

IView

36210, Rui Pedro Gama Franco, 933941557, 36210@alunos.isel.ipl.pt

40278, Diogo Mendes Aires, 962573900, 40278@alunos.isel.ipl.pt

Orientadores

Paula Graça, ISEL, mgraca@deetc.isel.pt

Frederico Ferreira, DoItLean, 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

Abril de 2018

Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

IView

40278 Diogo Mendes Aires

36210 Rui Pedro Gama Franco

Orientadores: Paula Graça, ISEL
Frederico Ferreira, DoItLean

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

Abril de 2018

Resumo

O projeto IView tem um objetivo principal, o desenvolvimento de uma aplicação de entrevistas que seja uma mais valia para uma empresa da área de informática e tecnologias e que permita ao candidato, ter um papel mais envolvido e simplificado.

Para a empresa, a aplicação irá não só garantir uma administração simples de entrevistas, candidatos, projetos, cliente e vagas, como também uma forma de se apresentar aos candidatos, demonstrando o que mais valida nos mesmos e como estes mais facilmente se podem apresentar. Por outro lado, irá oferecer aos candidatos, uma forma simples, interativa e apelativa de administrar as suas informações e realizar aplicações a vagas, que conseguira visualizar como também pesquisar tendo em conta os seus interesses.

Apesar do projeto ser focado no desenvolvimento de uma aplicação *web*, existirá também uma aplicação *mobile* que servirá como um suporte para os candidatos.

Durante o processo de desenvolvimento desta aplicação também iremos colocar os nossos conhecimentos da arquitetura de desenvolvimento OutSystems em prática, arquitetura cujo o conhecimento é cada vez mais requisitado no mercado informático e tecnológico Português e estrangeiro, o que torna qualquer experiencia sobre a mesma indispensável para o futuro.

Palavras-chave: OutSystems; Widgets; Candidatos; Entrevistas; Currículos; App;

Abstract

Abstract text (1 page).

Keywords: sorted keyword list, delimited by;.

Índice

RESUMO	IV
ABSTRACT	VI
LISTA DE FIGURAS	X
LISTA DE TABELAS	XII
1. INTRODUÇÃO	1
1.1 ENQUADRAMENTO	1
1.2 OBJETIVOS	1
1.3 ORGANIZAÇÃO DO DOCUMENTO	2
2. FORMULAÇÃO DO PROBLEMA	3
2.1 ESTADO DE ARTE	3
2.2 DESCRIÇÃO DO PROJETO	6
2.2.1 APLICAÇÃO WEB.....	7
2.2.2 APLICAÇÃO MOBILE	10
2.2.3 REQUISITOS NÃO FUNCIONAIS	11
2.3 DESCRIÇÃO DA PLATAFORMA, MODELOS DE DESENVOLVIMENTOS.....	11
2.3.1 MODELOS DE DESENVOLVIMENTO	13
3 SOLUÇÃO PROPOSTA	16
3.1 ARQUITETURA DO PROJETO.....	16
3.2 MODELO ENTIDADE-ASSOCIAÇÃO DA BASE DE DADOS	17
3.2.1 <i>Utilizadores</i>	17
3.2.2 <i>Vagas</i>	19
3.2.3 <i>Eventos</i>	20
3.3 WIREFRAMES DO PROJETO	21
4 AVALIAÇÃO EXPERIMENTAL	40
5 CONCLUSÕES.....	41
REFERÊNCIAS	42
A.1 DIAGRAMAS DA APLICAÇÃO	43
A.2 MODELOS DE DADOS	45

Lista de Figuras

Figura 1 – Visão geral do IView.....	6
Figura 2 - Casos de utilização, Utilizadores Não Registados.	7
Figura 3 - Casos de utilização, Colaborador 1.....	8
Figura 4 - Casos de utilização, Colaborador 2.....	9
Figura 5 - Casos de utilização, Candidato.	10
Figura 6 - Casos de utilização, Candidato Mobile.....	11
Figura 7 - Arquitetura OutSystems [5].A escolha desta plataforma foi devido a duas razões fundamentais:	12
Figura 8 - As Layers de Desenvolvimento Outsystems.....	13
Figura 9 - Estrutura 4 Layer Canvas.....	16
Figura 10 - Modelo EA, Candidatos.....	17
Figura 11 - Modelo EA, Vagas.....	19
Figura 12 - Modelo EA, Eventos.....	20
Figura 13 – Frames, Home Page.....	21
Figura 14 - Frames, Menu de Colaboradores.	22
Figura 15 - Frames, Menu de Candidatos.....	22
Figura 16 - Frame, MoreInfo Values Tab.....	23
Figura 17 - Frame, MoreInfo Contacts Tab.....	23
Figura 18 - Frame, MoreInfo Patnerships Tab.	24
Figura 19 - Frame, MoreInfo Personnel Tab.	24
Figura 20 - Frame, MoreInfo Client Tab.....	24
Figura 21 - My Curriculum, Show.....	25
Figura 22 - Frame, MyCurriculum, Edit.....	26
Figura 23 - Frames, MyProfile.	27
Figura 24 - Frames, MyProfile PopUp.	28
Figura 25 - Frames, Forms.....	29
Figura 26 - Frames, Applications.	30
Figura 27 - Frames, Vacancies.	30
Figura 28 - Frames, Events.....	31
Figura 29 - Frames, CheckCV.....	32
Figura 30 - Frames, Candidate.....	32
Figura 31 - Frames, Event.	33
Figura 32 - Frames, Application.....	34
Figura 33 - Frames, Vacancy General Information Tab.....	35
Figura 34 - Frames, Vacancy Steps Tab.....	36

Figura 35 - Frames, Vacancy Tools and Languages Tab.....	36
Figura 36 - Frames, FormEditAdd.....	37
Figura 37 - Frames, VacancyAdd Form Chosse PopUp.....	37
Figura 38 - Frames, Vacacncy Add Main Tab.....	38
Figura 39 - Frames, Vacacncy Add Sterps Tab.	38
Figura 40 - Frames, Vacacncy Add Tools Tab.....	38
Figura 41 - Frames, Vacacncy Add Languages Tab.....	39

Lista de Tabelas

1. Introdução

Neste capítulo iremos introduzir algumas ideias fundamentais sobre o projeto IView, inclui algumas inspirações e enquadramento no mercado de informática atual, os objectos gerais, a serem aprofundados em capítulos seguintes da aplicação e a organização do relatório.

1.1 Enquadramento

Muitas empresas de tecnologia portuguesas, têm um processo de entrevista bastante arcaico 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 competências¹ por correio eletrónico. Neste processo, qualquer marcação de entrevista é realizada por telemóvel, tendo como consequência a dependência do candidato num colaborador da empresa, que caso se esquece, ou não o informe de forma correta, a entrevista poder-se-á perder por falha de comunicação.

Uma das raras exceções que não se limita ao processo acima descrito, é a consultoria Mind Source, que fornece o “Portal de Emprego” [1]. 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” [2] desenvolvido pela empresa de recursos humanos Randstad, disponibiliza um formato simples e interativo de desenvolver o currículo e dossiê de capacidades, mas nada mais.

Atendendo às limitações consideradas nasceu o projeto IView, que tem como objetivo o não só expandir as funcionalidades das duas aplicações referidas, como também incluir outras funcionalidades úteis para a empresa que o utilize.

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

1.2 Objetivos

O projeto IView foca-se no desenvolvimento de duas aplicações, *web* e *mobile*, que funcionam em conjunto para garantir a comunicação entre candidato e entrevistador de uma forma simples e acessível, desde o momento em que uma nova vaga para um cargo na empresa é criada, até a 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 guardado, entre outros dados:

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

- Currículos, dossiês de competências e disponibilidades² de candidatos;
- Informações gerais dos colaboradores que participam na aplicação;
- Clientes e projetos com os mesmos;
- Vagas a novos cargos na empresa - Cada vaga inclui um processo de entrevista;
- Aplicações 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 para interagir com a empresa e aceder ao seu processo de candidatura.

1.3 Organização do documento

O restante relatório tem a seguinte composição de capítulos:

- Formulação do Problema – alguma informação sobre da aplicação IView, incluído algumas aplicações semelhantes, a capacidades das aplicações a desenvolver e a arquitetura onde o projeto será desenvolvido;
-

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

- Solução Proposta – neste capítulo será incluí a estrutura da aplicação, numa forma geral a base de dados da mesma e por último algumas ideias gerais da interface gráfica.

2. Formulação do Problema

Neste capítulo, na secção 2.1, é verificada a utilidade e 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 do IView. Por último, na secção 2.3, é descrita a plataforma de desenvolvimento OutSystems, com a qual o projeto foi implementado.

2.1 Estado de arte

Existe um considerável número de empresas, pelo menos as da área das tecnologias, que utilizam uma aplicação semelhante ao IView, mas com algumas limitações que pretendemos colmatar no IView. A mais marcante é a ausência de interação entre o candidato e o colaborador da empresa. Estas aplicações 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. Quando o candidato necessita de realizar qualquer alteração ao seu currículo e/ou dossiê de capacidades, tem que partilhar tais alterações com um colaborador por *email*, o que não garante que os ficheiros atualizados sejam utilizados em próximas partilhas com clientes.

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

Muitas vezes as bases do dossiê (*template* seguido na criação de um dossiê), não expressam que informações podem ser importantes para incluir no documento. Normalmente é requerido que o candidato expresse as suas capacidades em níveis, mas não é estabelecido o que cada nível implica. Mesmo que esta falta de informação não pareça importante, deve ser tomado em conta que o candidato possa ser alguém que esteja a entrar pela primeira vez no mercado de trabalho, e ainda não saiba como tratar essa situação.

As aplicações referidas também não incluem qualquer tipo de funcionalidade de marcação de entrevistas, embora algumas utilizem o Google Calendar. Aquelas que não utilizam nenhum mecanismo semelhante, arriscam que na ocorrência de qualquer falha de comunicação entre as duas partes interessadas, possa causar que a entrevista não se realize. Aquelas que utilizam o Google Calendar, conseguem ultrapassar esta limitação, mas o candidato fica completamente dependente de receber um convite para a entrevista, o que pode causar problemas. Um exemplo real que se passou com um dos alunos a realizar este projeto, demonstra esse problema. Foi marcada uma entrevista, mas o convite nunca chegou ao próprio, que só depois de conseguir entrar em contacto com o colaborador da empresa que geria o processo, foi encontrado o erro, ou seja, o convite tinha sido enviado para o *email* de outro indivíduo.

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

- Gerir ficheiros, currículos e dossiês, quebrando assim a necessidade de comunicar com um colaborador cada vez que se deseja realizar uma atualização de um dos ficheiros;
- Verificar as vagas disponíveis e aplicar-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 aplicações, a informação fornecida sobre a mesma é bastante limitada, sendo que só indica que aplicações estão a ocorrer, mas mais nada.

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 as 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 o perfil

do candidato, um extenso raio de informações que torna a sua formação um pouco mais complexa. Pensando na adequação do perfil para a área de informática, a aplicação da Randstad, não inclui no dossiê de capacidades, a lista de aplicações desenvolvidas pelo candidato, que reflete de alguma forma, a sua experiência e tipo de responsabilidades que tem vindo a assumir na sua experiência profissional. A aplicação não inclui também qualquer tipo de informação sobre vagas e candidaturas as mesmas, o que torna o início do processo de entrevista só possível pela parte da empresa, ou através do recurso do candidato a outras fontes, para dessa forma descobrir eventuais vagas.

