

**IView – Aplicação de Entrevistas e Currículos**

36210, Rui Pedro Gama Franco, 933941557, [36210@alunos.isel.ipl.pt](mailto:36210@alunos.isel.ipl.pt)  
40278, Diogo Mendes Aires, 962573900, [40278@alunos.isel.ipl.pt](mailto:40278@alunos.isel.ipl.pt)

**Orientadores**

Paula Graça, ISEL, [mgraca@deetc.isel.pt](mailto:mgraca@deetc.isel.pt)

Frederico Ferreirá, DoItLean, [frederico.ferreirá@doitlean.com](mailto:frederico.ferreira@doitlean.com)

|  |  |
| --- | --- |
|  |  |
|  |  |

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

Maio 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 Ferreirá, DoItLean |

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

Maio de 2018

# Resumo

O projeto IView tem como objetivo principal desenvolver uma plataforma de contratações, 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 irá garantir 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 para a qual estão à procura, facilitando assim o seu processo de seleção.

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

Esta plataforma é composta por duas vertentes: *Mobile* e *Web*, sendo que a *Mobile* disponibiliza notificações ao candidato sobre o estado dos vários processos, nos quais participa, bem como a consulta de informações relevantes sobre os mesmos; a vertente *Web*, além de disponibilizar as mesmas funções da app *Mobile*, disponibiliza também tanto as ofertas de emprego como a possibilidade de candidatura nelas.

Durante o processo de desenvolvimento desta aplicação também colocámos os nossos conhecimentos da arquitetura de desenvolvimento *OutSystems* em prática, arquitetura cujo o conhecimento é cada vez mais requisitado no mercado informático e tecnológico, tanto português como estrangeiro, o que torna qualquer experiência sobre a mesma uma mais valia para o futuro.

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

**Índice**

[Resumo iv](#_Toc519025009)

[Lista de Figuras viii](#_Toc519025010)

[Lista de Tabelas x](#_Toc519025011)

[1. Introdução 1](#_Toc519025012)

[1.1. Enquadramento 1](#_Toc519025013)

[1.2. Objetivos 1](#_Toc519025014)

[1.3. Organização do documento 2](#_Toc519025015)

[2. Formulação do Problema 3](#_Toc519025016)

[2.1. Estado da arte 3](#_Toc519025017)

[2.2. Descrição do Projeto 4](#_Toc519025018)

[2.2.1. Aplicação web 6](#_Toc519025019)

[2.2.2. Aplicação mobile 10](#_Toc519025020)

[2.3. Descrição da plataforma, modelos de desenvolvimentos 11](#_Toc519025021)

[2.3.1. Servidores Fundamentais 12](#_Toc519025022)

[2.3.2. Sincronismo 12](#_Toc519025023)

[2.4. Segurança de Infomação 13](#_Toc519025024)

[3. Solução Proposta 15](#_Toc519025025)

[3.1 Arquitetura do projeto 15](#_Toc519025026)

[3.2 Modelo entidade-associação da base de dados 16](#_Toc519025027)

[3.2.1 Utilizadores 16](#_Toc519025028)

[3.2.2 Vagas 18](#_Toc519025029)

[3.2.3 Eventos 19](#_Toc519025030)

[3.3 Wireframes do projeto 20](#_Toc519025031)

[3.4 Desenvolvimento Web 31](#_Toc519025032)

[3.4.1 Listagem 31](#_Toc519025033)

[3.4.3 Adicionar e Remover do Profile 33](#_Toc519025034)

[3.4.4 Introduzir novo candidato 33](#_Toc519025035)

[3.4.5 Verificar Eventos 34](#_Toc519025036)

[3.4.6 Adicionar Evento 38](#_Toc519025037)

[3.4.7 Adicionar Oferta 38](#_Toc519025038)

[3.4.8 Verificar percentagens de vaga 39](#_Toc519025039)

[3.4.9 Pesquisa de ofertas 40](#_Toc519025040)

[3.4.10 Continuação da candidatura 41](#_Toc519025041)

[3.4.11 Incluir disponibilidade 43](#_Toc519025042)

[3.5 Desenvolvimento Mobile 44](#_Toc519025043)

[3.5.1 Notificações 44](#_Toc519025044)

[3.5.2 Sincronismo 45](#_Toc519025045)

[4 Avaliação Experimental 47](#_Toc519025046)

[4.1 Candidatura Espontânea – Utilizador não registrado 47](#_Toc519025047)

[4.1.1 Ação 47](#_Toc519025048)

[4.1.2 Resultado Esperado 48](#_Toc519025049)

[4.2 Estabelecer / Recusar novos candidatos – Colaborador 48](#_Toc519025050)

[4.2.1 Ação 48](#_Toc519025051)

[4.2.2 Resultado Esperado 48](#_Toc519025052)

[4.3 Estabelecer novo candidato – Utilizador não registrado 49](#_Toc519025053)

[4.3.1 Ação 49](#_Toc519025054)

[4.4 Autentificação de utilizador – Utilizador 50](#_Toc519025055)

[4.5 Atualizar Disponibilidade – Candidato 53](#_Toc519025056)

[4.6 Estabelecer Empresa – Administrador 55](#_Toc519025057)

[4.7 Introduzir Projeto – Colaborador 56](#_Toc519025058)

[4.8 Introduzir Form – Colaborador 58](#_Toc519025059)

[4.9 Introduzir Oferta – Colaborador 59](#_Toc519025060)

[4.10 Pesquisar/Aplicar a oferta – Candidato 62](#_Toc519025061)

[4.11 Continuar/Acabar Candidatura – Colaborador 64](#_Toc519025062)

[4.12 Marcar Entrevista – Colaborador 67](#_Toc519025063)

[5. Conclusões 71](#_Toc519025064)

[Referências 73](#_Toc519025065)

[A.1 Diagramas da Aplicação 74](#_Toc519025066)

[A.2 Modelos de dados 76](#_Toc519025067)

# Lista de Figuras

[Figura 1 – Visão geral do IView. 5](#_Toc517738307)

[Figura 2 - Casos de utilização, Utilizadores Não Registados. 6](#_Toc517738308)

[Figura 3 - Casos de utilização, Colaborador 1. 7](#_Toc517738309)

[Figura 4 - Casos de utilização, Colaborador 2. 8](#_Toc517738310)

[Figura 5 - Casos de utilização, Candidato. 9](#_Toc517738311)

[Figura 6 - Casos de utilização, Candidato Mobile. 10](#_Toc517738312)

[Figura 7 - Arquitetura OutSystems [5]. 11](#_Toc517738313)

[Figura 8 - Estrutura 4 Layer Canvas. 13](#_Toc517738314)

[Figura 9 - Modelo EA, Candidatos. 15](#_Toc517738315)

[Figura 10 - Modelo EA, Vagas. 16](#_Toc517738316)

[Figura 11 - Modelo EA, Eventos. 18](#_Toc517738317)

[Figura 12 - Frames, Home Page 19](#_Toc517738318)

[Figura 13 - Frames, Menu de Colaboradores. 19](#_Toc517738319)

[Figura 14 - Frames, Menu de Candidatos. 20](#_Toc517738320)

[Figura 15 - Frame, MoreInfo Values Tab. 20](#_Toc517738321)

[Figura 16 - Frame, MoreInfo Patnerships Tab. 21](#_Toc517738322)

[Figura 17 - My Curriculum, Show. 21](#_Toc517738323)

[Figura 18 - Frame, MyCurriculum, Edit. 22](#_Toc517738324)

[Figura 19 - Frames, MyProfile. 22](#_Toc517738325)

[Figura 20 - Frames, MyProfile *Pop-up*. 23](#_Toc517738326)

[Figura 21 - Frames, Forms. 24](#_Toc517738327)

[Figura 22 - Frames, Applications. 25](#_Toc517738328)

[Figura 23 - Frames, Vacancies. 25](#_Toc517738329)

[Figura 24 - Frames, Events. 26](#_Toc517738330)

[Figura 25 - Frames, Candidate. 27](#_Toc517738331)

[Figura 26 - Frames, Vacancy General Information Tab. 27](#_Toc517738332)

[Figura 27 - Frames, Vacancy Steps Tab. 28](#_Toc517738333)

[Figura 28 - Frames, Vacancy Tools and Languages Tab. 28](#_Toc517738334)

[Figura 29 - Frames, FormEditAdd. 29](#_Toc517738335)

[Figura 30 - Frames, VacancyAdd Form Chosse *Pop-up*. 29](#_Toc517738336)

[Figura 31 - Candidatura a Applicação 45](#_Toc517738337)

[Figura 32 - Base de dados antes e depois da candidatura 45](#_Toc517738338)

[Figura 33 - Email recebido pelo utilizador não registrado 46](#_Toc517738339)

[Figura 34 - Email de utilizador estabelecido 46](#_Toc517738340)

[Figura 35 - Edição do currículo 46](#_Toc517738341)

[Figura 36 - Alteração da base de dados 47](#_Toc517738342)

[Figura 37 - Introdução de tecnologia 47](#_Toc517738343)

[Figura 38 - Alteração sobre a base de dados 47](#_Toc517738344)

[Figura 39 - Introdução de disponibilidade 48](#_Toc517738345)

[Figura 40 - Alteração a base de dados disponibilidade 48](#_Toc517738346)

[Figura 41 - Pesquisa de oferta, Project Manager 49](#_Toc517738347)

[Figura 42 - Pesquisa de oferta, Project Manager com Java 49](#_Toc517738348)

[Figura 43 - Pesquisa de oferta, Project Manager com Alemão 50](#_Toc517738349)

[Figura 44 - Pesquisas da base de dados 50](#_Toc517738350)

[Figura 45 - Criação de um Form 51](#_Toc517738351)

[Figura 46 - Alteração na base dados com a ciráção do Form 51](#_Toc517738352)

[Figura 47 - Questões do Form na base de dados 51](#_Toc517738353)

[Figura 48 - Criação de uma Parceirá 52](#_Toc517738354)

[Figura 49 - Alterações a base de dados com a criação de uma Parceirá 52](#_Toc517738355)

[Figura 50 - *Pop-up* para escolher parceirá 53](#_Toc517738356)

[Figura 51 - Criação de um projecto 53](#_Toc517738357)

[Figura 52 - Projectos da empresa "Plus One" 54](#_Toc517738358)

[Figura 53 - Alterações a base de dados com a criação de um projeto 54](#_Toc517738359)

[Figura 54 - *Pop-up* associar projeto a oferta 54](#_Toc517738360)

[Figura 55 - *Pop-up* associar form a passo de oferta 55](#_Toc517738361)

[Figura 56 - Criação de uma oferta 56](#_Toc517738362)

[Figura 57 - Ofertas do projeto 56](#_Toc517738363)

[Figura 58 - Alteração da base de dados com a criação da oferta 56](#_Toc517738364)

[Figura 59 - Candidaturas a oferta 57](#_Toc517738365)

[Figura 60 - Candidatura Individual, depois de continue 57](#_Toc517738366)

[Figura 61 - Passo de oferta 58](#_Toc517738367)

[Figura 62 - Passo na base de dados 59](#_Toc517738368)

[Figura 63 – Candidatura, com a contração 59](#_Toc517738369)

[Figura 64 - Alteração da base de dados, com a contração 59](#_Toc517738370)

[Figura 65 - Novo Colaborador 60](#_Toc517738371)

[Figura 66 – Eventos Mês 60](#_Toc517738372)

[Figura 67 - Eventos semana 61](#_Toc517738373)

[Figura 68 - *Pop-up* para novo evento, erro 61](#_Toc517738374)

[Figura 69 - *Pop-up* para novo evento, confirmação 62](#_Toc517738375)

[Figura 70 – *Pop-up* para escolher localidade 62](#_Toc517738376)

[Figura 71 - Alteração da base de dados com a criação do evento 62](#_Toc517738377)

# Lista de Tabelas

[Tabela 1 - Application 51](#_Toc517598376)

[Tabela 2 - ApplicationCurrentStep 51](#_Toc517598377)

[Tabela 3 – ApplicationInterview 51](#_Toc517598378)

[Tabela 4 – CandidateAcademics 51](#_Toc517598379)

[Tabela 5 – CandidateAppDeveloped 51](#_Toc517598380)

[Tabela 6 - CandidateAvailability 52](#_Toc517598381)

[Tabela 7 – CandidateCurriculum 52](#_Toc517598382)

[Tabela 8 - CandidateFormations 53](#_Toc517598383)

[Tabela 9 – CandidateFrameworks 53](#_Toc517598384)

[Tabela 10 - CandidateIDE 53](#_Toc517598385)

[Tabela 11 – CandidateLanguage 53](#_Toc517598386)

[Tabela 12 - CandidateTechnology 54](#_Toc517598387)

[Tabela 13 - CandidateWorkExp 54](#_Toc517598388)

[Tabela 14 – Capacity 54](#_Toc517598389)

[Tabela 15 – Client 54](#_Toc517598390)

[Tabela 16 – Discussion 54](#_Toc517598391)

[Tabela 17 – Employee 55](#_Toc517598392)

[Tabela 18 - Event 55](#_Toc517598393)

[Tabela 19 - EventGroup 55](#_Toc517598394)

[Tabela 20 - EventGroupMember 55](#_Toc517598395)

[Tabela 21 - EventType 55](#_Toc517598396)

[Tabela 22 - Form 56](#_Toc517598397)

[Tabela 23 - FormQuestion 56](#_Toc517598398)

[Tabela 24 - Interview 56](#_Toc517598399)

[Tabela 25 - InterviewAnswer 56](#_Toc517598400)

[Tabela 26 - JobTitle 56](#_Toc517598401)

[Tabela 27 – Language 56](#_Toc517598402)

[Tabela 28 - Location 57](#_Toc517598403)

[Tabela 29 - Photo 57](#_Toc517598404)

[Tabela 30 –Project 57](#_Toc517598405)

[Tabela 31 - ProjectClient 57](#_Toc517598406)

[Tabela 32 –ProjectResponsible 57](#_Toc517598407)

[Tabela 33 - SpontanousCurriculum 57](#_Toc517598408)

[Tabela 34 - SpontanousCurriculumFile 58](#_Toc517598409)

[Tabela 35 – StepState 58](#_Toc517598410)

[Tabela 36 - StepType 58](#_Toc517598411)

[Tabela 37 – User 58](#_Toc517598412)

[Tabela 38 – Tools 58](#_Toc517598413)

[Tabela 39 – TypeLanguage 59](#_Toc517598414)

[Tabela 40 – Vacancy 59](#_Toc517598415)

[Tabela 41 - VacancyLanguages 59](#_Toc517598416)

[Tabela 42 - VacancyStep 59](#_Toc517598417)

[Tabela 43 –VacancyTools 59](#_Toc517598418)

# Introdução

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

## Enquadramento

Muitas empresas de tecnologia portuguesas, têm um processo de entrevistas bastante desatualizado e limitado, não colocando os seus conhecimentos tecnológicos em utilização. Normalmente este processo envolve a transferência de ficheiros importantes, como por exemplo currículos e dossiês de competências[[1]](#footnote-1) por correio (eletrónico e/ou físico. A maioria das marcações de entrevistas neste processo ocorre por comunicação telemóvel ou email, o que torna o candidato dependente do colaborador, qualquer falha de comunicação poderá resultar na entrevista não ser marcada ou 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 funcionalidades úteis para as empresas que o utilizem.

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

## 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, desde o momento em que uma nova oferta é aberta, até ao momento final da eventual 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:

* Currículos, dossiês de competências e disponibilidades[[2]](#footnote-2) 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. Esta disponibiliza notificações ao candidato sobre o estado dos vários processos, nos quais participa, bem como a consulta de informações relevantes sobre os mesmos. Através da aplicação mobile também é possível realizar alterações aos dados pessoais do utilizador, tais como nome, morada, etc.

## Organização do documento

O restante relatório tem a seguinte composição de capítulos:

* Formulação do Problema – contém informação sobre da aplicação IView, incluído referências para algumas aplicações semelhantes, detalhe dos requisitos funcionais e plataforma tecnológica onde o projeto é desenvolvido;
* Solução Proposta - neste capítulo está incluída a arquitetura da aplicação, o modelo geral da base de dados, algumas ideias gerais da interface gráfica e alguns dos principais fluxos de implementação;
* Avaliação Experimental – onde será verificada a execução das várias funcionalidades da aplicação *web*, como também as alterações a base de dados que ocorrem com tal funcionalidade;
* Conclusões – o último capítulo deste relatório onde estabelecemos as conclusões que foram formadas com o desenvolvimento deste projeto.

# Formulação do Problema

Neste capítulo, na secção 2.1, é verificada a utilidade da nossa ferramenta bem como exemplos de aplicações semelhantes, incluindo algumas limitações de tais aplicações e como a IView as complementa. 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.

## 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 encontrámos servem normalmente para a empresa gerir candidatos, os seus currículos e a partilha dos mesmos com 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 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 este as partes interessadas, a entrevista pode 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.

