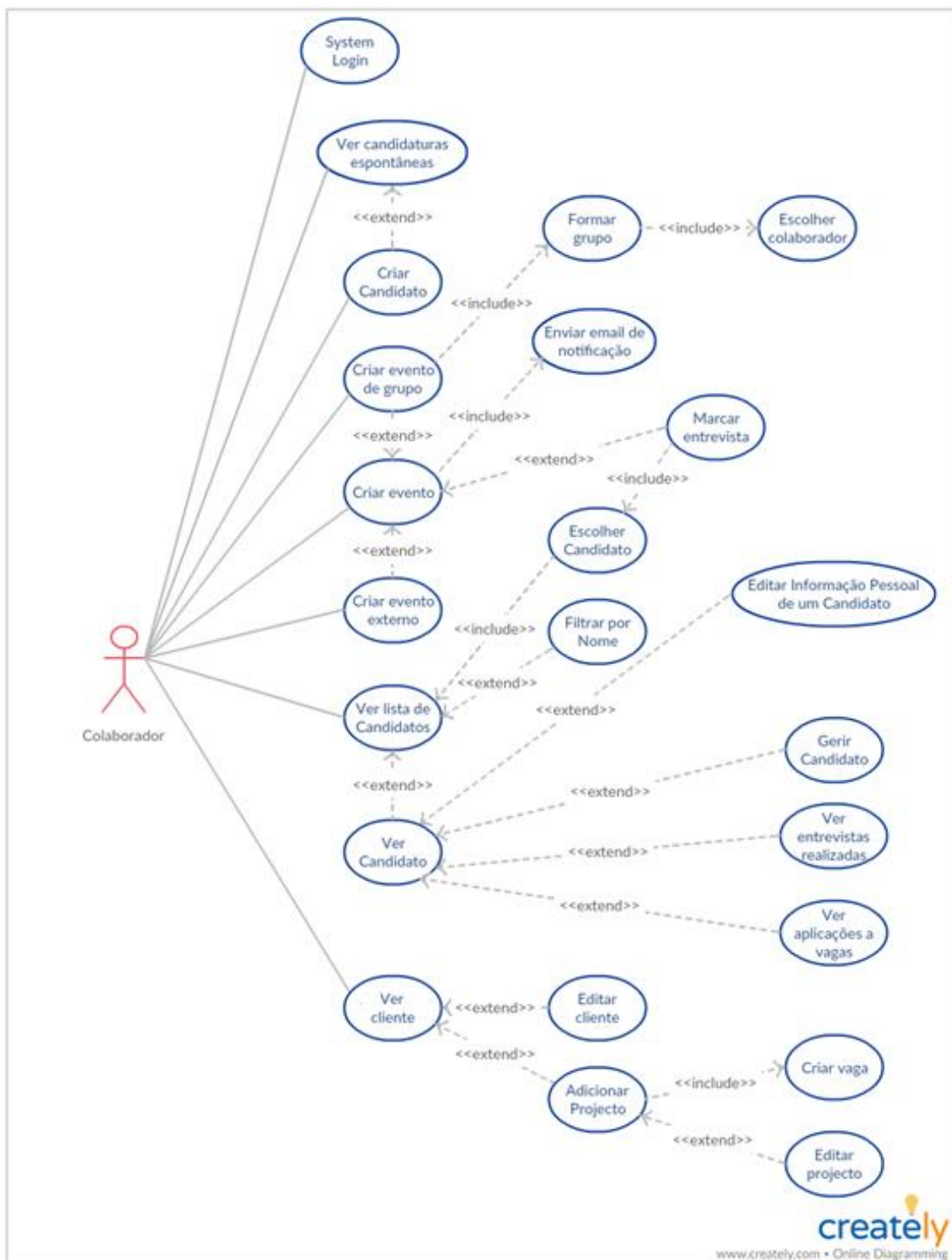


Figura 3 - Casos de utilização, Colaborador 1.

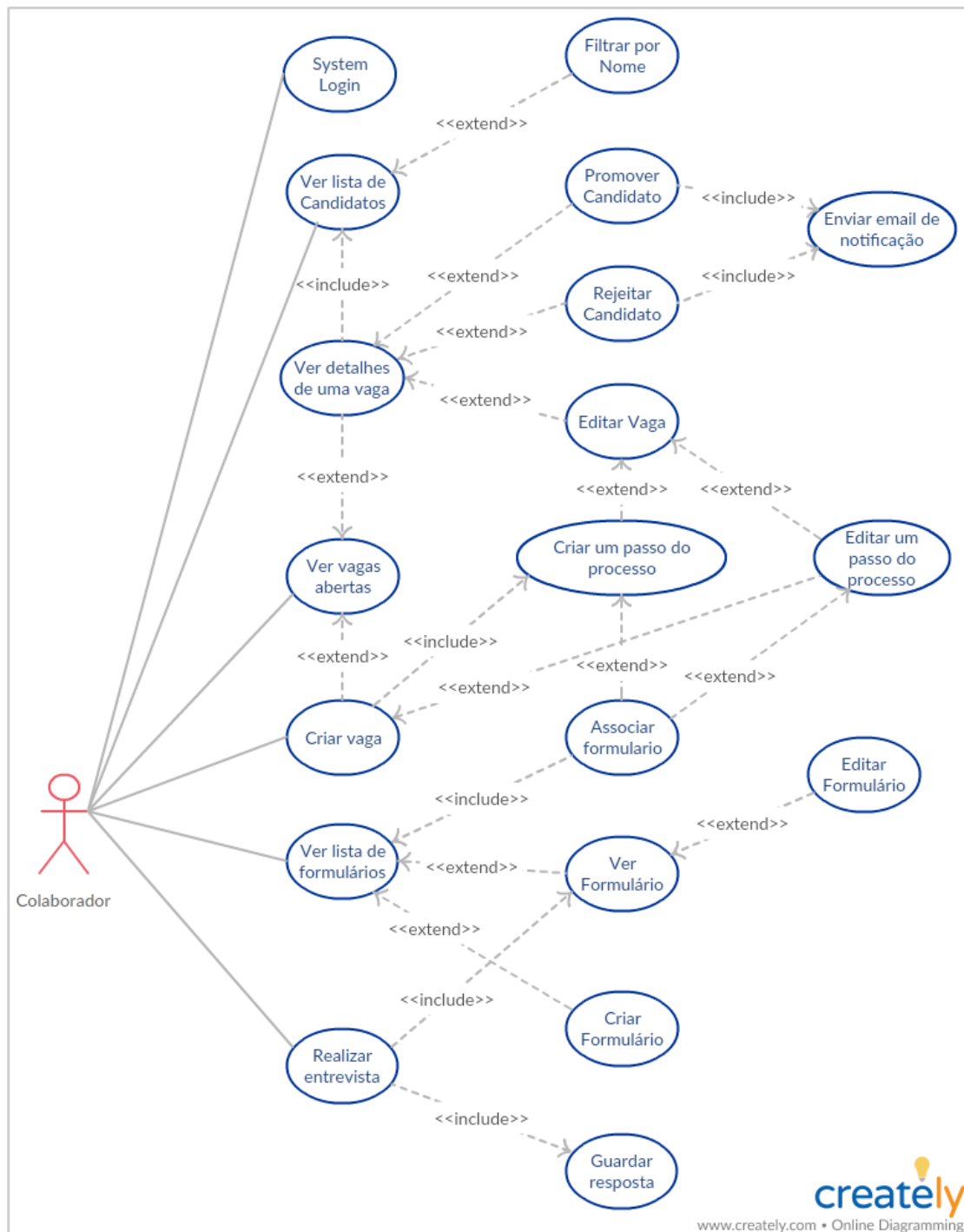


Figura 4 - Casos de utilização, Colaborador 2.

Os candidatos a novos cargos na empresa são capazes de:

- Gerir o seu currículo e dossiê de capacidades, com limitações e indicações estabelecidas pela empresa;
- Verificar/procurar anúncios aos quais se podem candidatar;
- Averiguar o estado de processos de entrevistas em que participam no momento;
- Verificar entrevistas marcadas, incluindo informações gerais da entrevista.

As funcionalidades referidas na aplicação *web*, podem ser verificadas na Figura 5.

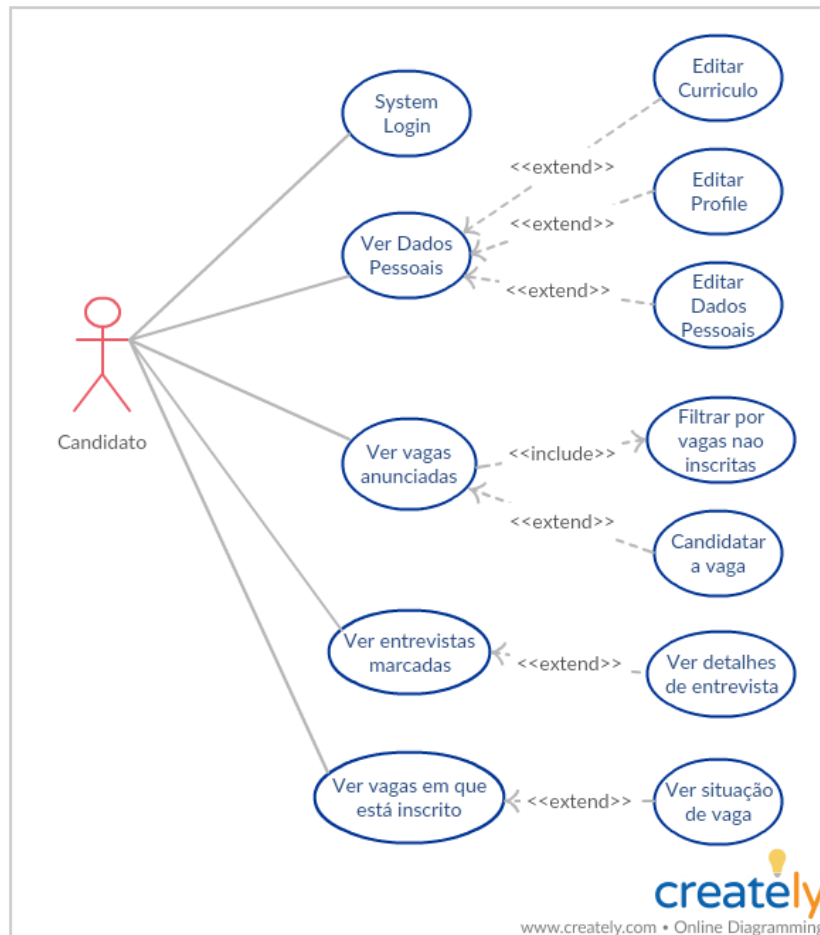


Figura 5 - Casos de utilização, Candidato.

Por último, temos o gerente que inclui todas as capacidades de um colaborador acrescentando mais algumas funcionalidades que podem ser verificadas na Figura 6. Estas capacidades extra servem principalmente para realizar alterações na base de dados que embora importantes, não devem ser acessíveis a todos os colaboradores. Estas capacidades extra são:

- Adicionar e remover empresas colaboradoras;
- Remover candidatos estabelecidos;
- Remover e adicionar colaboradores;
- Editar os eventos estabelecidos por colaboradores;
 - mais especificamente tem a capacidade de alterar a responsabilidade do evento para outro colaborador.
- Remover projetos já estabelecidos.

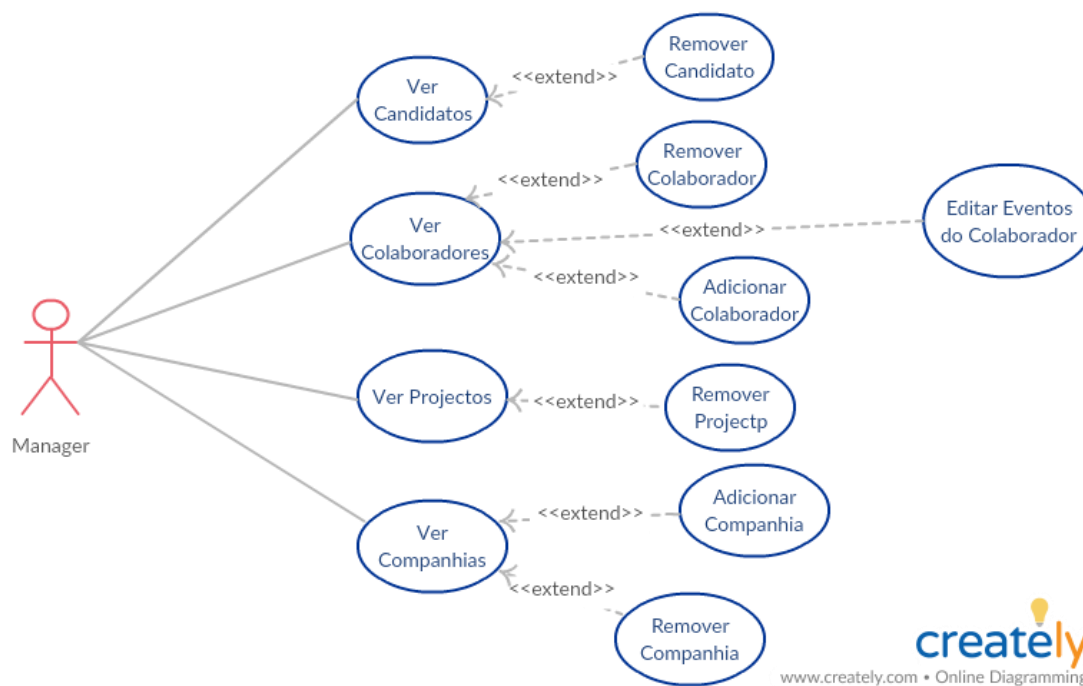


Figura 6 - Casos de utilização, Gerente.

2.2.2. Aplicação *mobile*

A vertente *mobile* da aplicação, tem como objetivo ser uma fonte de informação conveniente e *lightweight* para os candidatos que utilizam o IView, optando-se assim por manter nesta vertente as funcionalidades estritamente necessárias a este fim. Através desta aplicação, os candidatos são capazes de realizar as funcionalidades da Figura 7:

- Verificar e editar as suas informações pessoais básicas (nome, morada, contactos, etc.) no seu currículo;
- Ver o estado de processos aos quais se candidataram - receber notificações quando tais processos sofrem alteração, ou seja, se foram rejeitados ou passaram à próxima fase;
- Ver entrevistas marcadas e respetivas informações.

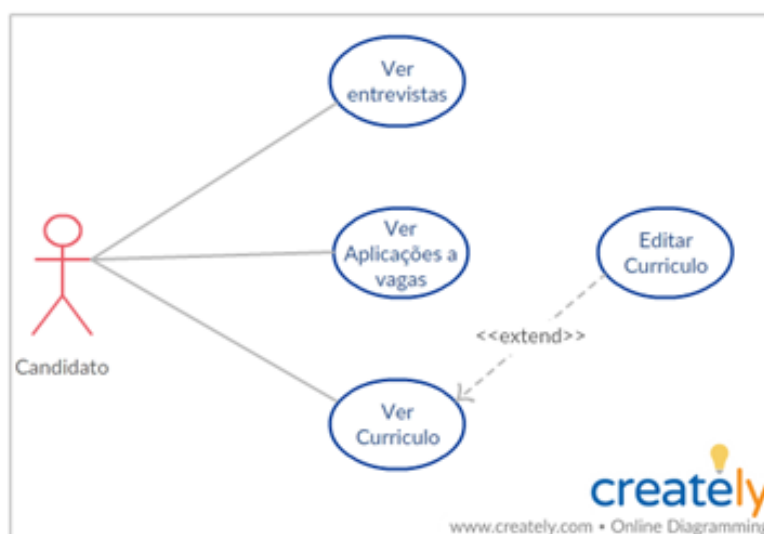


Figura 7 - Casos de utilização, Candidato *Mobile*.

A necessidade de uma aplicação mais leve deve-se, em parte, ao problema de sincronismo de aplicações móveis desenvolvidas em *OutSystems*. Ao manter só alguns acessos de leitura essenciais e ainda menos acessos de escrita, que em si são restritos a informações acessíveis unicamente a um candidato, a aplicação desenvolvida pode tomar partido dos padrões de sincronização mais simples, garantindo uma melhor *performance* quando comparada a aplicações *mobile* com um numero de leituras e escritas muito elevado. Este problema e as suas possíveis soluções são abordadas, numa forma simplificada, na subsecção 2.3.

2.3. Descrição da plataforma, modelos de desenvolvimentos

Para a implementação de ambas as aplicações do IView foi utilizada a plataforma de desenvolvimento *OutSystems*, cuja arquitetura se encontra ilustrada na Figura 8.

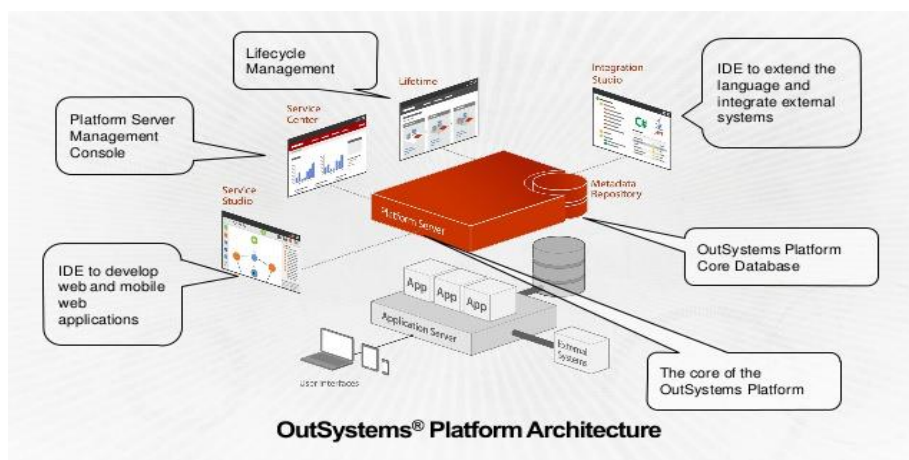


Figura 8 - Arquitetura OutSystems [5].

A escolha desta plataforma foi devido a duas razões fundamentais:

- Sendo capaz de desenvolver as duas vertentes (*web* e *mobile*), é possível aproximar as duas interfaces gráficas;
- Permite o desenvolvimento numa interface visualmente agradável e simples de utilizar, sem ter de se despendar muito tempo no desenvolvimento da mesma.

A plataforma *OutSystems* permite um desenvolvimento rápido de ambas as aplicações através de um modo visual quer na aplicação *web* quer na aplicação *mobile*. As aplicações são publicadas utilizando a *framework* .Net (linguagem C#). No *back-end*, utiliza uma base de dados *SQL Server* e padrões *JavaScript* para produzir o *front-end*.

2.3.1. Servidores Fundamentais

Para o desenvolvimento rápido e simples característico da aplicação, a *OutSystems* inclui três servidores fundamentais e especializados numa área diferente. O primeiro dos três servidores é o *Code Generator* que essencialmente realiza uma leitura da interface gráfica de desenvolvimento e a partir de informação da mesma:

- Verifica dependências externas e aplica otimizações à aplicação;

- Gera código nativo e *proxies* de integração;
- Agrupa as capacidades de gestão de sessão, autenticação e configurações para a aplicação.

No *Deployment Service* os componentes gerados pela aplicação são implementados num *server* de aplicações, garantido que a mesma é instalada consistentemente em cada servidor *front-end* da *farm* de servidores da organização dona da aplicação.

Por último existe o *Application Service* que fornece dois serviços importantes:

- O *Scheduler Service*, que administra a execução de *threads* planeadas;
- O *Log Service*, responsável por gerir erros, auditorias e o desempenho de uma aplicação.

2.3.2. Sincronismo

O desenvolvimento sobre a plataforma *mobile* regularmente envolve a utilização de duas fontes de informação: a base de dados (*online*) e o armazenamento local (*offline*). Estas fontes devem estar sempre que possível sincronizadas para garantir que informação da segunda fonte (*offline*) é a mais atualizada possível.

Por esta razão a *OutSystems* fornece uma *framework* de sincronismo, com cinco padrões principais que permitem o sincronismo de forma viável e constante, o mais constante possível.

- *Read-Only Data* – utilizado quando os utilizadores só precisam de ler informação, apaga toda a informação no recipiente e de seguida transfere toda a informação da base de dados, ideal para quando a informação pretendida é pouca;
- *Read-Only Data Optimized* – semelhante ao *Read-Only Data*, mas para dimensões de *data* mais extensiva, apenas atualiza informação modificada;
- *Read/Write Data Last Write Wins* – este padrão é útil quando existe alteração de *data* em modo *offline*, mas em que não existe conflito de escrita entre vários utilizadores;
- *Read/Write Data with Conflict Detection* – uma versão mais aprofundada do padrão anterior, permitindo a resolução de conflitos de escrita de múltiplos utilizadores;
- *Read/Write Data One-to-Many* – continuação dos dois padrões anteriores, este padrão permite que vários utilizadores alterem a mesma informação.

A aplicação *mobile* desenvolvida neste projeto utiliza os padrões *Read-Only Data* e *Read/Write Data Last Write Wins*. O primeiro para *screens* onde é verificada unicamente informação que não se pode alterar, exemplo entrevista e candidaturas a vagas. Já o segundo padrão é utilizado para alterar informações pessoais no currículo, que são unicamente acessíveis a um candidato e por isso não existirá concorrência.

2.4. Segurança de Informação

A aplicação *IView* não adiciona capacidades de segurança relevantes às fornecidas pelo *OutSystems*, no desenvolvimento normal das aplicações. Tal segurança é estabelecida

considerando vários problemas comuns nas aplicações *web*. Desta forma, serão verificados os problemas mais importantes e como a *OutSystems* os tenta corrigir.

- *Injection* - A *Outsystems* inclui vários filtros que removem troço de código que permita a injeção de informação que levante problema na base de dados;
- Autenticação – As características da autenticação da *OutSystems* garante que o identificador de utilizador não seja utilizado para gerar os cookies. As sessões têm uma validade e a sua identificação nunca é incluída em URL. Por último, as passwords são guardadas na base de dados com uma forte encriptação que utiliza outros valores da instância;
- *Cross-site scripting* – Semelhante a *Injection*, mas com a adição de uma segurança a recursos para que os mesmos não possam ser acedidos por domínios não reconhecidos;
- *Cross-site Request Forgery* – Cada página inclui um mecanismo *token-based* garantindo que a mesma foi gerada para um utilizador num ambiente seguro e reconhecível.

Só existem duas funcionalidades de segurança adicionados pela *IView* que possam ser verificados sendo que o mais simples dos dois é que qualquer utilização do identificador do utilizador para leitura ou escrita de valores associados ao mesmo são obtidos pela função *GetUserId()*. Este passo garante que tais leituras e escritas não ocorrem com um falso identificador.

Noutro lado, são utilizados três papeis, *Roles*, que limitam o acesso a certas páginas em função de cada *Role*. Os *Roles* utilizados representam o colaborador (*IViewEmployee*), o candidato (*IViewCandidate*) e por último o manager (*UserManager*). Note-se que os primeiros dois foram criados especificamente para a aplicação *IView* enquanto que o terceiro é predefinido pela *OutSystems*.