Apesar das limitações acima enumeradas, a aplicação da Randstad proporcionou-nos parte da inspiração gráfica para o projeto, sendo que, apesar de não termos a certeza, acreditamos que esta aplicação foi desenvolvida utilizando a plataforma OutSystems.

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 primeira a que nos referimos, trata-se da rede social LinkedIn [3], que permite gerir um perfil análogamente à aplicação da Randstad, mas de uma forma menos interativa. Esta aplicação, também não inclui qualquer tipo de informação sobre vagas, aplicações 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, não inclui qualquer possibilidade de gestão de perfis, não permitindo também o estabelecimento de registos de utilizador, ou qualquer processo de entrevista. As informações são registadas pela empresa que proporciona a vaga, não sendo acessíveis pela ITJobs.

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 dos candidatos conseguirem 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.

2.2 Descrição do Projeto

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

Na Figura 1 está representado um caso de utilização que descreve de forma geral as funcionalidades das aplicações para os vários perfis de utilizador.

Nesta figura é possível verificar a existência de quatro perfis de utilizador:

- Utilizador Não Registrado – o nível mais baixo que existe, com funcionalidades limitadas;
- Candidato – representa o candidato às vagas 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;
- 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

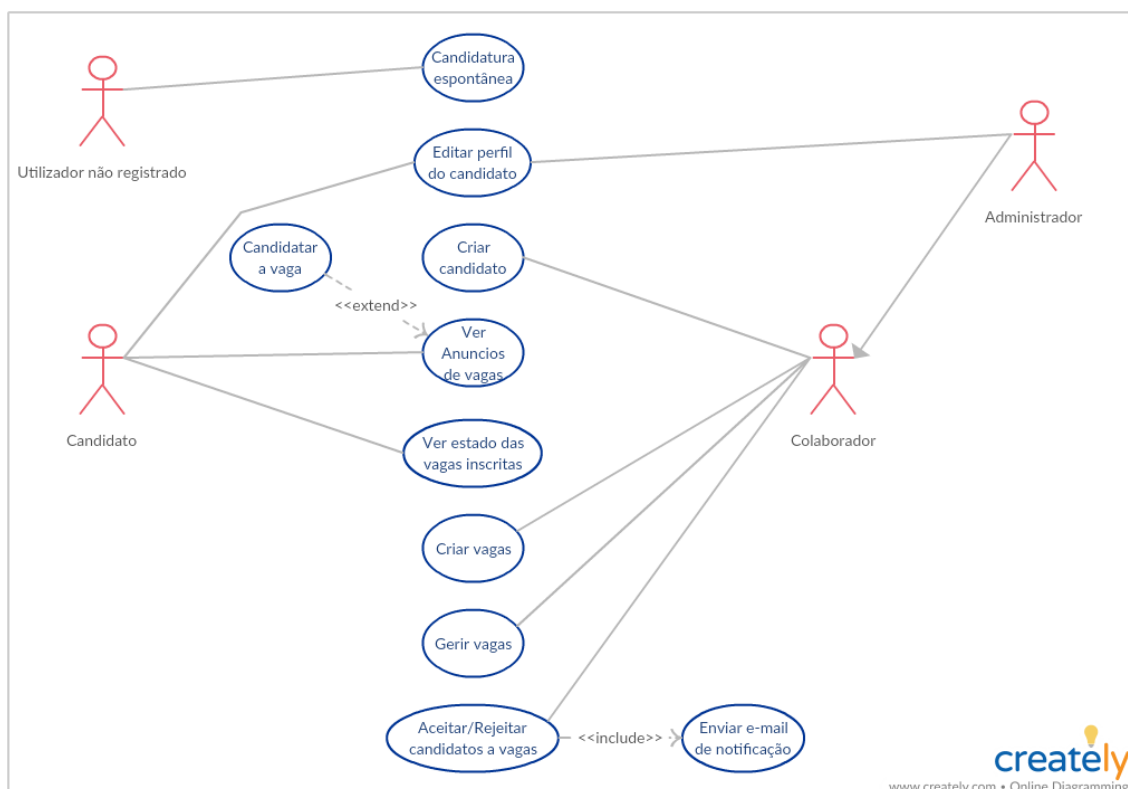


Figura 1 – Visão geral do IView.

Nas restantes 3 subsecções, descrevemos as funcionalidades de cada aplicação (*web* e *móvel*), apresentando casos de utilização, seguidos de uma ideia geral dos mesmos. Descrevemos também, 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 que apresenta o respetivo modelo entidade-associação.

2.2.1 Aplicação web

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

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

- Contactos;
- Missão e valores da empresa;
- Anúncios para novas colocações;
- Informação geral sobre os colaboradores;
- Possibilitar a realização de candidaturas espontâneas.

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

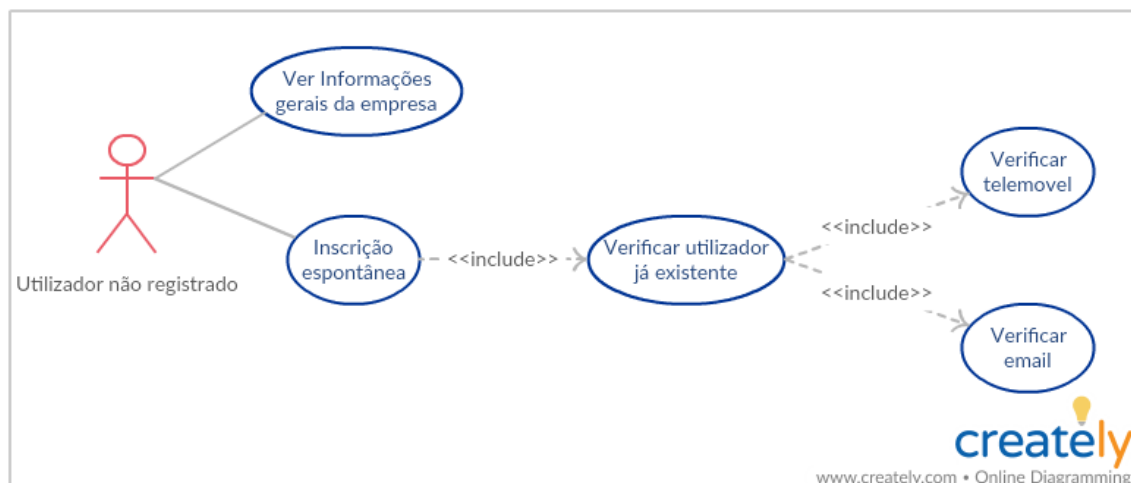


Figura 2 - Casos de utilização, Utilizadores Não Registrados.

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

- Estabelecer novos candidatos em função de candidaturas espontâneas;
- Formar os processos de entrevista para novos cargos na empresa como também os formulários para as entrevistas;
- Gerir os vários passos que compõem um processo de candidatura e os respetivos estados de acordo com a sua progressão;
- Verificar candidaturas a novos cargos, permitindo promover um candidato para o próximo passo do processo, ou terminar, sendo o candidato informado automaticamente, 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, mostrando a segunda figura as funcionalidades mais centradas nas entrevistas e outros eventos, sendo as restantes mostradas pela primeira.

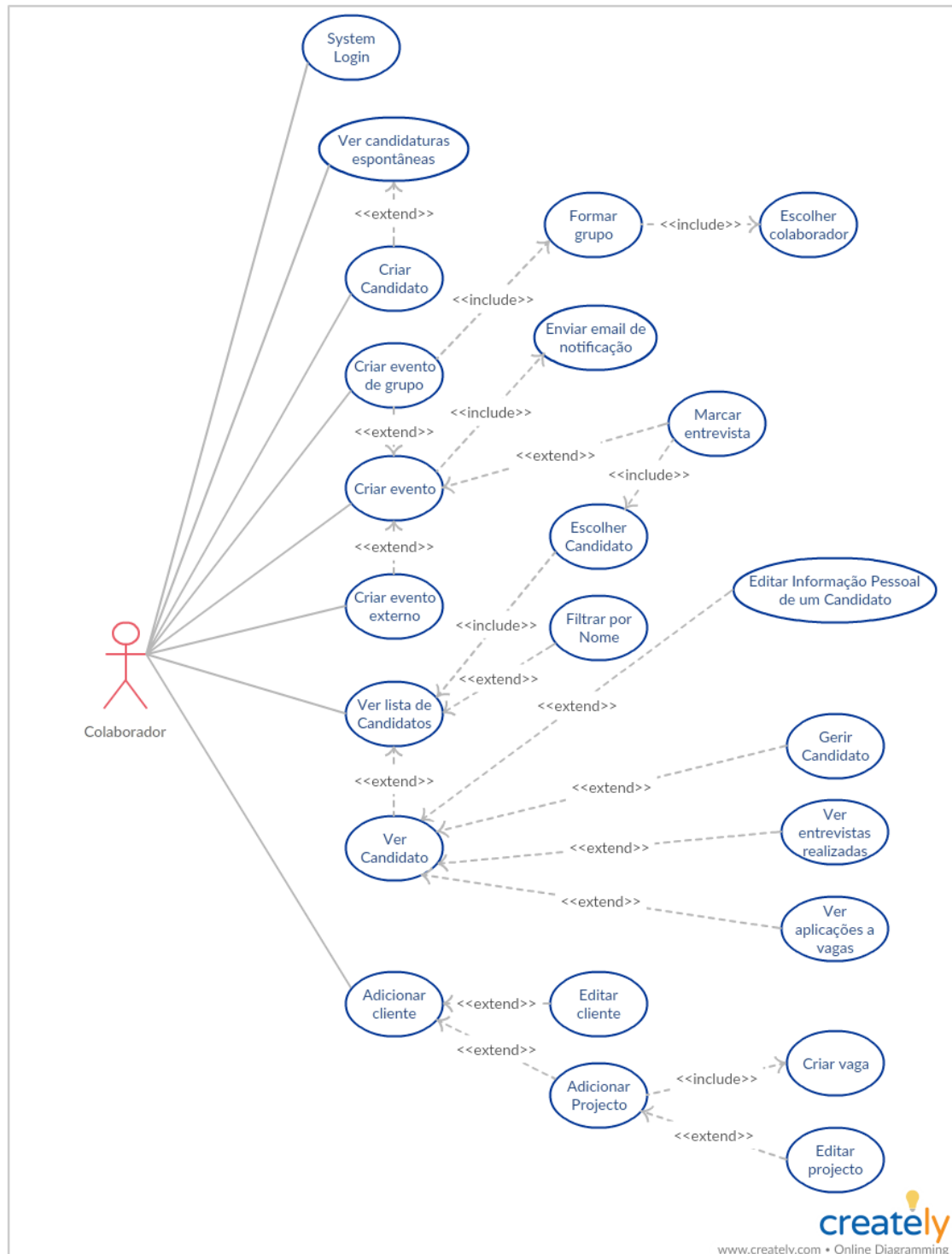


Figura 3 - Casos de utilização, Colaborador 1.