Na pesquisa que efetuámos para o desenvolvimento do IView, encontrámos, contudo, duas aplicações que permitem a comunicação entre candidato e colaborador. A primeirá, 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 é indicado quais os ficheiros que o candidato deve incluir, bem como a sua estrutura. Já para as candidaturas, a informação fornecida sobre as mesmas é bastante limitada, sendo que só indica que candidaturas estão a ocorrer, mas nada mais.

Outro exemplo é uma aplicação de gestão de perfil produzida pela *Randstad* [2], a qual não inclui qualquer capacidade de verificação de vagas e candidaturas às mesmas, estando limitado unicamente à construção e gestão do perfil do candidato. 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 limitações acima enumeradas, a aplicação da *Randstad* proporcionou-nos parte da inspiráção gráfica para o projeto.

Existem ainda duas outras aplicações que estão, de alguma forma, relacionadas com a IView, mas ao contrário das referidas anteriormente, estas não são limitadas a uma só empresa e são mais adequadas para a partilha de informação sobre perfis e vagas na área.

A primeirá a que nos referimos 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 em si é 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 da informática e tecnologias. 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 da área de informática e tecnologias, já dispõem de aplicações do tipo da IView;
* A possibilidade de os candidatos terem a capacidade de interagir com as empresas através dessas aplicações, é uma funcionalidade que não foi considerada, 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.

## 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, irá ser descrito as funcionalidades de cada aplicação (*web* e *móvel*), 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.

### 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 Registados.

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 segunda figura é demonstrado as funcionalidades mais centradas no processo de entrevista e outros eventos, sendo as restantes mostradas pela primeirá.

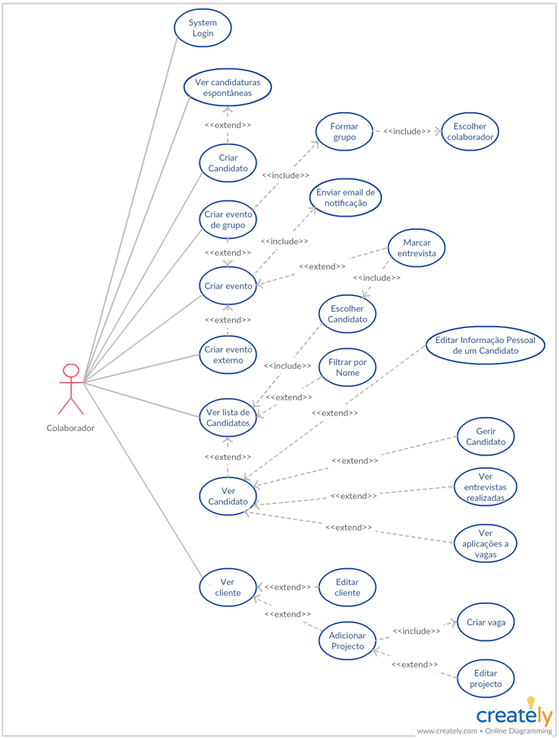


Figura 3 - Casos de utilização, Colaborador 1.



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 extras servem principalmente para realizar alterações notáveis na base de dados que embora importantes, não devem ser acessíveis a todos os colaboradores.

As novas capacidades 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.

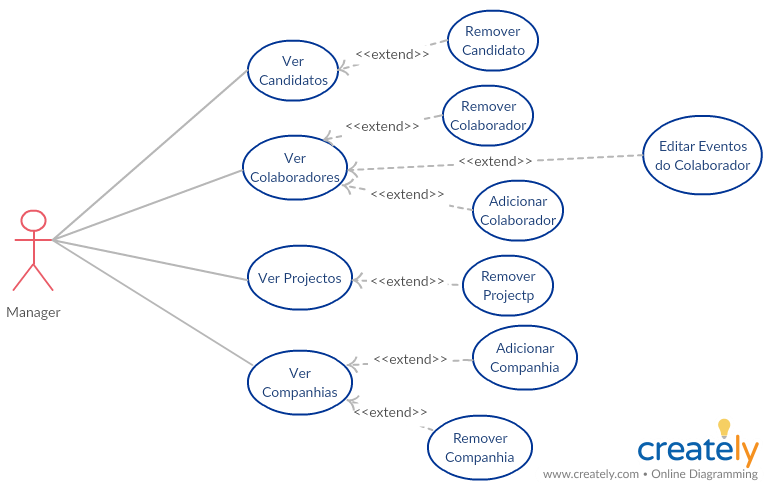


Figura 6 - Casos de utilização, Gerente.

### 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:

* Verificar e editar as suas informações pessoais 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.

Estas funcionalidades, podem ser verificadas Figura 6.



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. Este problema e as suas possíveis soluções são abordadas, numa forma simplificada, na subsecção 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 7.



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 duma interface visualmente agradável e simples de utilizar, sem ter de se despender 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.*

### Servidores Fundamentais

Para o desenvolvimento simples e rápido 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;
* Gerar *proxies* de integração;
* Agrupa as capacidades de gestão de sessão, autenticação e configurações para a aplicação;
* Adiciona recursos de registro e monitoramento 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.

### 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*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read-Only_Data) – utilizado quando os utilizadores precisam unicamente de ler informação e quando a dimensão da *data* a sincronizar é baixa;
* [*Read-Only Data Optimized*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read-Only_Data_Optimized) – semelhante ao *Read-Only Data*, mas para dimensões de *data* mais extensiva;
* [*Read/Write Data Last Write Wins*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read%2F%2FWrite_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*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read%2F%2FWrite_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*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read%2F%2FWrite_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 utilizará os padrões [*Read-Only Data Optimized*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read-Only_Data_Optimized)e[*Read/Write Data Last Write Wins*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read%2F%2FWrite_Data_Last_Write_Wins). O primeiro para ecrãs onde será verificado unicamente informação que não se pode alterar, exemplo entrevista e candidaturas a vagas, que podem incluir uma extensa quantidade de informação. Já o segundo padrão será utilizado para qualquer alteração realizada na aplicação, sendo que as informações a alterar (informações pessoais no currículo) são unicamente acessíveis a um candidato e por isso não existirá concorrência.

### Segurança de Infomação

A aplicação IView não adiciona capacidades de segurança notáveis às fornecidas pelo OutSystems no desenvolvimento normal das aplicações. Tal segurança é estabelecida considerando vários problemas comuns as aplicações web. Desta forma, serão verificados os problemas mais importantes e como a OutSystems os tenta corrigir.

* *Injection* - A Outsystems inclui vários filtros que garante qualquer eliminação de código HTML, JavaScript ou SQL que permitam injeção de informação que levante problema na base de dados;
* Autentificação – As características da autentificação da OutSystems garante que nenhum identificador de utilizador compõem os cookies. As sessões tem uma validade e a sua identificação nunca é incluída em URLs. Por último, as passwords são guardas 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-base* 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 é qualquer utilização do identificador do utilizador para leitura ou escrita de valores associados ao mesmo são obtidos pela *function* GetUserId(). Este passo garante que tais leituras e escritas não ocorrem com um falso identificador.

Por 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 Oustystems.

# Solução Proposta

O foco deste capítulo será a proposta do projeto gerado, referindo a estrutura da aplicação, que segue a arquitetura *4 Layers Canvas* estabelecida pela OutSystems 3.1, a base de dados da aplicação 3.2, os *wireframes* que demonstram alguns exemplos de páginas que compõem a aplicação *web 3.3* e, por último, alguma lógica para cumprir certas funcionalidades importantes a notar 3.4 e 3.5.

## Arquitetura do projeto

Como qualquer projeto desenvolvido 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 8 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* será composta pelo modulo IViewCore, onde é desenvolvido a base de dados e algumas estruturas partilhadas;
* *Library Layer* –este projeto inclui, mas não é limitado, as dependências:
  + *Liverpool Template*;
  + *Silk UI;*
  + *Html2PdfConverter*;
  + *FullCalendar2*;
  + Google Maps *Mobile*;
  + Google Maps *Web*;
  + In-App *Notification Core*;
  + *OneSignal Plugin*.

## Modelo entidade-associação da base de dados

Para facilitar a demonstração do modelo criado para o projeto, o modelo foi divido 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 irá ser 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 Dado.

### 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 9 demonstra o modelo entidade-associação desta componente do modelo.

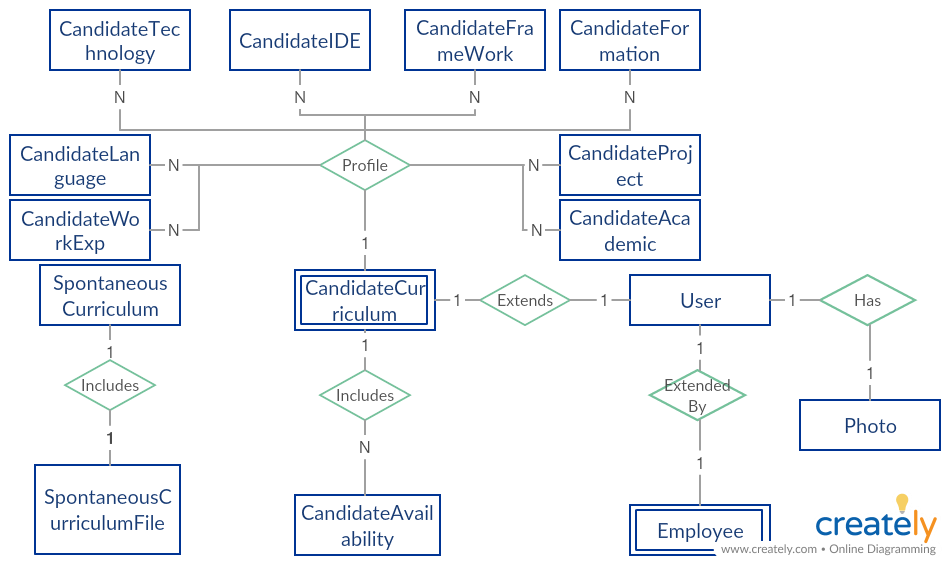


Figura 10 - Modelo EA, Candidatos.

As entidades da Figura 9 são:

* *User*, uma entidade padrão da OutSystems, representa os utilizadores registrados 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;
  + A razão por 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 (tempo de processamento) desnecessário sobre a aplicação, principalmente porque o ficheiro só é necessário numa forma mais individual
* *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 experiencia 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 experiencia 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.

### 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 10 demonstra estas entidades.

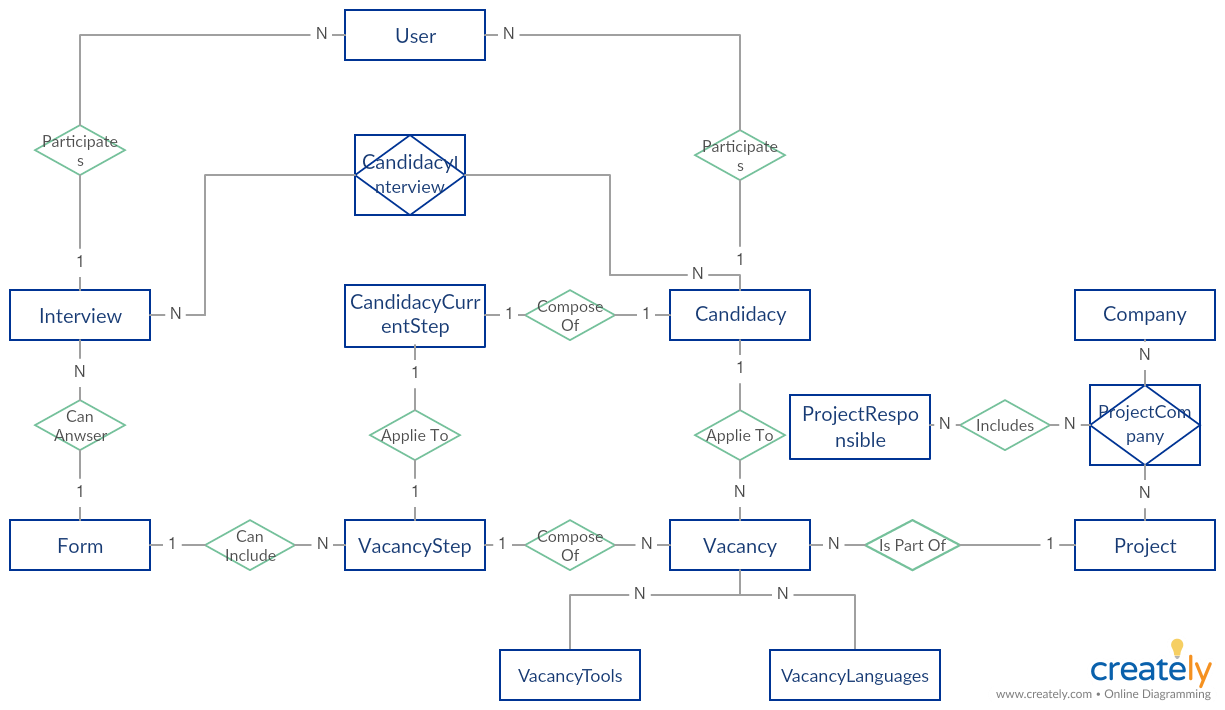


Figura 11 - Modelo EA, Vagas.

As entidades incluídas na Figura 10 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 vairás candidaturas;

### 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 11 que se segue.

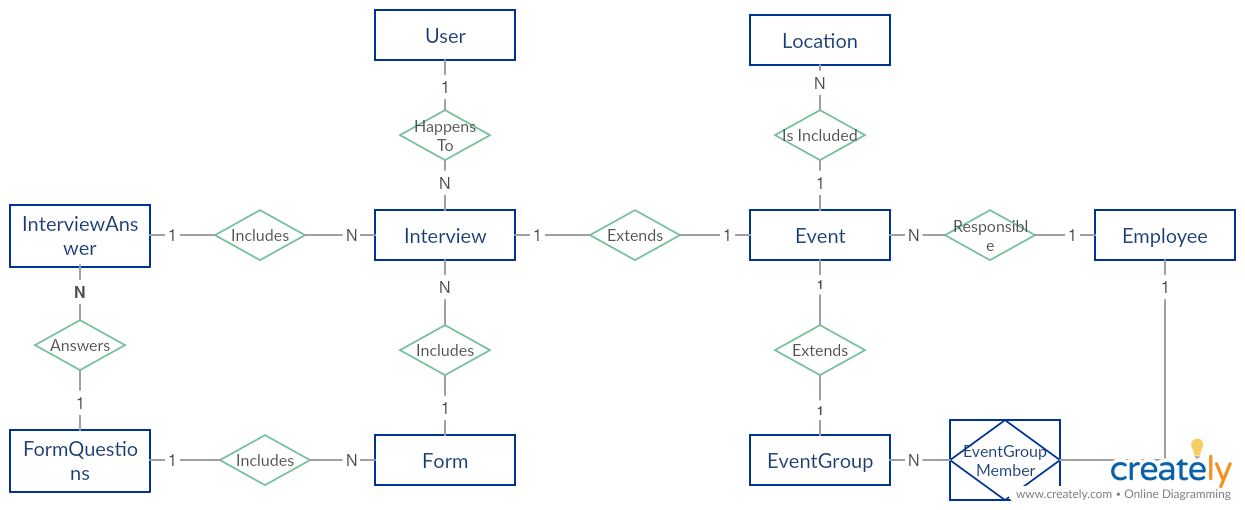


Figura 12 - Modelo EA, Eventos.

Na Figura 11 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.

## *Wireframes* do projeto

Para esta secção serão demonstradas algumas *frames* de páginas importantes, em que algumas tem uma estrutura única na aplicação e outras servem de exemplo de uma estrutura utilizada habitualmente em páginas com uma função especifica, 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 12**.** acompanhada pela Figura 13 e Figura 14 onde é demonstrado os dois possíveis menus de utilizador).



Figura 13 - Frames, Home Page



Figura 14 - Frames, Menu de Colaboradores.



Figura 15 - Frames, Menu de Candidatos.

Começando com o *header*, este inclui sempre um *link* para a *Home Page (*Figura 13*)* à esquerda, mais à direita aparece ou um *link* para a página de *login* ou a possibilidade de abrir um menu onde é possível aceder a página de informação geral do utilizador ou a página de *logout*.

Como se pode verificar pela Figura 14 e a Figura 15, existem dois menus diferentes, o primeiro demonstrado é o menu de colaboradores e o segundo o menu de candidatos. Cada menu é composto por um conjunto de *ButtonDropMenu* que incluem vários links para páginas mais importantes.

Originalmente foi considerado utilizar um *Ballon*, mas entre algumas limitações visuais como também uma má interação entre o *Ballon* e vários *plugins* da *Forge*, o *ButtonDropDown* tornou-se a opção viável.

Na Figura 15, Figura 16**.** é possível verificar parte da página de informação geral da PS IT que introduz dois elementos visuais fundamentais à aplicação.



Figura 16 - Frame, MoreInfo Values Tab.



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 dividas 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 15 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 17 e também verificar, e pela Figura 18, editar.



Figura 18 - My Curriculum, Show.



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 19 demonstra parte da página que introduz alguns elementos notáveis e regularmente utilizados na aplicação.