3. Solução Proposta

O foco deste capítulo é a proposta do projeto implementado, descrevendo a estrutura da aplicação e a arquitetura *4 Layer Canvas* estabelecida pela *OutSystems*, na secção 3.1, a base de dados da aplicação, na secção 3.2, os *wireframes* que demonstram alguns exemplos de páginas que compõem a aplicação *web*, na secção 3.3, e, por último, alguma lógica para implementar os requisitos funcionais mais relevantes, nas secções 3.4 e 3.5.

3.1 Arquitetura do projeto

Tal como grande parte dos projetos desenvolvidos na arquitetura *OutSystems*, *IView* é estruturada seguindo o padrão de *4 Layer Canvas*, que promove a abstração correta de serviços reutilizáveis, o isolamento correto de módulos funcionais distintos e a partilha dos mesmos por várias aplicações desenvolvidas em conjunto. A Figura 9 demonstra uma breve representação e descrição desta estrutura, descrevendo cada uma das camadas.

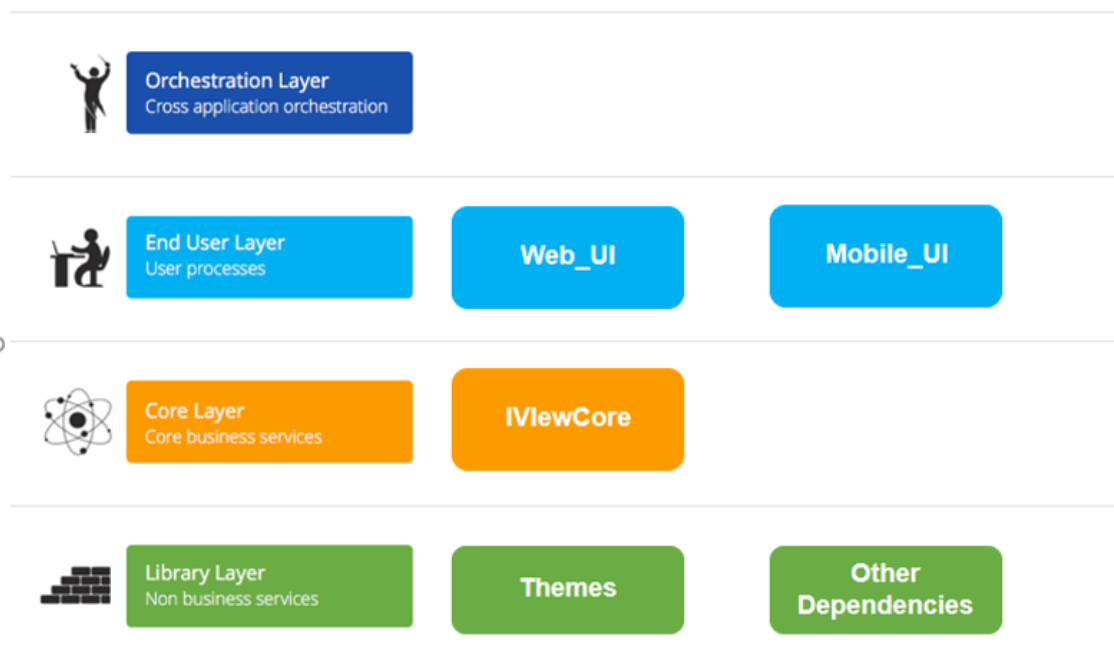


Figura 9 - Estrutura *4 Layer Canvas*.

Em termos da arquitetura da *IView* as quatro *layers* são:

- *Orchestration Layer* – este projeto não inclui qualquer módulo ou elemento que pertença a esta *layer*;
- *End User Layer* – sendo que o projeto é composto por duas aplicações, *web* e *mobile*, o projeto utiliza a *Web_UI* e *App_UI*;

- *Core Layer* – esta *layer* é composta pelo modulo *IViewCore*, onde é desenvolvida a base de dados e algumas estruturas partilhadas;
- *Library Layer* – este projeto inclui várias dependências, as mais notáveis são:
 - *Liverpool Template*;
 - *Silk UI*;
 - *Html2PdfConverter*;
 - *FullCalendar2*;
 - *Google Maps Mobile*;
 - *Google Maps Web*;
 - *In-App Notification Core*;
 - *OneSignal Plugin*.

3.2 Modelo entidade-associação da base de dados

Para facilitar a demonstração do modelo criado para o projeto, o modelo foi dividido em três partes como esta secção:

- Utilizadores - onde são verificadas as entidades que incluem informação dos Candidatos e Empregados, os utilizadores;
- Vagas - inclui as entidades sobre as vagas, projetos e candidaturas a vagas;
- Eventos - contém todas as entidades relacionadas com eventos e entrevistas.

Nesta secção só é verificado numa forma muito geral as entidades que compõem a base de dados, para verificar as mesmas numa forma mais abrangente temos no anexo a secção Modelos de Dados.

3.2.1 Utilizadores

Para esta componente do modelo estão incluídas todas as entidades que estendem a entidade *User*, que é incluída por padrão pela *OutSystems*, juntando informação importante, ou sobre o cargo do Candidato, ou sobre o currículo e capacidades do Candidato.

A Figura 10 demonstra o modelo entidade-associação desta componente do modelo.

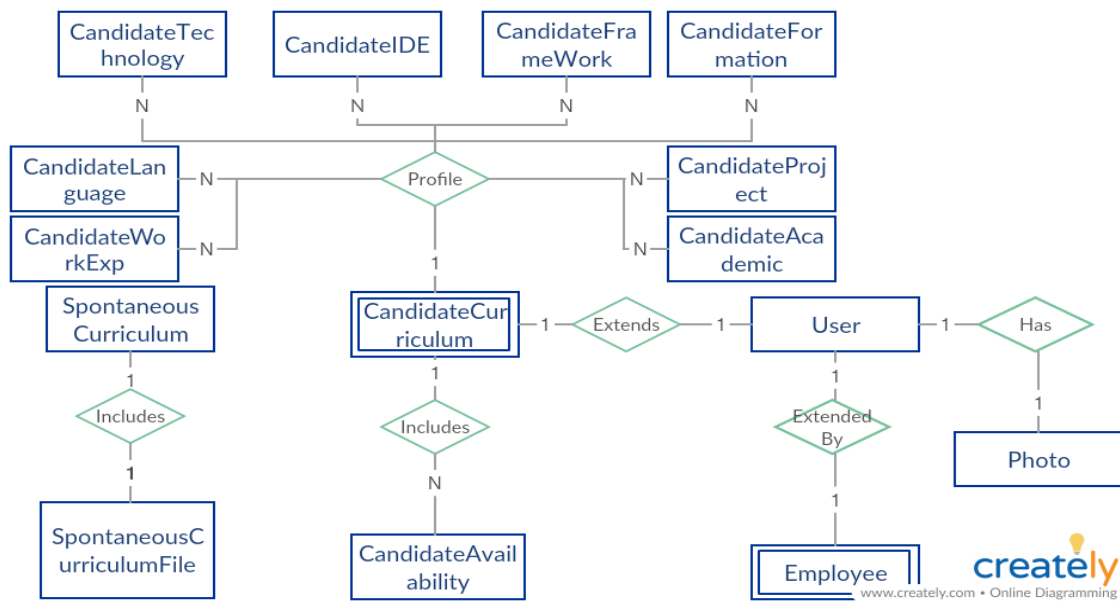


Figura 10 - Modelo EA, Candidatos.

As entidades da Figura 10 são:

- *User*, uma entidade padrão da *OutSystems*, representa os utilizadores registados na aplicação;
- *Photo*, todos os utilizadores podem incluir uma fotografia, que é representada por esta entidade;
- *SpontaneousCurriculum*, informações sobre candidatura espontânea;
- *SpontaneousCurriculumFile*, o currículo em si da candidatura espontânea;
- *Employee*, as instâncias desta entidade adicionam informação ao *User*, tal informação sendo relacionada com a posição e funções do utilizador na PS IT, sendo tal utilizador um colaborador da mesma;
- *CandidateCurriculum*, esta entidade serve como o centro de todas as entidades que formam as informações sobre os candidatos da aplicação;
- *CandidateAvailability*, instâncias desta entidade representam a disponibilidade que o candidato tem para entrevistas, associando a um dia de semana uma janela de tempo;
- *CandidateTechnology*, uma entidade que faz parte do dossiê de capacidades de candidatos, representa o nível da capacidade e experiência que um candidato tem com uma linguagem tecnológica, reconhecida pela aplicação;
- *CandidateFramework*, uma entidade que faz parte do dossiê de capacidades de candidatos, representa o nível da capacidade e experiência que um candidato tem com uma *framework*, reconhecida pela aplicação;
- *CandidateIDE*, uma entidade que faz parte do dossiê de capacidades de candidatos, representa o nível da capacidade e experiência que um candidato tem com uma IDE, reconhecida pela aplicação;

- *CandidateFormation*, uma entidade que faz parte do dossiê de capacidades de candidatos, representa uma formação oficial que o candidato tenha obtido, considerando as linguagens tecnológicas, *frameworks* e *IDEs* que a aplicação considera importantes;
- *CandidateLanguage*, uma entidade que faz parte do dossiê de capacidades de candidatos, representa a capacidade que um candidato tem com uma linguagem, reconhecida pela aplicação;
- *CandidateWorkExperience*, uma entidade que faz parte do dossiê de capacidades de candidatos, representa experiencia laboral que o candidato considere importante de notar;
- *CandidateAcademic*, uma entidade que faz parte do dossiê de capacidades de candidatos, representa o(s) curso(s) académico(s) que o candidato considere importante de notar;
- *CandidateProject*, uma entidade que faz parte do dossiê de capacidades de candidatos, que demonstra projetos desenvolvidos pelo candidato, que o mesmo considere importante de notar.

A razão para separar o *SpontaneousCurriculum* do *SpontaneousCurriculumFile* deve-se ao facto que as duas são normalmente visualizadas num grande conjunto e com a dimensão de um ficheiro *pdf*. Tal visualização poderá implicar um peso (em tempo de processamento) desnecessário sobre a aplicação, principalmente porque o ficheiro só é necessário numa forma mais individual.

3.2.2 Vagas

Nesta componente da base de dados estão incluídas todas as entidades que permitem estabelecer ofertas para candidatos se candidatarem, como também essas mesmas candidaturas. A Figura 11 demonstra estas entidades.

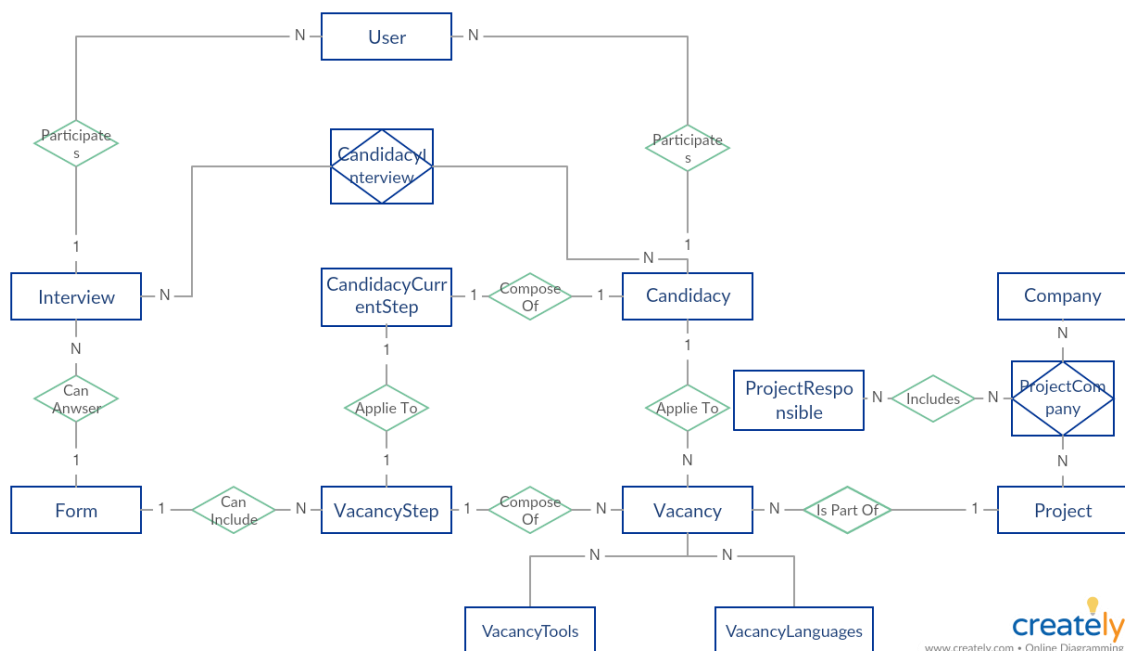


Figura 11 - Modelo EA, Vagas.

As entidades incluídas na Figura 11 são:

- *Company*, nesta entidade é possível incluir informações de empresas colaboradoras de projeto em que PS IT é também uma participante;
- *Project*, projetos a serem realizados pela PS Tec, por estes projetos é possível estabelecer vagas para candidatos;
- *ProjectCompany*, com esta entidade é possível associar uma empresa colaboradora a um projeto e também um cliente a vários projetos;
- *ProjectResponsible*, cada empresa a participar no projeto pode incluir um ou mais responsáveis a que informações dos candidatos escolhidos para vagas serão enviados, para que os mesmos possam decidir quem deve continuar no processo de entrevista, as instâncias desta entidade representam tal responsável;
- *Vacancy*, as vagas a preencher em si, podem ou não ser associadas a projetos;
- *VacancyTool*, com esta entidade é possível associar várias ferramentas reconhecidas pela aplicação a uma vaga, ferramentas cujo conhecimento pode ser visto como essencial para preencher a vaga;
- *VacancyLanguage*, cada instância desta entidade demonstra uma linguagem, reconhecida pela aplicação, cujo o conhecimento é visto como indispensável para preencher uma vaga;
- *VacancyStep*, cada vaga deve incluir um conjunto de passos do processo de entrevista, tais passos são representados por instâncias desta entidade;
- *Form*, sendo que um passo no processo de entrevista pode ser uma entrevista em si, o mesmo pode, não obrigatoriamente referir um *Form*, que em si é um formulário para uma entrevista;
- *Candidacy*, quando uma vaga é criada qualquer candidato da aplicação pode-se candidatar à mesma, tal candidatura a vaga é representada por uma instância desta entidade;
- *CandidacyCurrentStep*, esta entidade representa o passo atual duma candidatura;
- *CandidacyInterview* por instâncias desta entidade uma entrevista pode ser associada a uma candidatura, permitindo assim que uma entrevista possa ser utilizada para varias candidaturas;

3.2.3 Eventos

Neste componente do modelo EA são demonstradas as entidades relacionadas com eventos como também uma continuação das entidades *Interview* e *Form*.

Estas entidades podem ser verificadas na Figura 12 que se segue.

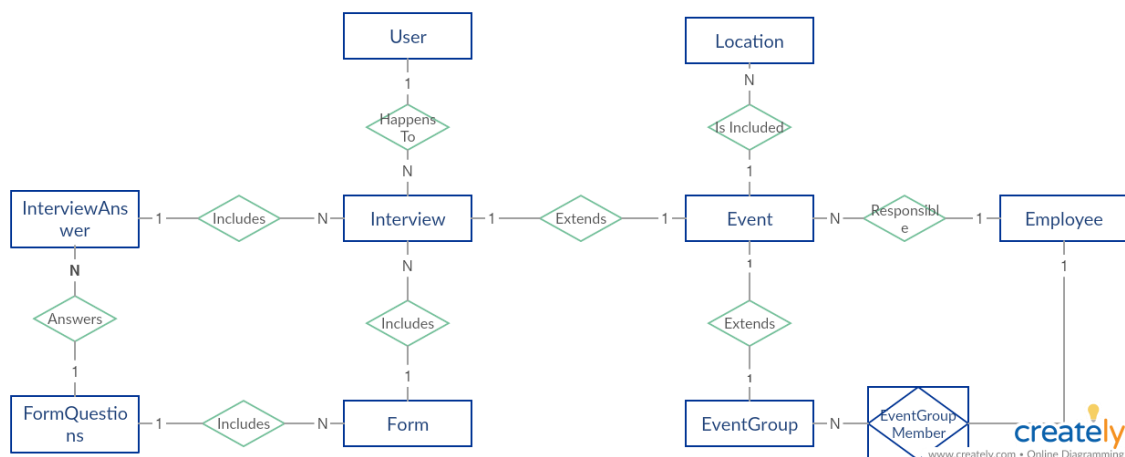


Figura 12 - Modelo EA, Eventos.

Na Figura 12 pode ser verificado as seguintes entidades:

- *Event*, esta entidade representa os vários eventos que podem ocorrer que envolvam pelo menos um colaborador, que esta entidade refere;
- *EventGroup*, grupo de colaboradores para um evento em grupo;
- *EventGroupMember*, membro individual de um grupo, para um evento;
- *Form*, formulário para uma entrevista;
- *Interview*, uma entrevista que tenha ocorrido/ ainda para ocorrer com um utilizador, que esta entidade refere. Deve referir o formulário utilizado;
- *FormQuestion*, cada instância desta entidade corresponde a uma questão de um formulário da *Form*;
- *InterviewAnswer*, na mesma forma que um formulário (*Form*) é composto por várias questões (*FormQuestions*), uma entrevista (*Interview*) é composta pelas várias respostas às questões, sendo estas representadas por instâncias desta entidade;
- *Location*, uma localização reconhecida pela aplicação, normalmente representa localidades onde um *Event* poder ocorrer.

Uma nota durante o desenvolvimento desta entidade considerou-se não incluir a referencia ao *Form*, sendo que *Interview* normalmente estará associada a um *CandidacyStep*, que já refere o *Form*. Mas sendo que queremos manter as instâncias de uma entrevista, mesmo depois de uma vaga ser preenchida, o formulário deve ser referido pela entrevista para a mesma manter a coerência da base de dados.

3.3 Wireframes do projeto

Nesta esta secção são demonstradas algumas *frames* de páginas importantes, em que algumas têm uma estrutura única na aplicação e outras servem de exemplo de uma estrutura utilizada habitualmente em páginas com uma função específica, também são utilizados estes exemplos para demonstrar alguns dos elementos mais comuns da aplicação, tanto *widgets* (*rich* e normais)

estabelecidos pela *OutSystems* como também elementos mais estruturados especificamente para a aplicação.

Os elementos mais comuns da aplicação são o *header* e o menu de utilizador, que aparecem sempre no topo das páginas (como se pode ver na Figura 13 acompanhada pela Figura 14 e Figura 15 onde é demonstrado os dois possíveis menus de utilizador).



Figura 13 - Frames, Home Page

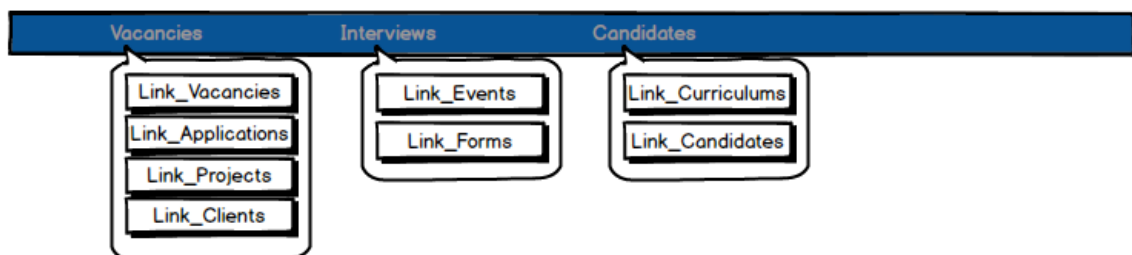


Figura 14 - Frames, Menu de Colaboradores.

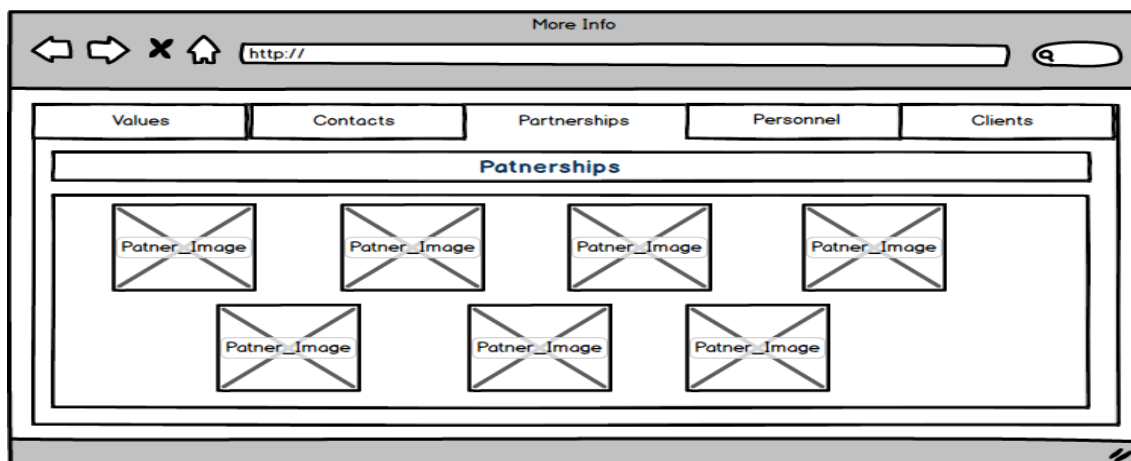


Figura 17 - Frame, MoreInfo Patnerships Tab.

Como referido anteriormente existem dois elementos importantes na aplicação que podem ser verificados nesta página, o primeiro desses é o *Tabs*. Este elemento permite dividir uma página até cinco partes, assim quando uma página inclui muita informação, em vez de ser extensa verticalmente, as informações estão divididas em *Tabs*.

O outro elemento utilizado nesta página a notar é o *List Records*, que permite demonstrar vários elementos obtidos na base de dados numa estrutura bastante livre, não forçando um formato em tabela.

Outra característica notável desta página é que várias palavras da *Tab* demonstrada na Figura 16 abre um *Ballon* com algum texto sobre o valor em si.

Na secção 3.2 foi estabelecido a existência de um currículo que um candidato pode verificar e alterar. Estas duas capacidades são exequíveis na página representada pela Figura 18 e também verificar, e pela Figura 19, editar.

Complete Name	Candidate Name
Date Of Birth	Candidate Date Of Birth
Native Tongue	Candidate Native Tongue
Locality	Candidate Locality
Accessibility	Candidate Accessibility

Figura 18 - My Curriculum, Show.

My Curriculum

Chang CV

Complete Name	Name_Input
Date Of Birth	Date_Of_Birth_Chosser
Native Tongue	Native_Tongue_ComboBox
Locality	Locality_Input
Accessibility	Accessibility_Input
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

Figura 19 - Frame, MyCurriculum, Edit.

Esta página serve como exemplo da utilização de alguns *inputs* da aplicação como também de dois *widgets* muitas vezes utilizados para demonstrar e alterar informação. Tais *widgets* são *Show Record* e *Edit Record*, ambos partilham o formato tabela e por isso são bastante utilizados em conjunto e também por vezes separados.

Na secção 3.2, também foi referido a capacidade de um candidato consegue controlar o seu dossiê de capacidades que é composto por várias entidades associadas ao colaborador. A Figura 20 demonstra parte da página que introduz alguns elementos notáveis e regularmente utilizados na aplicação.

My Profile

Technologies Frameworks IDEs Languages Academics Formations Work Experience Applications Developed

Technologies

Technologie	Pratical Exprience	Academic Exprience	Capacity
Technologie_Chosse	3	3	Capacitry_Chosse

Figura 20 - Frames, MyProfile.