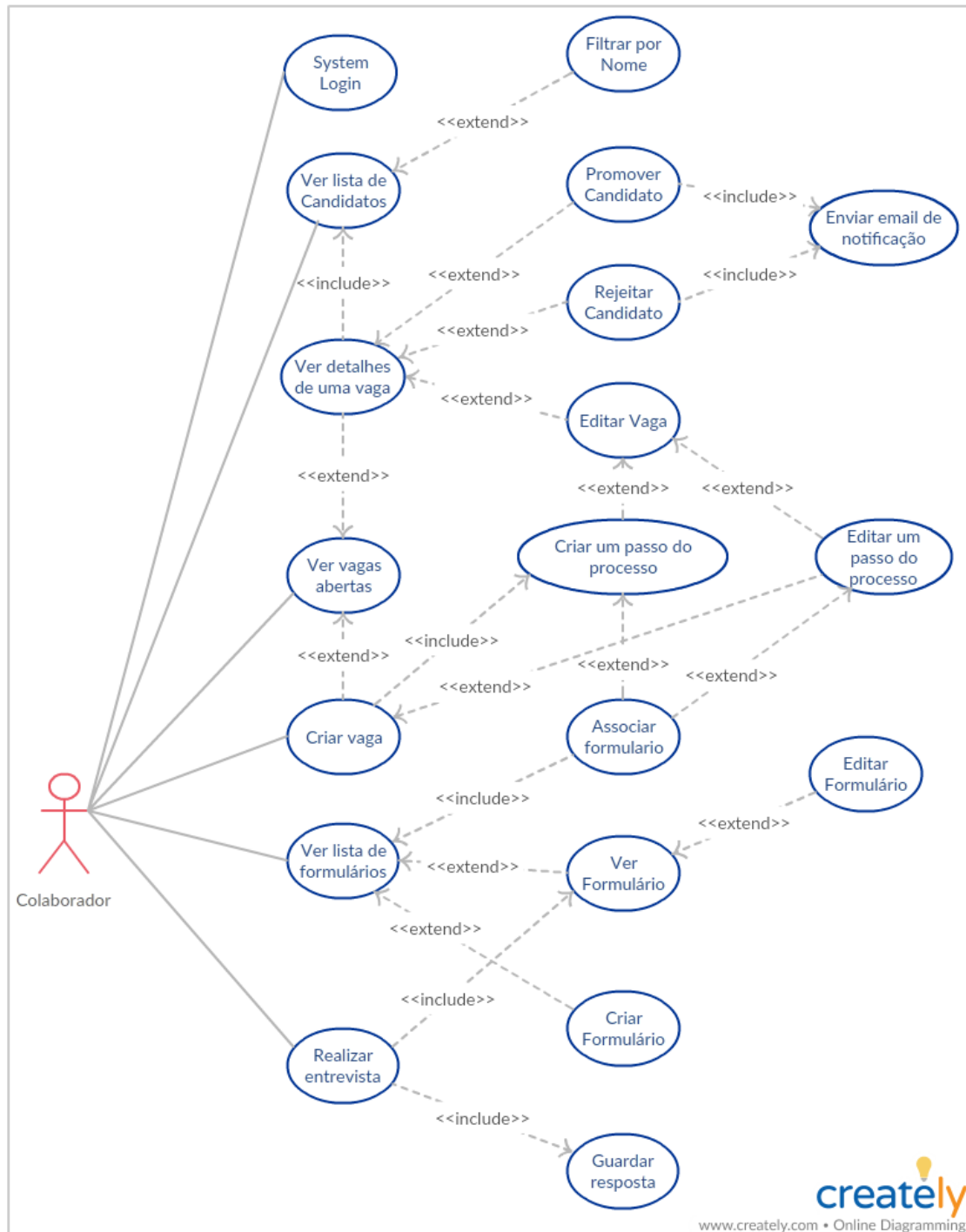


Figura 4 - Casos de utilização, Colaborador 2.

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 - o dossiê poderá incluir informação como: experiência com tecnologias (linguagens), *frameworks*, formação e educação na área, etc.;
- 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.

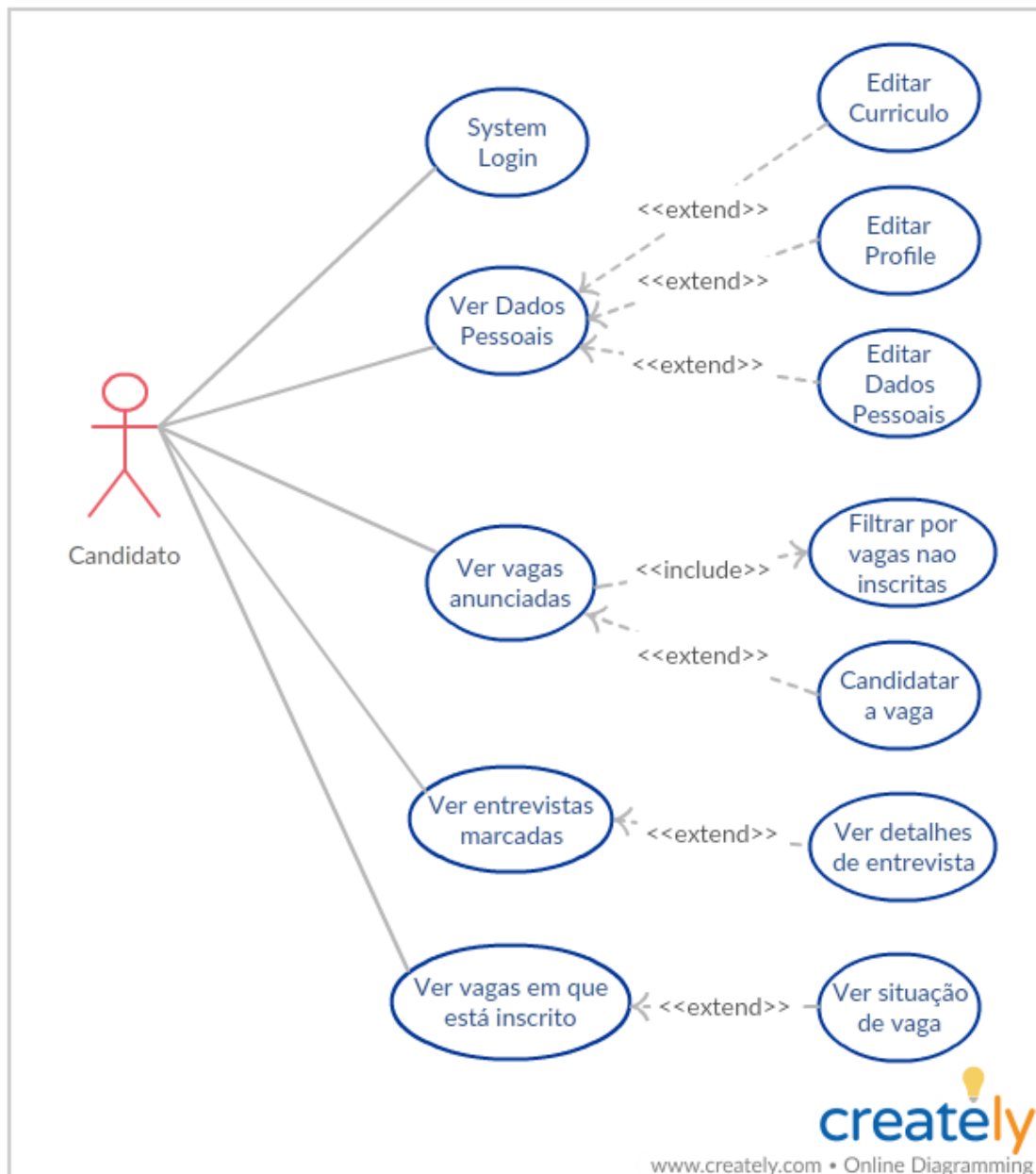


Figura 5 - Casos de utilização, Candidato.

2.2.2 Aplicação *mobile*

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

- Verificar e editar a informação 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 respectivas informações.

Estas funcionalidades, podem ser verificadas Figura 6.

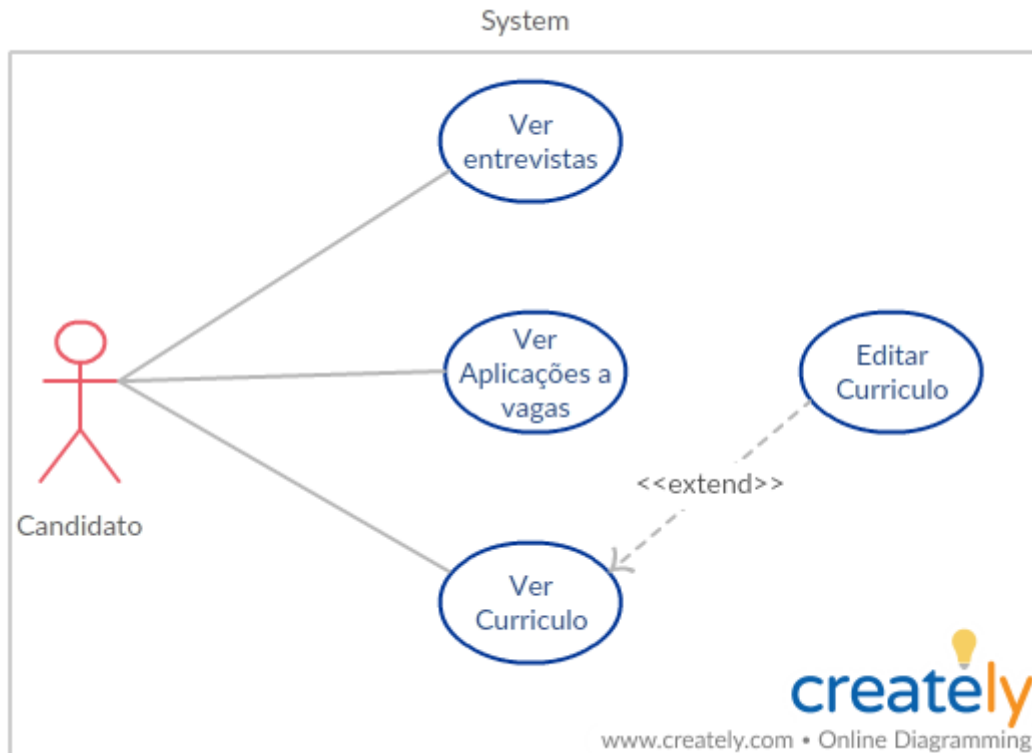


Figura 6 - Casos de utilização, Candidato Mobile.

A necessidade de uma aplicação mais leve deve-se, em parte, ao problema de sincronismo de aplicações moveis desenvolvidas na OutSystems, ao manter só alguns acessos de leitura essenciais e ainda menos acessos de escrita, que em si são restringidos 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 **Erro! A origem da referência não foi encontrada..**

2.2.3 Requisitos Não Funcionais

Para futuras melhorias, são ponderadas as seguintes funcionalidades:

- Permitir que a aplicação funcione com várias empresas;
- Usar a API do Google Calendar para marcação das entrevistas;
- Utilizar o Google Maps para demonstrar a localização da empresa;
- Utilizar uma base de dados SQL externa.

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.