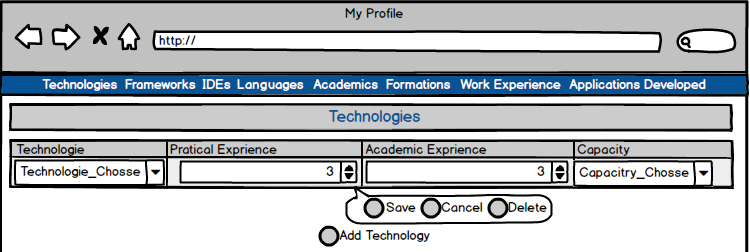


Figura 20 - Frames, MyProfile.

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]](#footnote-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 20.



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 21, que mostra todas as instâncias da entidade *Form*.



Figura 22 - Frames, Forms.

Esta página apesar de simples mostra os componentes mais comuns das páginas que demonstram as instância duma entidade, a primeirá 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 *Tabel* 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 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 21, 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 22, onde é verificada a página de listagem de aplicações a vagas existentes.



Figura 23 - Frames, Applications.

Como se pode ver entre a Figura 21 e a Figura 22 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 titulo 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 23.

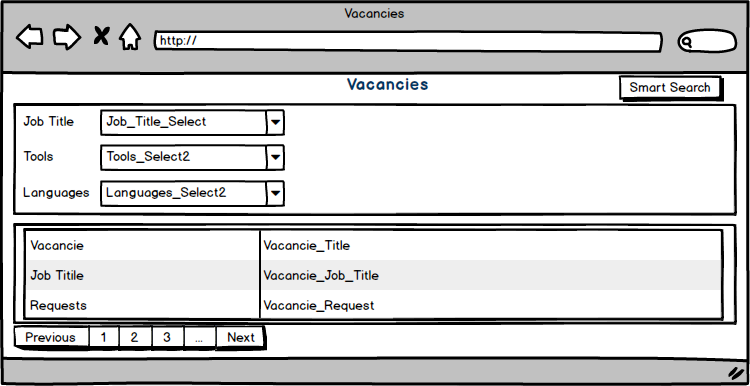


Figura 24 - Frames, Vacancies.

Como se pode observar, a pesquisa de vagas é realizado sem nenhum *input* de texto, sendo realizado unicamente por escolha de um valor de três *Combo Boxs*, sendo uma *Combobox* 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 24.



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 instância listadas, para uma página que demonstra a informação da instância mais detalhada. Um exemplo simples deste tipo de página pode ser verificado na Figura 25, que demonstra a página de um candidato.



Figura 26 - Frames, Candidate.

Por vezes estes tipo 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 páginausada.

Um exemplo da utilização das *Tabs* nestas páginas pode ser verificado na Figura 26, Figura 27 e Figura 28 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 27 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 em qualquer outra 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 29, que demonstra a página que permite introduzir um formulário.



Figura 30 - Frames, FormEditAdd.

A página da Figura 29, 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 será 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-up*s. Um exemplo de uma *Pop-up* utilizada neste tipo de páginas pode ser verificada na Figura 30.



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.

## *Desenvolvimento Web*

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

### 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 entidades[[4]](#footnote-4), 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

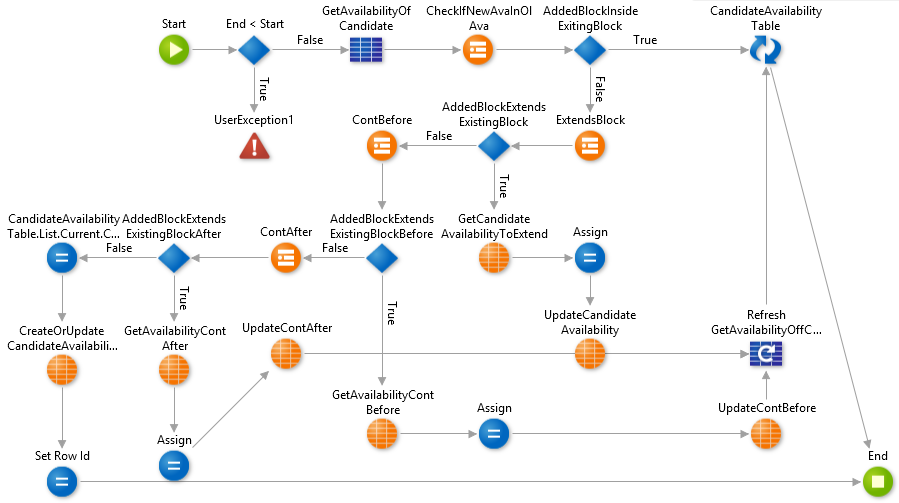
Grande parte desta páginas incluem o *rich* *widget* *List\_Navigation*, que fornece uma páginação simples da lista. Esta envolve sempre uma *action*, que realiza a alteração da informação em função da página. É realizado o *Aggregate* que obtém a informação da lista a páginar, seguido de um *Ajax Refresh*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



Código 2 - Páginação

* + 1. **Adicionar** disponibilidades

A página das disponibilidades de um candidato não inclui uma preparação particularmente notável, sendo que na mesma apenas é obtido as instâncias de *CandidateAvailability* associados ao candidato. Mas a introdução duma nova instância de *CandidateAvailability*, não é tão simples como criar uma nova instância e adiciona-la a base de dados, como se pode verificar na Código 3.



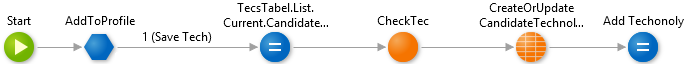
Código 3 - *Adicionar CandidateAvailability*

Esta complexidade deve-se ao facto de que quando um candidato introduz um novo bloco de disponibilidade, existe três formas deste novo bloco de 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 inico 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 intercessão.
  + Não ocorrendo qualquer intercessão, é adicionado o novo bloco a base de dados.

### Adicionar e Remover do *Profile*

Na Figura 19 observámos 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 Tabels* 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 dois *actions*: um para adicionar (Código 3) e outro para remover (Código 4), sendo que é usado um *Switch* que escolhe as ações a realizar tendo em conta o input que recebe. Ambas as imagem mostram parte da *actions*, sendo que tal é simplesmente repetida 8 vezes.



Código 4 - Adicionar ao Profile

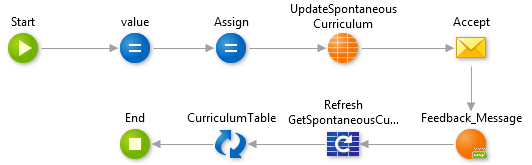


Código 5 - Remover do Profile

Ao adicionar é sempre verificado sé 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.

### 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.

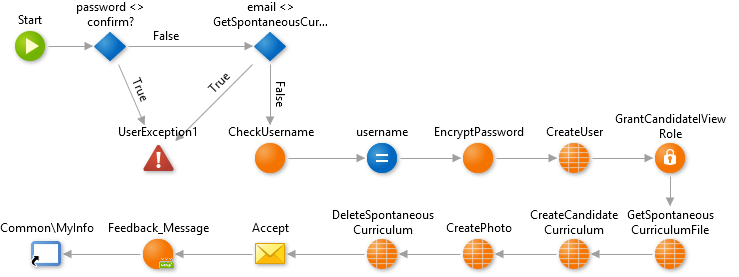


Código 6 - Começar a criação do Candidato

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

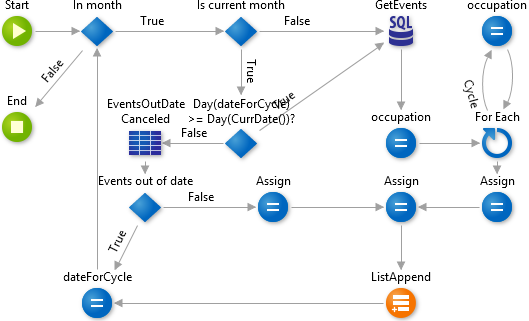
Da *action* do Código 7 o único passo a notar é a execução do *CheckUsername*, sendo que todos os outros passos criam um novo candidato, fornecendo o papel correto e remover qualquer instância da base de dados desnecessária.

O *CheckUsername* existe para tentar garantida 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 semelhante *username*. Caso tal aconteça, é adicionado ao *username* um número que é maior que o número no *username* igual mais recentemente criado.

### 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 7.



Código 7 - 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 titulo 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 8. 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 8 - Alteração do calendário de mês para semana

Esta *action* é bastante simples, mas a *FormEventsWeek* (Código 9) utilizada é focada num ciclo para criar *Events*, que representam os eventos em que o utilizador atual participa numa semana.



Código 9 - Formação dos Events para uma semana

Para tal a *function* começa por criar dois valores: *dateStart* - inico 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 obtidos 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 10, que esta associada a característica *OnNotify* do calendário semanal.



Código 10 - 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 acorra no passado é verificado 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:

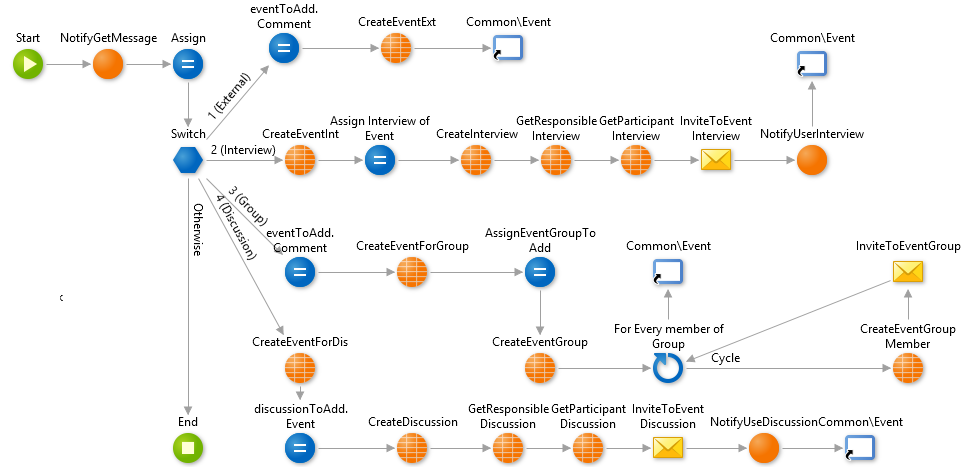
* O novo evento tem o início entre o início e fim de evento já existente, mas com o fim fora
* O novo evento tem o fim entre o início e fim de evento já existente, mas com o início fora
* O início do novo evento acontece antes do início de um evento e o fim depois do fim do mesmo evento

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

### 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 paramentos 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 certa características essenciais do *Event*, o mesmo pode utilizar o botão de *Add* que enfoca um *Pop-up* onde é possível selecionar uma localização, a única característica comum a todos os *Events* ainda a definir. Quando o mesmo é selecionado a *action* verificada na Código 11 ocorre.



Código 11 - 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*.

### 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 12.



Código 12 - Adição de *Vacancy*

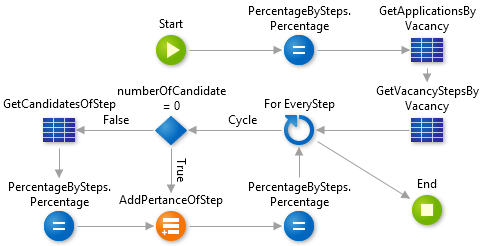
Nesta ação, depois de ser criada a instância de *Vacancy* que representa a oferta a adicionar, são executados três ciclos diferentes. No primeiro acontece para a lista de passos adicionados, sendo adicionado um *VacancyStep* para cada passo

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.

### 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 Figura 28. Os valores que preenchem este *Pie Chart* são formados pela *action* demonstrada na Código 13.

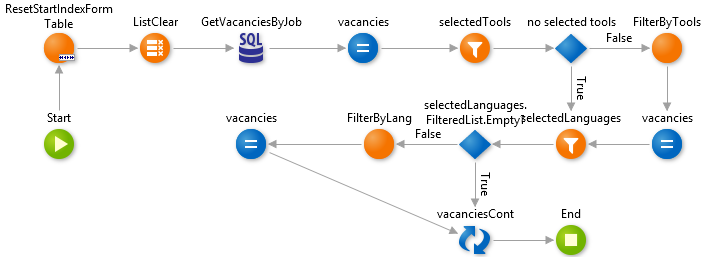


Código 13 - Geração de percentagens para o *Pie Chart*

Para formar tais informações a *action* começa por obter todas as candidaturas a oferta a demonstrar *no Pie Chart*. Para cada passo é obtido o número de candidaturas nesse mesmo passo, que em conjunto com o número de candidaturas obtidas no início, permite obter a percentagem de candidaturas nesse passo. Esta percentagem mais um *label* são depois adicionadas à lista que fornecem a informação para a *Pie Chart*. Na situação de não haver uma só candidatura então é realizado o mesmo ciclo, mas de forma que o primeiro passo tenha 100% e todos os outros 0%.

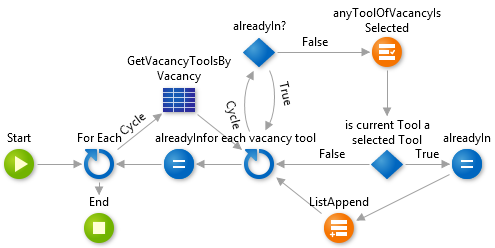
### Pesquisa de ofertas

Qualquer candidato ou utilizador não registrado consegue verificar todas as ofertas disponíveis pela PS IT como também realizar uma pesquisa inteligente de ofertas pelo o tipo de posição, ferramentas e linguagens relativas a vaga. Esta pesquisa ocorre através da interação com um formulário que com cada alteração realiza a *action* verificada no Código 14.

****

Código 14 - Pesquisa de *Vacancy*

A pesquisa começa por obter todas as vagas cuja posição associada é a escolhida pelo candidato, excluindo as vagas às quais o candidato já se encontra inscrito. Destas, são obtidas todas as que incluem pelo menos uma linguagem e pelo menos uma ferramenta selecionada. Para verificar se uma oferta inclui uma ferramenta ou linguagem selecionada são executadas duas *actions* semelhantes, com um exemplo verificado no Código 15.

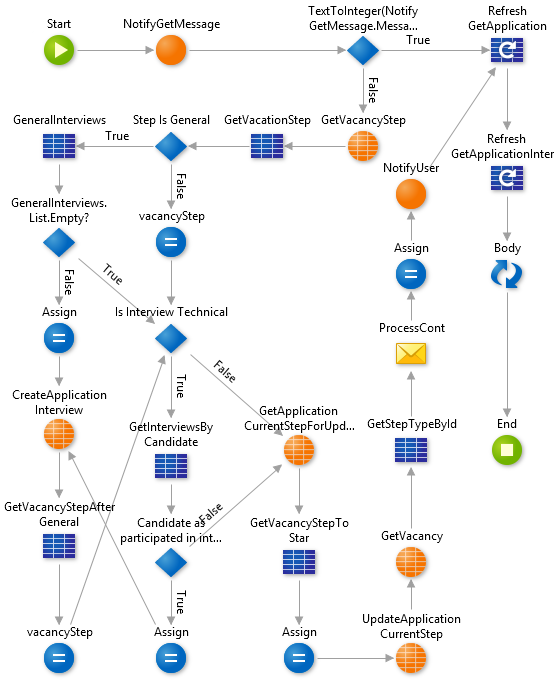


Código 15 - Verificar *Vacancy* em função das suas ferramentas

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.

### 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 16 que se segue.



Código 16 - Continuação duma Candicy

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 *VacanyCurrentStep* 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*.

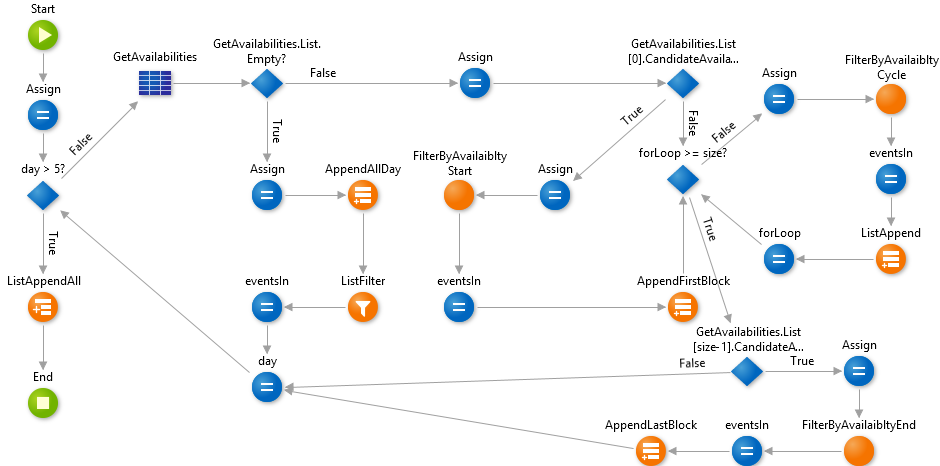
Tendo chegado ao passo em que o candidato ainda tem de realizar, a candidatura é atualizada para tal passo e o candidato é avisado da continuação da sua candidatura, tanto por email e por notificação, acabando de seguida a *action.*

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.