O elemento a notar é a *Editable Table*, uma tabela que permite adicionar elementos à mesma, como também alterar elementos já existentes. No caso desta página, remover, guardar e alterar elementos da tabela não só altera a tabela como também altera diretamente a base de dados.

A única entidade que compõe o dossiê que não é alterado por uma *Editable Table* é a entidade *CandidateAppDeveloped*, sendo que mesma envolve informação extensa de mais para uma só linha de texto, por isso é utilizado uma *ListRecord*, com cada elemento utilizando um *EditRecord* para alterar a aplicação.

Já para adicionar uma nova instância a *CandidateAppDeveloped* é utilizado o *Pop-up Editor*³. No caso da *MyProfile* quando o botão “Add” é pressionado, um *Pop-up* abre que permite introduzir as informações do projeto desenvolvido. Este *Pop-up* pode ser verificado na Figura 21.

The screenshot shows a window titled "Add Application Developed". It contains a form with the following fields and controls:

- Name:** A text input field labeled "Name_Input".
- Start Date:** A date input field with two empty boxes and a calendar icon.
- End Date:** A date input field with two empty boxes and a calendar icon.
- Responsibilities:** A text input field labeled "Responsibilities_Input".
- Technologies:** A text input field labeled "Technologies_Input".
- Accomplishments:** A text input field labeled "Accomplishments_Input".
- Summarization:** A text input field labeled "Summarization_Input".
- Buttons:** "Cancel" and "Add" buttons at the bottom right.

Figura 21 - Frames, MyProfile *Pop-up*.

Uma funcionalidade regular das páginas da aplicação *web* da IView é de listar um conjunto de instâncias estabelecidas duma entidade da base de dados. Um exemplo simples destas páginas é o demonstrado na Figura 22, que mostra todas as instâncias da entidade *Form*.

The screenshot shows a web page titled "Forms". It contains the following elements:

- Search Bar:** A search input field labeled "Search_Form_Title_Input", a "Search" button, and a "Reset" button.
- Table:** A table with the following structure:

Title	Number Of Questions	Created
Form_Title_Link	Form_Number_Of_Questions	Form_Date_Of_Creation
- Pagination:** A bar with buttons for "Previous", "1", "2", "3", "...", and "Next".
- Buttons:** An "Add Form" button in the top right corner.

Figura 22 - Frames, Forms.

Esta página apesar de simples mostra os componentes mais comuns das páginas que demonstram as instâncias duma entidade, a primeira é a utilização da *Table Records* para demonstrar as instâncias em si.

Este *widget* é semelhante a *List Records* pois ambos demonstram instâncias duma lista, mas ao contrário da *List*, a *Table* mostra a informação no formato de tabela, composta de instâncias com informação pouco extensa e por isso que possam ser demonstradas numa só linha de texto.

Esta tabela é formada para demonstrar um conjunto limitado de instâncias da base de dados, por causa desta limitação é necessário utilizar outro elemento habitual deste tipo de páginas: a

³ *Rich widget* que permite criar uma página tipo *popup* a partir de uma página já estabelecida na aplicação.

lista de botões de navegação. Esta navegação é realizada com o *rich widget List_Navigation*, que produz a lista de botões de navegação como necessário tendo a conta a tabela, sem de precisar de código extra da parte da aplicação.

Por último temos o botão de adição, no caso da Figura 22, o botão “*Add Form*”, este botão não é utilizado sempre neste tipo de páginas, mas em quase todas, e serve sempre para navegar para uma página onde é possível adicionar um novo elemento.

Existem algumas exceções nas páginas de listagem das instâncias, algumas introduzindo um formato alternativo a pesquisa, outros demonstram a informação numa forma bastante diferente. Um exemplo dessas exceções pode ser visto na Figura 23, onde é verificada a página de listagem de aplicações a vagas existentes.

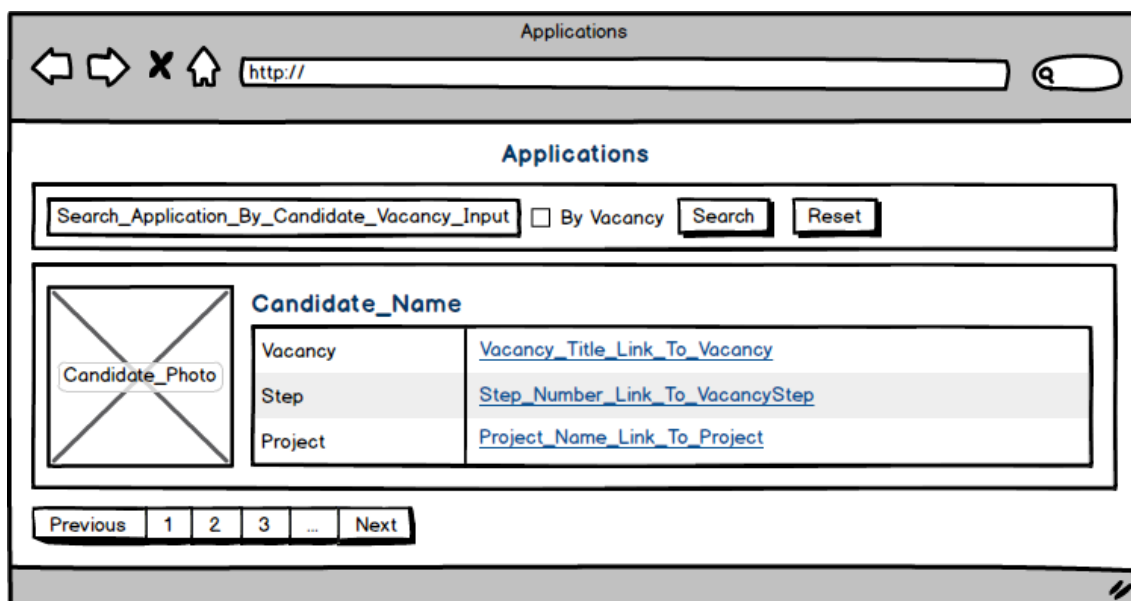


Figura 23 - Frames, Applications.

Como se pode ver entre a Figura 22 e a Figura 23 existem duas diferenças notáveis: a tabela é bastante diferente sendo utilizado um *List Record* em vez de uma *Table Record*, com o elemento demonstrado sendo uma combinação dos *widgets CardLeftImage* e *Show Record*.

Por último o formulário de pesquisa inclui a escolha de pesquisa por nome do candidato ou título da vaga, utilizando um *Check Box* para escolher entre um e outro, este quando é alterado força a pesquisa com o valor na *input*.

Outra exceção do formulário de pesquisa encontra-se na página de listagem de vagas, que pode ser verificada na Figura 24.

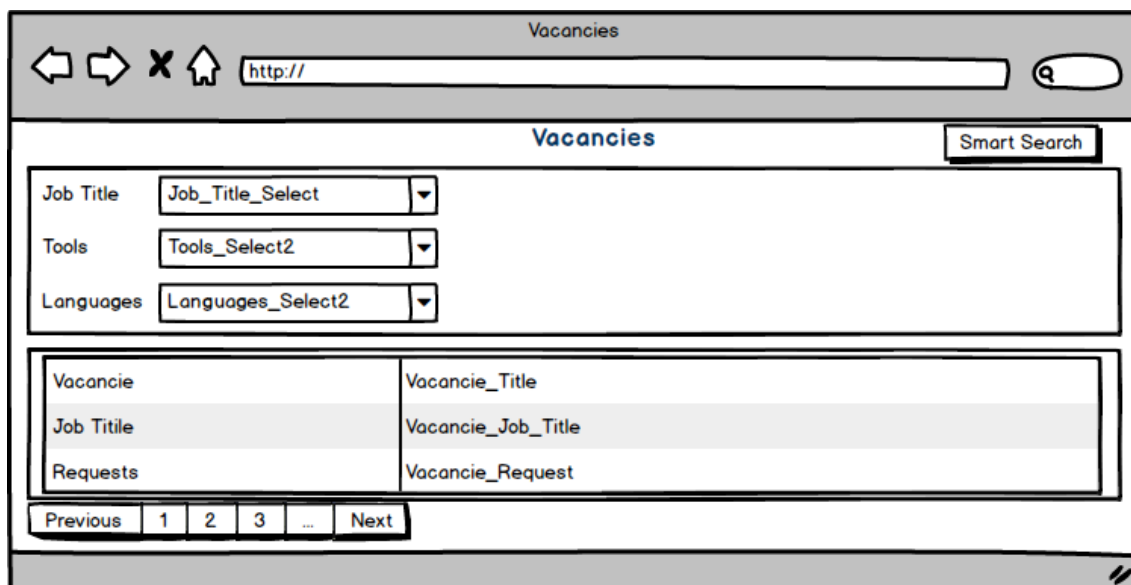


Figura 24 - Frames, Vacancies.

Como se pode observar, a pesquisa de vagas é feita sem nenhum *input* de texto, sendo realizado unicamente por escolha de um valor de três *Combo Boxes*, sendo uma *Combo box* simples, permitindo a seleção de um só elemento e duas com *Select2* que permite escolher uma ou mais opções.

Outra exceção a notar é a página de listagem de eventos que pode ser verificada na Figura 25.

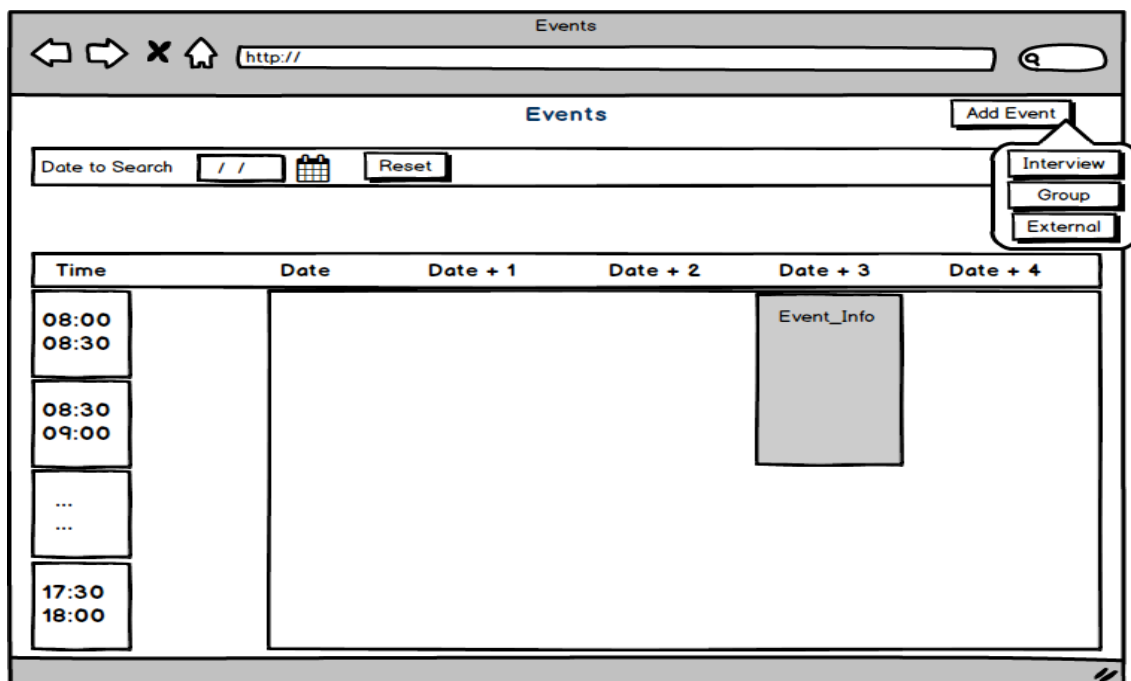


Figura 25 - Frames, Events.

O fator mais notável desta exceção é o facto de que os eventos são listados com um bloco que ocupa a janela de tempo entre o início e o fim do evento, na data correta, algo que é possível utilizando o *pluggin FullCalendar2*.

Anteriormente foi estabelecido que as páginas de listagem normalmente incluem um *link* nas instâncias listadas, para uma página que demonstra a informação da instância mais detalhada. Um exemplo simples deste tipo de página pode ser verificado na Figura 26, que demonstra a página de um candidato.

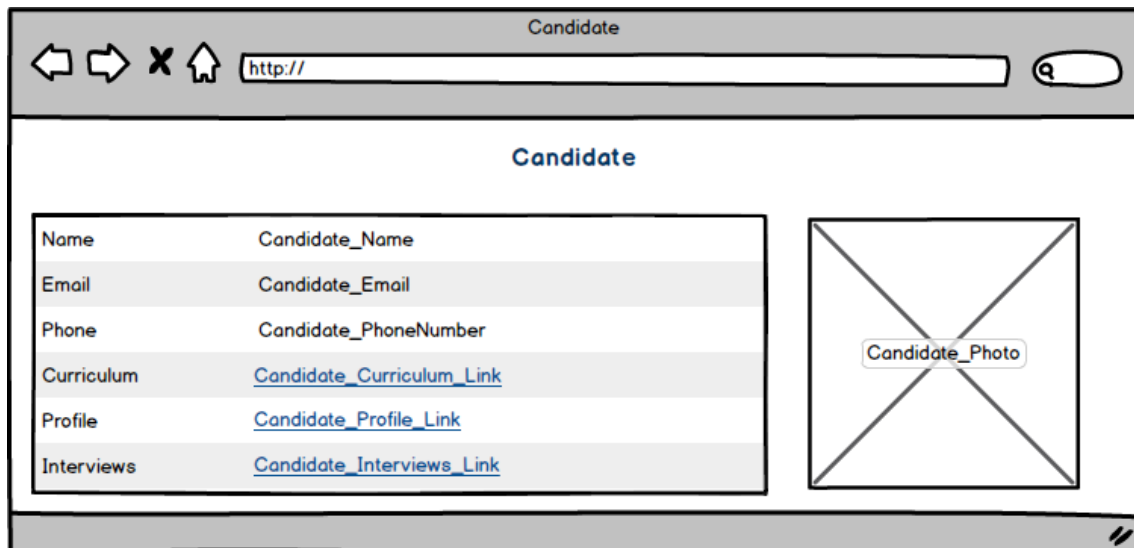


Figura 26 - Frames, Candidate.

Por vezes estes tipos de páginas demonstram informação bastante extensa, por este facto decidimos usar *Tabs* para dividir blocos de informação. Desta forma, facilitamos a visualização e organização da página usada.

Um exemplo da utilização das *Tabs* nestas páginas pode ser verificado na Figura 27, Figura 28 e Figura 29 onde se pode ver a página que demonstra uma vaga, que inclui a informação geral da mesma, os vários passos para a vaga e por últimos a lista das ferramentas e linguagens consideradas importantes para a vaga.

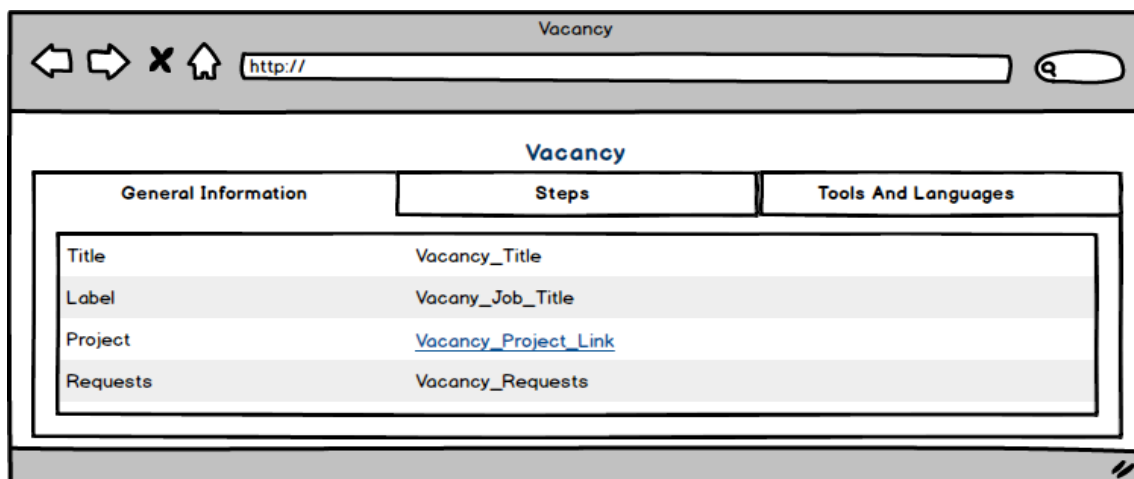


Figura 27 - Frames, Vacancy General Information Tab.

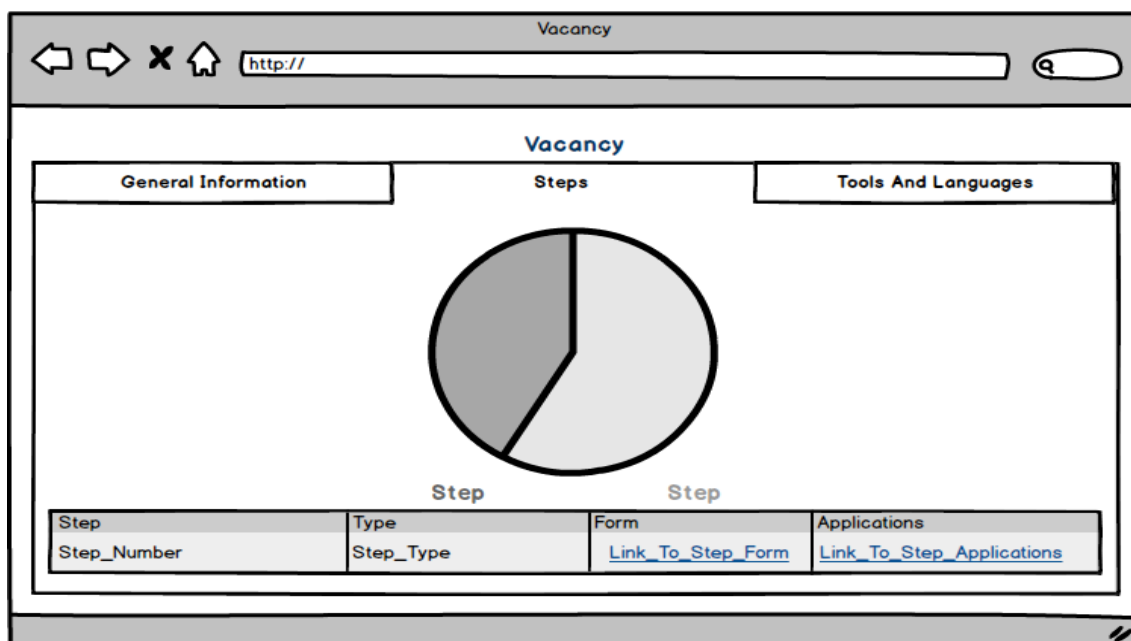


Figura 28 - Frames, Vacancy Steps Tab.

Figura 29 - Frames, Vacancy Tools and Languages Tab.

O único outro fator a notar da página da Figura 28 é o facto que é utilizado um *Pie Chart* para demonstrar a divisão por passos (em percentagem) dos candidatos participantes da vaga. Este *widget* não é utilizado mais nenhuma página.

Por último deve ser notado páginas que servem para introduzir novas instâncias das entidades da base de dados, um exemplo bastante simples destas páginas é a verificada na Figura 30, que demonstra a página que permite introduzir um formulário.

Figura 30 - Frames, FormEditAdd.

A página da Figura 30, demonstra os dois *widgets* mais comuns a páginas para adicionar instâncias, o primeiro sendo o *Edit Record*, este *widget* é normalmente utilizado quando se quer introduzir a informação que compõe a instância a introduzir. O outro *widget* regular é o *Edit Table*, que é utilizado regularmente para introduzir outras instâncias sobre uma entidade relacionada com a instância central a adicionar.

Outros dois elementos regulares destas páginas são os botões “*Create*” e “*Cancel*”, com o primeiro servindo para criar a instância formulada na página e o segundo para cancelar a criação da instância.

Em algumas situações uma página simples não é suficiente para estruturar uma instância a adicionar a base de dados, nesses casos uma de duas soluções são utilizadas: a utilização de *Tabs* ou de *Pop-ups*. Um exemplo de uma *Pop-up* utilizada neste tipo de páginas pode ser verificada na Figura 31.

Title	Number Of Questions	Created	Check Form	Chosse
Form_Title	Form_Number_Of_Questions	Form_Date_Created	Link_To_Form	Chosse_Form

Figura 31 - Frames, VacancyAdd Form Chosse *Pop-up*.

No caso das figuras anteriores, o *Pop-up* serve para escolher o *Form* para cada passo da vaga, demonstrando todos os *Forms* existentes, como também disponibilizando uma pesquisa dos *Forms* pelo nome.

3.4 Desenvolvimento Web

Para facilitar a demonstração das implementações mais notáveis, esta secção está dividida num conjunto de subsecções. Da mesma forma, na descrição, utilizamos o termo *action* quando se está a referir a uma *Screen Action* ou *Interface*, e o termo *function* no caso de uma *Function*, *Logic*.

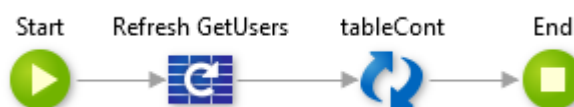
3.4.1 Listagem

Grande parte das páginas que compõem a aplicação *web* servem para visualizar uma lista de elementos, utilizando os *widgets Table Records* ou *List Records*, *widgets* que são preenchidos pela utilização de um *Aggregate*. Este módulo essencialmente obtém todos os elementos duma ou mais entidades⁴, cujos campos estejam de acordo com fatores de filtragem estabelecidos. Um exemplo simples deste *Aggregate* pode ser verificado no Código 1.



Código 1 - Criação da lista

Grande parte destas páginas incluem o *rich widget List_Navigation*, que fornece uma paginação simples da lista. Esta envolve sempre uma *action*, que realiza a alteração da informação em função da página. É realizado o *Aggregate* que obtém a informação da lista a paginar, seguido de um *Ajax Refresh* para reformar a lista na página escolhida, como se pode verificar no exemplo no Código 2.