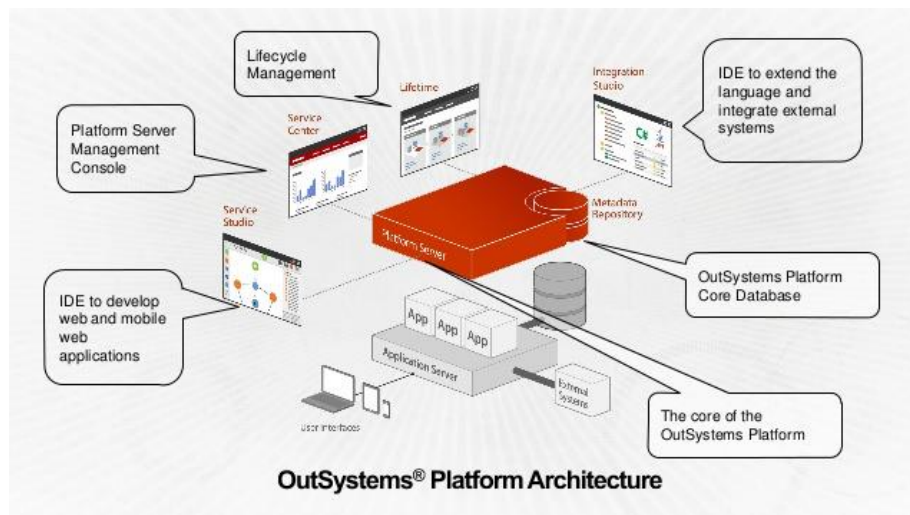


Figura 7 - Arquitetura OutSystems [5]. A escolha desta plataforma foi devido a duas razões fundamentais:

- 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 colocar muito tempo no desenvolvimento da mesma.

A plataforma OutSystems permite um desenvolvimento rápido das 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*.

Para tal desenvolvimento simples e rápido, 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 a aplicação;
- Gera código nativo;
- Gerar *proxies* de integração;
- Agrupar as capacidades de gerir de sessão, autenticação e configurações para a aplicação;
- Adicionar recursos de registo e monitoramento a aplicação.

No Deployment Service os componentes gerados de uma aplicação são aplicados 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 ultimo existe o Application Service que fornece dois serviços importantes;

- O Scheduler Service, que administra a execução de *threads* planeadas;

- O Log Service, responsável por gerir erros, auditorias e o desempenho de uma aplicação.

2.3.1 Modelos de Desenvolvimento

A OutSystems Service Studio, onde é possível desenvolver tanto aplicações *web* como *mobile*, divide o desenvolvimento em três *layers*, cada uma dividida num conjunto de componentes. Estes *layers* são verificados na Figura 8, note-se que no caso desta figura os *layers* são os de desenvolvimento de aplicações *mobile*.

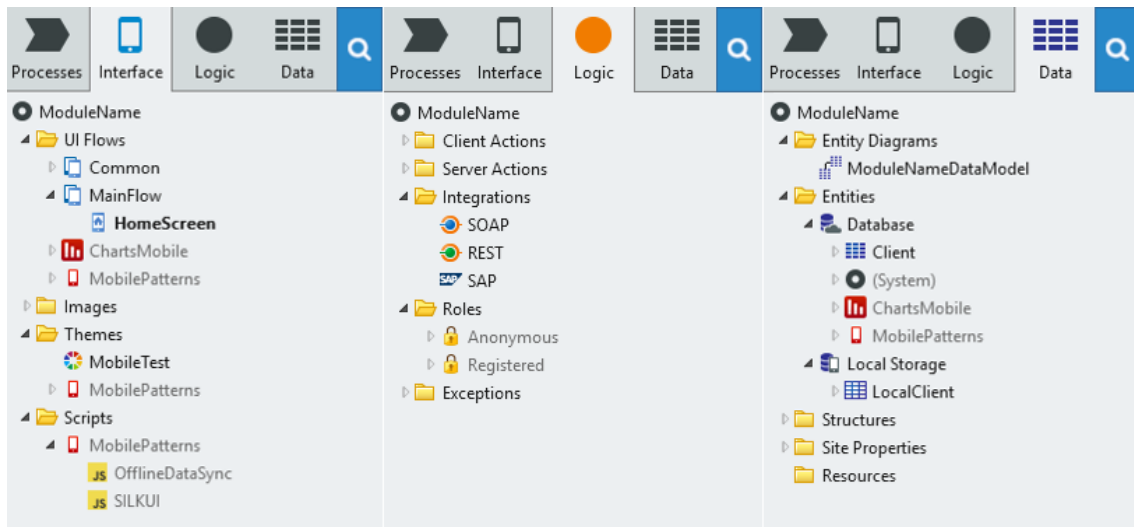


Figura 8 - As Layers de Desenvolvimento Outsystems.

Primeiro temos a Interface Layer, que se centra na interface gráfica da aplicação e que é composta pelos seguintes componentes:

- UI Flows – onde estão incluídos tanto os ecrãs que compõem a interface gráfica da aplicação desenvolvida, como também padrões utilizados;
- Images – uma pasta das imagens, gráficos e *icons* que podem ser utilizadas como imagens estáticas nos ecrãs;
- Themes – temas que estrutura a interface gráfica da aplicação;
- Scripts – recursos JavaScript scripts, único nas aplicações moveis;
- Multilingual Locals – permite gerir *locales* que em si garante a tradução da aplicação em varias línguas, único nas aplicações *Web*.

Depois temos a Logic Layer, responsável pelo desenvolvimento de ações que podem correr durante o processo da aplicação:

- Client Actions – ações que ocorrem no lado do cliente;
- Server Actions – ações que ocorrem no lado do servidor, existe unicamente no desenvolvimento *mobile*;

- Integrations – permite configurar integrações que a aplicação pode utilizar, permite integrações em servidores SOAP, REST e SAP;
- Roles – onde é possível incluir papéis de segurança;
- Exceptions – onde é incluído as excepções desenvolvidas para a aplicação.

Por ultimo temos a Data Layer, que permite estabelecer data importante para a aplicação, esta é composta por:

- Entity Diagrams – diagramas das entidades que servem como uma representação viável das entidades que compõem a base de dados;
- Database – inclui as entidades que guardam dados, tanto online, como offline, no caso de desenvolvimento mobile;
- Structures – é possível gerir e guardar estruturas de apoio neste componente da Data Layer;
- Session Variables – variáveis de secções estabelecidas, existe unicamente no desenvolvimento de aplicações mobile;
- Site Properties – propriedades do site;
- Resources – recursos disponíveis na aplicação.

Web LifeCycle, o ciclo de vida de uma página Web desenvolvida é normalmente composto por três passos, com o primeiro sendo opcional:

- Preparação – onde é obtido e gerido dados demonstrados na página, este passo é opcional sendo que se a informação da página pode ser completamente estática;
- Composição – a estrutura da página e os seus elementos HTML são geridos, também são obtidos recursos como CSS, JS e IMG;
- Interação – a partir deste ponto o utilizador pode interagir com a página, incluindo invocar ações Screen e/ou Server.

Mobile LifeCycle, apesar do ciclo de vida dum ecrã *mobile* poder ser visto como também dividido nos três passos do ciclo de vida anterior, este ciclo é algo mais complexo. Sendo que não é incluída a ação Preparation, em vez disso, ecrãs incluem *fetches* e *event handlers*.

O primeiro permite obter os dados a demonstrar no ecrã, sendo possível obter os dados da base de dados, do armazenamento local ou até de fontes terceiras, estes *fetches* são realizados de forma assíncrona e paralela a outros *fetches* do mesmo ecrã.

Em termos de event handlers cada ecrã tem a possibilidade de incluir ações associadas a quatro eventos, pelos quatro handlers:

- OnInitialize - ação ocorre depois de se verificar a possibilidade de acesso pelo utilizador, mas antes de qualquer *fetch*, permite inicializar o ecrã, definindo valores *default*;

- OnReady – ação ocorre quando ecrã está pronto, mas antes do *render* de o mesmo começar, manipula o DOM do ecrã;
- OnRender – ação para correr quando o ecrã termina o seu *render*, ou cada vez que um elemento do ecrã muda;
- OnDestroy – ação ocorre antes de um ecrã ser destruído.

Sincronismo, desenvolvimento sobre a plataforma *mobile* regularmente envolve a utilização de duas fontes de informação, a base de dados, *online*, e o armazenamento local, *offline*. Mas estas fontes devem sempre, ou pelo menos sempre que possível, sincronizadas para garantir que informação da segunda fonte é a mais atualizada possível.

Por esta razão a OutSystems fornece uma *framework* de sincronismo, alguns dos exemplos de sincronismo possíveis são:

- Read-Only Data – utilizável para quando utilizadores precisam unicamente de ler informação e quando a dimensão da data a sincronizar é baixa;
- Read-Only Data Optimized – semelhante ao Read-Only Data, mas para dimensões de data mais extensiva;
- Read/Write Data Last Write Wins – este padrão é útil quando existe alteração de data, modo *offline*, mas em que não existe conflito de escrita entre vários utilizadores;
- Read/Write Data with Conflict Detection – uma versão mais aprofundada do padrão anterior, permitindo a resolução de conflitos de escrita de múltiplos utilizadores;
- Read/Write Data One-to-Many – continuação dos dois padrões anteriores, este padrão permite que vários utilizadores alterem a mesma informação.

A aplicação *mobile* desenvolvida neste projeto utilizará os padrões Read-Only Data Optimized e Read/Write Data Last Write Wins.

O primeiro para ecrãs onde será verificado unicamente informação que não se pode alterar, exemplo entrevista e aplicações a vagas, que podem incluir uma extensa quantidade de informação. Já o segundo padrão será utilizado em qualquer alteração realizada na aplicação, sendo que as informações a alterar, currículo e possivelmente dossiê de capacidades, são únicos acessíveis a um candidato e por isso não existirá concorrência.

3 Solução Proposta

Neste capítulo é demonstrada a arquitetura do projeto em 3.1, de seguida é descrita a base de dados e os seus modelos relacionais no 3.2 e por fim são mostrados vários exemplos de páginas da aplicação no subcapítulo 3.3.

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, a base de dados da aplicação, os *wireframes* que demonstram alguns exemplos de páginas que compõem a aplicação web e o por ultimo alguma lógica para cumprir certas funcionalidades importantes de notar.

3.1 Arquitetura do projeto

Como qualquer aplicação desenvolvida na arquitetura OutSystems, IView é estruturada seguindo o padrão de 4 Layer Canvas, esta promove a abstração correta de serviços reutilizáveis, o isolamento correto de módulos funcionais distintos e a partilha dos mesmos por varias aplicações desenvolvidas em conjunto.

A Figura 9 demonstra uma breve representação e descrição desta estrutura, descrevendo cada uma das camadas.

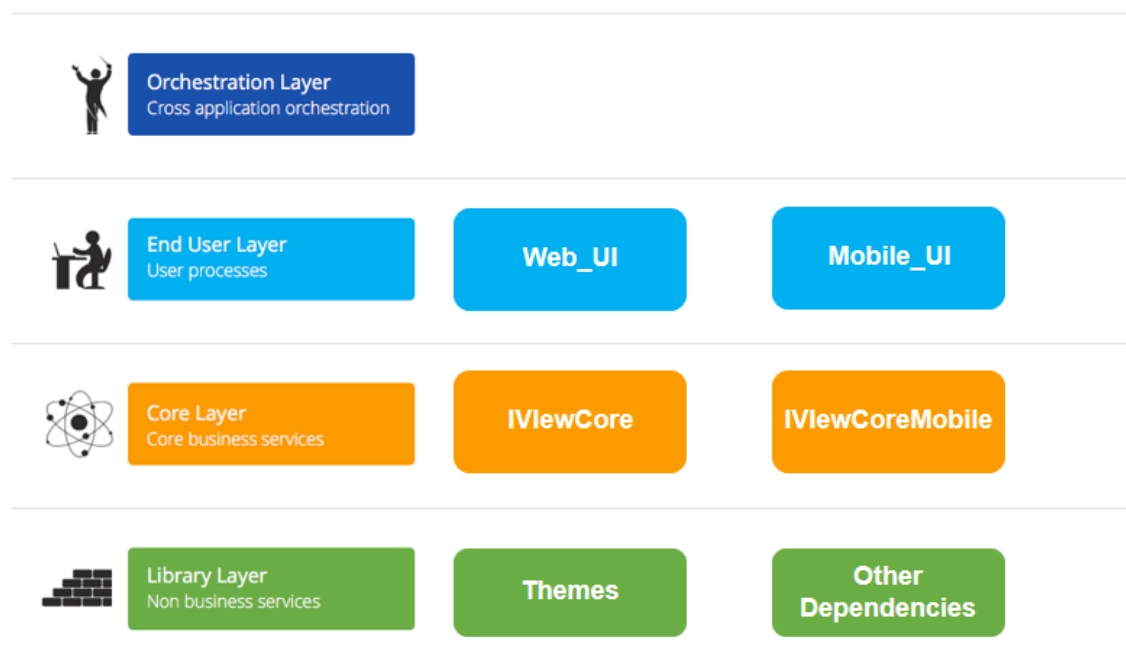


Figura 9 - Estrutura 4 Layer Canvas.