### 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.

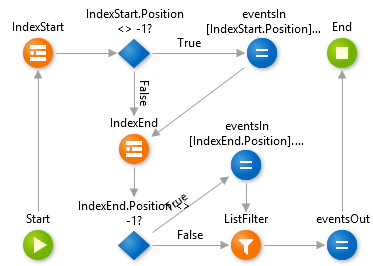
Por isso a aplicação permite a criação de blocos temporários, que podem ser utilizados para limitar a criação de novos Eventos. Para tal depois de se obter os eventos duma semana, usando o código verificado no Código 9, é realizado a *actions* verificada no Código 17.



Código 17 - Adicionar disponibilidade

Nesta *action* são obtidas todas as disponibilidades do candidato por cada dia da semana e para cada é introduzido 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 18.



Código 18 - 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.

## *Desenvolvimento Mobile*

Como a secção anterior, esta secção irá demonstrar 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.

### 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 dipositivo registrado no mesmo servidor, associando os mesmos à aplicação já registrada.

Por isso a ação de *login* que ocorre na aplicação *mobile*, que pode ser averiguada no Código 18 , 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 registro da mesma no servidor da *OneSignal*.



Código 19 - Registro de utilizador na *OneSignal*

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

### Sincronismo

Na secção 2.3 foi referido o conceito de sincronismo que será aplicado na vertente *mobile* do projeto, sendo a mesma bastante simples e leve. Esta sincronização ocorrera em dois padrões: a primeirá, e mais simples das duas, *Read-Only*, ocorre pela ação *SyncOnLogin*, que ocorre no *login*, como se pode verificar no Código 19.

Esta ação, que pode ser verificada no Código 20, 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 20 - 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*](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/Read%2F%2FWrite_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*.

# Avaliação Experimental

Neste capítulo iremos apresentar uma série de teste que têm como objetivo demonstrar as funcionalidades mais importantes da IView, através de uma série de *screen shots* da aplicação *web* e *mobile*, da tabela da base de dados que demonstra a alteração realizada sobre a mesma e por último, em algumas situações, dos emails enviados pela aplicação.

Para isso cada funcionalidade importante inclui um teste onde é verificado os valores fornecidos pelo utilizador e o resultado na base de dados e na aplicação *web* em si.

## 4.1 Candidatura Espontânea – Utilizador não registrado

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

* + 1. **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 servirá como o seu currículo, como se pode ver na Figura 32.

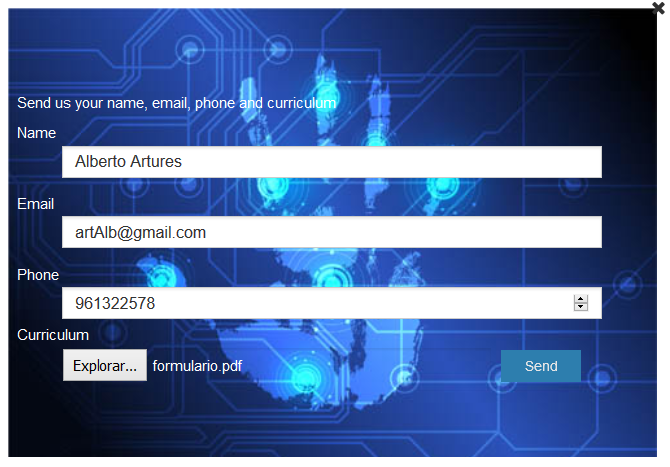


Figura 32 - Candidatura Espontânea

* + 1. **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 da candidatura ser registrada corretamente, uma das mensagens da Figura 36.

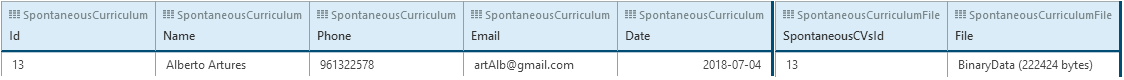


Figura 34 - Introdução de SpontaneousCurriculum e de SpontaneousCurriculumFile

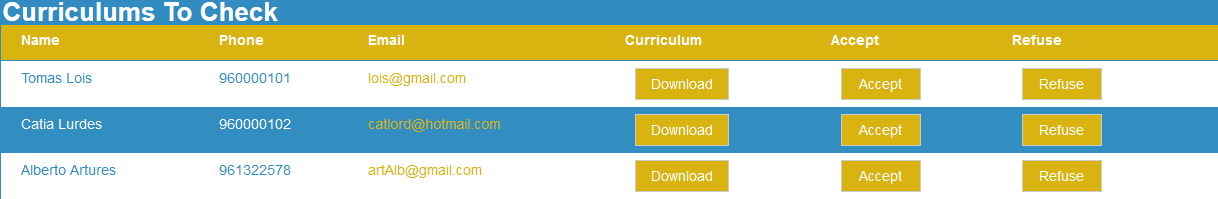


Figura 34 - Candidaturas Espontâneas

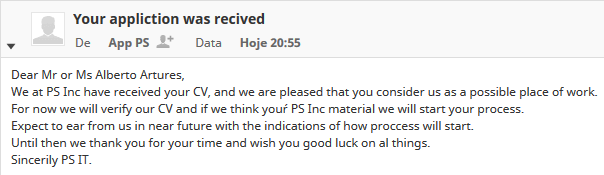


Figura 35 - Email de submissão da candidatura espontânea

C:\Users\Diogo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\mensagens.png

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”.

* + 1. **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.

* + 1. **Resultado Esperado**

Com a ação do “Accept” a instância de *SponatnousCurriculum* é alterado para incluir o valor de segurança, como se pode verificar na Figura 37 como também no envio do email da Figura 40. Já a ação do “Refuse” simplesmente resulta no enviou do email da Figura 41.

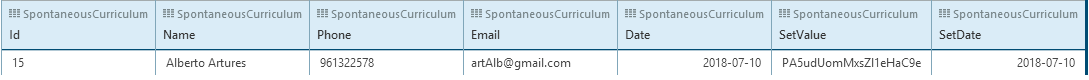


Figura 37 - Alteração sobre SponatnousCurriculum

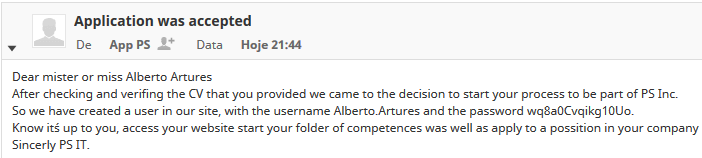


Figura 38 - Email de estabelecimento do candidato

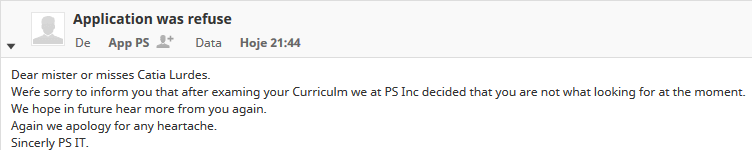


Figura 39– Email de recusa da candidatura espontânea

## 4.3 Estabelecer novo candidato – Utilizador não registrado

Tendo recebido o email da Figura 40, um utilizador ainda não registrado pode aceder a uma página onde lhe será possibilitado o seu estabelecimento como candidato da aplicação.

* + 1. **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 42.

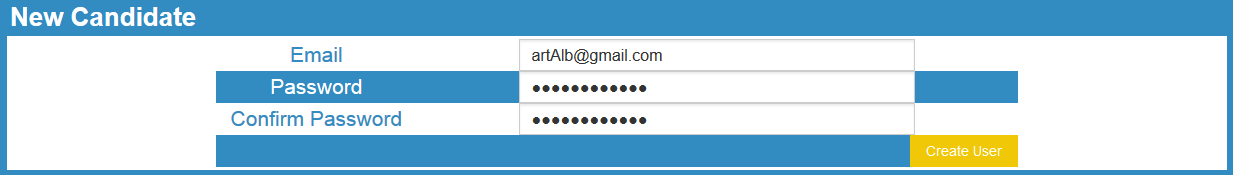


Figura 40 - Criação do novo utilizador

* + 1. **Resultado Esperado**

Se os valores fornecidos estiverem corretos, então uma instância da entidade *User* e outra da entidade *CandiadateCurriculum* serão iniciadas, Figura 41 e Figura 42, como também é enviado o email, da Figura 43, onde é incluindo o *username* e *password* do novo *user*.

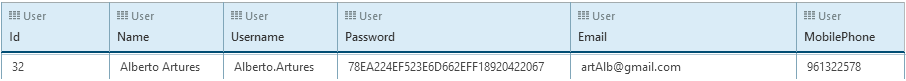


Figura 41 - Estabelecimento de User

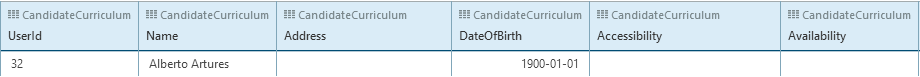


Figura 42 - Estabelecimento de CandidateCurriculum

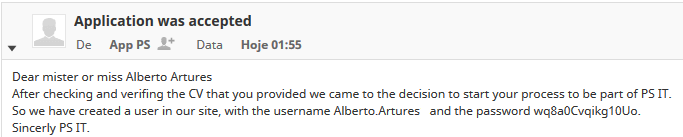


Figura 43 - Email com novo utilizador

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.

C:\Users\Diogo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\messaegneerr.png

Figura 44 - Mensagens de erro

## 4.2 Autentificação 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.

**4.4.1 Ação**

Pela página verificada na Figura 45, o utilizador consegue fornecer os seus valores de autentificação. Este será 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 autentificação da aplicação *mobile*.



Figura 45 - Autentificação Web

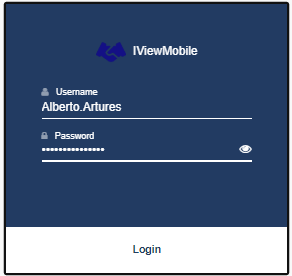


Figura 46 - Autentificação Mobile

## 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.

* + 1. **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*, dai os valores alterados na primeirá podem ser verificados na segunda.

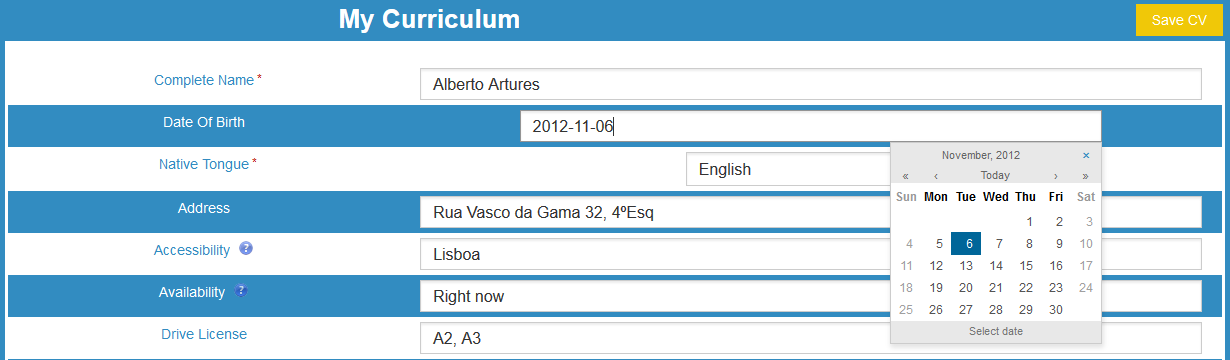


Figura 47 - Alterar Currículo, *Web*

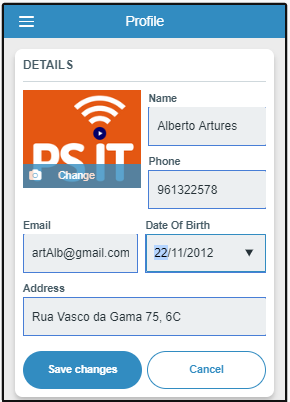


Figura 48 - Alterar Currículo, Mobile

* + 1. **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

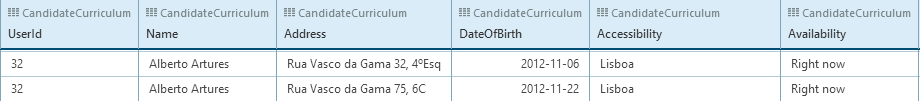


Figura 49 – Alterações a *CandidateCurriculum*



Figura 50 – Currículo depois da primeirá alteração

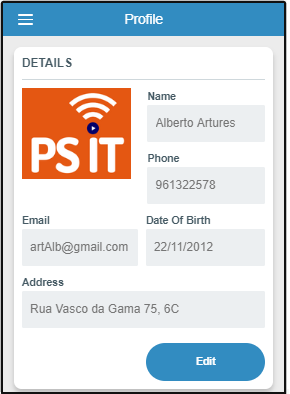


Figura 51 - Currículo depois da segunda alteração

### Atualizar Disponibilidade – 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.

* + 1. **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.

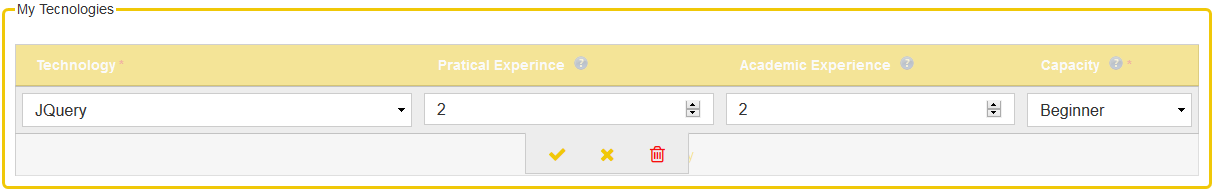


Figura 52 - Introduzir Tecnologia

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.

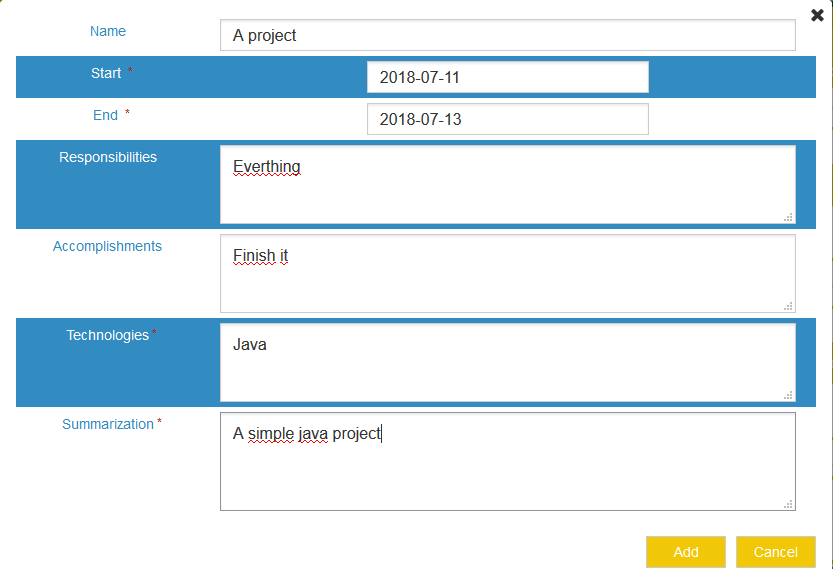


Figura 53 - Introduzir Projeto

* + 1. **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.



Figura 54 - Alteração a *CandidateTechnology*

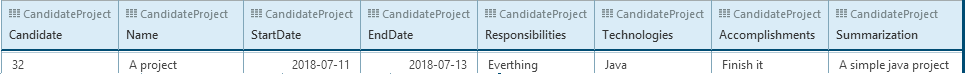


Figura 55 - Alteração a *CandidateProject*

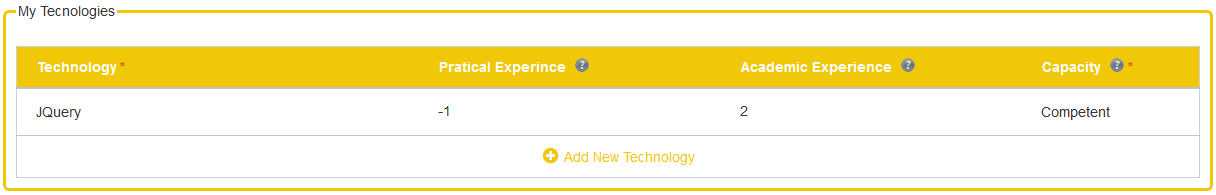


Figura 56 - Tecnologias do Candidato

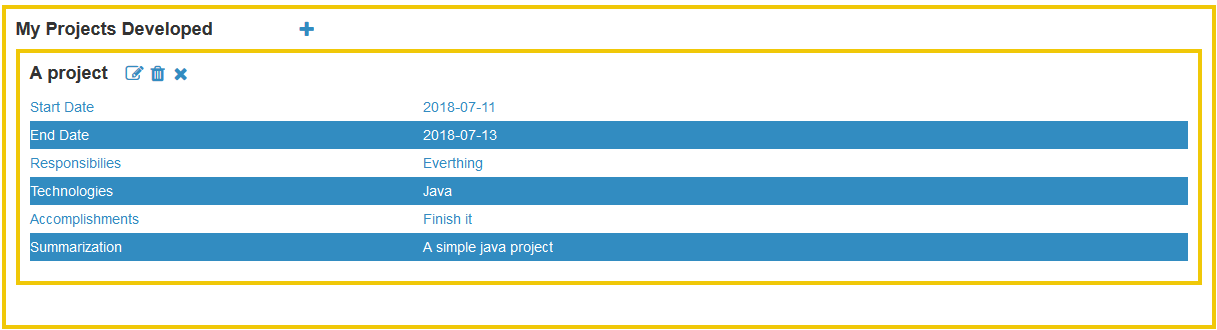


Figura 57 - Projetos do Candidato

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 outa já existente.

tecnologias - erros

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.

projectos - erros

Figura 59 - Mensagens de Erro de Projetos

### 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.

* + 1. **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 pre-definidos, pela a utilização de *selects*.