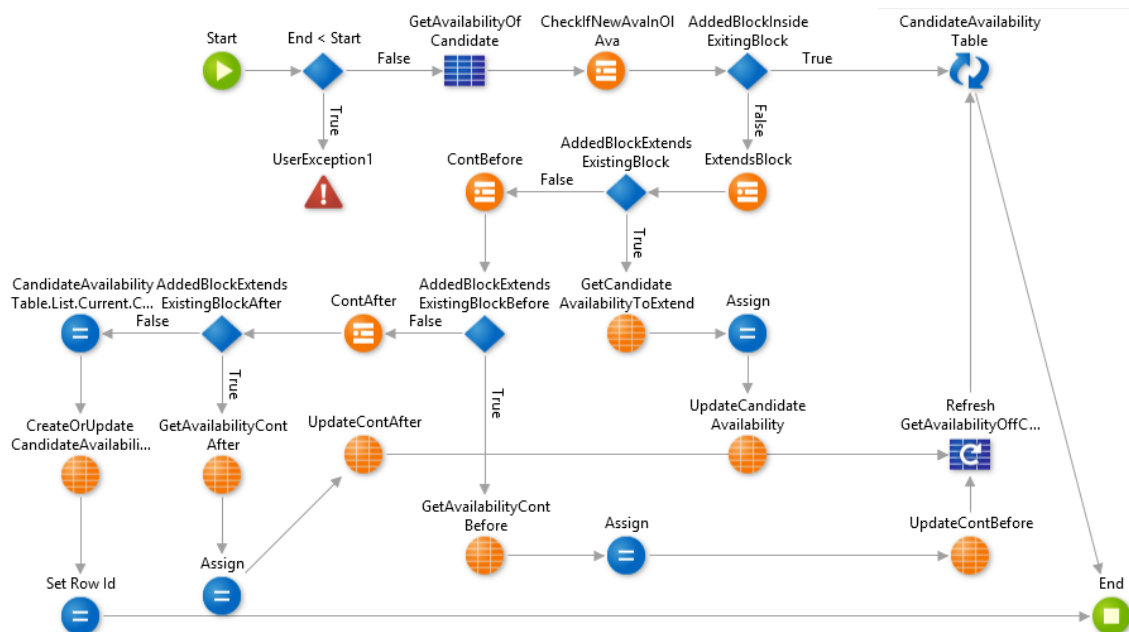


Código 2 - Paginação

3.4.2 Adicionar disponibilidades

A página das disponibilidades de um candidato não inclui uma preparação particularmente notável, sendo que na mesma apenas são obtidas as instâncias de *CandidateAvailability* associadas ao candidato. Mas a introdução de uma nova instância de *CandidateAvailability*, não é tão simples como criar uma nova instância e adicioná-la a base de dados, como se pode verificar no Código 3.

⁴ Se for obtida mais que uma entidade, estas são unidas através de ações *join*



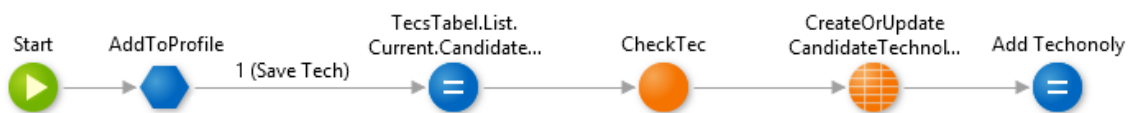
Código 3 - Adicionar CandidateAvailability

Esta complexidade deve-se ao facto de que quando um candidato introduz um novo bloco de disponibilidade, existem três formas deste novo bloco interferir com blocos já existentes. Por esta razão, qualquer adição de bloco temporal, começa por verificar se o bloco interfere com qualquer outro bloco existente e dependentemente de qual interseção acontecer é realizada uma alteração diferente a base de dados.

- O bloco a adicionar corre dentro de um bloco já estabelecido, o início do bloco a adicionar acontece depois do início do bloco estabelecido e o fim do bloco a adicionar acontece antes do fim do bloco estabelecido;
 - Neste caso o bloco a adicionar é ignorado e não é realizado qualquer alteração a base de dados.
- O bloco a adicionar começa antes de um bloco já existente, e acaba no meio desse bloco, o fim do novo bloco acontece entre o início e o fim do bloco existente, o início do mesmo acontece fora;
 - Neste caso o início do bloco estabelecido é alterado para o início do bloco a adicionar.
- O bloco a adicionar começa no meio de um bloco já existente, e acaba depois de tal bloco, o início do novo bloco acontece entre o início e o fim do bloco existente, o fim do mesmo acontece fora;
 - Neste caso o fim do bloco estabelecido é alterado para o início do bloco a adicionar.
- Não acontece qualquer interseção.
 - Não ocorrendo qualquer interseção, é adicionado o novo bloco a base de dados.

3.4.3 Adicionar e Remover do Profile

Na Figura 20 observamos a página de dossier de capacidades onde se pode verificar que inclui todas as capacidades do candidato. Capacidades estas que, em grande parte, são demonstradas por um conjunto de *Editable Tables* permitindo adicionar elementos à mesma, mas não inclui qualquer mecanismo próprio para escrever ou remover o elemento a base de dados. Para tal são utilizados duas *actions*: uma para adicionar (Código 4) e outra para remover (Código 5), sendo que é usado um *Switch* que escolhe as ações a realizar tendo em conta o *input* que recebe. Ambas as imagens mostram parte das *actions*, sendo que tal é simplesmente repetida 8 vezes.



Código 4 - Adicionar ao Profile

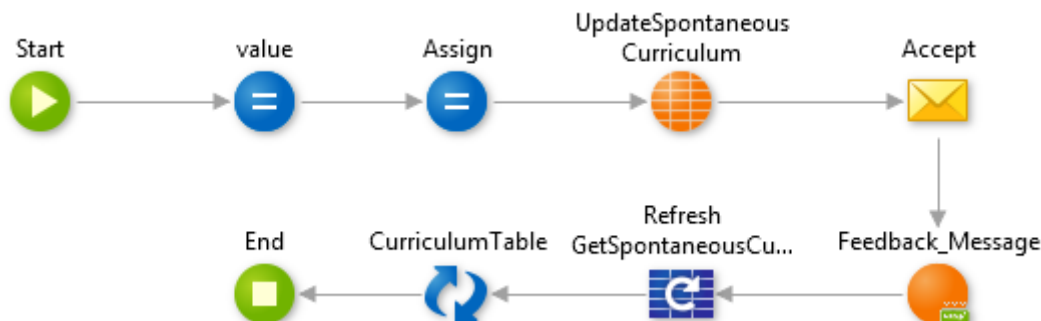


Código 5 - Remover do Profile

Ao adicionar é sempre verificado se é possível adicionar a instância garantindo que não existe repetições, utilizado uma *action* diferente. Neste processo, são verificados certos fatores chave como por exemplo: repetições e datas corretas. Caso não for ocorra qualquer exceção, então é criado ou atualizado o objeto a escrever. Já o remover não inclui qualquer verificação, simplesmente remove o elemento da entidade indicada.

3.4.4 Introduzir novo candidato

Os colaboradores têm acesso a uma página onde são observadas todas as candidaturas espontâneas ainda a verificar, com cada candidatura incluindo três botões que executam ações diferentes. Das mesmas, a única a tomar especial atenção é que a aceita uma candidatura, que inicia o processo no estabelecimento de um novo candidato, como é demonstrado no Código 6.

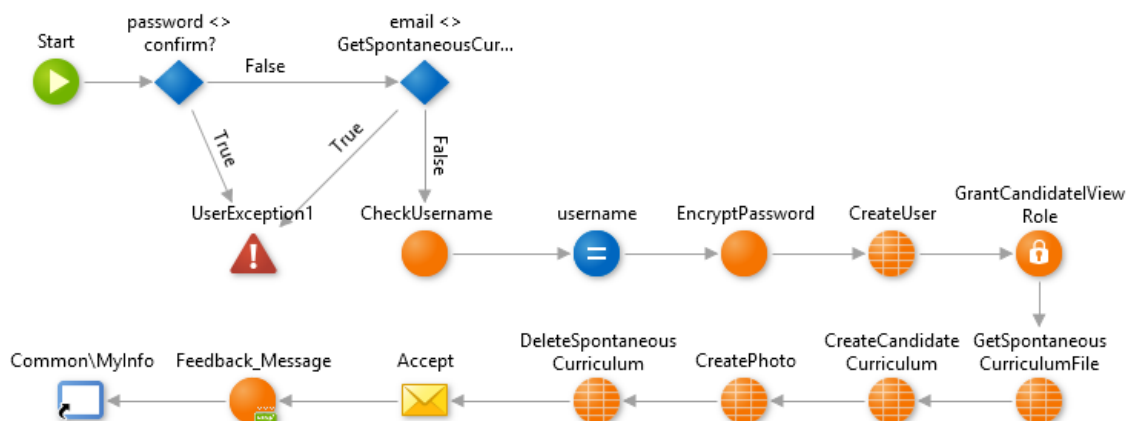


Código 6 - Começar a criação do Candidato

Na *action* do código anterior é estabelecido um valor de segurança: um texto de 20 caracteres completamente aleatórios. Este valor é adicionado ao *SpontaneousCurriculum* escolhido e é adicionado, como *input*, ao *link* do *email* que é enviado para *email* do *SpontaneousCurriculum*.

Este *link* levará o novo candidato para uma página que o permite estabelecer-se como um candidato. Esta página utiliza o valor do *input* para obter o *SpontaneousCurriculum*, alterado no Código 6. Desta forma, garantimos que no caso de um não existir, o utilizador é redirecionado para a página de erro interno.

Caso tal problema não ocorra, o utilizador tem de fornecer o seu *email*, sendo que este é o mesmo associado ao *SpontaneousCurriculum*, e a sua password duas vezes, podendo depois executar a *action* do Código 7.



Código 7 - Estabelecimento do novo candidato

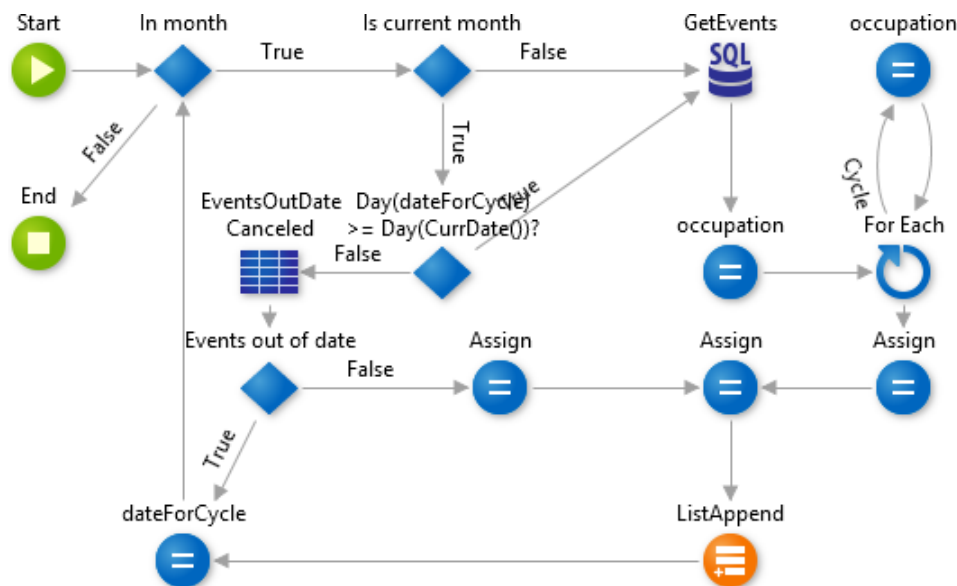
Da *action* do Código 7 o único passo a notar é a execução do *CheckUsername*, sendo que todos os outros passos criam um novo candidato, fornecendo o papel correto e remover qualquer instância da base de dados desnecessária.

O *CheckUsername* existe para tentar garantir a unicidade do *usernames* na aplicação. Para tal, após o *username* ser criado, se o nome do utilizador tiver espaços, estes serão trocados por pontos finais. De seguida, é verificado se já existe um utilizador com um *username* semelhante. Caso tal aconteça, é adicionado ao *username* um número que é maior que o número no *username* igual mais recentemente criado.

3.4.5 Verificar Eventos

Na página da Figura 25, um colaborador consegue visualizar o seu calendário de eventos, tanto por mês como por dias da semana, sendo que cada opção de visualização é gera um calendário diferente. Os calendários utilizados para os eventos são fornecidos pela extensão *FullCalendar2*. Esta extensão tem uma estrutura para produzir o evento chamado *Event*, o mesmo nome que a entidade *Event* representando um evento. Para distinguir os dois, quando é utilizado a estrutura do *FullCalendar2* a mesma é referida como *FullCalendarEvent*.

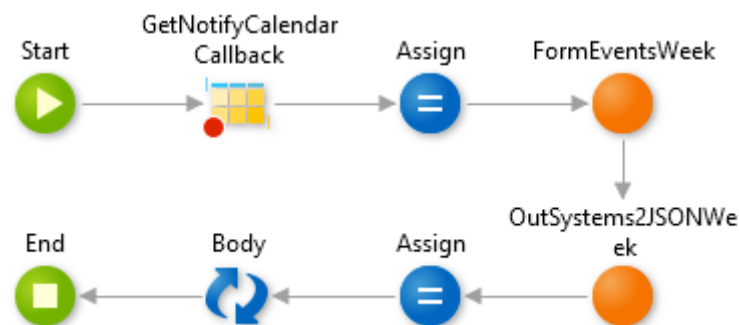
A preparação desta página envolve a *function FormEventsMonthEmployee* que gera os *FullCalendarEvents*. Esta mostra o estado de ocupação dos dias do mês atual em função dos eventos já estabelecidos em que o utilizador atual participa. Esta *function* pode ser verificada no Código 8.



Código 8 - Formação dos *Events* para um mês

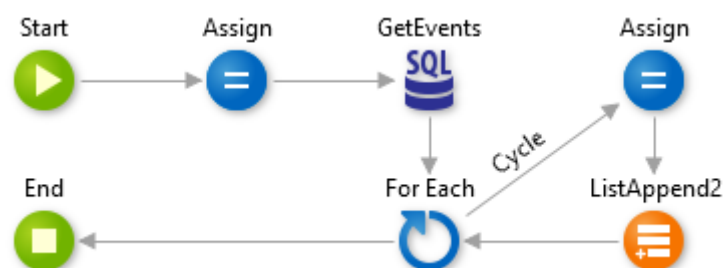
A *function* em si foca-se num ciclo em que é obtido todos os dias de um mês, não incluindo dias já ultrapassados. Dentro do ciclo é realizado um *SQL Query* que obtém todos os *Events*, em que o utilizador atual participa para o dia atual do ciclo. O tempo de todos os *Events* obtidos são acumulados numa variável, gerando assim o valor de ocupação que é utilizada para definir tanto a cor como o título do *FullCalendarEvent* do dia verificado.

Outra funcionalidade observável nesta página, podemos introduzir novos eventos no calendário do utilizador atual. Para tal, quando uma data é selecionada no calendário, o mesmo mostra a semana do dia escolhido, permitindo introduzir o novo evento. Para este fim, é associado uma *action* ao calendário de mês, observável em Código 9. Esta *action* começa com a utilização do *GetNotifyCalendarCallback*, para verificar a interação que ocorreu entre o utilizador e o calendário, mais particularmente as datas selecionadas. Com a informação das datas selecionadas é executada a *function FormEventsWeek* que devolve a lista de *FullCalendarEvents*, que representam todos os eventos em que o utilizador atual participa na semana selecionada.



Código 9 - Alteração do calendário de mês para semana

Esta *action* é bastante simples, mas a *FormEventsWeek* (Código 10) utilizada é focada num ciclo para criar *Events*, que representam os eventos em que o utilizador atual participa numa semana.

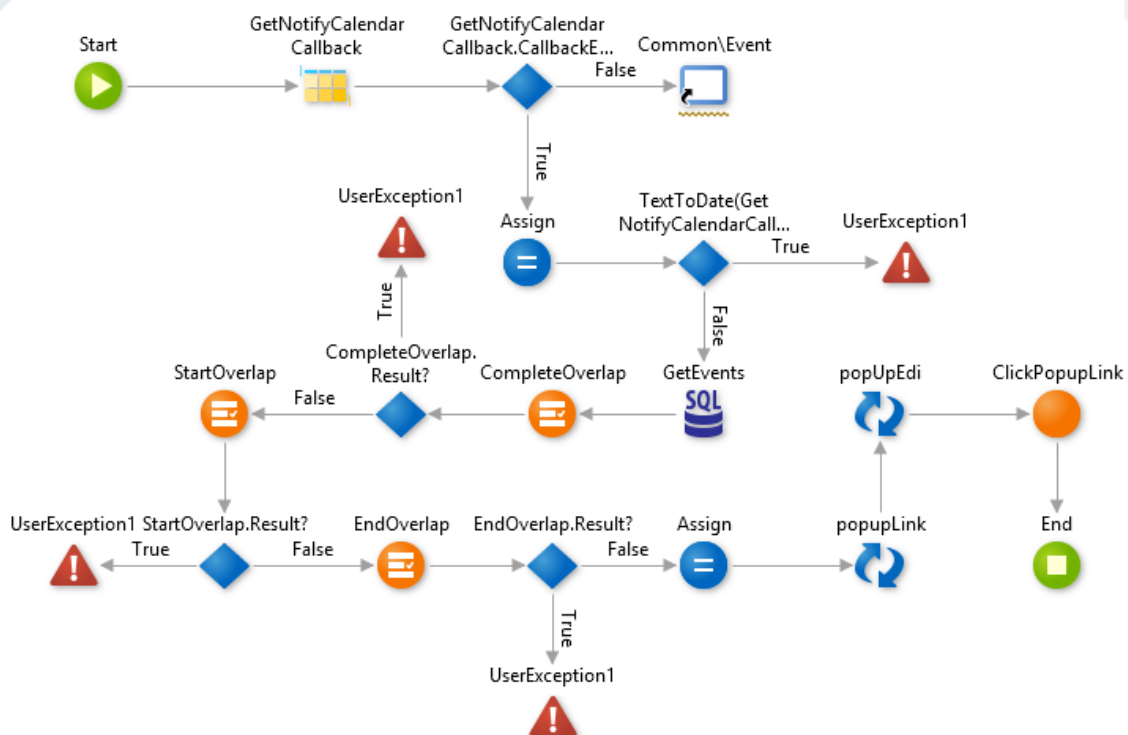


Código 10 - Formação dos Events para uma semana

Para tal a *function* começa por criar dois valores: *dateStart* - início da semana- e *dateEnd* -o fim da semana. O primeiro é igual a subtrair a data que a *function* recebe, a data selecionada no calendário, ao número do dia da semana menos um, essencialmente devolvendo sempre a data do início da semana. Já a segunda data só envolve adicionar 5 dias a data inicial.

Tendo as duas datas, são obtidas todas estâncias de *Events* em que o utilizador atual participa, dentro das duas datas, inclusive para ambas. Para cada evento obtido, é criado um *FullCalendarEvent* com a informação do *Event* do ciclo, que é adicionado a lista que a *FormEventsWeek* devolve.

Por último deve ser verificado o que ocorre quando um utilizador interage com o calendário da semana. Para tal existe a *action* do Código 11, que esta associada a característica *OnNotify* do calendário semanal.



Código 11 - Início de criação de um *Event*, ou escolha de um *Event* existente

Esta *action* começa por verificar que interação ocorreu entre utilizador e calendário. Se um evento já estabelecido foi pressionado, então é aberta a página que demonstra o evento duma forma mais detalhada.

Caso o utilizador crie um bloco, por *click and drag*, um *Event* é criado baseado no bloco criado. Mas primeiro é verificado se o bloco pode ser produzido, sendo que o bloco pode ser criado num dia que já passou ou no dia atual, mas numa hora que já passou. Caso um destes aconteça então é levantado uma exceção.

Caso o *Event* não ocorra no passado são verificadas três possibilidades de interseção entre o novo evento e eventos já estabelecidos. Para tal é obtido todos os *Events* que ocorrem no dia do evento a adicionar e é verificado se:

- O novo evento tem o início entre o início e fim de evento já existente, com o fim fora;
- O novo evento tem o fim entre o início e fim de evento já existente, com o início fora;
- O início do novo evento acontece antes do início de um evento e o fim depois do fim do mesmo evento.

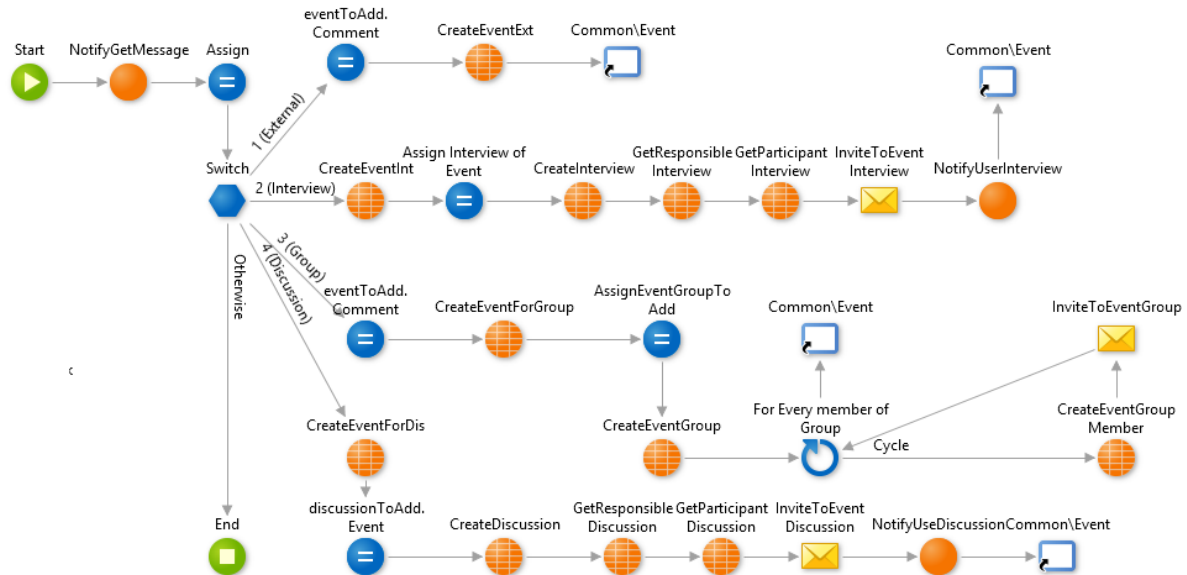
Caso nenhuma interferência ocorra, então o utilizador pode ser redirecionado a página que o permite criar o *Event*, cujas ações são verificadas na subseção seguinte.

3.4.6 Adicionar Evento

Os colaboradores têm acesso a uma página que permite introduzir um novo *Event*, que ocorrerá na data e tempos que a página recebe como parâmetros de entrada. A única *action* a notar desta

página, é a associada a criação do *Event*, que em si só é notável devido a sua utilização de um *Pop-up*.

Depois de um colaborador definir certas características essenciais do *Event*, o mesmo pode utilizar o botão de *Add* que enfoca um *Pop-up* onde é possível seleccionar uma localização, a única característica comum a todos os *Events* ainda a definir. Quando o mesmo é seleccionado a *action* verificada no Código 12 ocorre.



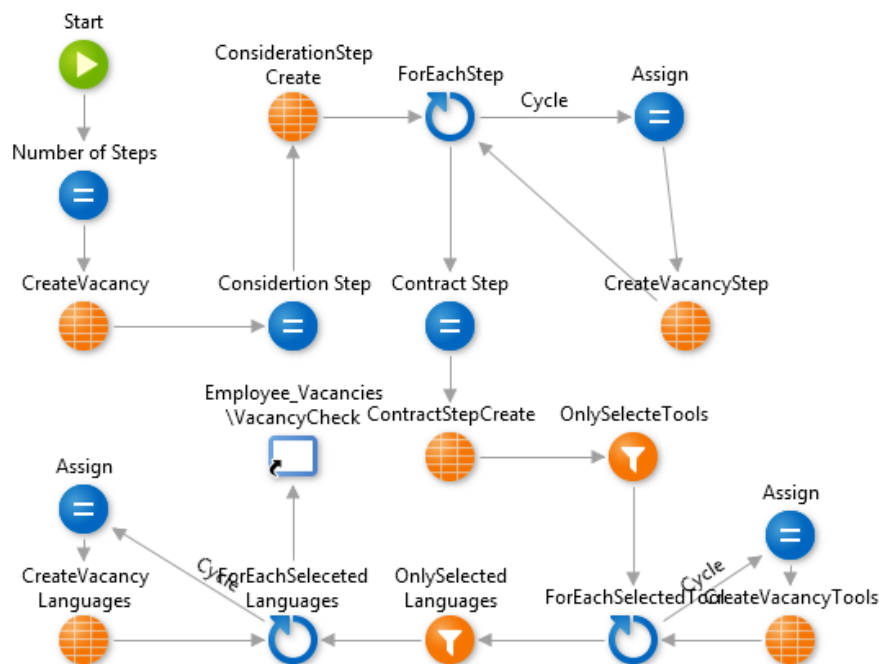
Código 12 - Adição de um novo Event

A *action* começa por obter a localidade escolhida pelo colaborador, a mensagem do *Pop-up* que é utilizada para definir a *Location* do *Event* a adicionar, cujos outros campos são depois declarados em função de outras informações estabelecidas pelo colaborador.