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

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

- Core Layer – esta *layer* será composta por dois módulos:
 - IViewCore – onde é desenvolvido a base de dados e algumas estruturas partilhadas;
 - IViewMobileCore – base de dados local.
- Library Layer – para este projeto utilizamos como dependências:
 - Liverpool Template;
 - Silk UI;
 - FileSystem;
 - Html2PdfConverter.

3.2 Modelo entidade-associação da base de dados

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

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

3.2.1 Utilizadores

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

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

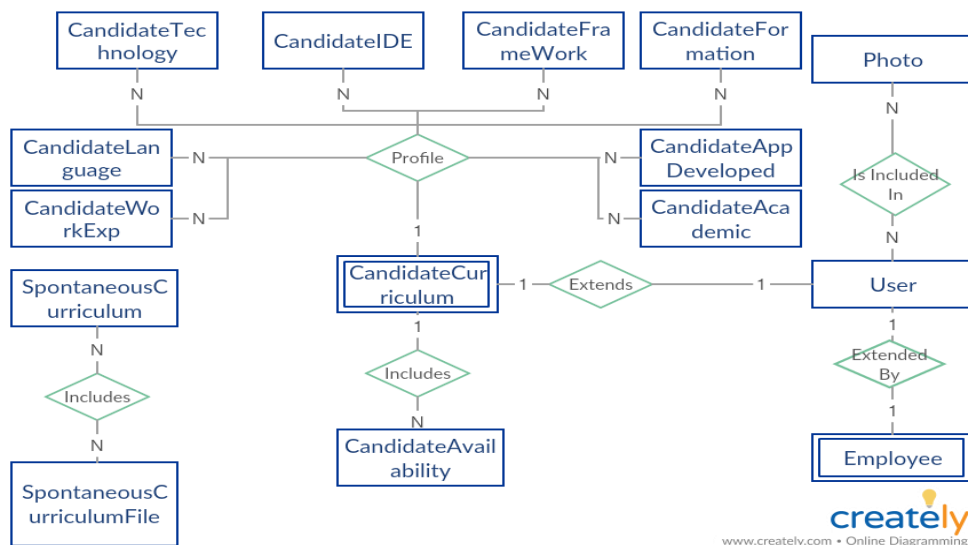


Figura 10 - Modelo EA, Candidatos.

As entidades da Figura 10 são:

- User, uma entidade padrão da OutSystems, representa os utilizadores 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;
- Employee, as instancias desta entidade adicionam informação ao User, tal informação sendo relacionada com a posição e funções do utilizador na PS Tec;
- CandidateCurriculum, uma das entidades que adiciona informação a utilizadores candidatos, representando o currículo do mesmo;
- CandidateAvailability, instancias desta entidade representam a disponibilidade que o candidato tem para entrevistas, associando a um dia de semana uma janela de tempo;
- CandiadteTechnology, uma entidade que faz parte do dossiê de capacidades de candidatos, representa a capacidade que um candidato tem com uma linguagem de tecnologia, reconhecida pela aplicação;
- CandiadteFramework, uma entidade que faz parte do dossiê de capacidades de candidatos, representa a capacidade que um candidato tem com uma *framework*, reconhecida pela aplicação;
- CandiadteIDE, uma entidade que faz parte do dossiê de capacidades de candidatos, representa a capacidade que um candidato tem com uma IDE, reconhecida pela aplicação;
- CandiadteFormation, 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 curso académicos que o candidato considere importante de notar;

- CandidateAppDeveloped, uma entidade que faz parte do dossiê de capacidades de candidatos, representa aplicações desenvolvidas pelo candidato que o candidato considere importante de notar.

3.2.2 Vagas

Nesta componente da base de dados estão incluídas todas as entidades que permitem estabelecer vagas para candidatos se aplicarem, como também essas mesmas aplicações de candidatos.

A Figura 11 demonstra estas entidades.

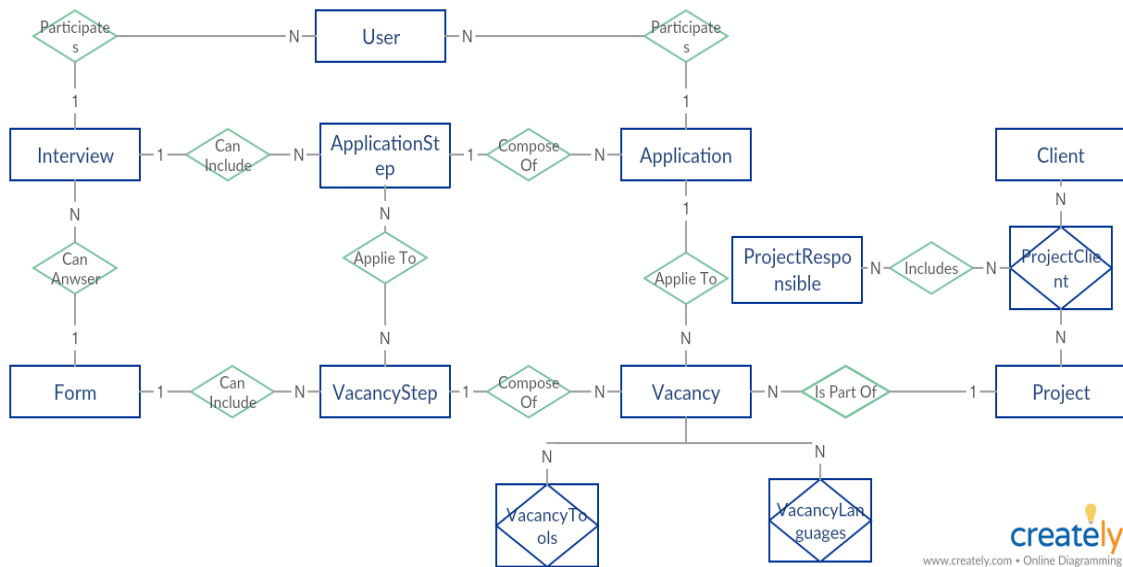


Figura 11 - Modelo EA, Vagas.

As entidades incluídas na Figura 11 são:

- Client, nesta entidade é possível incluir informações dos clientes da PS Tec, que podem iniciar projetos com a PS Tec;
- Project, projetos a serem realizados pela PS Tec, por estes projetos é possível estabelecer vagas para candidatos;
- ProjectClient, com esta entidade é possível associar vários clientes a um projeto, e claro um cliente a vários projetos;
- ProjectResponsible, cada cliente 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 instancia 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 instancias 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;
- Application, quando uma vaga é criada qualquer candidato da aplicação pode-se aplicar à mesma, tal aplicação a vaga é representada por uma instancia desta entidade;
- ApplicationStep, como uma vaga, Vacancy, pode ser composta por vários passos, uma aplicação a uma vaga ocorrerá em vários passos, tais passos são representados por esta entidade;
- Interview, uma entrevista que tenha ocorrido/ ainda para ocorrer com um utilizador, que refere. Possivelmente devido a um passo duma aplicação que também refere. Também refere um Form, o formulário utilizado da entrevista

3.2.3 Eventos

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

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

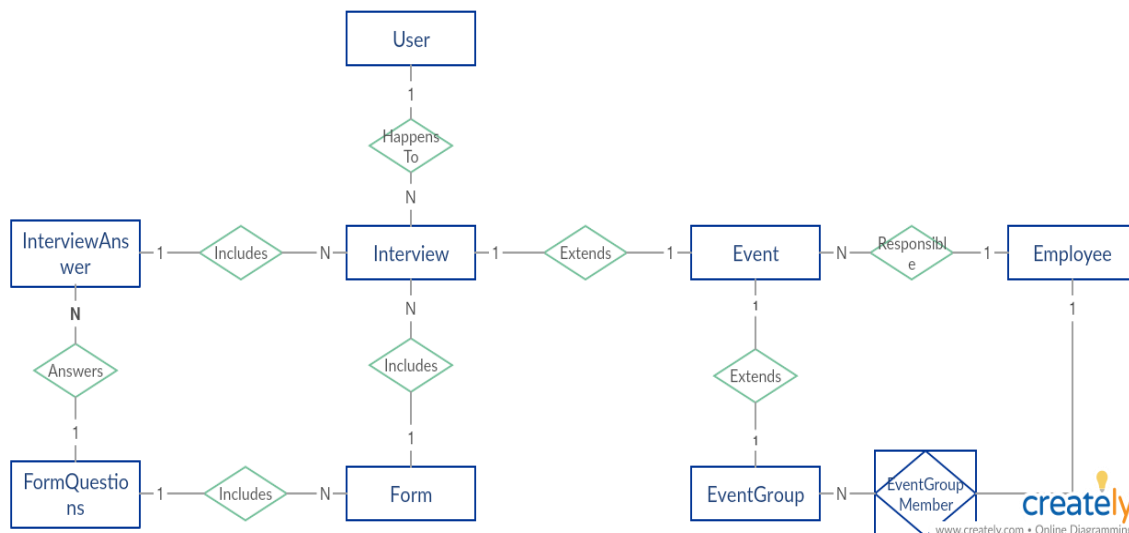


Figura 12 - Modelo EA, Eventos.

Na Figura 12 pode ser verificado as seguintes entidades:

- Event, esta entidade representa os vários eventos que podem ocorrer que envolvam pelo menos um colaborador, que 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 refere. Deve referir o formulário utilizado;
- FormQuestion, cada instancia 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 instancias desta entidade.

3.3 Wireframes do projeto

Para esta secção serão demonstradas algumas *frames* para páginas importantes, em que algumas são únicas na aplicação e outras servem de exemplo que demonstram o formato habitual de várias páginas, também são utilizados estes exemplos para demonstrar alguns dos elementos mais comuns da aplicação, tanto *widgets* (*rich* e normais) estabelecidos pela OutSystems como também elementos mais estruturados especificamente para a aplicação.