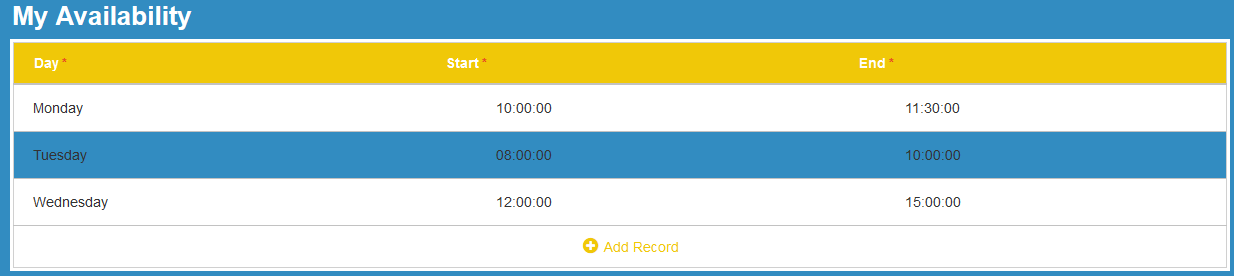


Figura 60 - Inserir Disponibilidades

* + 1. **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(Turesday) 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.

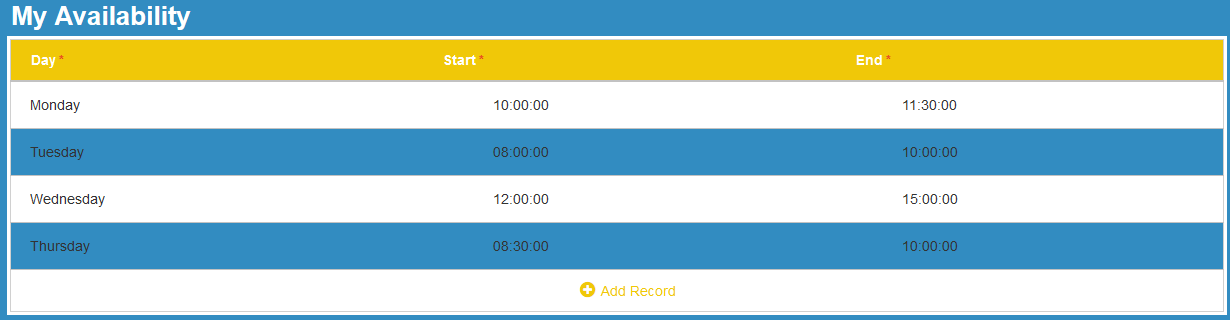


Figura 61 - Disponibilidades depois da introdução

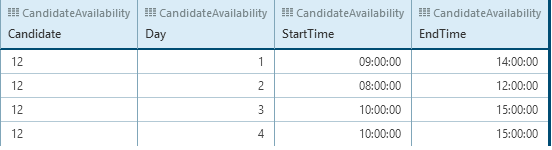


Figura 62 - Alteração a *CandidateAvailability*

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.

error

Figura 63 - Mensagem de erro ao inserir disponibilidade

### Estabelecer Empresa – Administrador

Os administradores da IView podem estabelecer empresas parceirás, 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.

* + 1. **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.

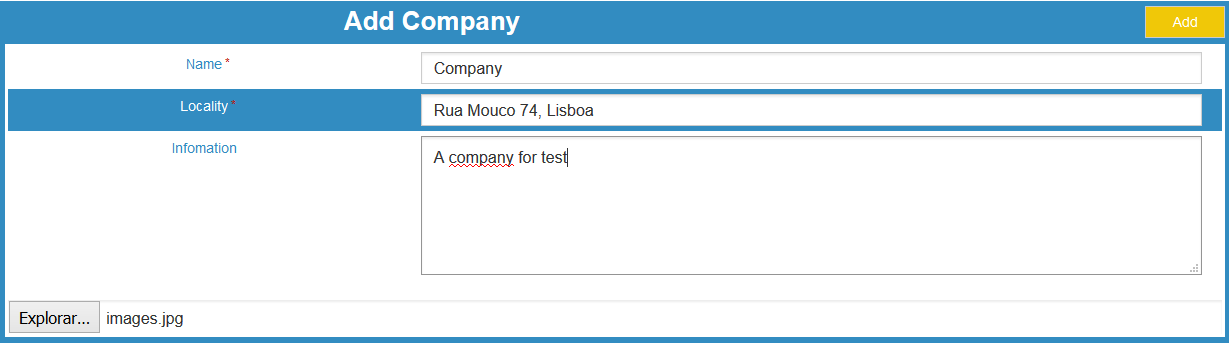


Figura 64 - Introduzir Empresa

* + 1. **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.

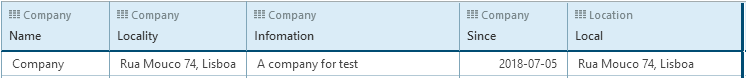


Figura 65 – Nova instância de *Company*

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.

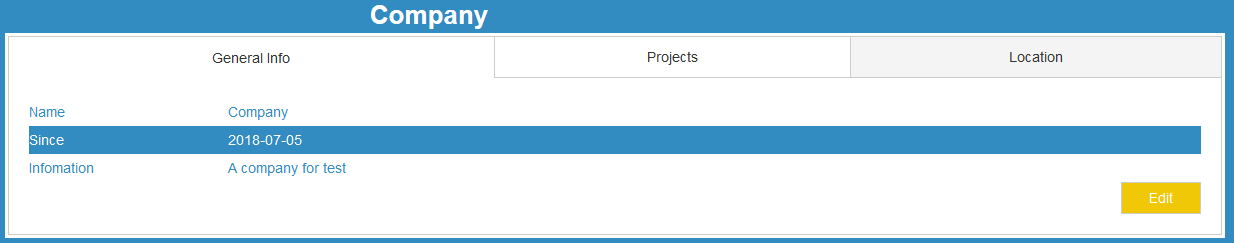


Figura 66 - Empresa introduzida, Informação geral

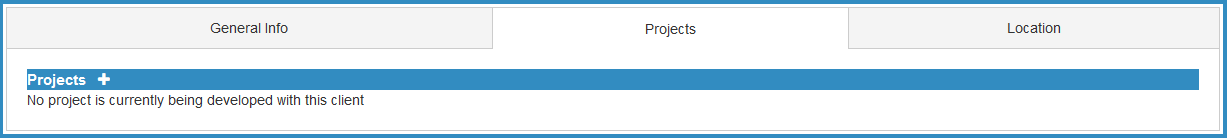


Figura 67 - Empresa Introduzida, Projectos

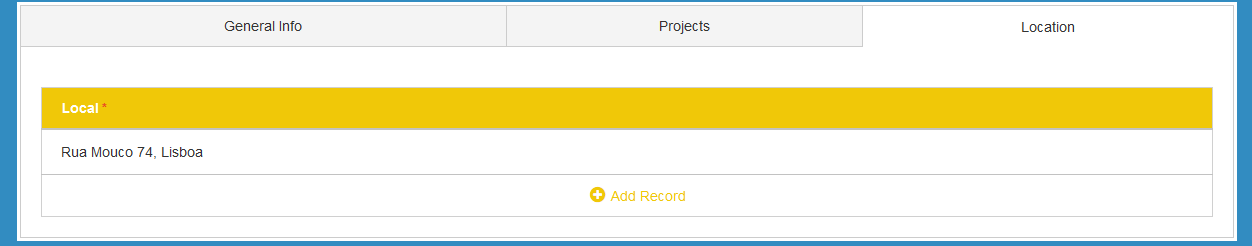


Figura 68 - Empresa Introduzida, Localidades

### Introduzir Projeto – Colaborador

No subcapítulo anterior foi referido o facto que as instâncias da entidade *Company* representam empresas parceirás, 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.

* + 1. **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.

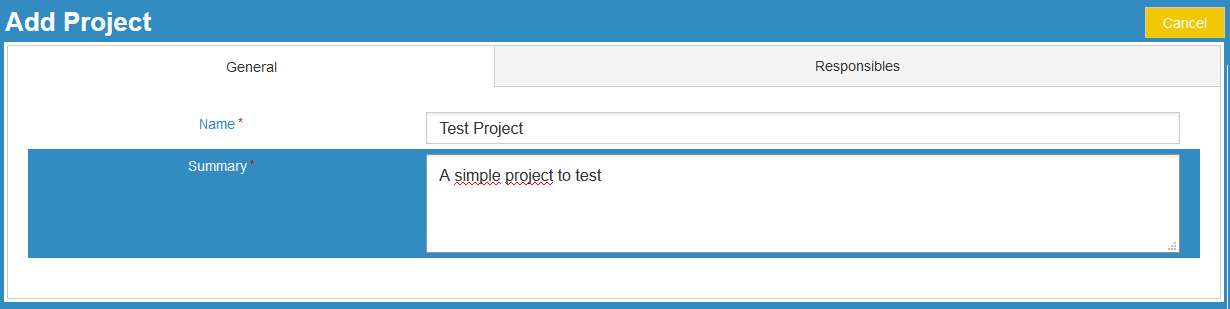


Figura 69 - Introduzir Projeto, informação geral

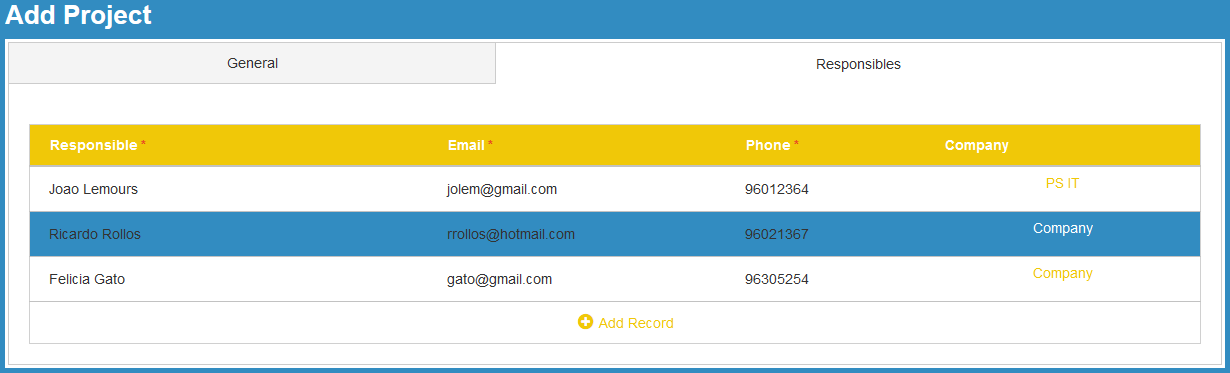


Figura 70 - Introduzir Projeto, responsáveis

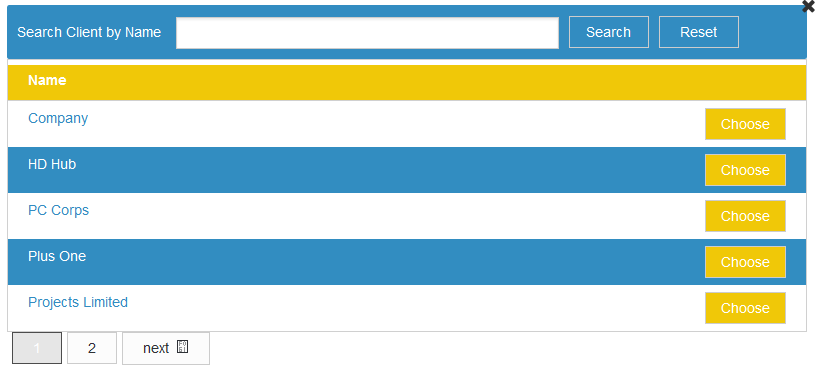


Figura 71 - Adicionar Projeto, escolher empresa de responsável

* + 1. **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 *ProjectResponsable*, uma para cada responsável incluído na adição anterior, figura Figura 74. Apesar de serem atribuído três responsáveis, só são associada 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.

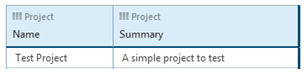


Figura 72 – Nova Instância de Project

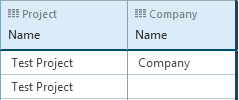


Figura 73 – Novas Instâncias de ProjectCompany

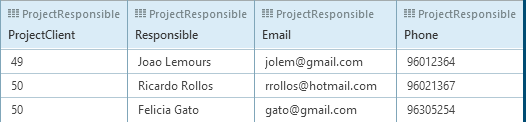


Figura 74 – Novas Instâncias de ProjectResponsible

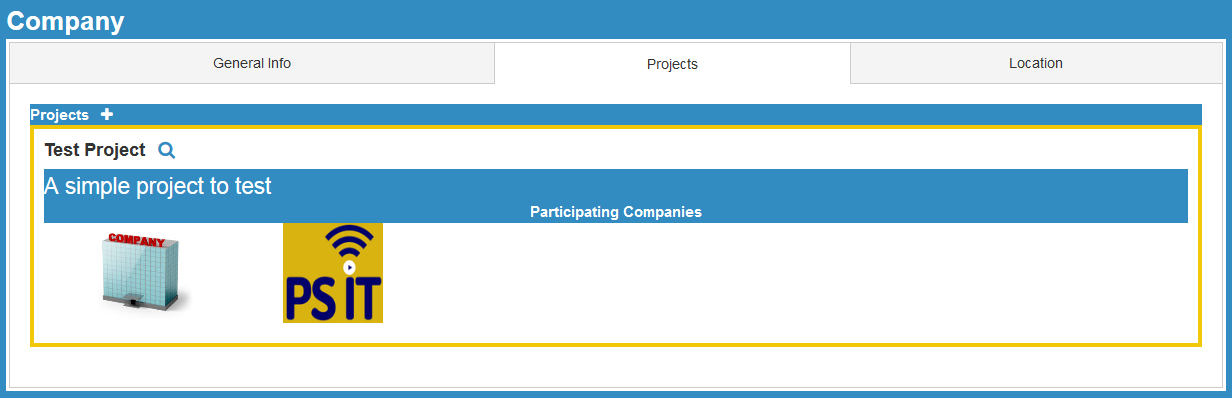


Figura 75 – Projeto na lista da empresa

### 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.

* + 1. **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.

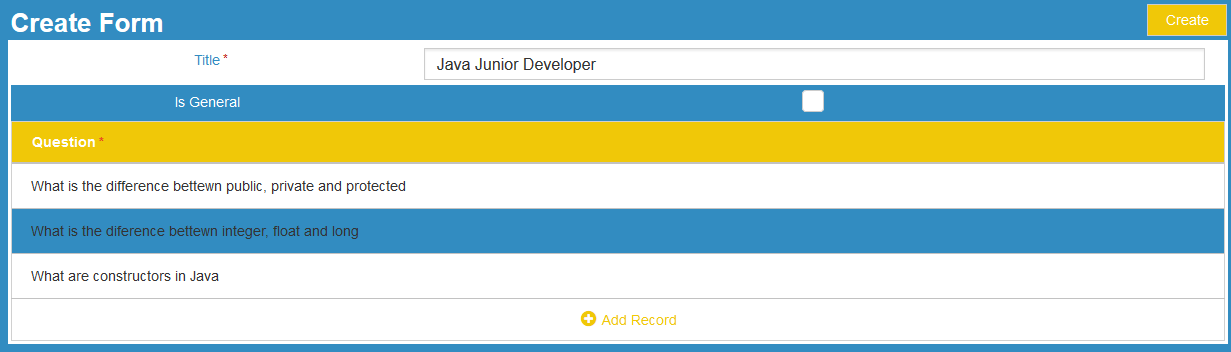


Figura 76 - Criar formulário

* + 1. **Resultado Esperado**

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

form

Figura 77 - Nova instância de *Form*

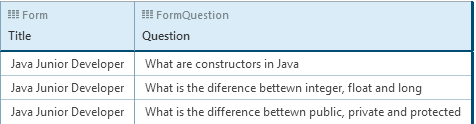


Figura 78 Novas instâncias de *FormQuestions*

### Introduzir Oferta – Colaborador

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

* + 1. **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 pagina do projeto diretamente.