Todos os participantes do *Event*, com exceção o responsável, são informados do evento por *email*, como também, no caso de candidatos, são notificados pelas notificações da aplicação *mobile*.

3.4.7 Adicionar Oferta

A aplicação inclui uma página, acessível unicamente a colaboradores, que permite a introdução de uma nova vaga, definindo os diferentes passos do mesmo incluindo o formulário de cada e as várias ferramentas e linguagens cujo conhecimento é indispensável para o cargo anunciado pela vaga. A única ação a notar desta página é que a mesma permite adicionar a nova vaga em si, que pode ser verificada no Código 13.



Código 13 - Adição de Vacancy

Nesta ação, depois de ser criada a instância de *Vacancy* que representa a oferta a adicionar, são executados três ciclos diferentes. No primeiro acontece para a lista de passos adicionados, sendo adicionado um *VacancyStep* para cada passo

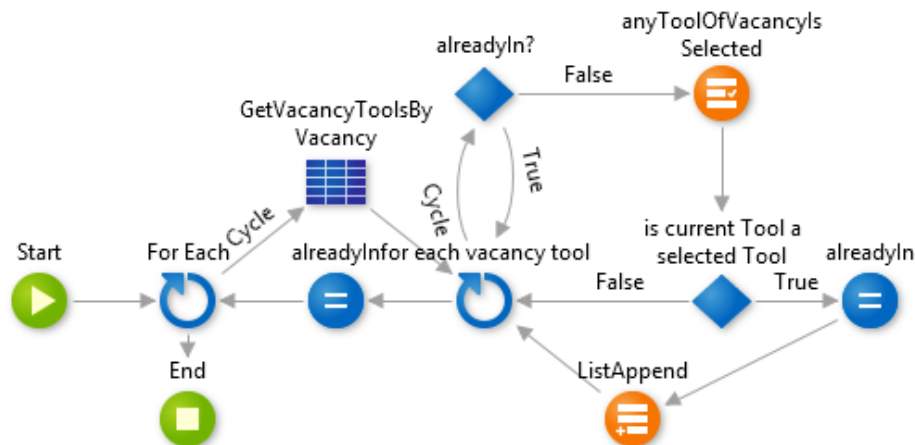
Depois todas as ferramentas que foram selecionadas, obtidas com o *OnlySelectedTools* que filtra a lista de ferramentas possíveis obtendo unicamente as selecionadas, são utilizadas para criar uma instância de *VacancyTools*.

Por último é realizado um ciclo semelhante ao anterior, mas em vez de ferramentas são utilizadas as linguagens selecionadas para produzir instâncias da entidade *VacancyLanguages*, O *OnlySelectedLanguages* tem uma funcionalidade idêntica a *OnlySelectedTools*, mas para as linguagens em vez das ferramentas.

3.4.8 Verificar percentagens de vaga

Na página da Figura 27 um colaborador pode verificar uma oferta, mais as candidaturas a esta. Com tais candidaturas, estas são demonstradas num *Pie Chart*, como se pode verificar na Figura 28. Os valores que preenchem este *Pie Chart* são formados pela *action* demonstrada na Código 14.

verificar se uma oferta inclui uma ferramenta ou linguagem selecionada são executadas duas *actions* semelhantes, com um exemplo verificado no Código 16.

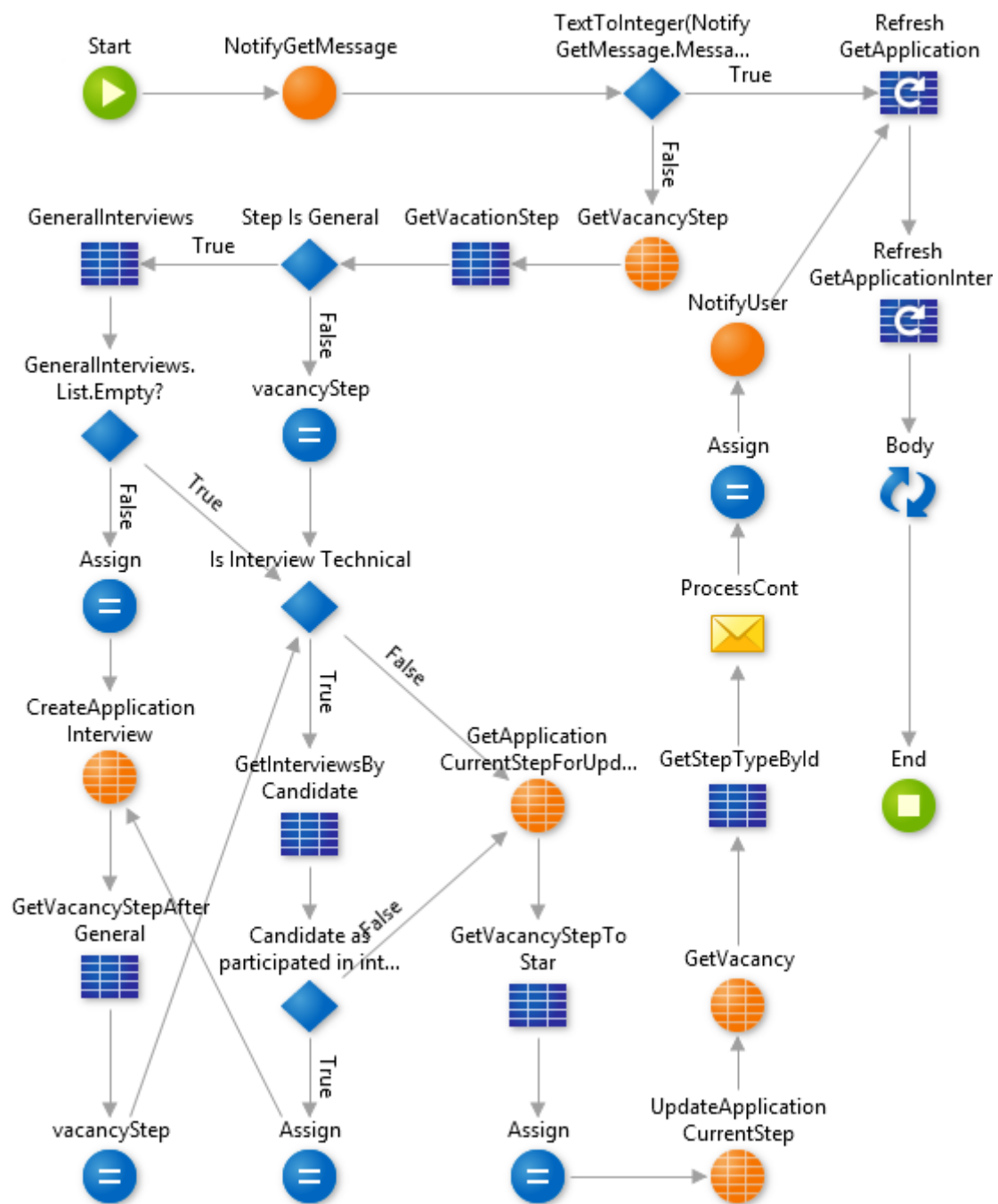


Código 16 - Verificar *Vacancy* em função das suas ferramentas

Esta verificação envolve a obtenção de todas as ferramentas ou linguagens de cada oferta obtidas até ao ponto. Se uma dessas for umas das ferramentas ou linguagens escolhidas pelo utilizador, então a ferramenta é adicionada a lista de ofertas que continua para o próximo passo da pesquisa.

3.4.10 Continuação da candidatura

Para garantir a inexistência de repetições de entrevistas o processo de continuar uma candidatura do passo atual para o próximo envolve algum código, mas especificamente o código visualizado no Código 17 que se segue.



Código 17 - Continuação duma Candidacy

Essencialmente o que acontece é se o próximo passo for uma entrevista geral e se o candidato já realizou uma entrevista geral, então o candidato pode saltar este passo e é iniciado um ciclo, que também acontece se o passo não for geral. Já no caso de o passo ser uma entrevista geral, mas o candidato nunca realizou uma entrevista geral a *action* passa para um ponto mais à frente em que define *VacancyCurrentStep* para o passo atual.

No ciclo já referido é verificado se o candidato já realizou uma entrevista com o mesmo formulário que é utilizado no passo, caso tal aconteça o candidato salta este passo e o ciclo continua, caso contrário o ciclo termina e a *action* define o *VacancyCurrentStep*.

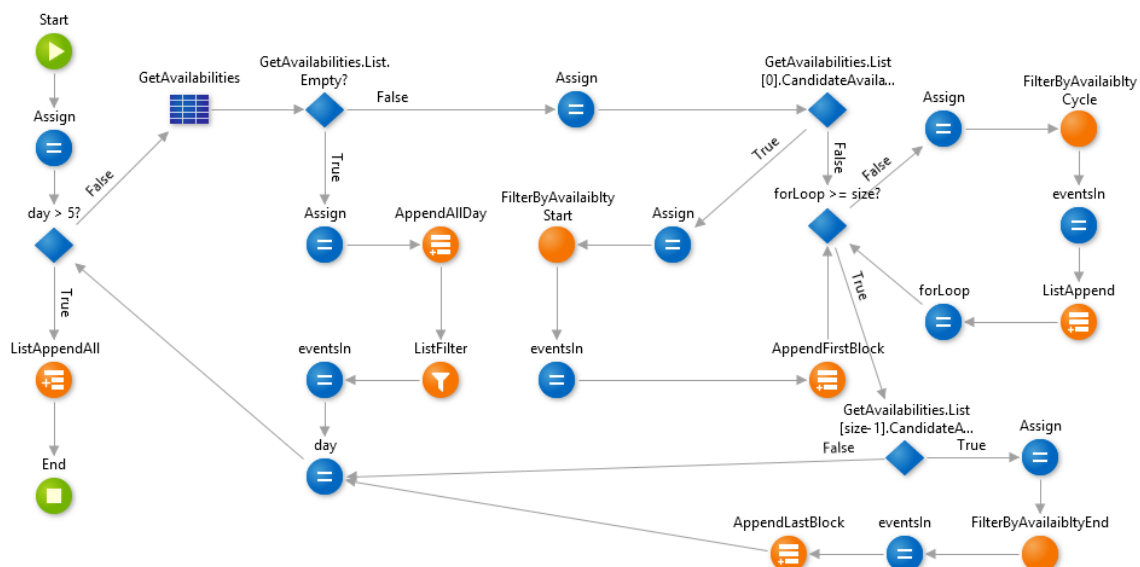
Tendo chegado ao passo em que o candidato ainda tem de realizar, a candidatura é atualizada para tal passo e o candidato é avisado da continuação da sua candidatura, tanto por *email* e por notificação, acabando de seguida a *action*.

Deve ser notado que o ciclo referido anteriormente tem sempre um final, sendo que na situação de um passo não incluir qualquer formulário o ciclo termina definindo a candidatura para tal passo. Sendo por *default* todos os passos de candidatura terminam num passo de contratação que não inclui um formulário, existe sempre um passo sem formulário.

3.4.11 Incluir disponibilidade

Como referido anteriormente um candidato é capaz de demonstrar a sua disponibilidade para entrevistas, o que permite interferir no mecanismo de marcação de eventos, quando o mesmo, o evento, envolve o candidato.

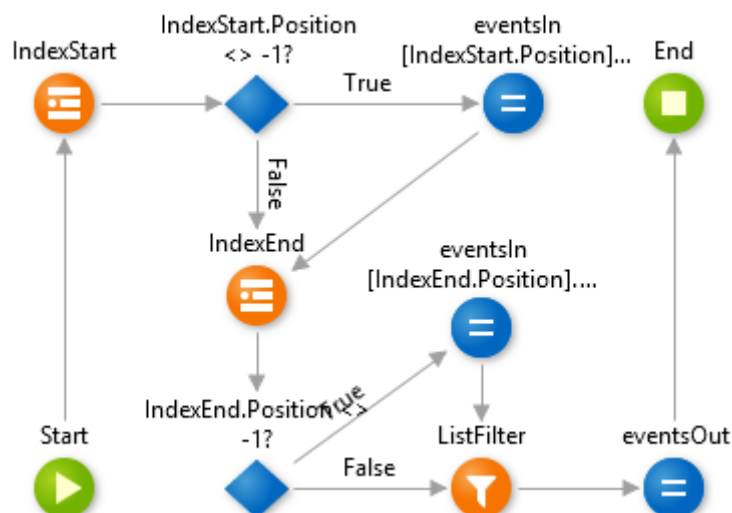
Por isso a aplicação permite a criação de blocos temporários, que podem ser utilizados para limitar a criação de novos Eventos. Para tal depois de se obter os eventos duma semana, usando o código verificado no Código 10, é realizada a *actions* verificada no Código 18.



Código 18 - Adicionar disponibilidade

Nesta *action* são obtidas todas as disponibilidades do candidato por cada dia da semana e para cada são introduzidos blocos fora da disponibilidade e é removido qualquer evento que ocorre dentro dos blocos de disponibilidade. Para o caso de não existir qualquer disponibilidade no dia então é criado um bloco que ocupa o dia inteiro e remove-se o todos os eventos do dia.

Se existir pelo menos um ou mais blocos de disponibilidade, é executado um ciclo que obtém um bloco de disponibilidade como também o bloco seguinte e é criado um bloco temporal que começa no fim do primeiro e acaba no início do seguinte, removendo todos os eventos nesse período usando a *action* verificado no Código 19.



Código 19 - Filtrar Eventos

Nesta *action* é verificado se existe qualquer evento que a *action* recebe, que comece ou acabe dentro de um bloco de disponibilidade, cuja duração é fornecida a *action*. Se existirem então estes são alterados das seguintes formas:

- Se o evento começar dentro do bloco, o seu início passa ao fim do bloco de disponibilidade a ser verificado;
- Se o evento acabar dentro do bloco o seu fim passa ao início do bloco de disponibilidade;
- Qualquer evento que começa e acaba dentro do bloco de disponibilidade é simplesmente removido.

3.5 Desenvolvimento Mobile

Como a secção anterior, esta secção demonstra as implementações que podem ser consideradas como mais notáveis no desenvolvimento da componente *mobile* do projeto. Estas implementações são restringidas às notificações e ao sincronismo dos dados entre a base de dados local e remota.

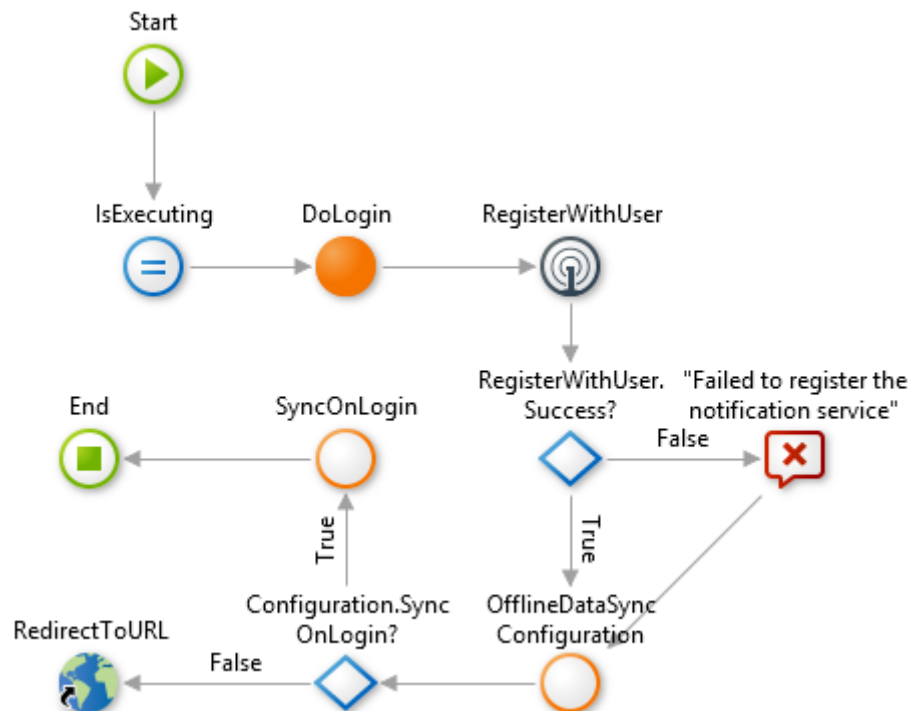
3.5.1 Notificações

Como já foi estabelecido, a aplicação *mobile* inclui nas suas funcionalidades a visualização de notificações que irão demonstrar informações importantes. Para realizar esta funcionalidade foi escolhido a extensão *OneSignal* que permite realizar as notificações desejadas com pouca adição de lógica ou peso para a aplicação.

Os únicos requisitos para utilizar esta extensão são o facto que a aplicação tem de ser registada nos serviços da *OneSignal*, o que ocorre fora da aplicação, que cada utilizador tenha o seu dispositivo registado no mesmo servidor, associando os mesmos à aplicação já registada.

Por isso a ação de *login* que ocorre na aplicação *mobile*, que pode ser averiguada no Código 20, inclui a utilização da ação *RegisterWithUser* que realiza o registo do utilizador nos servidores

de *OneSignal* utilizando o identificador da aplicação *mobile* desenvolvida, obtida no registo da mesma no servidor da *OneSignal*.



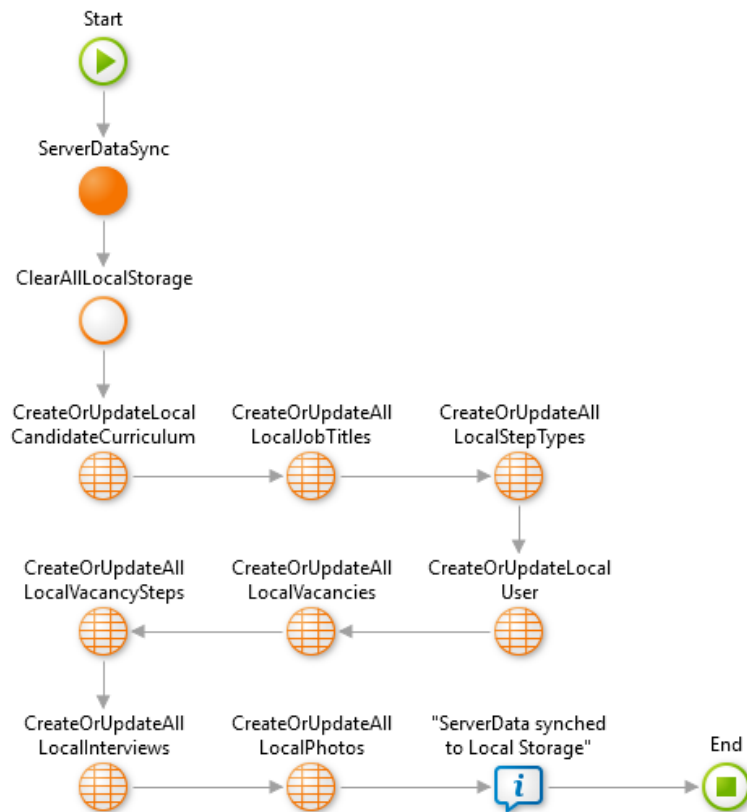
Código 20 - Registo de utilizador na *OneSignal*

Também se pode verificar que é incluindo uma verificação de erro depois do registo, garantindo assim que se o registo não ocorrer, o utilizador é informado da situação utilizando o uma mensagem de erro.

3.5.2 Sincronismo

Na secção 2.3 foi referido o conceito de sincronismo que foi aplicado na vertente *mobile* do projeto, sendo a mesma bastante simples e leve. Esta sincronização ocorrera em dois padrões: a primeira, e mais simples das duas, *Read-Only*, ocorre pela ação *SyncOnLogin*, que ocorre no *login*, como se pode verificar no Código 20 - Registo de utilizador na *OneSignal*.

Esta ação, que pode ser verificada no Código 21, começa por utilizar a ação *ServerDataSync* que obtém todos os dados aos quais a aplicação acede à base de dados ou a *local storage* já estabelecida, caso não exista a um acesso a internet. Tendo as informações a sincronizar, os dados da *local storage* são limpos para serem restabelecidos com as várias ações *CreateOrUpdate* que seguem.



Código 21 - Sincronismo *Read-Only*

A única situação, já estabelecida, que não envolve um sincronismo do tipo *Read-Only*, ocorre quando um colaborador quer alterar o seu currículo. Sendo uma escrita em que não existe conflito, é utilizado o padrão de sincronismo *Read/Write Data Last Write Wins*. Como se pode observar, o currículo é um dos vários valores adicionados ao *local storage*, por isso a alteração do currículo não só ocorre na base de dados, mas também ocorre no *local storage*.

4 Avaliação Experimental

Neste capítulo, iremos apresentar uma série de casos de teste que têm como objetivo demonstrar as funcionalidades principais do IView, através de uma série de *screen shots* da aplicação *web* e *mobile*, das alterações realizadas na base de dados, e por último, nas situações em que ocorram, dos *emails* enviados pela aplicação. Para isso, cada uma das funcionalidades apresentadas, inclui-se um teste onde são verificados os valores fornecidos pelo utilizador, e o resultado esperado na aplicação, na base de dados.

4.1 Candidatura Espontânea – Utilizador não registrado

Um utilizador ainda não registrado pode realizar uma candidatura espontânea que poderá resultar no seu registo na aplicação como candidato.

Ação

Para se inscrever na aplicação, um utilizador deve fornecer quatro informações: o seu nome, o seu *email*, o seu número de telemóvel e um *.pdf* que serve como o seu currículo, como se pode ver na Figura 32.

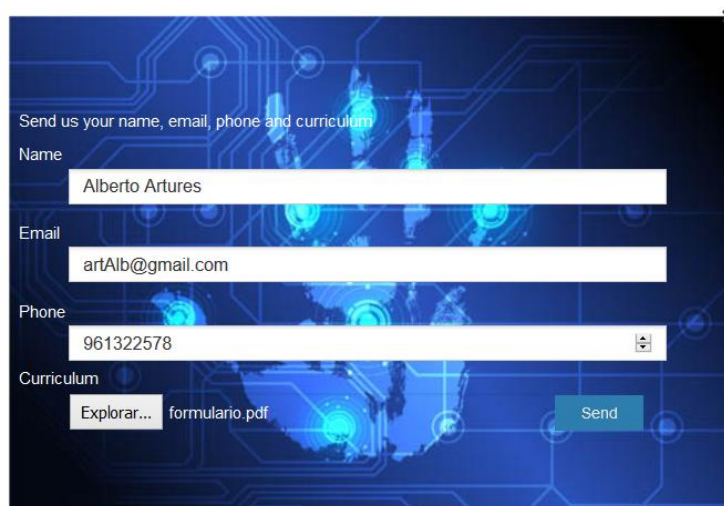
A screenshot of a web application interface for a spontaneous application. The background is dark blue with a circuit-like pattern. The form is white and contains the following fields: 'Name' with the value 'Alberto Artures', 'Email' with the value 'artAlb@gmail.com', 'Phone' with the value '961322578', and 'Curriculum' with a file selection button labeled 'Explorar...' and the filename 'formulario.pdf'. A 'Send' button is located at the bottom right of the form. Above the form, the text 'Send us your name, email, phone and curriculum' is displayed.