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

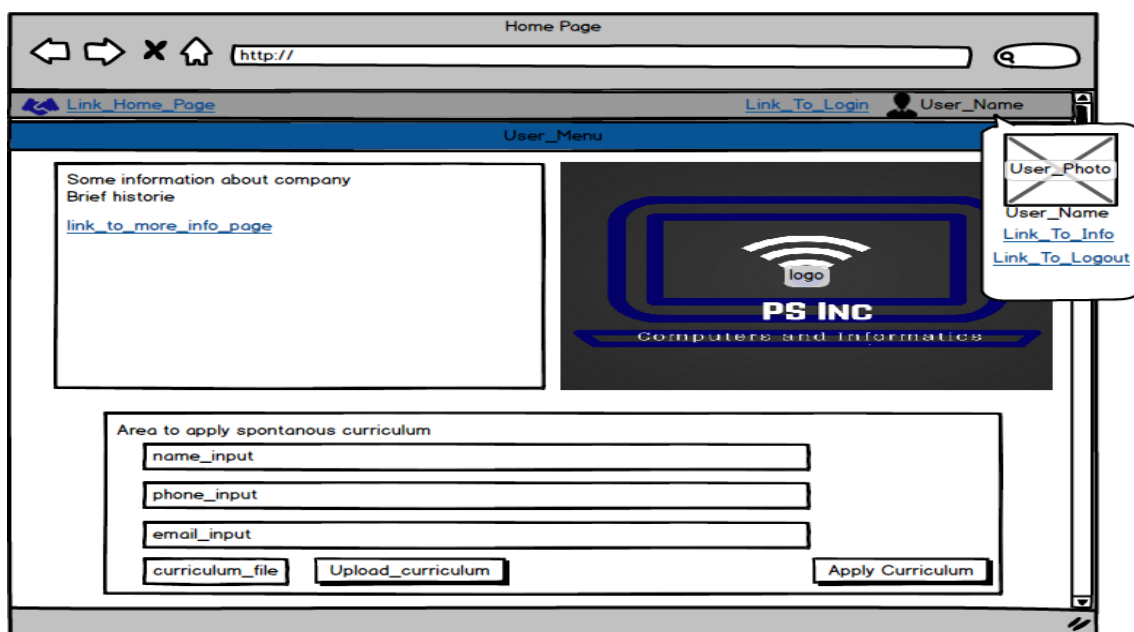


Figura 13 – Frames, Home Page.

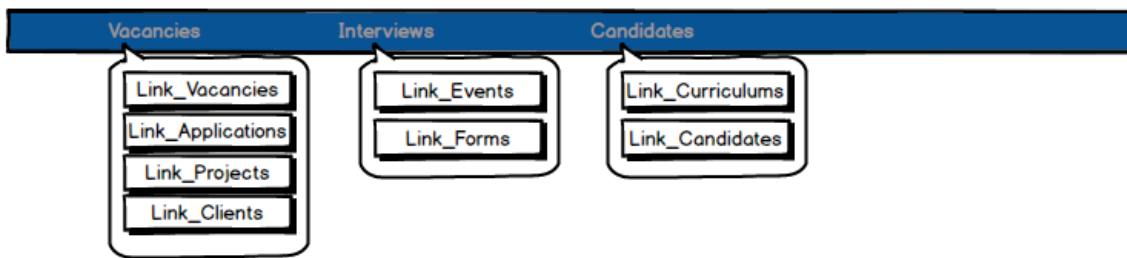


Figura 14 - Frames, Menu de Colaboradores.

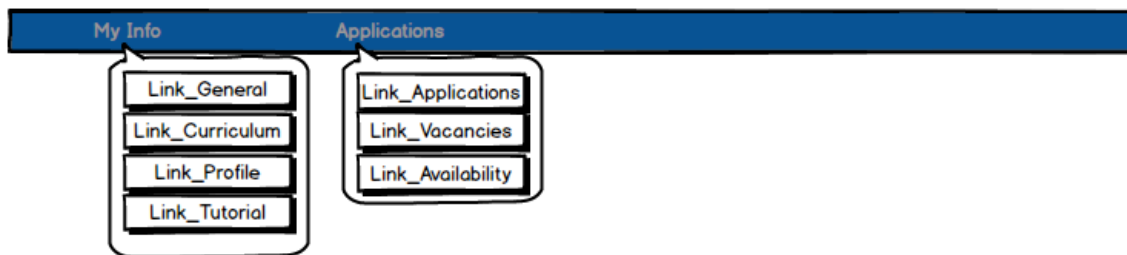


Figura 15 - Frames, Menu de Candidatos.

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*. A escolha das duas possibilidades é feita em função do utilizador ser não registrado (primeira possibilidade) ou estar registrado (segunda possibilidade).

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 de candidatos. Deve ser notado que os menus só aparecem para utilizadores já registrados, cada menu é composto por um conjunto de elementos que abram e fechem um Balloon³ associado, onde é incluindo um conjunto de links para paginas consideradas importantes.

A página principal, como se pode verificar na Figura 13, inclui o logotipo da PS Tec, como também um simples resumo da mesma, onde é incluindo um *link* para página de informações, também é nesta página que é possível realizar a candidatura espontânea, pela utilização de três *inputs* (um texto, outro *email* e outro *phone*) onde o candidato pode introduzir algumas informações gerais, mais um *upload input* onde o candidato pode introduzir o seu currículo. Estes dados são todos fornecidos à aplicação por um só botão.

Na Figura 16, Figura 17, Figura 18, Figura 19 e Figura 20 é possível verificar a página de informação geral da PS Tec que introduz dois elementos visuais fundamentais à aplicação, mais outro menos utilizável, note-se que as cinco figuras representam uma só pagina, mas cada demonstra uma Tab diferente da pagina.

³ Rich Widget desenvolvido pela OutSystems que pode ser associado a um elemento na pagina, desde que tenha um nome. Quando o elemento é pressionado um balão de dialogo é aberto.

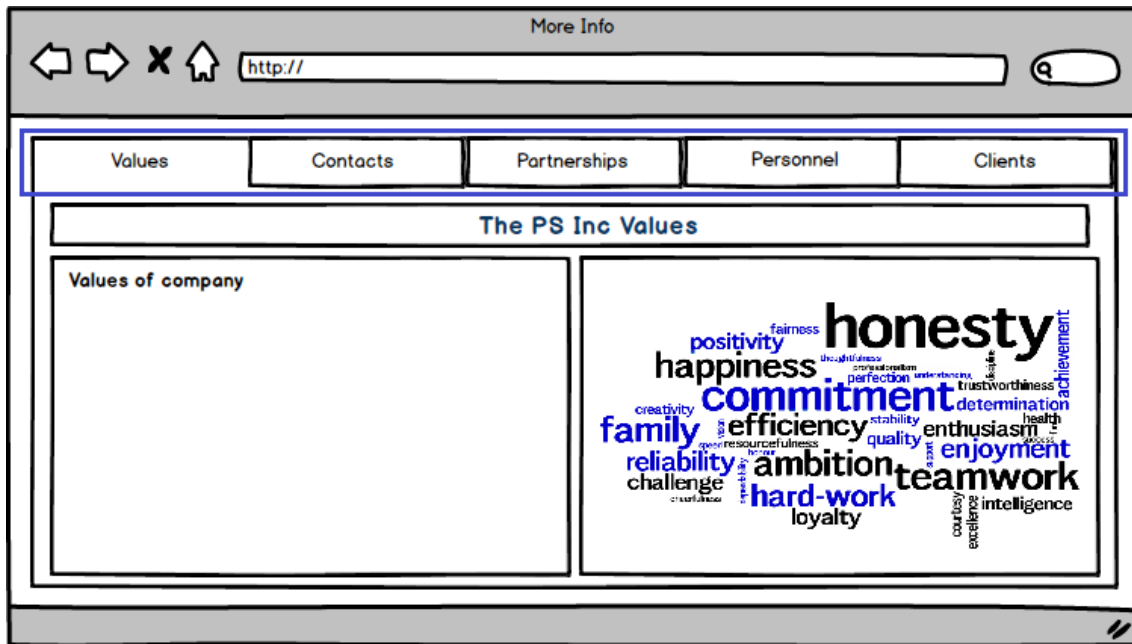


Figura 16 - Frame, MoreInfo Values Tab.

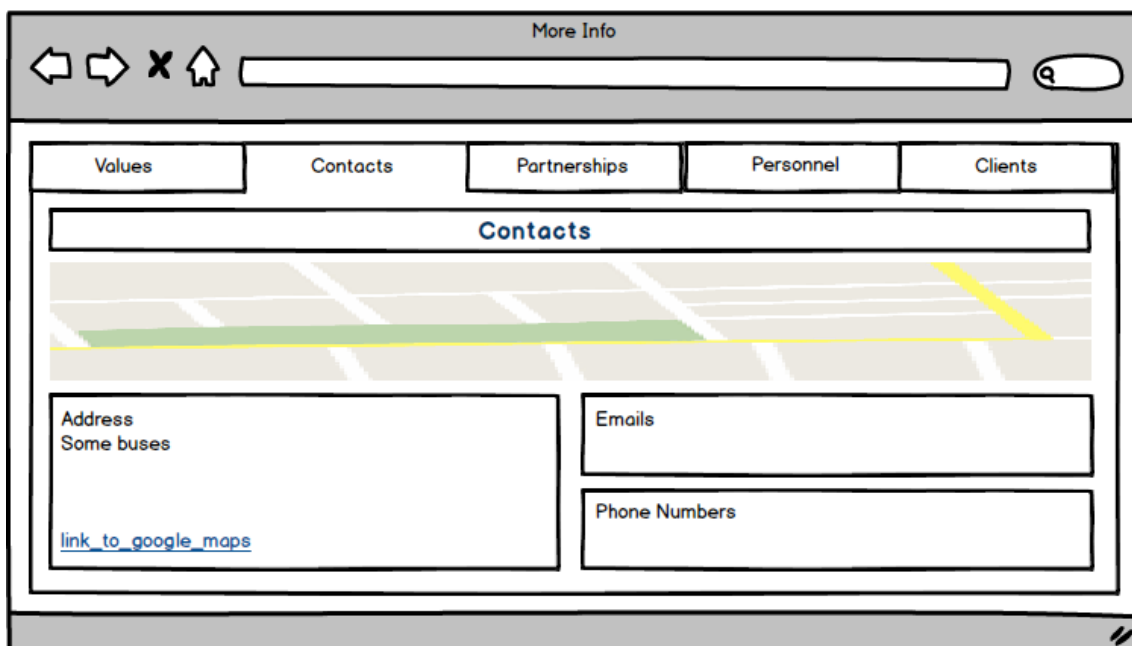


Figura 17 - Frame, MoreInfo Contacts Tab.

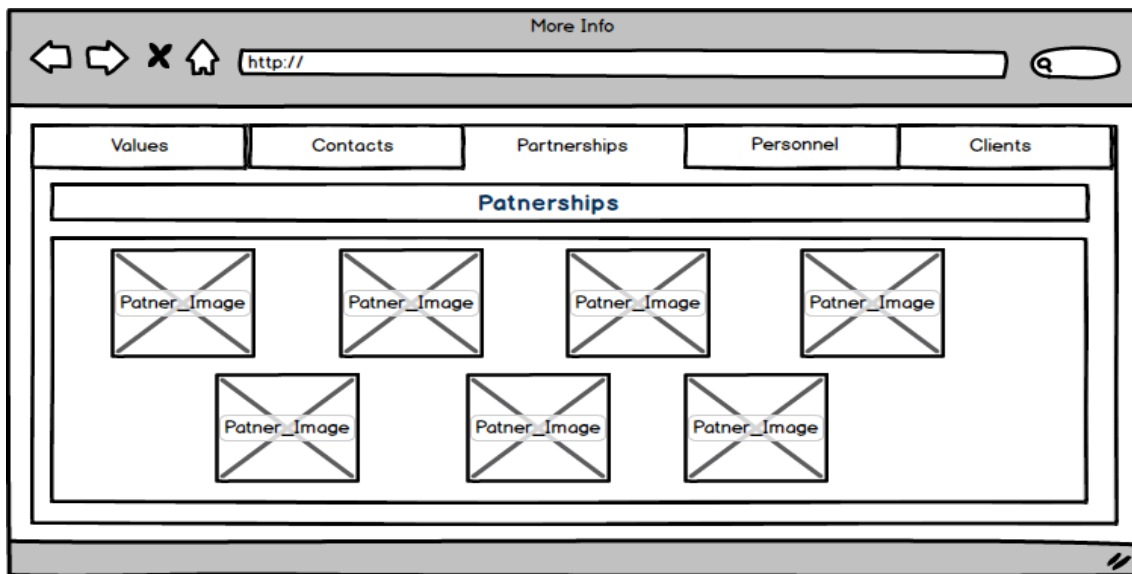


Figura 18 - Frame, MoreInfo Patnerships Tab.