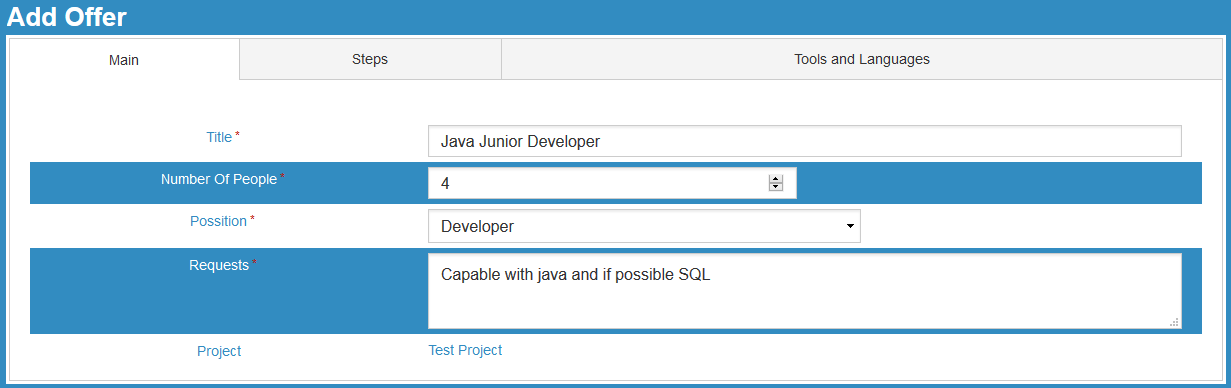


Figura 79 - Estabelecer Oferta, informação geral

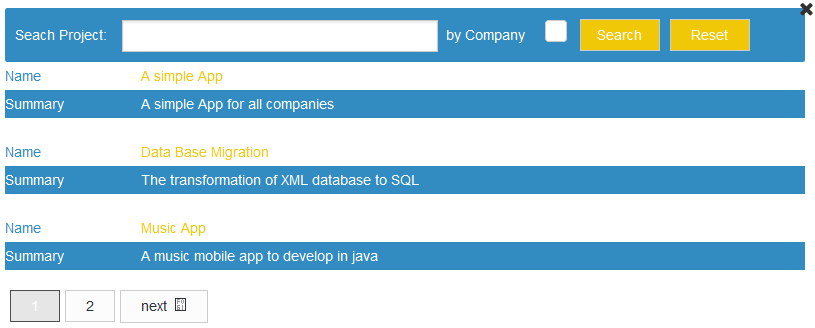


Figura 80 - Estabelecer Oferta, escolha de projeto

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 ate 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.

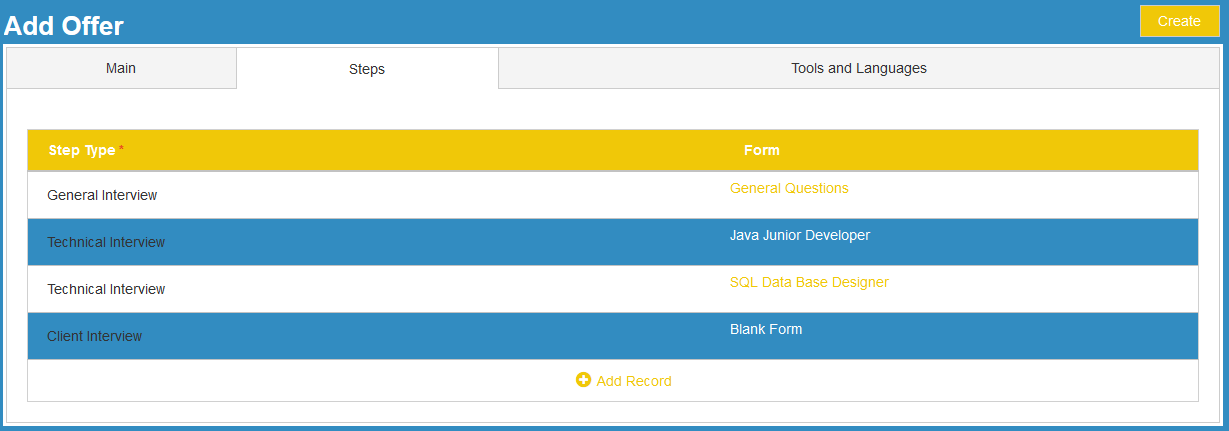


Figura 81 - Estabelecer Oferta, tecnologias e linguages

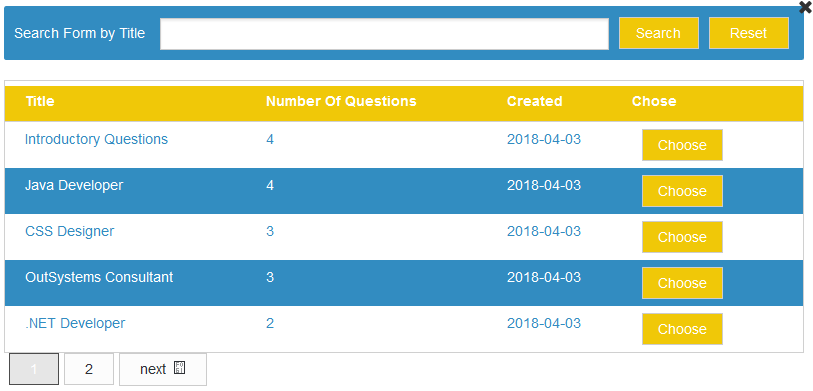


Figura 82 - Estabelecer Oferta, escolher formulário

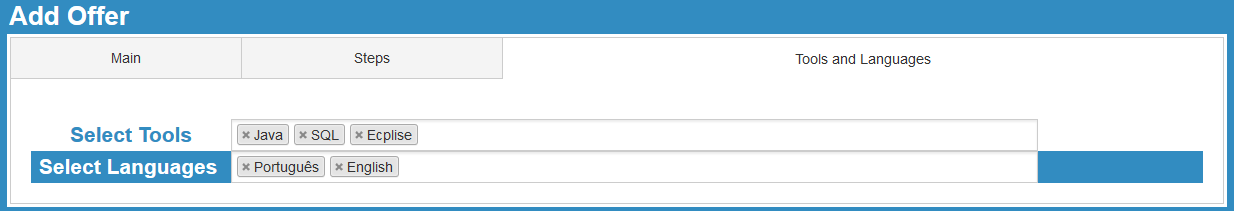


Figura 83 - Estabelecer Oferta, tecnologias e linguagens

* + 1. **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 87 e Figura 86 respetivamente.

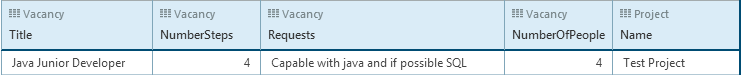


Figura 84 - Nova instância de *Vacancy*



Figura 85 - Novas instâncias de *VacancySteps*

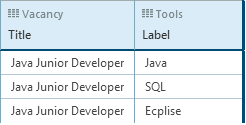


Figura 86 - Novas instâncias de *VacancyTools*



Figura 87 - Novas instâncias de *VacancyLanguages*

### 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.

* + 1. **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 88 mostra o resultado da pesquisa quando é procurado qualquer oferta para uma posição como *Developer*, já a Figura 89 adiciona a tal pesquisa o filtro de linguagens e por último a Figura 90 demonstra a primeira pesquisa mais o filtro por tecnologias.

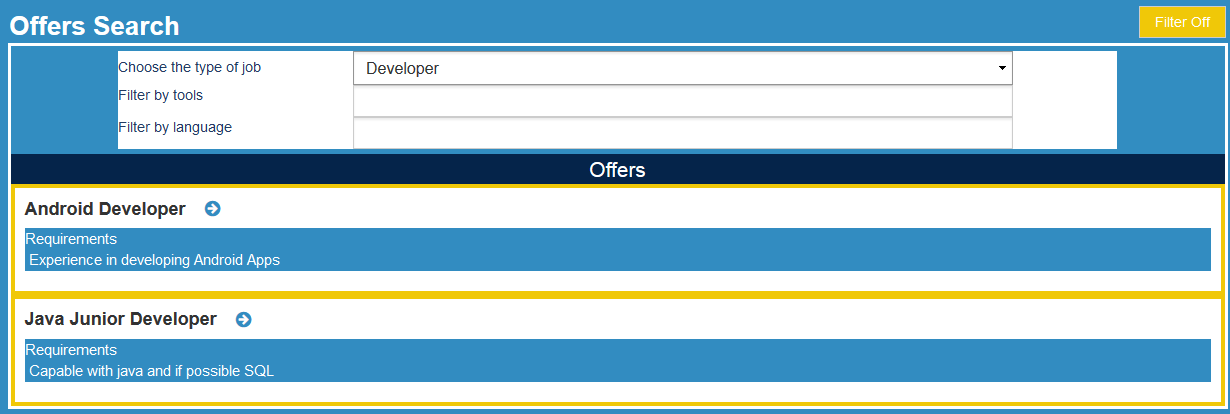


Figura 88 - Pesquisa por posição

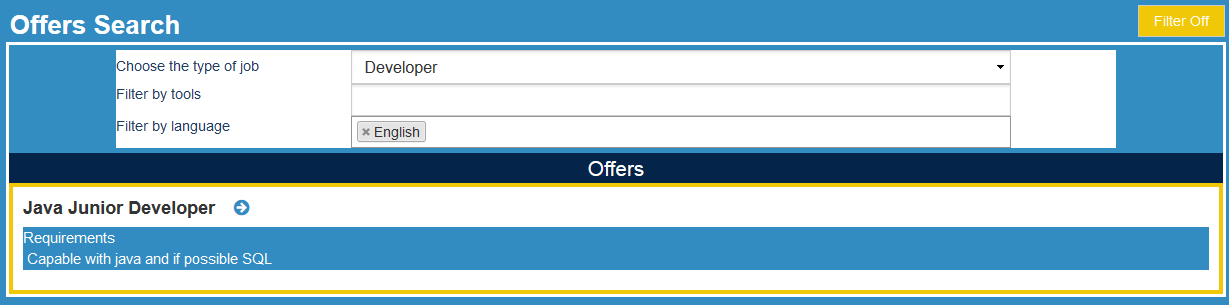


Figura 89 - Pesquisa por posição e linguagem

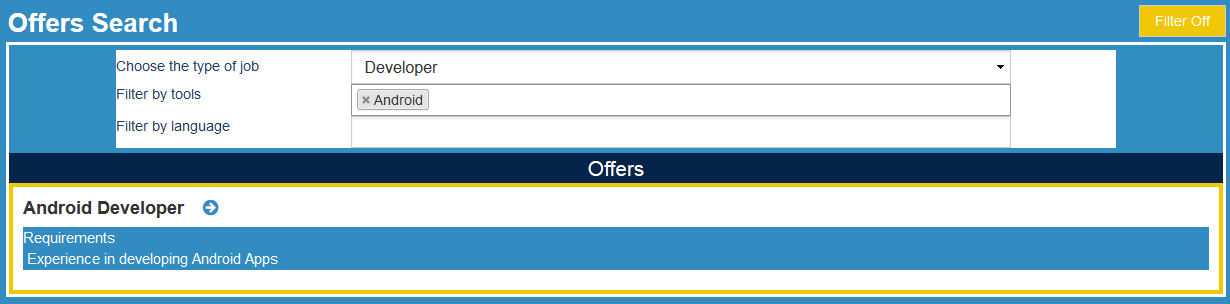


Figura 90 - Pesquisa por posição e tecnologia

* + 1. **Resultado Esperado**

Quando um candidato se candidata a uma vaga, é criada uma instância da entidade *Candidacy*, Figura 91, que representa a candidatura e uma da entidade *CandidacyStep*, Figura 92, que representa a situação da candidatura. Note-se que as instâncias de *CandidacySteps* começam sempre no passo “*Consideration*” e no estado “*ToProcess*”.

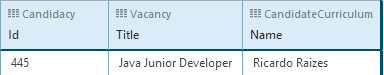


Figura 91 - Nova instância de Candidacy

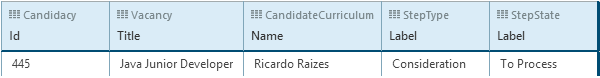


Figura 92 - Nova instância de CandidacyStep

Esta nova candidatura pode ser observada tanto na aplicação *web,* Figura 89,como também na aplicação *mobile*, Figura 94,



Figura 93 - Candidatura, *web*



Figura 94 - Candidatura *mobile*

### 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.

* + 1. **Ação**

Um colaborador consegue verificar qualquer candidatara, 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 95.

Desta é possível escolher um passo em particular, Figura 96, onde pode verificar todas as candidaturas nesse passo, o que permite verificar as candidaturas, como por exemplo a da Figura 97.

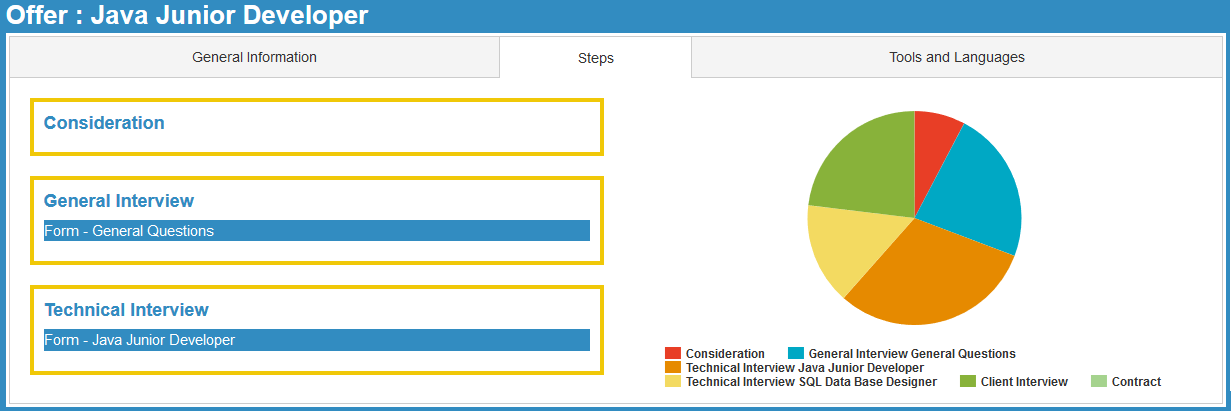


Figura 95 - Passos de uma oferta

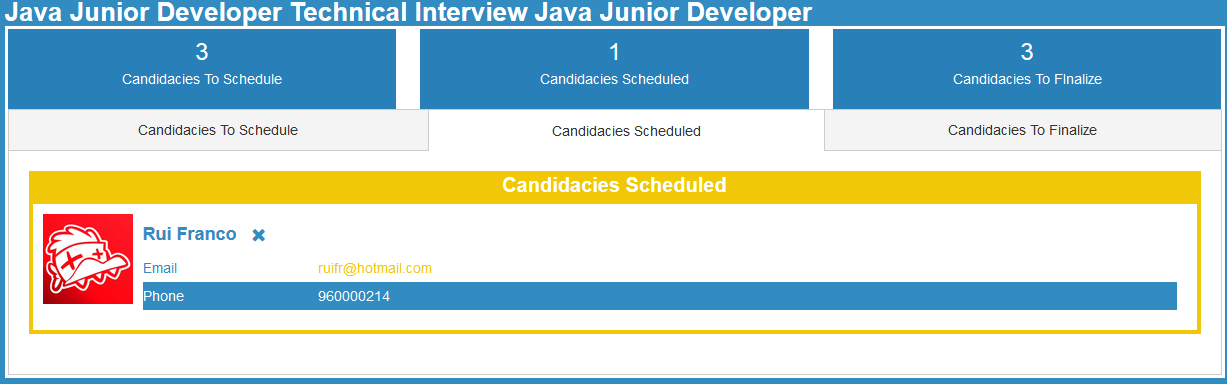


Figura 96 - Passo de uma oferta

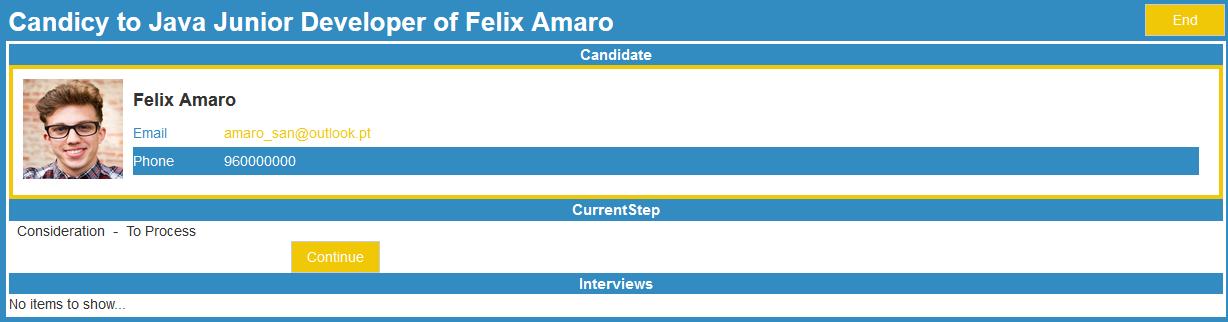


Figura 97 - Candidatura

* + 1. **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 98 como também por uma notificação da aplicação *mobile* Figura 99.