Figura 32 - Candidatura Espontânea

Resultado Esperado

Esta candidatura resulta na criação das instâncias *SpontaneousCurriculum* e *SpontaneousCurriculumFile*, Figura 33, que são demonstradas na tabela da Figura 34 e que foi enviado o *email* da Figura 35. No caso de erro, um *pdf* não é fornecido ou no caso de a candidatura ser registrada corretamente, uma das mensagens da Figura 36.

SpontaneousCurriculum Id	SpontaneousCurriculum Name	SpontaneousCurriculum Phone	SpontaneousCurriculum Email	SpontaneousCurriculum Date
13	Alberto Artures	961322578	artAlb@gmail.com	2018-07-04

SpontaneousCurriculumFile SpontaneousCVsid	SpontaneousCurriculumFile File
13	BinaryData (222424 bytes)

Figura 33 - Introdução de SpontaneousCurriculum e de SpontaneousCurriculumFile

Curriculums To Check					
Name	Phone	Email	Curriculum	Accept	Refuse
Tomas Lois	960000101	lois@gmail.com	Download	Accept	Refuse
Catia Lurdes	960000102	catlond@hotmail.com	Download	Accept	Refuse
Alberto Artures	961322578	artAlb@gmail.com	Download	Accept	Refuse

Figura 34 - Candidaturas Espontâneas

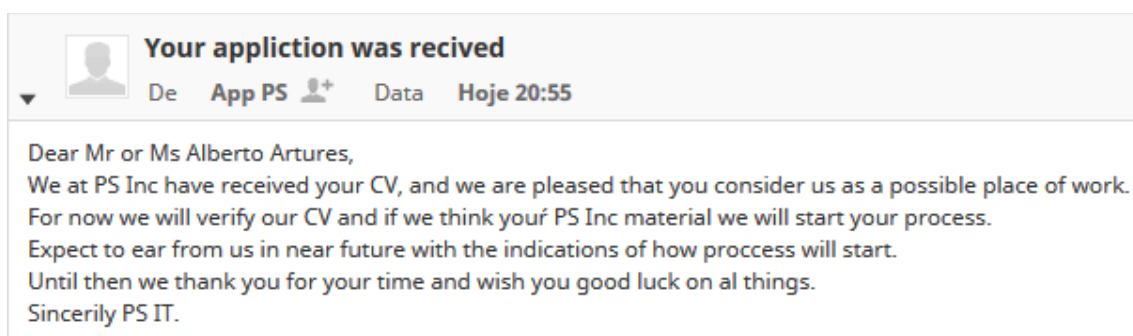


Figura 35 - Email de submissão da candidatura espontânea



Figura 36 - Mensagem de erro e de sucesso da candidatura

4.2 Estabelecer / Recusar novos candidatos – Colaborador

A lista verificada na Figura 34 não só mostra as candidaturas espontâneas ainda a tratar, como também permite estabelecer um novo candidato, pelo botão “Accept” ou terminar a candidatura pelo botão “Refuse”.

Ação

O colaborador não precisa de fornecer qualquer informação para estabelecer o novo candidato, só precisa de escolher que candidatura quer aceitar, criar utilizador para o candidato, ou recusar.

Resultado Esperado

Com a ação do “Accept” a instância de *SponatnousCurriculum* é alterado para incluir o valor de segurança, como se pode verificar na **Erro! A origem da referência não foi encontrada.** como também no envio do *email* da Figura 38. Já a ação do “Refuse” simplesmente resulta no enviou do *email* da Figura 39.

SpontaneousCurriculum Id	SpontaneousCurriculum Name	SpontaneousCurriculum Phone	SpontaneousCurriculum Email	SpontaneousCurriculum Date
15	Alberto Artures	961322578	artAlb@gmail.com	2018-07-10

SpontaneousCurriculum SetValue	SpontaneousCurriculum SetDate
PA5udUomMxsZ1teHaC9e	2018-07-10

Figura 37 - Alteração sobre SponatnousCurriculum

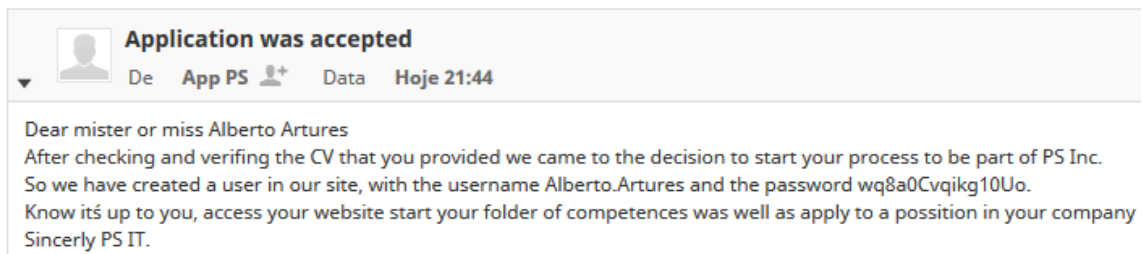


Figura 38 - Email de estabelecimento do candidato

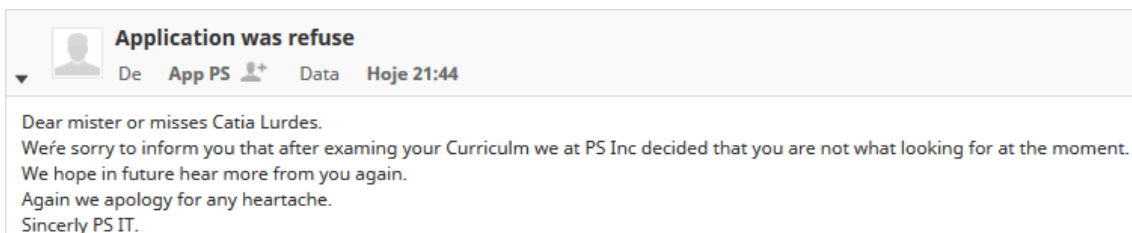


Figura 39 - Email de recusa da candidatura espontânea

4.3 Estabelecer novo candidato – Utilizador não registrado

Tendo recebido o *email* da Figura 38, um utilizador ainda não registrado pode aceder a uma página onde lhe será possibilitado o seu estabelecimento como candidato da aplicação.

Ação

A página referida é composta unicamente por um *form* onde o utilizador tem de fornecer o seu *email*, o *email* associado à sua candidatura espontânea, e repetir a sua password duas vezes, a página pode ser verificada na Figura 40.

New Candidate	
Email	artAlb@gmail.com
Password
Confirm Password
Create User	

Figura 40 - Criação do novo utilizador

Resultado Esperado

Se os valores fornecidos estiverem corretos, então uma instância da entidade *User* e outra da entidade *CandiadateCurriculum* serão iniciadas, Figura 37 e Figura 42, como também é enviado o *email*, da Figura 43, onde é incluindo o *username* e *password* do novo *user*.

User Id	User Name	User Username	User Password	User Email	User MobilePhone
32	Alberto Artures	Alberto.Artures	78EA224EF523E6D662EFF18920422067	artAlb@gmail.com	961322578

Figura 41 - Estabelecimento de User

CandidateCurriculum UserId	CandidateCurriculum Name	CandidateCurriculum Address	CandidateCurriculum DateOfBirth	CandidateCurriculum Accessibility	CandidateCurriculum Availability
32	Alberto Artures		1900-01-01		

Figura 42 - Estabelecimento de CandidateCurriculum

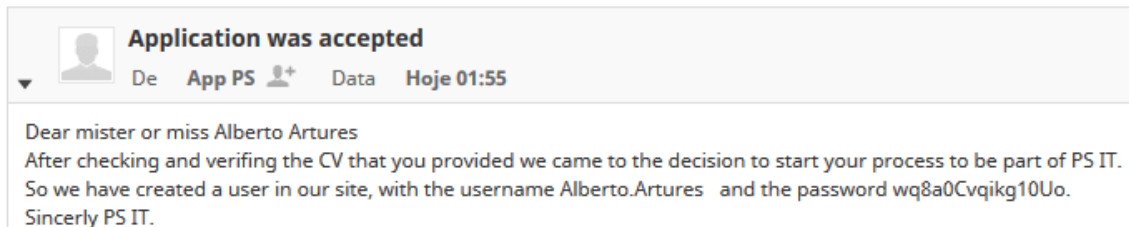


Figura 43 - Email com novo utilizador

No caso de algum erro, o *email* não ser o associado à candidatura, ou que o valor de *Password* e o valor de confirmação não forem iguais, então uma das mensagens da Figura 44 é lançada.



Figura 44 - Mensagens de erro

4.4 Autenticação de utilizador – Utilizador

Um utilizador estabelecido pode ser autenticado pelo o seu *username* e *password*, tendo assim acesso as várias capacidades do seu papel na aplicação.

Ação

Pela página verificada na Figura 45, o utilizador consegue fornecer os seus valores de autenticação. Este é depois redirecionado, se os valores forem corretos, para a página principal da aplicação *web*. Já nesta página (Figura 46) é possível verificar o *screen* de autenticação da aplicação *mobile*.

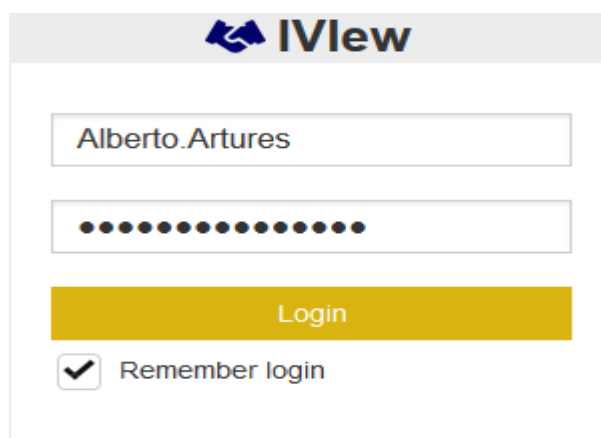


Figura 45 - Autenticação Web

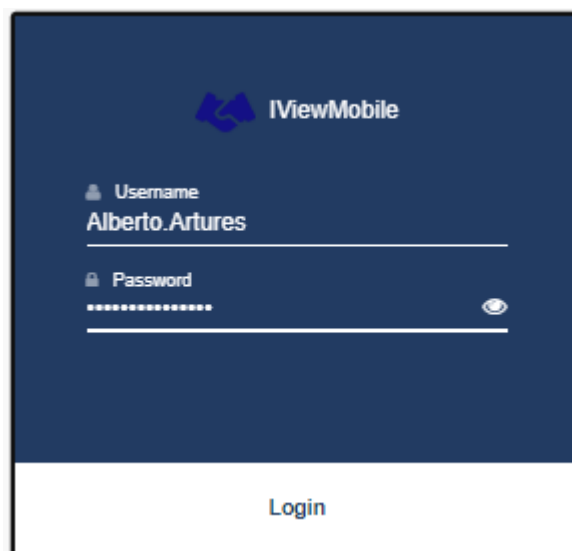


Figura 46 - Autenticação Mobile

4.5 Atualizar Currículo – Candidato

O candidato tem a capacidade de introduzir, remover e alterar os vários componentes que formam o seu dossier de capacidades, demonstrando assim as suas capacidades e experiências na área da tecnologia como também alguma informação fora da área.

Ação

Para realizar esta alteração, o candidato tem de fornecer os dados que quer alterar na página da Figura 47, ou no *screen* da Figura 48.

A Figura 47 não demonstra por completo a página do currículo, sendo que a mesma é extensa demais para demonstrar numa figura. O que a figura demonstra são os primeiros elementos do currículo.

Também podemos observar que a alteração do currículo realizado pela *web* ocorre antes da alteração pela aplicação *mobile*, daí os valores alterados na primeira podem ser verificados na segunda.

Figura 47 - Alterar Currículo, Web

Figura 48 - Alterar Currículo, Mobile

Resultado Esperado

Ao alterar o currículo, a instância da entidade *CandidateCurriculum* associado ao candidato Alberto Artures, foi alterada duas vezes, para os valores verificados na Figura 49, que são demonstrados na página da Figura 50 e no *screen* da Figura 51

CandidateCurriculum UserId	CandidateCurriculum Name	CandidateCurriculum Address	CandidateCurriculum DateOfBirth	CandidateCurriculum Accessibility	CandidateCurriculum Availability
32	Alberto Artures	Rua Vasco da Gama 32, 4ºEsq	2012-11-06	Lisboa	Right now
32	Alberto Artures	Rua Vasco da Gama 75, 6C	2012-11-22	Lisboa	Right now

Figura 49 – Alterações a *CandidateCurriculum*

Figura 50 – Currículo depois da primeira alteração

Figura 51 - Currículo depois da segunda alteração

4.6 Atualizar *Profile* – Candidato

O candidato tem a capacidade de introduzir, remover e até alterar os vários componentes que formam o seu dossier de capacidades, demonstrando assim as suas capacidades e experiências na área da tecnologia como também alguma informação fora da área.

Ação

Existem vários valores que pertencem ao dossier, que o candidato pode alterar. Mas só serão verificados dois deles: as tecnologias e os projetos, sendo que grande parte dos componentes do dossier tem uma introdução ou alteração igual ao primeiro, com o segundo o único elemento que não segue o padrão. Para realizar qualquer alteração na lista de tecnologias o candidato simplesmente tem de introduzir os valores corretos, como é verificado na Figura 52.

Figura 52 - Introduzir Tecnologia

Já para introduzir ou alterar um projeto é utilizado uma *Edit Records* que no caso do verificado na Figura 53, pode aparecer ou num *Pop-up* quando o projeto é para adicionar numa lista, quando é para alterar.

Name

A project

✕

Start *

2018-07-11

End *

2018-07-13

Responsibilities

Everthing

Accomplishments

Finish it

Technologies *

Java

Summarization *

A simple java project

Add

Cancel

Figura 53 - Introduzir Projeto

Resultado Esperado

As alterações anteriores resultaram na introdução de uma instância da entidade *CandidateTechnologies*, Figura 54, e uma da entidade *CandidateProjects*, Figura 55, que podem também ser verificadas nas tabelas da Figura 56 e da Figura 57.

CandidateTechnology ExperiencePratical	CandidateTechnology ExperienceAcademic	Capacity Label	Tools Label
2	2	Competent	JQuery

Figura 54 - Alteração a *CandidateTechnology*

CandidateProject Candidate	CandidateProject Name	CandidateProject StartDate	CandidateProject EndDate	CandidateProject Responsibilities	CandidateProject Technologies	CandidateProject Accomplishments	CandidateProject Summarization
32	A project	2018-07-11	2018-07-13	Everthing	Java	Finish it	A simple java project




Figura 55 - Alteração a *CandidateProject*

My Technologies

Technology	Pratical Experience	Academic Experience	Capacity
JQuery	-1	2	Competent
Add New Technology			

Figura 56 - Tecnologias do Candidato

My Projects Developed +

A project   

Start Date	2018-07-11
End Date	2018-07-13
Responsibilities	Everthing
Technologies	Java
Accomplishments	Finish it
Summarization	A simple java project

Figura 57 - Projetos do Candidato

Estas ações podem levantar alguns erros. Nas tecnologias, um candidato pode tentar inserir experiências negativas, o que não é possível, ou tentar introduzir uma tecnologia já existente. Nestas situações as mensagens de erro da Figura 58 aparecem. Deve ser notado que a segunda mensagem pode ser lançada com a alteração duma tecnologia para outra já existente.

✖ Pratical Experince Must Be Possitive ✖

✖ You Already Added The Technology JQuery ✖

Figura 58 - Mensagens de Erro de Tecnologias

Já os projetos só lançam uma mensagem de erro, quando o candidato tenta criar, ou alterar, um projeto de forma que o início do projeto aconteça depois do seu fim, tal mensagem pode ser verificada na Figura 59.

✖ End date of a Project must be after start date ✖

Figura 59 - Mensagens de Erro de Projetos

4.7 Atualizar Disponibilidade – Candidato

Por esta pagina, qualquer candidato pode estabelecer a sua disponibilidade para participar em qualquer entrevista, adicionado ou alterando blocos temporais que associam dois tempos (início e fim do bloco) a um dia da semana.

Ação

O candidato consegue estabelecer o bloco pela a página verificada na Figura 60, onde é possível escolher o início do bloco, o fim do mesmo e um dia da semana de um conjunto de valores pré-definidos, pela a utilização de *selects*.

My Availability

Day *	Start *	End *
Monday	10:00:00	11:30:00
Tuesday	08:00:00	10:00:00
Wednesday	12:00:00	15:00:00


 Add Record

Figura 60 - Inserir Disponibilidades

Resultado Esperado

Ao contrário da capacidade anterior, introduzir valores nesta tabela não resultam unicamente na introdução de uma instância na base de dados, sendo que um valor introduzido poder interferir com valores já existentes, o que resulta na alteração dos valores já estabelecidos.

Para testar foram introduzidos 4 blocos, um na segunda (*Monday*) das nove as quatorze, terça(*Tuesday*) das nove as doze, quarta(*Wednesday*) das dez as doze e meia e, por último, na quinta(*Thursday*) das dez as quinze. Estas introduções resultam nas instâncias da *CandidateAvailability* verificadas na Figura 61 e na Figura 62.

My Availability		
Day	Start	End
Monday	10:00:00	11:30:00
Tuesday	08:00:00	10:00:00
Wednesday	12:00:00	15:00:00
Thursday	08:30:00	10:00:00
Add Record		

Figura 61 - Disponibilidades depois da introdução

CandidateAvailability	CandidateAvailability	CandidateAvailability	CandidateAvailability
Candidate	Day	StartTime	EndTime
12	1	09:00:00	14:00:00
12	2	08:00:00	12:00:00
12	3	10:00:00	15:00:00
12	4	10:00:00	15:00:00

Figura 62 - Alteração a *CandidateAvailability*

É possível que a introdução de uma disponibilidade demonstre um bloco malformado, com o início a acontecer depois do fim o que resulta na mensagem de erro da Figura 63.



Figura 63 - Mensagem de erro ao inserir disponibilidade

4.8 Estabelecer Empresa – Administrador

Os administradores da IView podem estabelecer empresas parceiras, ou seja, empresas que participam em projetos que PS IT também participa. Estes elementos também podem demonstrar os clientes da PS IT, informação geral que pode ser verificada por qualquer visitante ao site.

Ação

Para estabelecer uma nova empresa um administrador só precisa de fornecer um nome, uma localidade e uma breve descrição da empresa. Também pode, mas não é obrigatório, fornecer uma fotografia. No caso de não ser fornecida, uma imagem *default* é utilizada. A introdução duma nova empresa pode ser verificada na Figura 64.

Figura 64 - Introduzir Empresa

Resultado Esperado

Com a introdução duma empresa, uma instância da entidade *Company* é criada, como também uma da entidade *Locality*, que representa a localidade fornecida, no caso da introdução anterior as instâncias criadas são as verificadas na Figura 65.

Company Name	Company Locality	Company Information	Company Since	Location Local
Company	Rua Mouco 74, Lisboa	A company for test	2018-07-05	Rua Mouco 74, Lisboa

Figura 65 – Nova instância de *Company*

Pela Figura 66, Figura 67 e Figura 68 é possível verificar a página completa da empresa criada, que inclui a informação geral, os projetos onde é possível adicionar um novo projeto, indiretamente, e por último as localidades, onde é possível adicionar novas localidades, como também alterar já existentes, duma forma direta.

Figura 66 - Empresa introduzida, Informação geral

Figura 67 - Empresa Introduzida, Projectos

Figura 68 - Empresa Introduzida, Localidades

4.9 Introduzir Projeto – Colaborador

No subcapítulo anterior foi referido o facto que as instâncias da entidade *Company* representam empresas parceiras, participando em projetos com a IView. Tais projetos podem ser adicionados

pelo colaborador como agora vai ser verificado. Estes mesmos projetos podem depois ter um ou mais ofertas associadas.

Ação

Para adicionar um projeto um colaborador deve fornecer não só a informação geral do projeto, pela *tab* representada na Figura 69, como também um ou mais responsáveis, Figura 70. A cada responsável pode ser associado uma empresa, utilizando o *Pop-up* demonstrado na Figura 71.