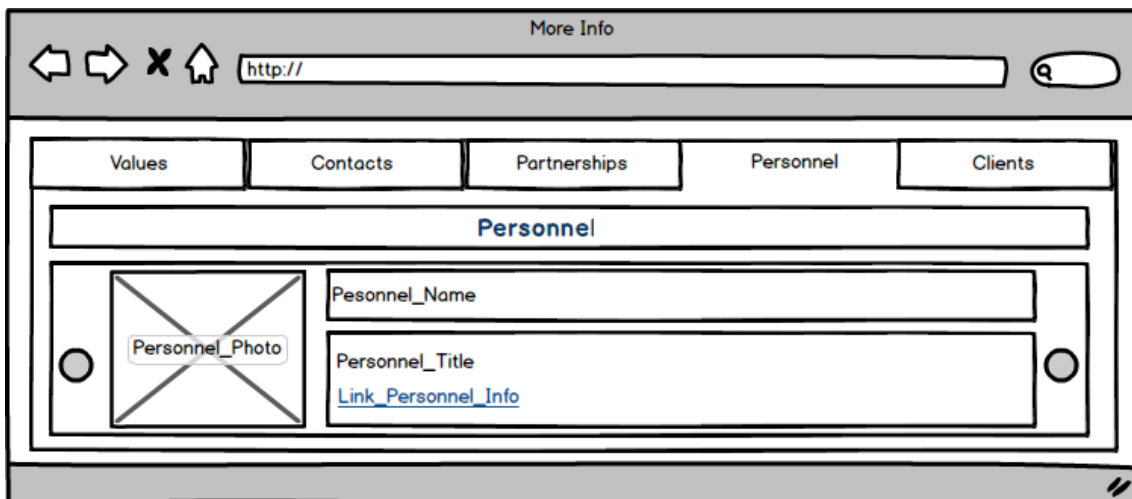


Figura 19 - Frame, MoreInfo Personnel Tab.

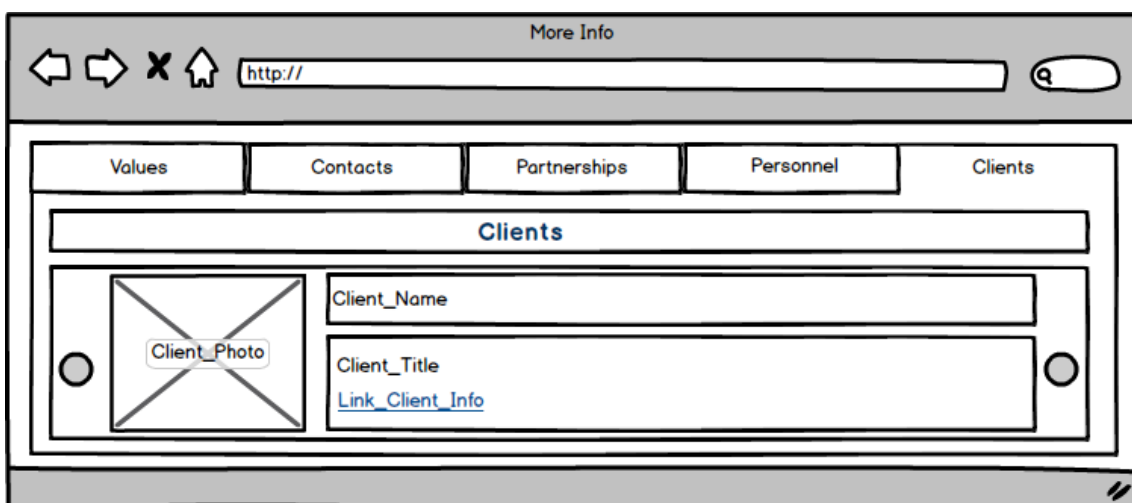


Figura 20 - Frame, MoreInfo Client Tab.

Esta página demonstra:

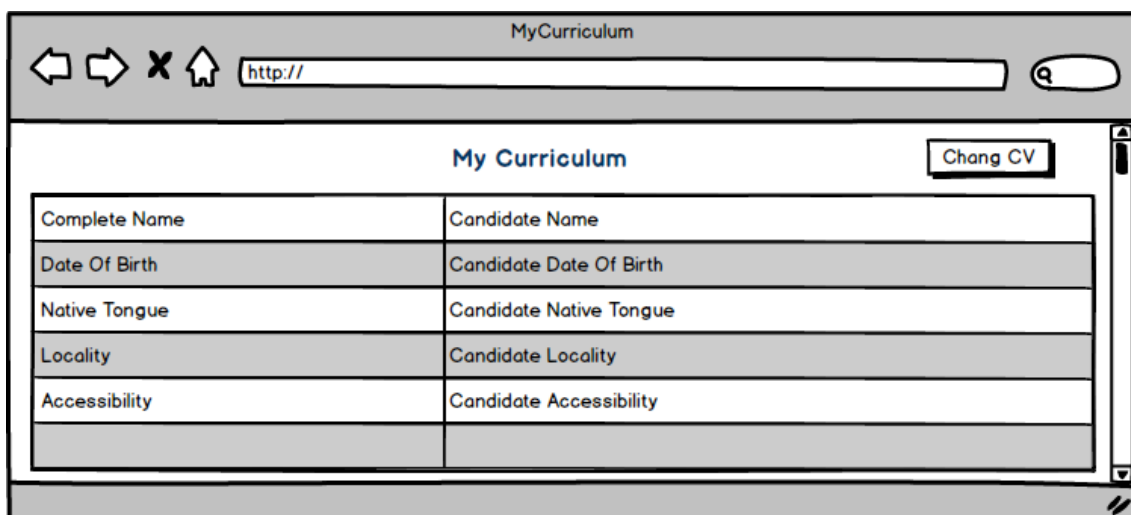
- Valores e ambições da PS Tec;
- Contactos;
- Parcerias importantes;
- Colaboradores registrados na aplicação;
- Clientes já registrados.

Como referido anteriormente existem dois elementos importantes na aplicação que podem ser verificados nesta página, o primeiro desses é o *rich widget* Tabs, que pode ser verificado no topo da página. Este elemento permite dividir uma página até cinco partes, com uma sempre visualizada e as outras quatro escondidas, assim quando uma página inclui muita informação, em vez de ser extensa verticalmente, as informações estão divididas em *tabs*.

O outro elemento utilizado nesta página a notar é o *widget* List Records, que permite demonstrar vários elementos obtidos na base de dados, numa estrutura bastante livre, não forçando um formato tipo tabela, a List Records é normalmente utilizada em situações em que se quer demonstrar informação que possa ocupar mais que uma linha ou inclua pelo menos uma imagem.

No caso da Figura 19 e da Figura 20, esta *widget* é utilizada com a *rich widget* Caroussel para demonstrar os clientes e colaboradores num formato simples, cada um dos colaboradores e clientes demonstrados são representados por uma instancia do *rich widget* CardLeftImage, que permite demonstra uma figura a esquerda e informação geral a direita, este *rich widget* é utilizado varias vezes, principalmente em listas de Users e clientes.

Na secção **Modelo entidade-associação da base de dados** foi estabelecido a existência de um currículo que um candidato pode verificar e alterar, estas duas capacidades são possíveis na página representada pela Figura 21, verificar, e pela Figura 22, editar.



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

Figura 21 - My Curriculum, Show.

Figura 22 - Frame, MyCurriculum, Edit.

Esta página serve como exemplo da utilização de alguns *inputs* da aplicação como também de dois *widjets* muitas vezes utilizados para demonstrar e alterar informação, tais *widjets* são Show Record e Edit Record, ambos partilham o formato tabela e por isso são bastante utilizados em conjunto e às vezes separados. O formato apesar de simples combina visualmente com qualquer página e por isso tem uma utilidade regular quando se quer demonstrar ou introduzir informação extensa.

Nesta página são utilizados dois *inputs* que irão aparecer algumas vezes, primeiro é o Combo Box, verificado a frente de “Native Tongue”, da Figura 22, este *input* permite associar a uma variável um valor duma lista estabelecida, assim limitando o valor escolhido a valores na lista. No exemplo da Figura 22, a língua nativa do currículo é limitada a línguas reconhecidas pela aplicação, instancias da entidade Languages. O outro *input* é na verdade a combinação de um *input* simples, mais o *rich widjet* Input Calendar, que faz aparecer um calendário onde se pode escolher uma data cada vez que o *input* normal é pressionado, com a data escolhida sendo demonstrada nesse mesmo *input*.

Como também foi referido na secção **Modelo entidade-associação da base de dados**, um candidato consegue controlar o seu dossiê de capacidades que é composto de várias entidades associadas ao colaborador. A **Erro! A origem da referência não foi encontrada.** demonstra esta página que introduz alguns elementos notáveis e regularmente utilizados na aplicação.

Frameworks

Framework	Practical Experience	Academic Experience	Capacity
Framework_Chosse	3	3	Capacity_Chosse

☐ Save
☐ Cancel
☐ Delete

☐ Add Framework

IDEs

IDE	Practical Experience	Academic Experience	Capacity
IDE_Chosse	3	3	Capacity_Chosse

☐ Save
☐ Cancel
☐ Delete

☐ Add IDE

Academics

Course	University	Complete	Average	Start Date	End Date
Course_Input	University_Input	<input type="checkbox"/>	Average_Input	/ /	/ /

☐ Save
☐ Cancel
☐ Delete

☐ Add Academic

Formations

Tool	Provider	Date
Tool_Chosse	Provider_Input	/ /

☐ Save
☐ Cancel
☐ Delete

☐ Add Formation

Work Experience

Position	Company	Start	End
Position_Input	Company_Input	/ /	/ /

☐ Save
☐ Cancel
☐ Delete

☐ Add Work Experience

Applications Developed

Name	Name_Input
Start Date	/ /
End Date	/ /
Responsibilities	Responsibilities_Input
Technologies	Technologies_Input
Accomplishments	Accomplishments_Input
Summarization	Summarization_Input

Figura 23 - Frames, MyProfile.

O primeiro elemento a notar é a Editable Table, uma tabela que permite adicionar elementos à mesma, como também alterar elementos já existentes, estas tabelas incluem sempre um botão

que adiciona um vazio a tabela, que pode ser alterado. Quando um elemento desta tabela é selecionado é possível alterar o mesmo, sendo disponível três botões, um cancela qualquer alteração, outro remove a instancia selecionada e último guarda as alterações à mesma. No caso desta página, o remover e guardar elementos 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. Sendo que ListRecord não permite a utilização direta de EditRecord, o mesmo é na verdade parte de um WebBlock. Esta combinação de ListRecord e WebBlock é utilizada varias vezes para listar elementos com informação extensa.