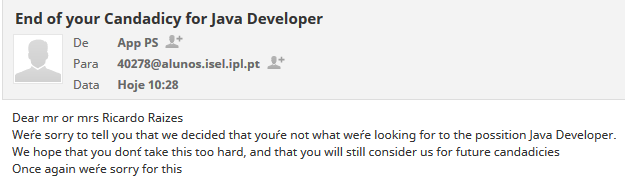


Figura 98 - Finalização de uma candidatura, email

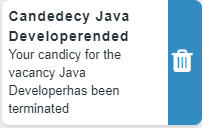


Figura 99 - Finalização de uma candidatura, notificação

Já continuar uma candidatura resulta sempre na alteração da *VacancyStep* para demonstrar o passo a realizar, passando para o estado “*ToScehdule*”, Figura 101.

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 97, a mesma salta três passos, criando as instâncias da *VacancyInterview* da Figura 100.

Estas alterações são verificadas na página da candidatura, como se pode ver na Figura 102.

Independentemente do que ocorre com a continuação, o candidato é informado da continuação em si tanto por email, Figura 103 como por notificação Figura 104.

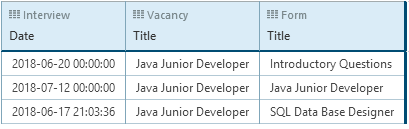


Figura 100 - Novas instâncias de *VacancyInterview*



Figura 101 - Alteração sobre *VacancyStep*

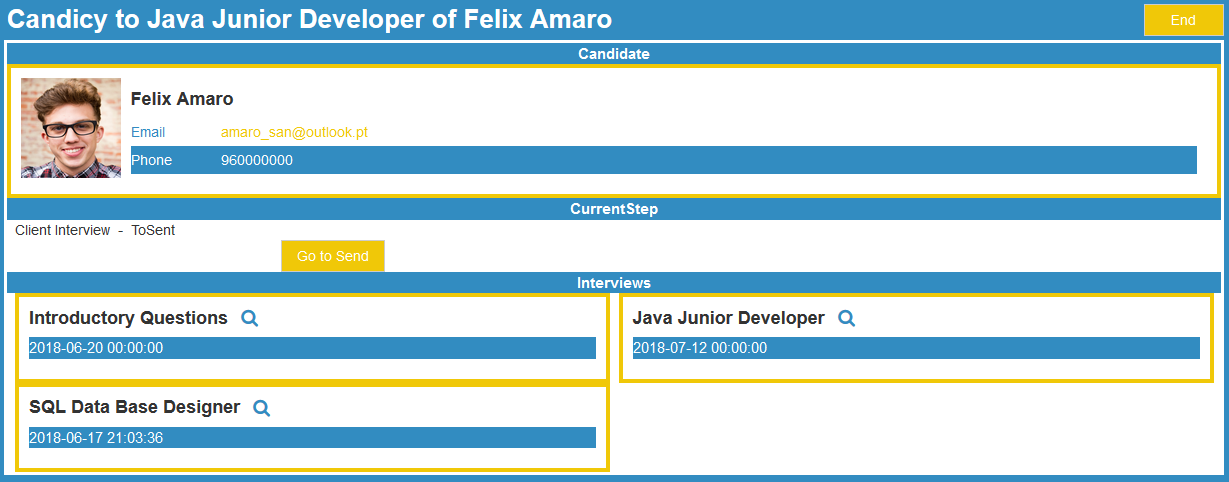


Figura 102 - Candidatura depois

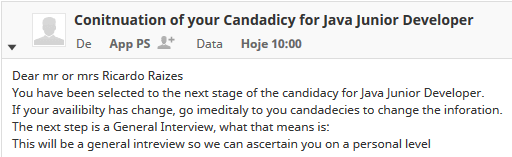


Figura 103 - Continuação da candidatura, email



Figura 104 - Continuação da candidatura, notificação

### 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.

* + 1. **Ação**

Pela candidatura um colaborador consegue abrir o *Pop-up* da Figura 105, 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 106. A partir deste *Pop-up* é também possível criar um novo bloco que representa o início e fim da entrevista a marcar.

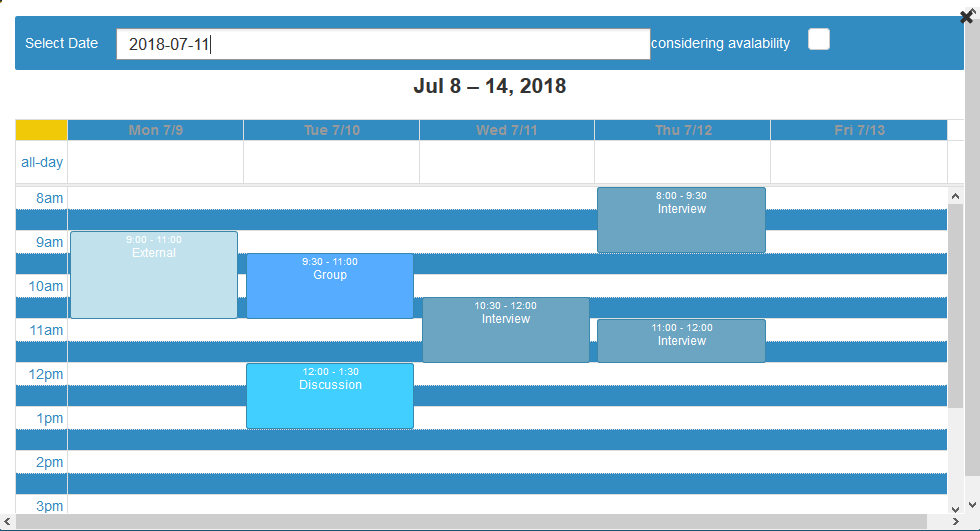


Figura 105 – Eventos

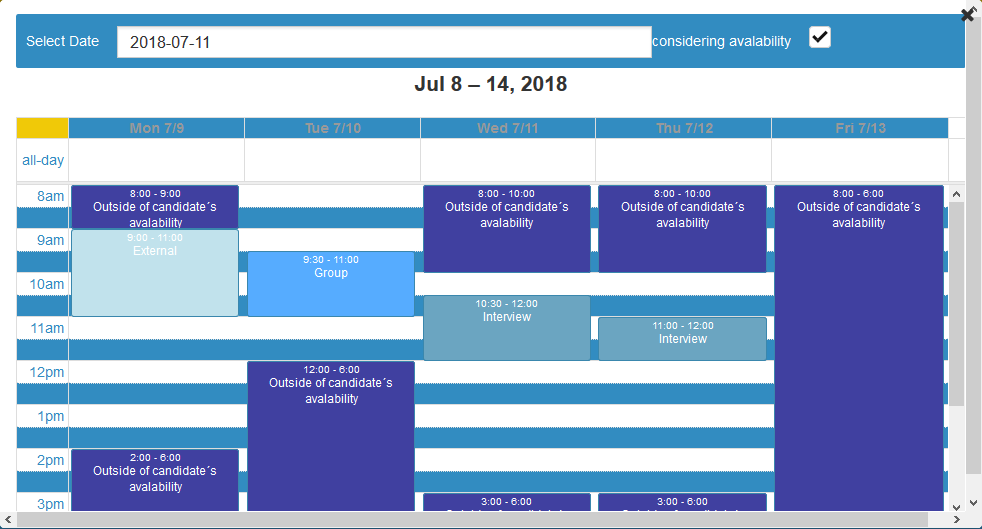


Figura 106 - Eventos com disponibilidade

* + 1. **Resultado Esperado**

Ao escolher os tempos para o evento, o *Pop-up* da Figura 107 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 108 aparece, o que permite escolher a localidade onde a entrevista deverá ocorrer.

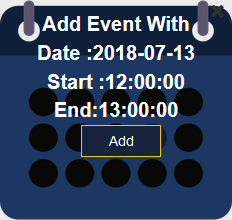


Figura 107 - *Pop-up* do novo evento

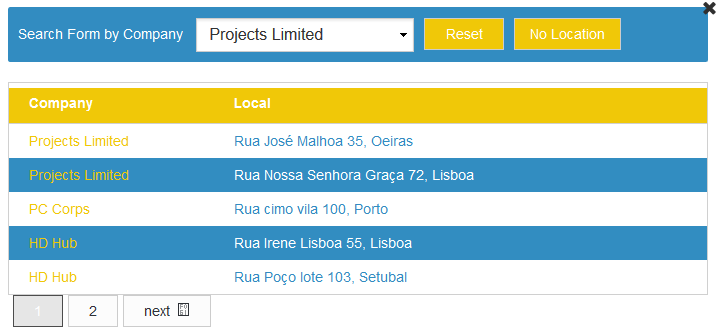


Figura 108 - *Pop-up* para escolher localidades

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 110, Figura 111 e Figura 112,

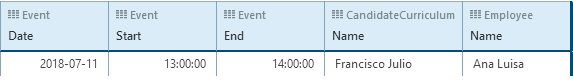


Figura 109 - Nova instância de *Event* e *Interview*

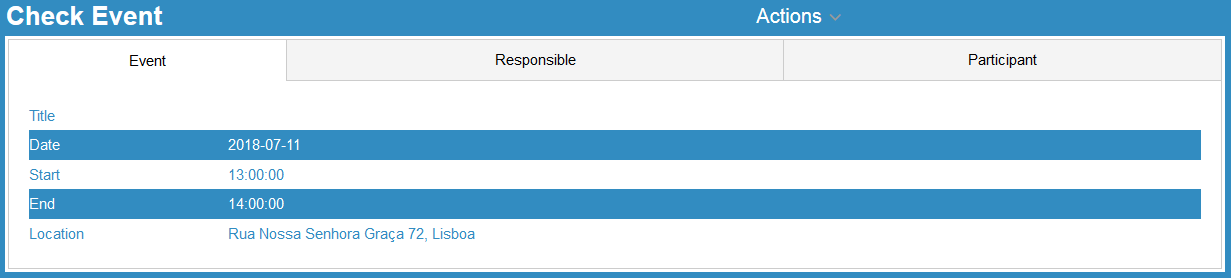


Figura 110 - Nova entrevista, informação geral

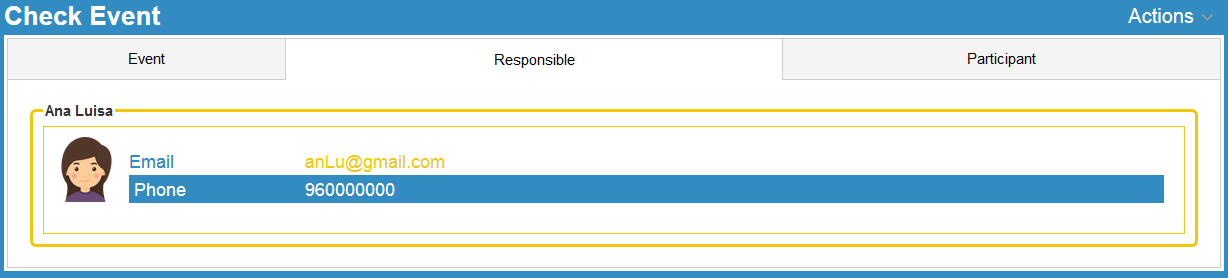


Figura 111 - Nova entrevista, responsável

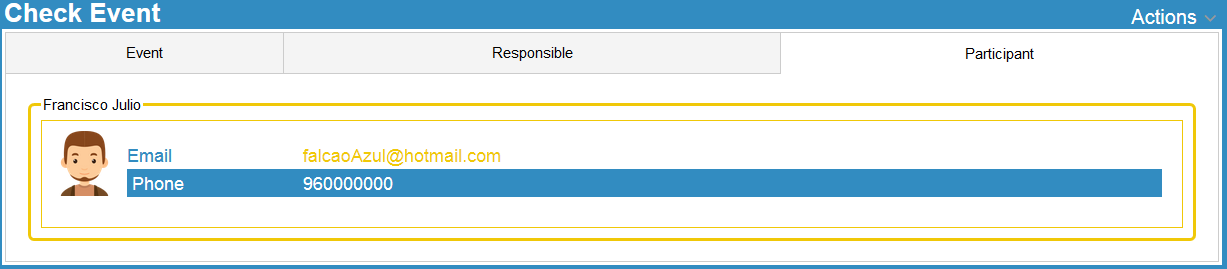


Figura 112 - Nova entrevista, participante

O candidato participante é informado da nova entrevista tanto por um email, Figura 113 ou por uma notificação Figura 114.

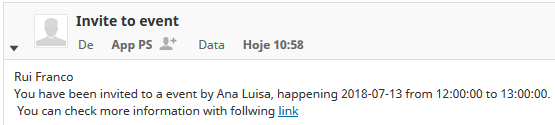


Figura 113 - Convite da entrevista, email

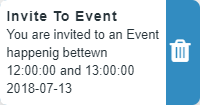


Figura 114 - Convite da entrevista, notificação

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 115 aparecerem.

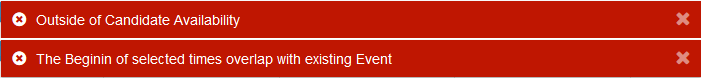


Figura 115 - Erro na marcação

# Conclusões

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

Quando a aplicação foi primeiro considerada era uma simples ajuda a candidatos gerar o seu profile e currículo, mas durante o desenvolvimento a aplicação cresceu mais e mais permitindo assim ser mais valia não só para candidatos como também para colaboradores. Desde a proposta novos objetivos não opcionais foram adicionados tanto aos candidatos como aos colaboradores e todos estes como também os originais foram compridos, com pequenas melhorias realizadas, tanto na área de interface como também na funcionalidade.

Dos requisitos funcionalidades opcionais só uma foi completada, a inclusão do Google Maps, que esta incluído tanto na aplicação *web*, na área de contactos nas informações gerais e na aplicação *mobile*.

Das três não executadas a única que tivemos dificuldades a executar e dai não a ter cumprido foi incluir um acesso a Googel Calendar. Tal acesso pode, em parte, ser realizado por várias extensões da OutSystems, incluído a extensão utilizada FullCalendar2. Mas todas as extensões tinham um problema em comum, permitiam a leitura na Google Calendar mas não a escrita sobre a mesma. Apesar dessa capacidade ser interessante tendo uma limitação tao notável simplesmente vez a inclusão desta capacidade algo que desnecessária.

O requisito de passar toda a base de dados para uma base externa foi algo que foi basta averiguado mas entre o facto que introduzia uma dependência a uma base externa, tornava o acesso mais lento e tornava qualquer alteração a estrutura base de dados mais complexa demonstrou que não tinha uma mais valia quando comparada a alternativa mais simples. A únicas vantagens que encontramos seria a dimensão da base de dados e o facto 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 o requisito de estruturar a aplicação de forma a englobar várias empresas, durante o desenvolvimento apercebemos que seria simplesmente poderia introduzir problemas que tornava o requisito simplesmente contraprodutivo. Para começar a base de dados seria muito extensa tendo em conta todas as informações que poderiam ser incluídas. Mas o problema principal seria a partilha de informação, com a utilização da aplicação uma empresa incluiria informação algo que sensível, desde fatores como projetos, eventos ou até candidaturas, numa base de dados e aplicação acessível a outros grupos. Não seria particularmente difícil de incluir um filtro que garanta que qualquer acesso a esta informação só fosse possível por elementos duma empresa, mas não conseguimos ver muitos grupos a querem pertencer a uma aplicação em que a sua informação mais sensível estive-se acessível a outros grupos. Uma possível solução para este problema seria não incluir essa informação, mas se toda a informação que um poderia considerar sensível o que teríamos no final seria uma pouco mais que um LinkeIn [3] com a capacidade de um utilizador formar o seu currículo.

Para futuras temos algumas considerações, para começar com incluir o Google Calendar e melhorar a interface gráfica, tentar tornar a informação mais fácil de seguir o possível.

Outra funcionalidade interessante seria incluir o lançamento de notificações quando uma oferta possivelmente interessante a um candidato fosse criada. Tal poderia ser possivelmente executado com a introdução de preferências para candidatos em função de ferramentas, assim quando uma nova oferta incluía tal tecnologia o candidato era informado da oferta.

Outra melhoria considerada seria formar os formulários de forma a que o salto de passos de um processo de entrevista não estivesse completamente dependente de utilizar um particular formulário. Possivelmente adicionar um nível de exigência e as ferramentas principais no formulário assim se um candidato já tenha executado uma entrevista com um formulário na mesma exigência com a mesma tecnologia a sua candatura podia saltar o passo.