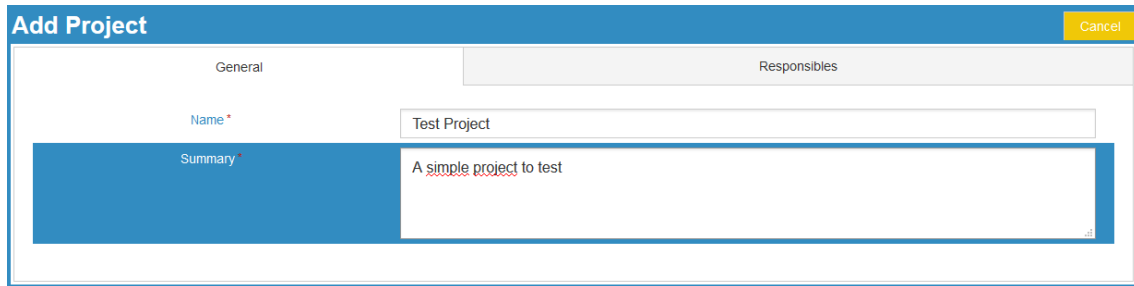
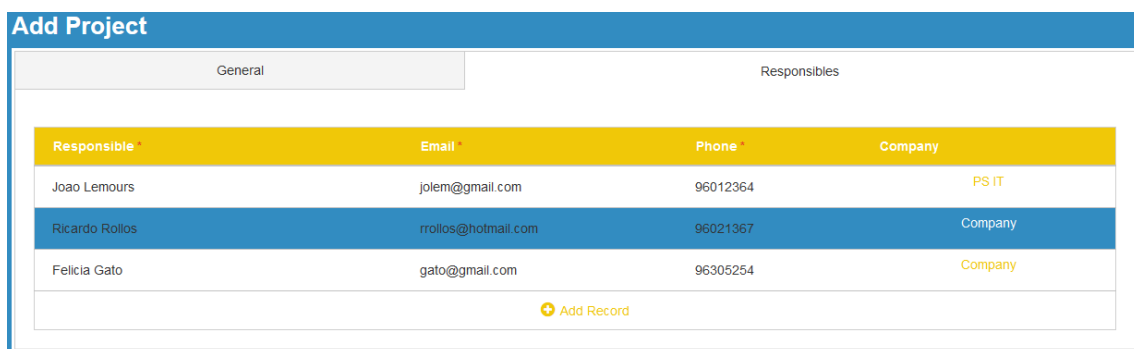


Figura 69 - Introduzir Projeto, informação geral



Responsible	Email	Phone	Company
Joao Lemours	jolem@gmail.com	96012364	PS IT
Ricardo Rolos	rrollos@hotmail.com	96021367	Company
Felicia Gato	gato@gmail.com	96305254	Company

Figura 70 - Introduzir Projeto, responsáveis

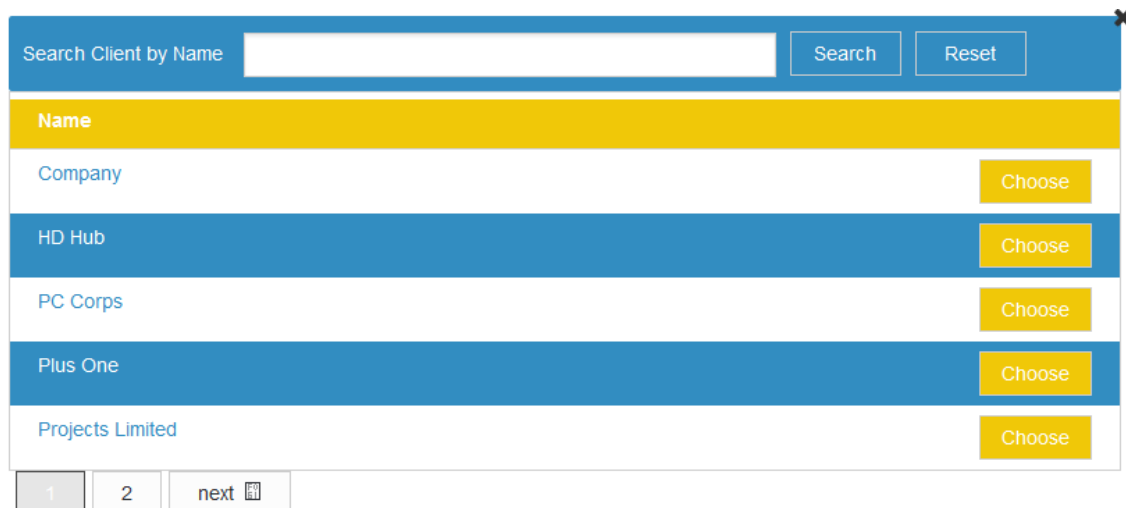


Figura 71 - Adicionar Projeto, escolher empresa de responsável

Resultado Esperado

A introdução de um projeto leva à criação de instâncias da entidade *Project*, duas instâncias de *ProjectCompany*, Figura 73, uma para cada empresa participante (incluindo PS IT) e três instâncias da entidade *ProjectResponsible*, uma para cada responsável incluído na adição anterior, figura Figura 74. Apesar de serem atribuído três responsáveis, só são associadas duas

empresas ao projeto, sendo que Ricardo Rollos e Felicia Gato partilham o mesmo *ProjectCompany*.

Tendo em conta que um dos responsáveis é da empresa *Company*, a página da mesma já inclui o projeto criado, como se pode verificar na página representada na Figura 75.

Project Name	Project Summary
Test Project	A simple project to test

Figura 72 – Nova Instância de Project

Project Name	Company Name
Test Project	Company
Test Project	

Figura 73 – Novas Instâncias de ProjectCompany

ProjectResponsible ProjectClient	ProjectResponsible Responsible	ProjectResponsible Email	ProjectResponsible Phone
49	Joao Lemours	jolem@gmail.com	96012364
50	Ricardo Rollos	rrollos@hotmail.com	96021367
50	Felicia Gato	gato@gmail.com	96305254

Figura 74 – Novas Instâncias de ProjectResponsible

The screenshot shows a web application interface for a company. At the top, there's a blue header with the word 'Company'. Below it, there are three tabs: 'General Info', 'Projects', and 'Location'. The 'Projects' tab is active. Inside the 'Projects' tab, there's a search bar with 'Test Project' and a magnifying glass icon. Below the search bar, the project details are shown: 'Test Project' and 'A simple project to test'. Underneath, there's a section titled 'Participating Companies' which displays two logos: one for 'COMPANY' (a blue cube) and one for 'PSIT' (a yellow square with a blue 'PSIT' text and a Wi-Fi symbol).

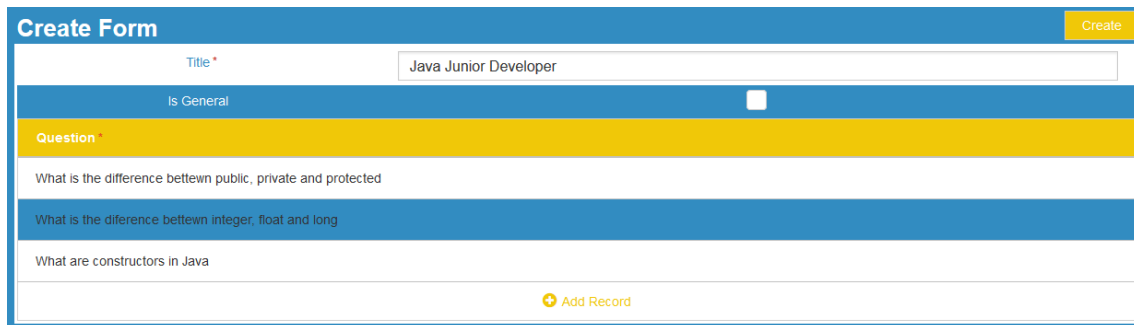
Figura 75 – Projeto na lista da empresa

4.10 Introduzir Form – Colaborador

Um formulário é uma peça importante do estabelecimento do processo de contratação de uma oferta, sendo que não só pode servir como uma indicação das capacidades necessárias para a oferta, como também permite diminuir o número de entrevistas necessárias, sendo que num processo de entrevista, a aplicação considera entrevistas já realizadas, com a consideração sendo realizada em função do formulário em si. Estes formulários podem ser criados e alterados por qualquer colaborador como for necessário.

Ação

Para criar um formulário um colaborador deve fornecer o nome e as várias questões que compõem o formulário, como se pode verificar na Figura 76. O nome fornecido deve indicar até certo ponto que tipo de capacidade que a utilização do formulário procura e, se possível, a tecnologia principal.



Create Form Create

Title *

Is General ☒

Question *

What is the difference between public, private and protected

What is the difference between integer, float and long

What are constructors in Java

[+ Add Record](#)

Figura 76 - Criar formulário

Resultado Esperado

Ao criar o formulário anterior, uma instância da entidade *Form* foi criada, Figura 77. Para representar o cabeçalho do formulário que é referida pelas três novas instâncias da entidade *FormQuestions* que foram criadas, para representar cada questão introduzida, Figura 78.

Form NumberOfQuestions	Form Title	Form Created	Form isGeneral
3	Java Junior Developer	2018-07-06	false

Figura 77 - Nova instância de *Form*

Form Title	FormQuestion Question
Java Junior Developer	What are constructors in Java
Java Junior Developer	What is the difference between integer, float and long
Java Junior Developer	What is the difference between public, private and protected

Figura 78 Novas instâncias de *FormQuestions*

4.11 Introduzir Oferta – Colaborador

Um dos focos principais da aplicação é capacidade de estabelecer novas ofertas que podem estar, ou não, associadas a projetos estabelecidos.

Ação

O estabelecimento duma oferta não só envolve estabelecer a informação geral da oferta, a suas necessidades, o seu projeto, as tecnologias e linguagens procuradas, como também são estabelecidos os passos do processo de contratação para preencher a oferta o que pode, ou não, envolver o estabelecimento do formulário para o passo.

Pela Figura 79 é possível verificar o *form* para a informação geral, que envolve um *link* para o *Pop-up* da Figura 80, que permite escolher o projeto a associar a entrevista. Pode ser notado que é possível estabelecer uma oferta através da página do projeto diretamente.

The 'Add Offer' form has three tabs: 'Main', 'Steps', and 'Tools and Languages'. The 'Main' tab is active. It contains the following fields:

- Title ***: Java Junior Developer
- Number Of People ***: 4
- Position ***: Developer
- Requests ***: Capable with java and if possible SQL
- Project**: Test Project

Figura 79 - Estabelecer Oferta, informação geral

The 'Add Offer' form has three tabs: 'Main', 'Steps', and 'Tools and Languages'. The 'Main' tab is active. It contains the following fields:

- Search Project:** [Text Input] by Company [Checkbox] [Search] [Reset]
- Name**: A simple App
- Summary**: A simple App for all companies
- Name**: Data Base Migration
- Summary**: The transformation of XML database to SQL
- Name**: Music App
- Summary**: A music mobile app to develop in java
- Navigation**: 1 2 next

Figura 80 - Estabelecer Oferta, escolha de projeto

Já pela Figura 83 é possível verificar os passos da entrevista, que pode utilizar o *Pop-up* da Figura 82, para escolher o formulário para o passo corrente. Uma oferta não pode ser adicionada até ser estabelecido pelo menos um passo.

Por último, pela Figura 83, o colaborador pode escolher utilizando dois *selects* todas as ferramentas e linguagens a associar à oferta.

The 'Add Offer' form has three tabs: 'Main', 'Steps', and 'Tools and Languages'. The 'Tools and Languages' tab is active. It contains the following fields:

- Step Type ***: General Interview, Technical Interview, Technical Interview, Client Interview
- Form**: General Questions, Java Junior Developer, SQL Data Base Designer, Blank Form
- Add Record**: [Button]

Figura 81 - Estabelecer Oferta, tecnologias e linguagens

Search Form by Title

Search

Reset

Title	Number Of Questions	Created	Chose
Introductory Questions	4	2018-04-03	<div>Choose</div>
Java Developer	4	2018-04-03	<div>Choose</div>
CSS Designer	3	2018-04-03	<div>Choose</div>
OutSystems Consultant	3	2018-04-03	<div>Choose</div>
.NET Developer	2	2018-04-03	<div>Choose</div>

1

2

next

Figura 82 - Estabelecer Oferta, escolher formulário

Add Offer

Main

Steps

Tools and Languages

Select Tools

✖ Java

✖ SQL

✖ Eclipse

Select Languages

✖ Português

✖ English

Figura 83 - Estabelecer Oferta, tecnologias e linguagens

Resultado Esperado

Com a ação anterior é introduzida na base de dados: uma instância de *Vacancy* que representa a informação geral da oferta, Figura 84; seis instâncias de *VacancyStep* que inclui os quatro passos introduzidos mais o passo de consideração, passo inicial e o passo de contratação o passo final, Figura 85; duas instâncias da *VacancyLanguages* uma para cada linguagem escolhida e três instâncias da *VacancyTools* uma para cada ferramentas escolhida, Figura 86.

Vacancy Title	Vacancy NumberSteps	Vacancy Requests	Vacancy NumberOfPeople	Project Name
Java Junior Developer	4	Capable with java and if possible SQL	4	Test Project

Figura 84 - Nova instância de *Vacancy*

Vacancy Title	VacancyStep Step	StepType Label	Form Title	Vacancy NumberOfPeople
Java Junior Developer	0	Consideration		4
Java Junior Developer	1	General Interview	General Questions	4
Java Junior Developer	2	Technical Interview	Java Junior Developer	4
Java Junior Developer	3	Technical Interview	SQL Data Base Designer	4
Java Junior Developer	4	Client Interview		4
Java Junior Developer	5	Contract		4

Figura 85 - Novas instâncias de *VacancySteps*

Vacancy Title	Tools Label	Vacancy Title	Language Label
Java Junior Developer	Java	Java Junior Developer	Português
Java Junior Developer	SQL	Java Junior Developer	English
Java Junior Developer	Eclipse		

Figura 86 - Novas instâncias de *VacancyTools* e *VacancyLanguages*

4.12 Pesquisar/Aplicar a oferta – Candidato

Para tornar a candidatura a uma oferta mais acessível aos desejos de qualquer candidato, a aplicação inclui um sistema de pesquisa de ofertas que permite colocar filtros em função dos desejos do candidato. As ofertas verificadas com a pesquisa não são todas as ofertas existentes, mas sim todas as ofertas existentes a que o candidato ainda não se candidatou.

Ação

Para filtrar as ofertas, um candidato precisa apenas de escolher o tipo de trabalho que procura dos valores reconhecidos, utilizando um *select*. Também é possível adicionar a esse filtro, um novo filtro em função de tecnologias e linguagens.

A Figura 87 mostra o resultado da pesquisa quando é procurado qualquer oferta para uma posição como *Developer*, já a Figura 88 adiciona a tal pesquisa o filtro de linguagens e por último a Figura 89 demonstra a primeira pesquisa mais o filtro por tecnologias.

The screenshot shows the 'Offers Search' interface. At the top, there's a header 'Offers Search' with a 'Filter Off' button. Below it, a form allows filtering by job type (set to 'Developer'), tools, and language. The results section, titled 'Offers', lists two job offers: 'Android Developer' and 'Java Junior Developer'. Each offer shows its requirements: 'Experience in developing Android Apps' for Android and 'Capable with java and if possible SQL' for Java.

Figura 87 - Pesquisa por posição

This screenshot is similar to the previous one but includes an additional filter. In the 'Filter by language' section, 'English' has been selected, indicated by a small 'x' icon. Consequently, the results section now only displays the 'Java Junior Developer' offer, as the 'Android Developer' offer likely does not match the 'English' language filter.

Figura 88 - Pesquisa por posição e linguagem

Offers Search Filter Off

Choose the type of job: Developer

Filter by tools: Android

Filter by language:

Offers

Android Developer ➔

Requirements:
Experience in developing Android Apps

Figura 89 - Pesquisa por posição e tecnologia

Resultado Esperado

Quando um candidato se candidata a uma vaga, é criada uma instância da entidade *Candidacy*, Figura 90, que representa a candidatura e uma da entidade *CandidacyStep*, Figura 91, que representa a situação da candidatura. Note-se que as instâncias de *CandidacySteps* começam sempre no passo “*Consideration*” e no estado “*ToProcess*”.

[[[Candidacy Id	[[[Vacancy Title	[[[CandidateCurriculum Name
445	Java Junior Developer	Ricardo Raizes

Figura 90 - Nova instância de *Candidacy*

[[[Candidacy Id	[[[Vacancy Title	[[[CandidateCurriculum Name	[[[StepType Label	[[[StepState Label
445	Java Junior Developer	Ricardo Raizes	Consideration	To Process

Figura 91 - Nova instância de *CandidacyStep*

Esta nova candidatura pode ser observada tanto na aplicação *web*, Figura 92, como também na aplicação *mobile*, Figura 93,

My Candidacies

Java Developer
Requirements:
Good with Java and JavaBeans

Java Junior Developer
Requirements:
Capable with java and if possible SQL

Figura 92 - Candidatura, *web*

Vacancy Applications

Java Developer
Job Title: Developer

Java Junior Developer
Job Title: Developer

Locations Profile Home Jobs Interviews

Figura 93 - Candidatura *mobile*

4.13 Continuar/Acabar Candidatura – Colaborador

Depois de uma oferta ser estabelecida, um colaborador consegue controlar as candidaturas sobre estas, principalmente pelas ações de marcar uma entrevista, a verificar no próximo subcapítulo, terminar a candidatura em si ou continuar a mesma, ou seja, continuar a candidatura para o próximo passo de contratação.

A continuação, como mostrada anteriormente, no Código 17, permite saltar vários passos do processo de contratação desde que o candidato tenha participado numa entrevista que tenha utilizado o mesmo formulário que o passo saltado.

Ação

Um colaborador consegue verificar qualquer candidatura, pesquisando-a numa página ou consegue obter todas as candidaturas a uma particular oferta. Para a segunda um colaborador tem de primeiro escolher uma oferta, o que o permite verificar a *tab* da Figura 94.

Desta é possível escolher um passo em particular, Figura 95, onde pode verificar todas as candidaturas nesse passo, o que permite verificar as candidaturas, como por exemplo a da Figura 96.

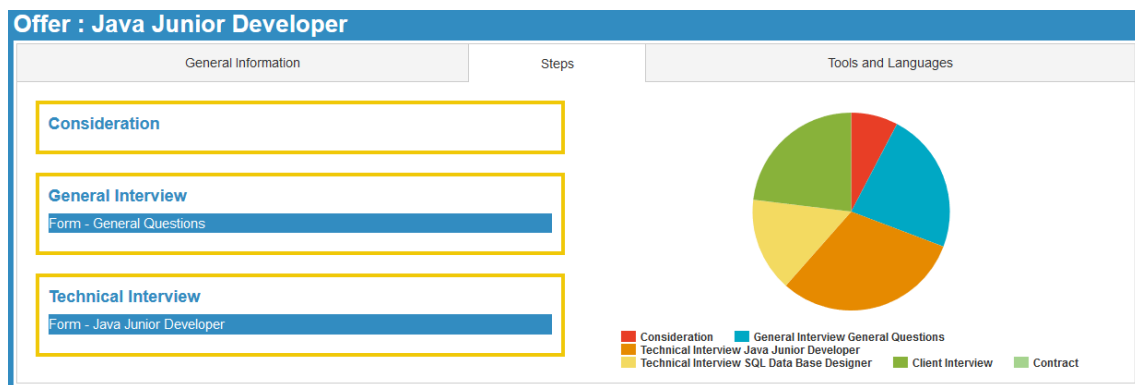


Figura 94 - Passos de uma oferta



Figura 95 - Passo de uma oferta

Figura 96 - Candidatura

Resultado Esperado

Ao escolher terminar uma candidatura, a *Vacancy* correspondente como também qualquer *VacancyCurrentStep* e *VacancyInterview* associadas a essa *Vacancy*, são removidas da base de dados e o candidato é informado na terminação da candidatura tanto por um *email*, Figura 97 como também por uma notificação da aplicação *mobile* Figura 98.

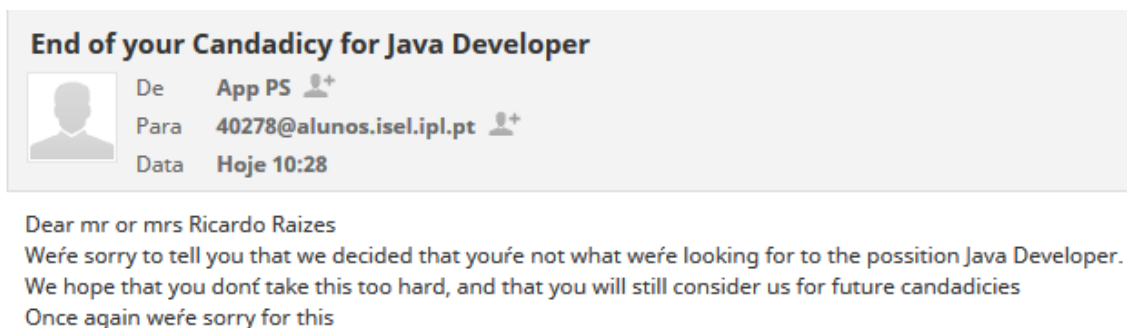


Figura 97 - Finalização de uma candidatura, *email*



Figura 98 - Finalização de uma candidatura, notificação

Já continuar uma candidatura resulta sempre na alteração da *VacancyStep* para demonstrar o passo a realizar, passando para o estado “*ToScehdule*”, Figura 100.

Também é possível que a continuação resulte na criação de novas instâncias de *VacancyInterview*, o que acontece quando o candidato já realizou entrevistas com os formulários dos passos saltados. No caso da continuação da candidatura da Figura 96, a mesma salta três passos, criando as instâncias da *VacancyInterview* da Figura 99.

Estas alterações são verificadas na página da candidatura, como se pode ver na Figura 101.

Independentemente do que ocorre com a continuação, o candidato é informado da continuação em si tanto por *email*, Figura 102 como por notificação Figura 103.

Interview Date	Vacancy Title	Form Title
2018-06-20 00:00:00	Java Junior Developer	Introductory Questions
2018-07-12 00:00:00	Java Junior Developer	Java Junior Developer
2018-06-17 21:03:36	Java Junior Developer	SQL Data Base Designer


Figura 99 - Novas instâncias de *VacancyInterview*

CandidacyCurrentStep State	CandidacyCurrentStep Step	CandidateCurriculum Name	Vacancy Title	StepState Label
5	4	Felix Amaro	Java Junior Developer	ToSent

Figura 100 - Alteração sobre *VacancyStep*

Candicy to Java Junior Developer of Felix Amaro
End

Candidate



Felix Amaro
Email: amaro_san@outlook.pt
Phone: 960000000

CurrentStep

Client Interview - ToSent
Go to Send


Interviews

Introductory Questions
2018-06-20 00:00:00

SQL Data Base Designer
2018-06-17 21:03:36

Java Junior Developer
2018-07-12 00:00:00

Figura 101 - Candidatura depois


Conitnuation of your Candadicy for Java Junior Developer
De App PS Data Hoje 10:00

Dear mr or mrs Ricardo Raizes

You have been selected to the next stage of the candidacy for Java Junior Developer.

If your availibilty has change, go imeditaly to you candadecies to change the inforation.

The next step is a General Interview, what that means is:

This will be a general intreview so we can ascertain you on a personal level

Figura 102 - Continuação da candidatura, *email*

Candicy Java Junior Developer continuation
Your Candicy to the vacancy Java Junior Developer has been continued to the next step



Figura 103 - Continuação da candidatura, notificação

4.14 Marcar Entrevista – Colaborador

Uma entrevista pode ser marcada de várias formas, mas a mais importante é a marcação que pode ser realizada em função de uma candidatura, permitindo assim marcar uma entrevista completa fornecendo pouca informação.

Ação

Pela candidatura um colaborador consegue abrir o *Pop-up* da Figura 104, onde pode não só verificar os eventos já estabelecidos em que participa, como também pode verificar a disponibilidade do colaborador a participar na entrevista, Figura 105. A partir deste *Pop-up* é também possível criar um novo bloco que representa o início e fim da entrevista a marcar.

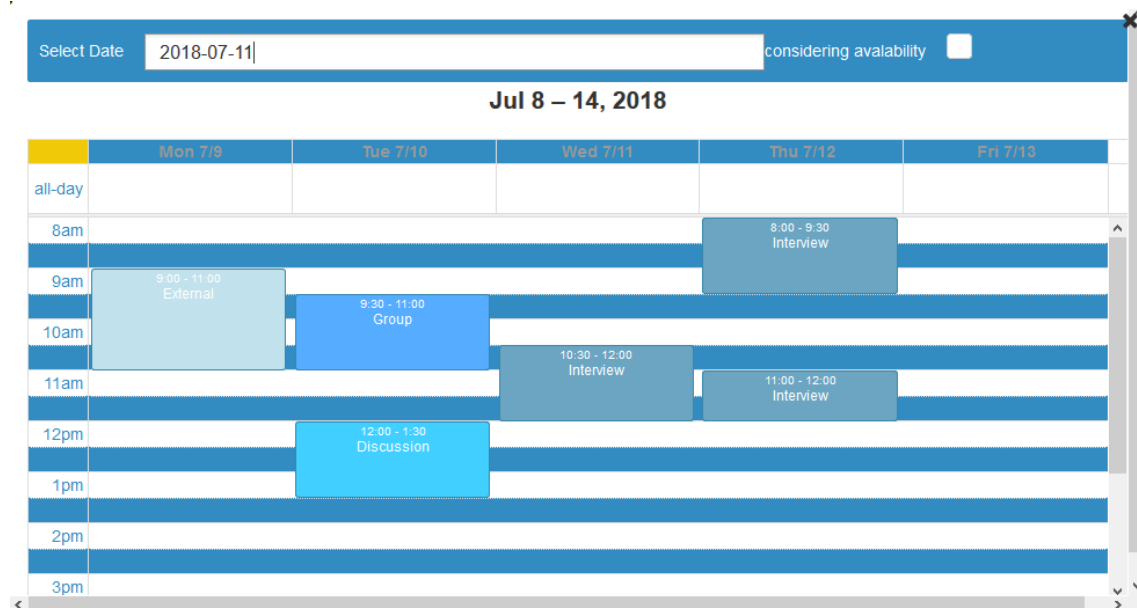


Figura 104 – Eventos

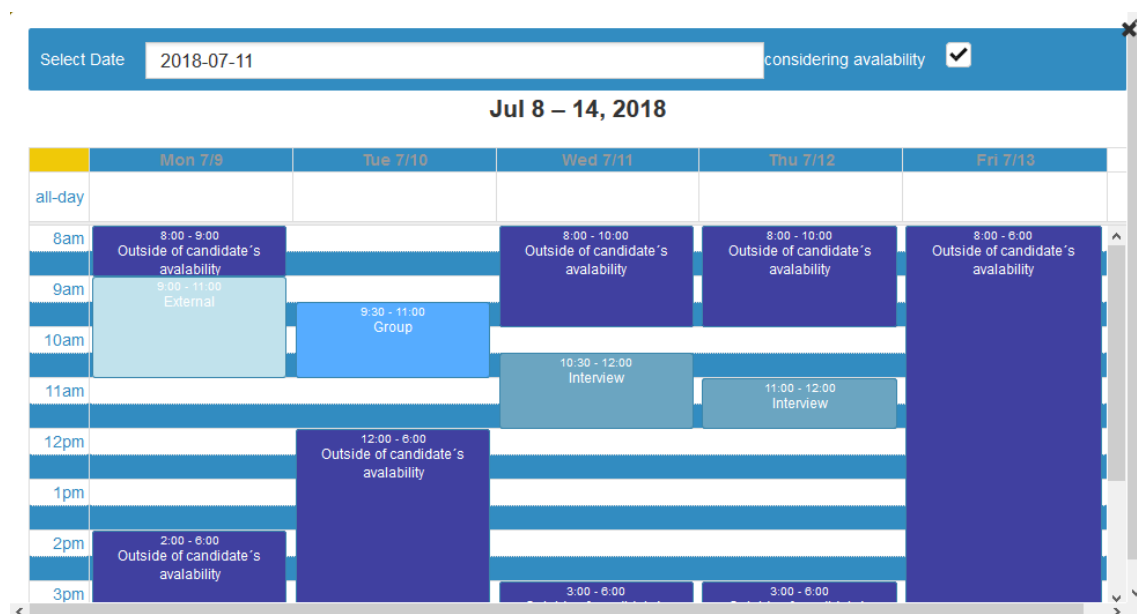


Figura 105 - Eventos com disponibilidade

Resultado Esperado

Ao escolher os tempos para o evento, o *Pop-up* da Figura 106 aparece permitindo que o colaborador escolha criar um evento com os tempos escolhidos. Se o mesmo escolher criar a entrevista então o *Pop-up* da Figura 107 aparece, o que permite escolher a localidade onde a entrevista deverá ocorrer.

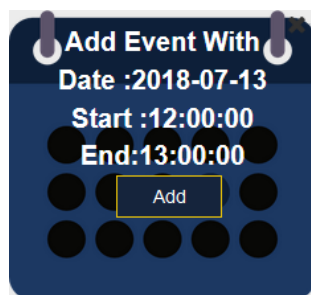


Figura 106 - *Pop-up* do novo evento

Search Form by Company
Projects Limited
Reset
No Location

Company	Local
Projects Limited	Rua José Malhoa 35, Oeiras
Projects Limited	Rua Nossa Senhora Graça 72, Lisboa
PC Corps	Rua cimo vila 100, Porto
HD Hub	Rua Irene Lisboa 55, Lisboa
HD Hub	Rua Poço lote 103, Setubal

1
2
next

Figura 107 - *Pop-up* para escolher localidades

Depois de escolher o tempo e a localidade da entrevista, uma nova instância da entidade *Event* e uma de *Interview* são criadas, que podem ser demonstradas na página do próprio evento, cujas *tabs* estão na Figura 109, Figura 110 e Figura 111,

Event Date	Event Start	Event End	CandidateCurriculum Name	Employee Name
2018-07-11	13:00:00	14:00:00	Francisco Julio	Ana Luisa

Figura 108 - Nova instância de *Event* e *Interview*

Check Event
Actions

Event	Responsible	Participant
Title		
Date	2018-07-11	
Start	13:00:00	
End	14:00:00	
Location	Rua Nossa Senhora Graça 72, Lisboa	

Figura 109 - Nova entrevista, informação geral

Check Event Actions ▾

Event	Responsible	Participant
<div style="border: 2px solid yellow; padding: 10px;"> <div style="display: flex; align-items: center;"> <div> <p>Ana Luisa</p> <p>Email: anLu@gmail.com</p> <p>Phone: 960000000</p> </div> </div> </div>		

Figura 110 - Nova entrevista, responsável

Check Event Actions ▾

Event	Responsible	Participant
<div style="border: 2px solid yellow; padding: 10px;"> <div style="display: flex; align-items: center;"> <div> <p>Francisco Julio</p> <p>Email: falcaoAzul@hotmail.com</p> <p>Phone: 960000000</p> </div> </div> </div>		

Figura 111 - Nova entrevista, participante

O candidato participante é informado da nova entrevista tanto por um *email*, Figura 112 ou por uma notificação Figura 113.

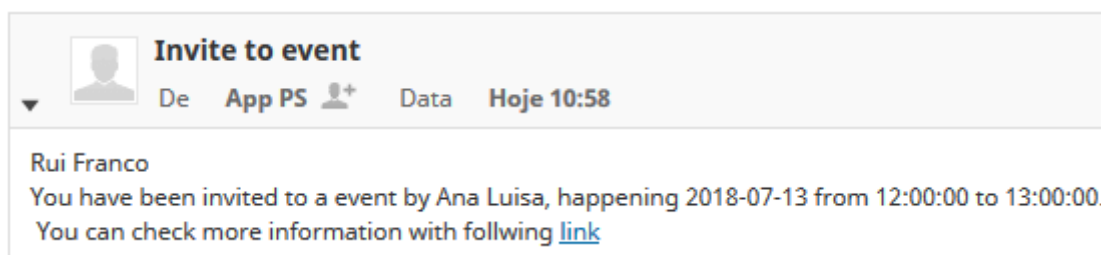


Figura 112 - Convite da entrevista, email

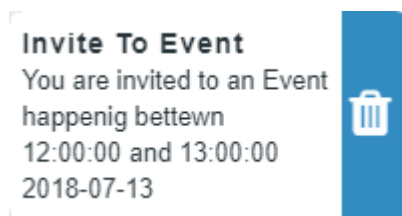


Figura 113 - Convite da entrevista, notificação

Durante a marcação duma entrevista, dois erros podem ocorrer: ou a entrevista marcada ocorre ao mesmo tempo que uma entrevista já marcada, ou é marcada fora da disponibilidade do colaborador, o que faz as mensagens da Figura 114 aparecerem.

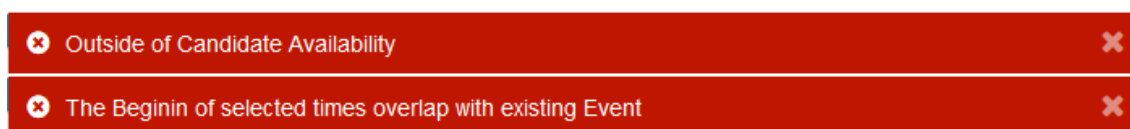


Figura 114 - Erro na marcação

5. Conclusões e Trabalho Futuro

Durante o processo de desenvolvimento deste projeto tivemos um contacto formal e sólido com a arquitetura *Outsystems*, uma arquitetura cujo o conhecimento tem sido cada vez mais requisitado no mercado atual. O que por e si é uma experiência positiva e uma mais valia para o futuro.

Quando a aplicação *IView* foi inicialmente pensada, seria uma simples ajuda a candidatos para gerar o seu *profile* e currículo, mas durante o desenvolvimento do projeto, novas ideias foram surgindo e a aplicação foi enriquecendo, permitindo assim, poder vir a ser, uma mais valia não só para candidatos como também para colaboradores. Assim, novos objetivos não opcionais foram adicionados tanto aos candidatos como aos colaboradores, tendo sido cumpridos todos os objetivos, tanto nas funcionalidade propostas, como na área das interfaces de utilização.

Dos requisitos “funcionalidades opcionais” só uma foi completada, a inclusão do *Google Maps*, que está incluído tanto na aplicação *web*, na área de contactos nas informações gerais e na aplicação *mobile*. Das restantes não executadas, a única em que tivemos dificuldades, e daí não a termos incluído, foi o acesso a *Google Calendar*. Este pode em parte ser realizado por várias extensões da *OutSystems*, incluído a extensão utilizada *FullCalendar2*. Mas todas as extensões têm um problema em comum, permitem a leitura na *Google Calendar*, mas não a escrita. Apesar dessas capacidades serem interessante, tal limitação invalidou a sua inclusão no *IView*.

Foi bastante explorada a possibilidade de passar todos os dados para uma base de dados externa, mas além de introduzir uma dependência adicional, tornava o acesso mais lento bem como qualquer alteração à estrutura da base de dados mais complexa, demonstrando assim, que não tinha uma mais valia quando comparada a alternativa mais simples. A únicas vantagens que encontrámos, foi em relação ao aumento da sua dimensão, e ao facto de que poderia ser mais fácil introduzir e remover valores da base de dados sem ter de utilizar a aplicação, mas tais vantagens eram simplesmente menores que as desvantagens.

Por último, em relação ao requisito de estruturar a aplicação de forma a englobar várias empresas, apercebemo-nos durante o desenvolvimento, que poderia introduzir problemas que tornavam o requisito contraprodutivo. A base de dados seria muito extensa tendo em conta todas as informações que poderiam ser incluídas. Contudo, o problema principal seria a partilha de informação, pois com a utilização regular da aplicação, as empresas incluiriam informação sensível, desde informação de projetos, eventos ou até candidaturas, que ficariam acessíveis a outros grupos.

Para futuras melhorias do *IView*, temos algumas considerações, tais como a inclusão do *Google Calendar* e a melhoria da interface gráfica, tentando tornar a informação o mais fácil de seguir possível. Outra funcionalidade interessante, seria a inclusão do lançamento de notificações, quando fosse criada uma oferta eventualmente interessante a um determinado perfil de candidatos. Outra melhoria a considerar, seria a formação dos formulários de forma a que, o salto de passos

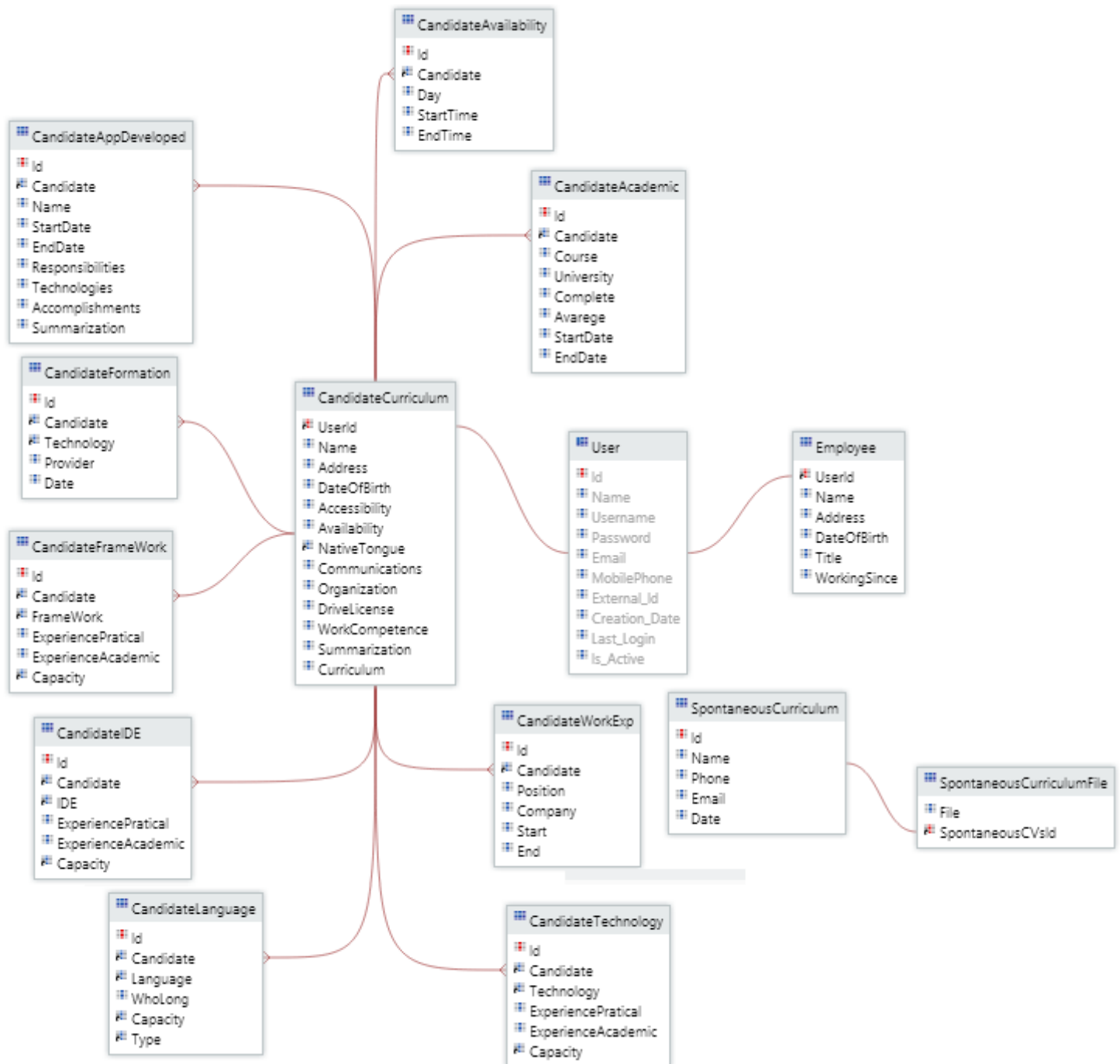
de um processo de entrevista, não estivesse completamente dependente de um formulário em particular. Interessante seria também, a possibilidade de incluir na aplicação *web*, a capacidade de receber notificações. Deste modo, as notificações seriam enviadas tanto para aplicação *mobile* como para a *web*. Como última melhoria a considerar, seria a capacidade de criar uma via de comunicação diretamente na aplicação, permitindo assim uma mais fácil marcação de entrevistas, podendo até ser possível realizar as entrevistas por essa via.

Referências

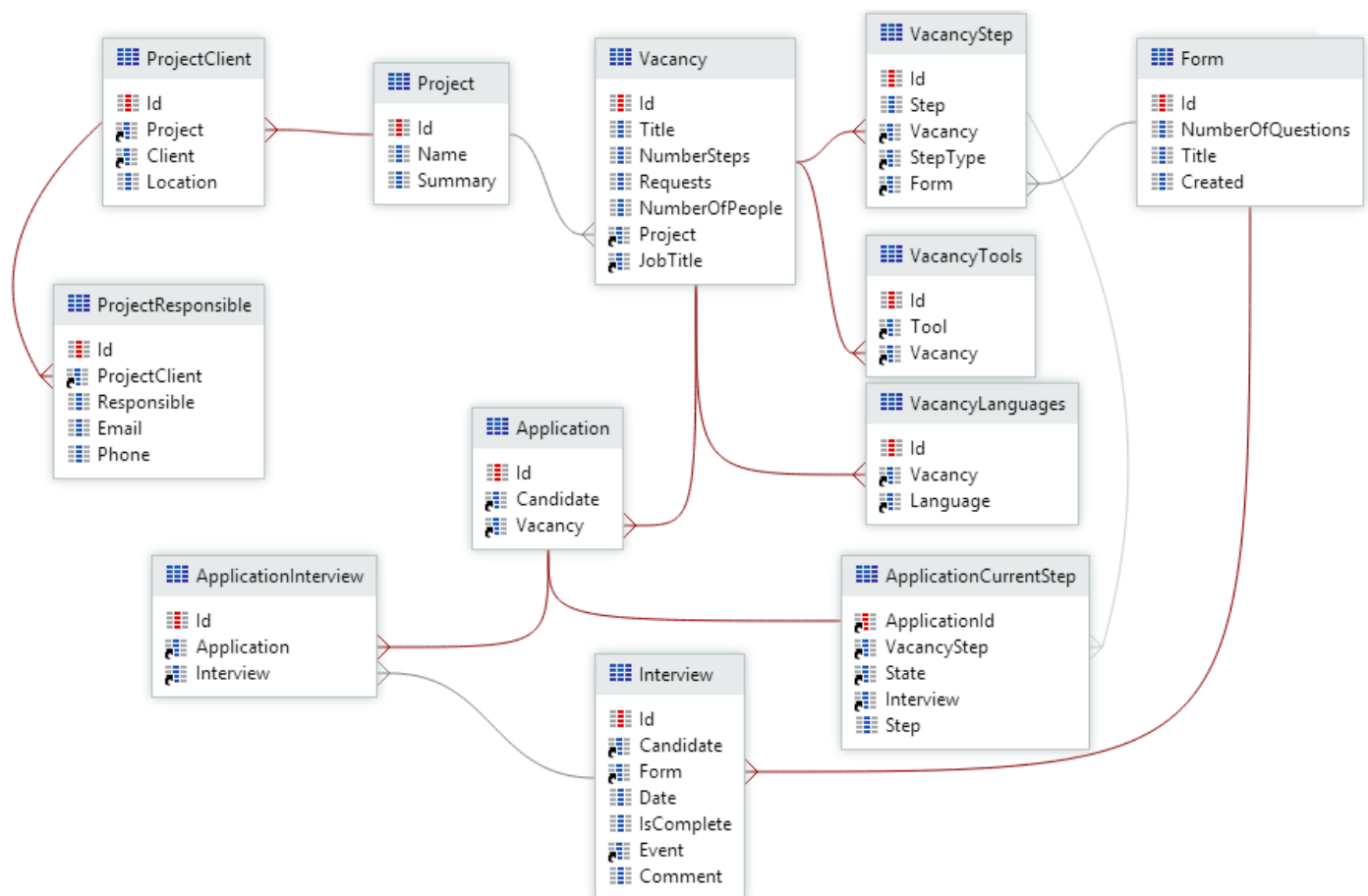
- [1] Mind Source, “Portal de Emprego,” Mind Source, [Online]. Available: <https://emprego.mindsources.pt/>. [Acedido em Abril 2018].
- [2] Randstrand, “My Profile,” Randstrand, [Online]. Available: <https://myprofile.randstad.pt/>. [Acedido em Abril 2018].
- [3] LinkedIn Corporation, “LinkedIn,” LinkedIn Corporation, 05 Maio 2003. [Online]. Available: <https://pt.linkedin.com/>. [Acedido em Abril 2018].
- [4] ITJobs Portugal, “ITJobs,” ITJobs Portugal, [Online]. Available: <https://www.itjobs.pt/>. [Acedido em Abril 2018].
- [5] OutSystems, “OutSystems Architecture,” OutSystems, [Online]. Available: https://success.outsystems.com/Evaluation/Architecture/2_OutSystems_Platform_architecture. [Acedido em Abril 2018].

A.1 Modelos de dados

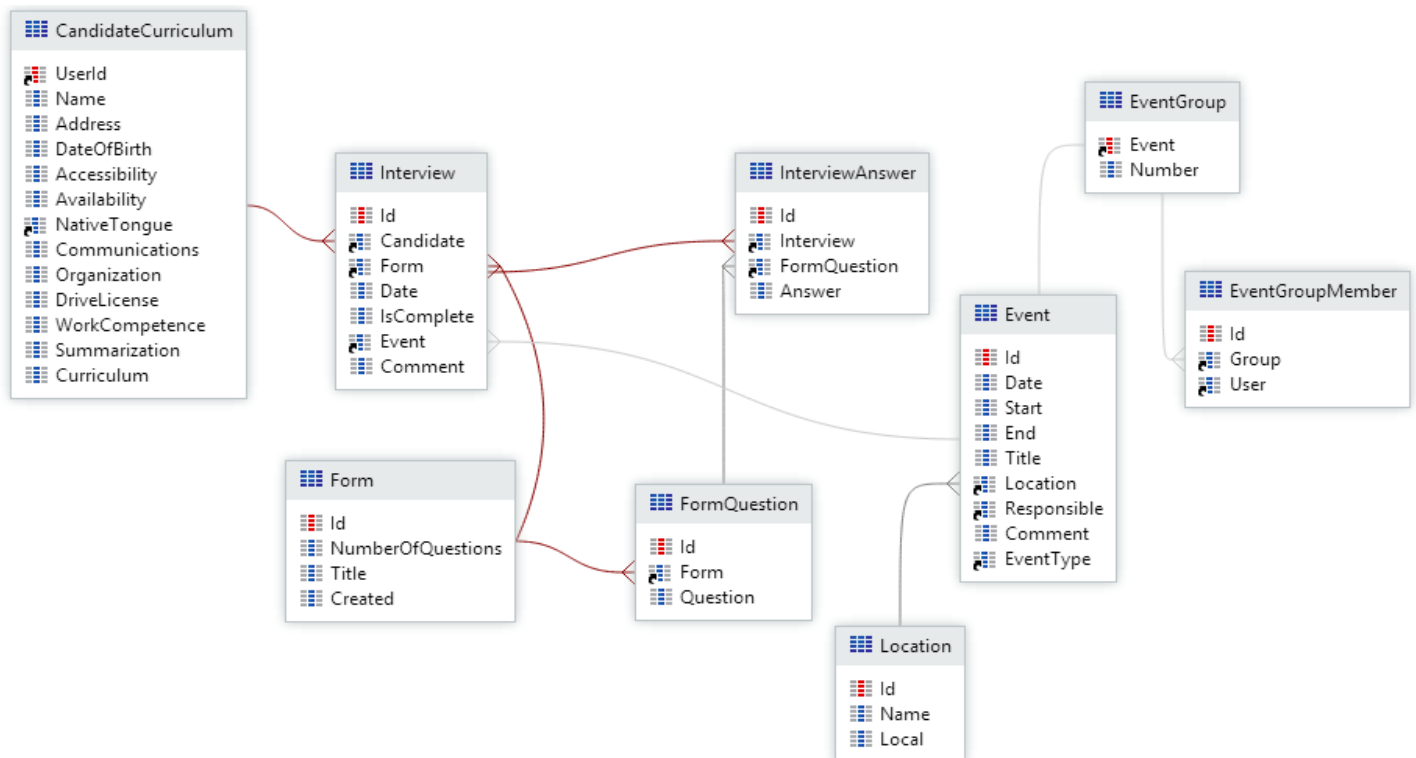
Sendo que a base de dados desenvolvida para este projeto inclui um notável número de entidades, o modelo de dados é demonstrado em três submodelos, com a divisão das entidades sendo igual a verificada na secção 3.2.



Modelos De Dados 1 - Users



Modelos De Datos 2 - Vacancies



Modelos De Datos 3 - Events