A possibilidade de incluir uma capacidade de notificações para a aplicação *web* como na aplicação *web*, utilizando o componente Web InApp Notifications fornecido pela OutSystems. Assim notificações seriam enviadas tanto para aplicação *mobile* como também para a *web*.

Por último uma melhoria a considerar seria a capacidade de criar uma via de comunicação diretamente na aplicação em si permitindo uma mais fácil marcação de entrevistas, até possivelmente realizar as entrevistas pela via. Para tal seria utilizada a extensão In-App Chat Web para a comunicação *web* e a extensão In-App Chat Mobile para a aplicação *mobile*.

# Referências

Referências

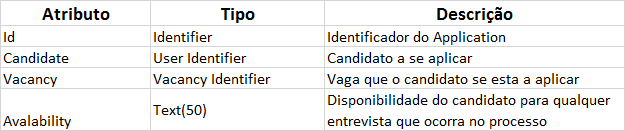
|  |  |
| --- | --- |
| [1] | Mind Source, “Portal de Emprego,” Mind Source, [Online]. Available: https://emprego.mindsource.pt/. [Acedido em Abril 2018]. |
| [2] | Randstrand, “My Profile,” Randstrand, [Online]. Available: https://myprofile.randstad.pt/. [Acedido em Abril 2018]. |
| [3] | LinkedIn Corporation, “LinkeId,” 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 Diagramas da Aplicação

# A.2 Modelos de dados

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

Modelos De Dados 1 - *Users*



Modelos De Dados 3 - *Events*

Modelos De Dados 2 - *Vacancies*

**Tabela 1 - *Application***

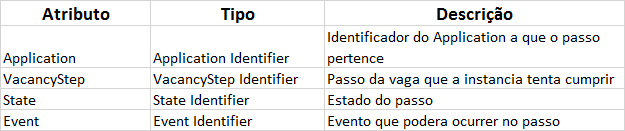


Tabela 2 - *ApplicationCurrentStep*

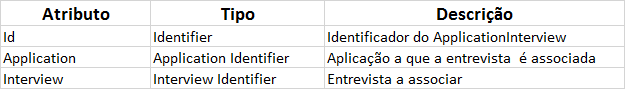


Tabela 3 – *ApplicationInterview*

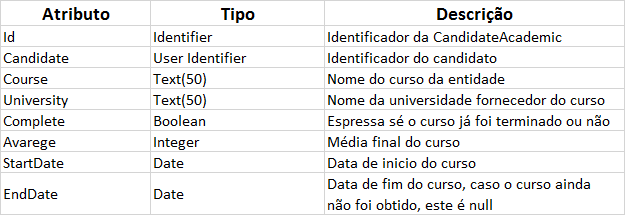


Tabela 4 – *CandidateAcademics*

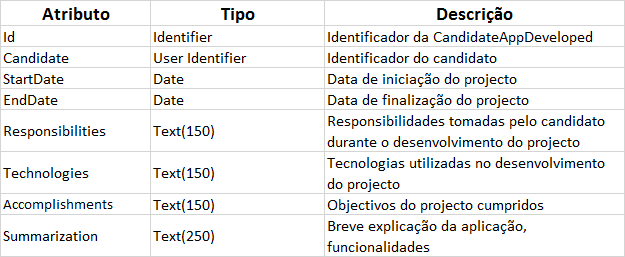
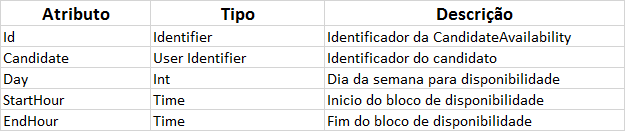
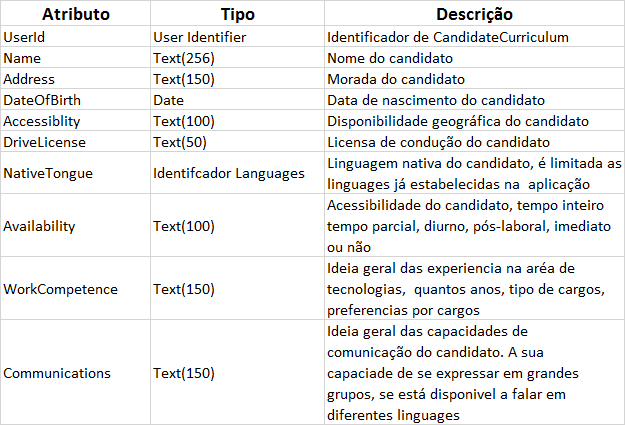


Tabela 5 – *CandidateAppDeveloped*



**Tabela 6 - *CandidateAvailability***



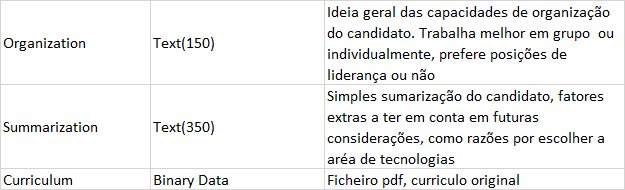


Tabela 7 – *CandidateCurriculum*

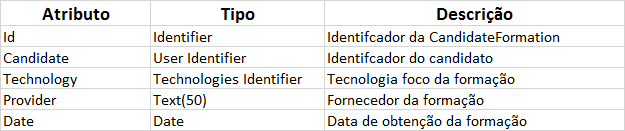


Tabela 8 - *CandidateFormations*



Tabela 9 – *CandidateFrameworks*

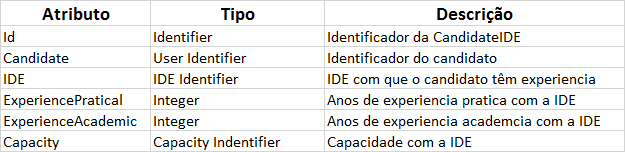


Tabela 10 - *CandidateIDE*

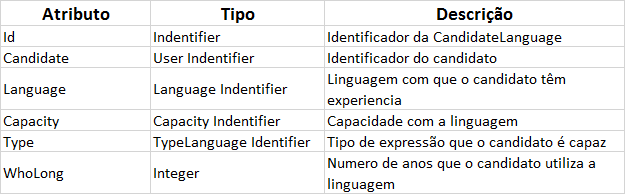
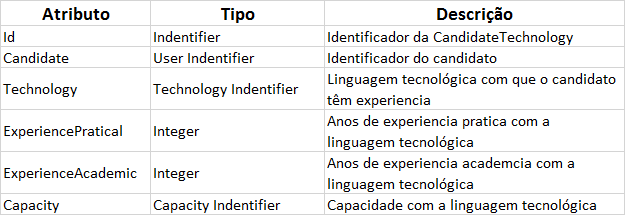
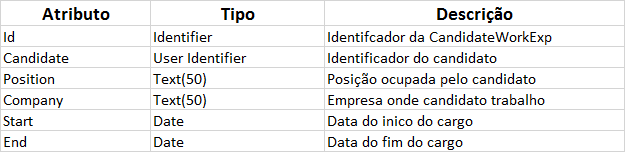


Tabela 11 – *CandidateLanguage*



**Tabela 12 - *CandidateTechnology***



**Tabela 13 - *CandidateWorkExp***

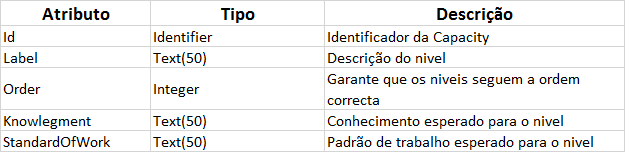
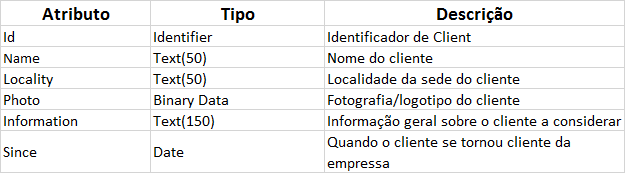


Tabela 14 – *Capacity*



**Tabela 15 – *Client***

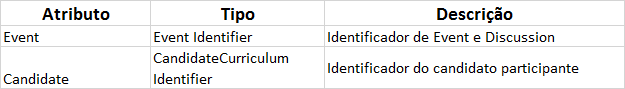


Tabela 16 – *Discussion*

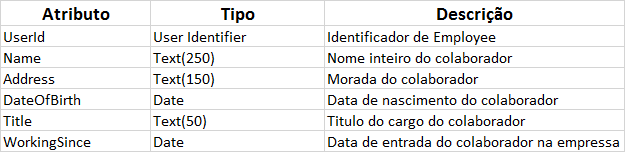
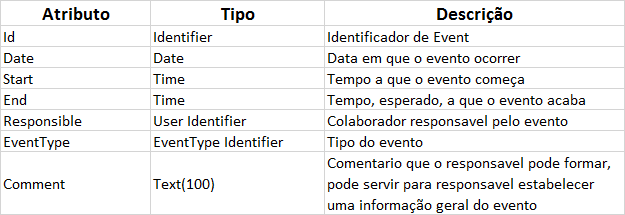
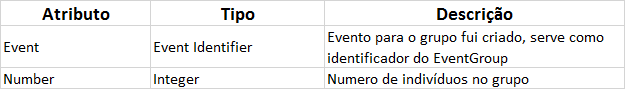


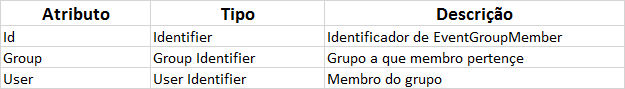
Tabela 17 – *Employee*



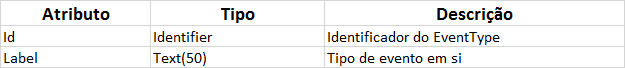
**Tabela 18 - *Event***



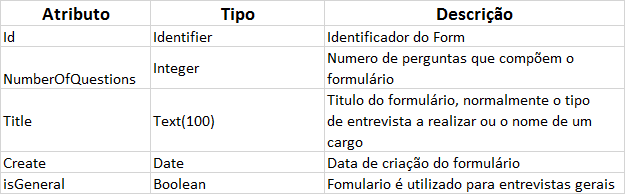
**Tabela 19 - *EventGroup***



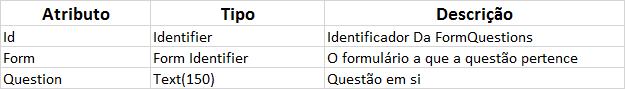
**Tabela 20 - *EventGroupMember***



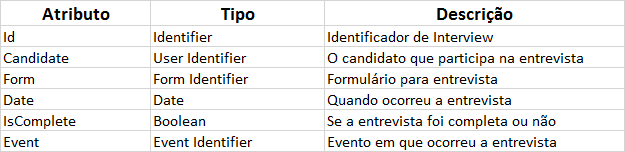
**Tabela 21 - *EventType***



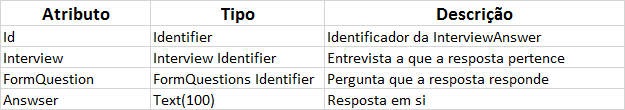
**Tabela 22 - Form**



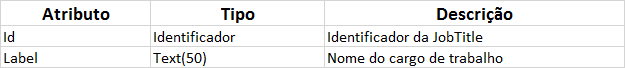
**Tabela 23 - *FormQuestion***



**Tabela 24 - *Interview***



**Tabela 25 - *InterviewAnswer***



**Tabela 26 - *JobTitle***

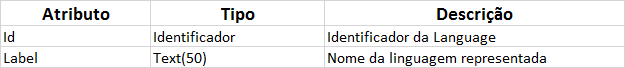
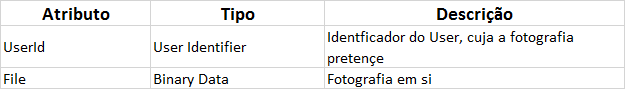


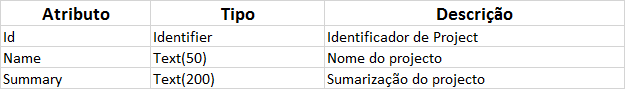
Tabela 27 – *Language*



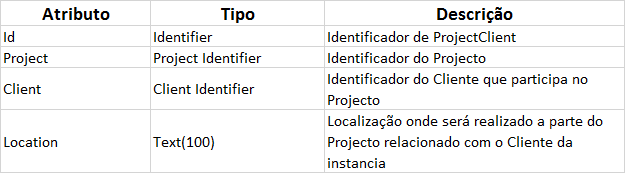
Tabela 28 - *Location*



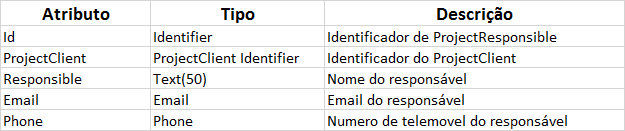
**Tabela 29 - *Photo***



**Tabela 30 –*Project***



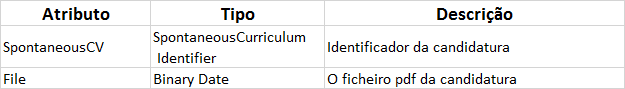
**Tabela 31 - *ProjectClient***



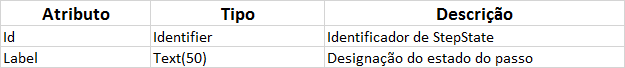
**Tabela 32 –*ProjectResponsible***



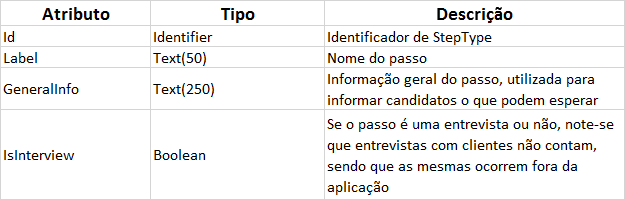
**Tabela 33 - *SpontanousCurriculum***



**Tabela 34 - *SpontanousCurriculumFile***



**Tabela 35 – *StepState***



**Tabela 36 - *StepType***

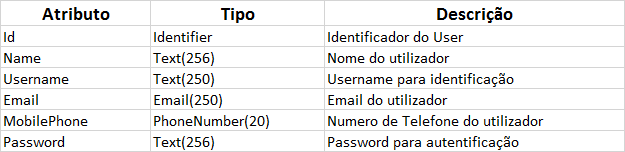


Tabela 37 – *User*

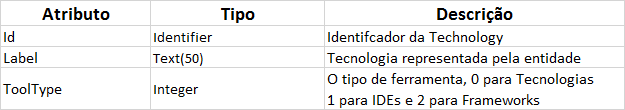


Tabela 38 – *Tools*

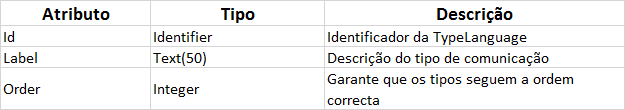
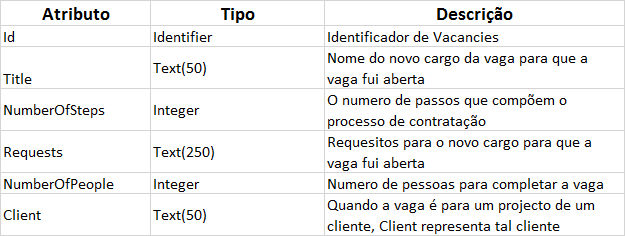
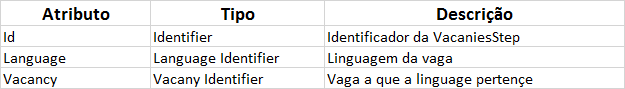


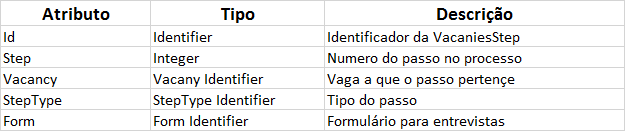
Tabela 39 – *TypeLanguage*



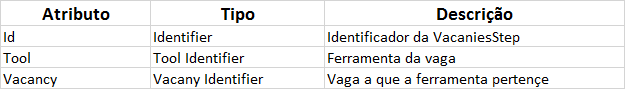
**Tabela 40 – *Vacancy***



**Tabela 41 - *VacancyLanguages***



**Tabela 42 - *VacancyStep***



**Tabela 43 –*VacancyTools***

1. Ficheiro em que um candidato consegue demonstrar as suas competências em tecnologias, idiomas e onde, normalmente, pode incluir aplicações que desenvolveu [↑](#footnote-ref-1)
2. Por disponibilidades, referimos a janelas de tempo associados a dias de semana que o candidato considerar fiáveis para participar numa entrevista [↑](#footnote-ref-2)
3. R*ich widget* que permite criar uma página tipo *popup* a partir de uma página já estabelecida na aplicação. [↑](#footnote-ref-3)
4. Se for obtida mais que uma entidade, estas são unidas através de ações *join* [↑](#footnote-ref-4)