Já para adicionar uma nova instancia a CandidateAppDeveloped é utilizado o PopUp Editor⁴. No caso da MyProfile, quando o botão “Add” é pressionado um PopUp abre que permite introduzir as informações da aplicação desenvolvida, normalmente PopUps não são utilizados para introduzir novos valores a uma entidade, mas sim para facilitar a escolha de instancias duma tabela.

Este PopUp pode ser verificado na Figura 24.

Name	Name_Input
Start Date	// [Calendar Icon]
End Date	// [Calendar Icon]
Responsibilities	Responsibilities_Input
Technologies	Technologies_Input
Accomplishments	Accomplishments_Input
Summarization	Summarization_Input
<input type="button" value="Cancel"/> <input type="button" value="Add"/>	

Figura 24 - Frames, MyProfile PopUp.

Uma funcionalidade regular das páginas da aplicação web da IView é de listar um conjunto de instancias estabelecidas duma entidade da base de dados. Um exemplo simples destas páginas é o demonstrado na Figura 25, que mostra todas as instancias da entidade Form.

⁴ Rich widget que permite criar uma página tipo *popup* a partir de uma página já estabelecida na aplicação.

Figura 25 - Frames, Forms.

Esta página apesar de simples mostra os componentes mais comuns das páginas que demonstram as instâncias de uma entidade, a primeira é a utilização da Table Records para demonstrar as instâncias em si. Esta *widget* é semelhante a List Records, ambas demonstram instâncias de uma lista, mas ao contrário da List a Table demonstra a informação no formato de tabela. Esta *widget* serve principalmente para demonstrar instâncias com informação pouco extensa e por isso que possam ser demonstradas numa só linha de texto. Normalmente os elementos desta tabela incluem um *link* que permite aceder à página onde a instância pode ser demonstrada numa forma mais extensa, onde também é possível alterar ou destruir a instância.

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 em conta a tabela, sem de precisar de código extra da parte da aplicação.

Outro elemento bastante comum nas páginas de instâncias é um formulário de pesquisa, que normalmente é composto por um *input* e dois botões, no *input* o utilizador pode introduzir um valor de pesquisa, normalmente um valor texto da entidade demonstrada na página, sobre os botões um permite pesquisa reexecutando o Aggregate que obtém a informação da tabela, com o valor da *input*, outro permite restabelecer a tabela, com o valor de pesquisa *null*.

Por último temos o botão de adição, no caso da Figura 25, o botão “Add Form”, este botão não é utilizado sempre neste tipo de páginas, mas aparece em grande parte 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 à pesquisa, outros demonstram a informação numa forma bastante diferente e por último outros introduzem mais ações possíveis por cada instância da entidade listada, um exemplo dessas exceções pode ser visto na Figura 26, onde é verificada a página de listagem de aplicações a vagas existentes.

Figura 26 - Frames, Applications.

Como se pode ver entre a Figura 25 e a Figura 26 existem três diferenças notáveis, a primeira é a falta do botão de adição, o que faz sentido sendo que uma aplicação só pode ser adicionada por um colaborador, utilizador que não tem acesso a página da Figura 26.

A tabela também 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 ultimo 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 27.

Figura 27 - Frames, Vacancies.

Como se pode ver a pesquisa de vagas é realizado sem nenhum *input* de texto, sendo realizado unicamente por escolha de um valor de uma Combo Box, que demonstra as instancias de JobTitle e pela escolha de ferramentas e /ou linguagens, este formulário, como se pode notar não inclui qualquer botão de pesquisa, qualquer alteração ao Combo Box ou da seleção resulta numa pesquisa. Também deve-se notar que o botão “Smart Search” altera a visualização do formulário, e caso o botão torne o formulário invisível, então pesquisa é cancelada e as vagas demonstradas não são limitadas pelos valores no formulário.

Outra exceção a notar é a página de listagem de eventos que pode ser verificada na Figura 28.

Figura 28 - Frames, Events.

Uma alteração notável desta página quando comparada as outras páginas já referidas é que a paginação de eventos é realizada em função do utilizador atual. Os eventos listados na página são os eventos em que o utilizador participa, como responsável ou como participante, em eventos em grupo, também pode ser notado que a pesquisa é realizada, não com um *input* simples, mas com um Input Calendar e que os eventos listados não são só os eventos do dia da pesquisa, mas também os próximos quatro dias, não incluindo fim-de-semana.

Mas o fator mais notável de exceção é 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, não com a utilização de uma Table Records, mas uma List Records onde cada elemento listado resulta num bloco ocupado ou livre.

A ultima exceção a notara é verificada na Figura 29, que não envolve o formulário de pesquisa e adiciona capacidades a cada instancia da entidade. Esta figura demonstra a listagem de todas as candidaturas espontâneas ainda não tratadas.

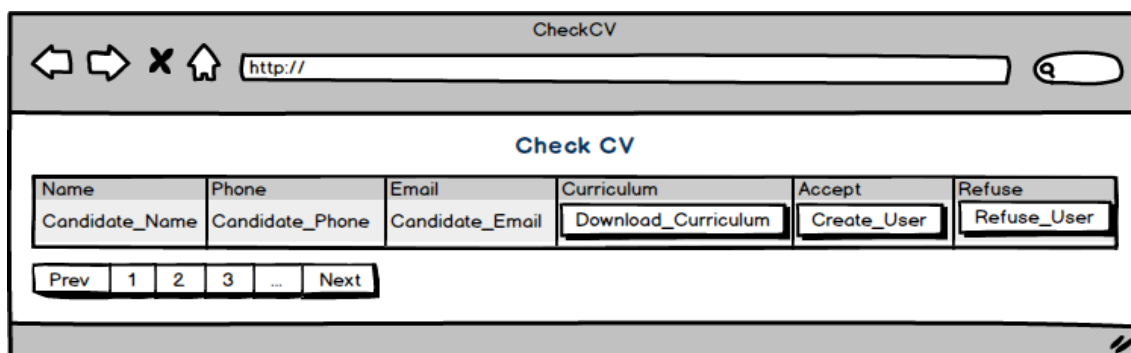


Figura 29 - Frames, CheckCV.

Como se pode ver cada instancia da tabela da Figura 29, inclui três botões com cada tendo a sua funcionalidade:

- Download, faz download do currículo enviado na candidatura;
- Accept, aceita a candidatura e é criado um utilizador, candidato, com a informação da candidatura, remove a candidatura da base de dados;
- Refuse, recusa a candidatura, remove a candidatura da base de dados.

Anteriormente foi estabelecido o facto que as páginas de listagem normalmente incluem um *link* nas instancia listadas, para uma página que demonstra a informação da instancia mais detalhada. Um exemplo simples deste tipo de página pode ser verificado na Figura 30, que demonstra a página de um candidato.

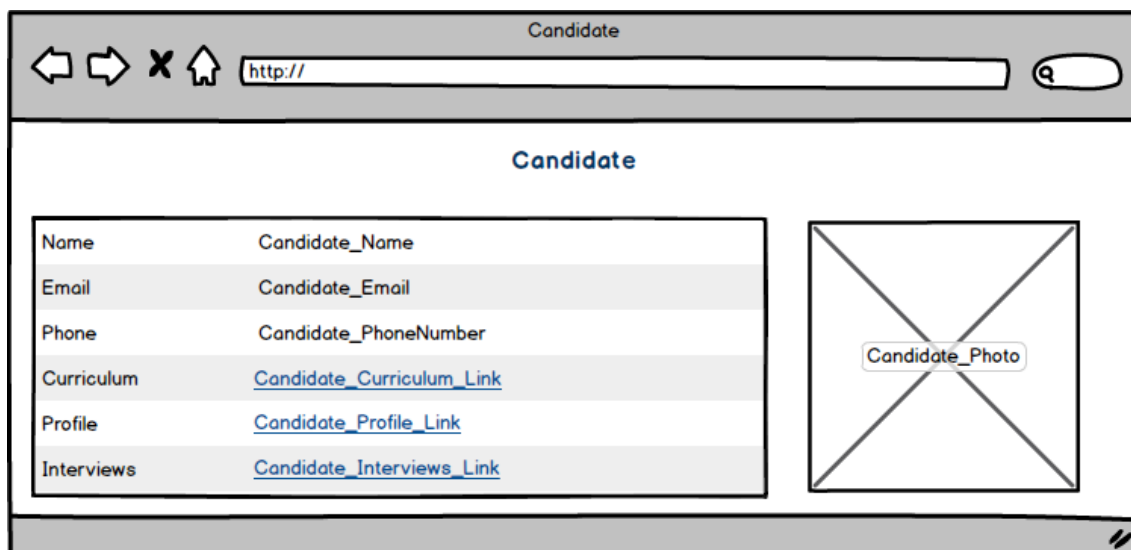


Figura 30 - Frames, Candidate.

Esta página, como muitas outras que demonstram uma instancia, utiliza o Show Record para demonstrar a informação da instancia em si, muitas vezes as páginas deste tipo não são tão simples, sendo que algumas instancias em si estão associadas a vários elementos de outra

entidade, nesses casos a página também demonstra essas instancias de outra entidade, como se pode ver na Figura 31, onde se pode verificar a página que demonstra alguma informação de um evento, mais os seus participantes.

The image shows a web browser window with the title "Event". The address bar contains "http://". The main content area is divided into several sections. The top section is titled "Event" and contains a date range: "Date : Event_Date Start: Event_Start End: Event_End". Below this, there are two main sections: "Responsible" and "Participant". Each section contains a table with fields like Name, Email, and Phone, and a placeholder for a photo (labeled "Responsible_Photo" and "Participant_Photo" respectively). At the bottom, there's a "Group" section with a "Participant" table and a photo placeholder, followed by an "Event Comment" text area.

Figura 31 - Frames, Event.

Algo a notar na página é que a mesma inclui sempre os blocos Responsible e Event Comment, podendo incluir o bloco Participant ou Group, caso for uma entrevista ou evento de grupo. No primeiro caso de um dos blocos possíveis, é simplesmente a utilização duma imagem e um Show Record para demonstrar o participante, mas no caso de grupo para demonstra um elemento individual de tal grupo, também é utilizado a combinação, mas o mesmo está numa List Record que lista todos os elementos de um grupo para um evento.

Às vezes a listagem dos elementos pode ser mais complexa que a verificada na Figura 31, sendo que em certas listagens o que é demonstrado depende de vários fatores da instancia a ser listada, um bom exemplo de uma página com esta complexidade é a página da Figura 32, que demonstra a página de uma aplicação, o que envolve a listagem dos passos dessa aplicação que já ocorreram e o passo atualmente a correr.

Figura 32 - Frames, Application.

Neste caso o valor na lista depende do estado do passo e possivelmente no tipo do passo, com cada estado permitindo executar uma ação diferente sobre o passo. Deve ser notado que os textos “If_Step_State...” e “If_Step_Is/Not” não aparecem na página, só servem na frame para indicar que estado do passo demonstra o conteúdo relacionado ao texto, neste caso o estado da aplicação demonstra um botão diferente que permite uma ou mais opções.

- Estado ToSchedule, como o passo está para marcar, a opção permite marcar a entrevista/evento;
- Estado Scheduled, a aplicação tem uma entrevista marcada, a opção permite demonstrar essa entrevista;
- Estado ToProcess, o processo pode ser continuado para o próximo passo, neste caso há varias opções:
 - O passo é entrevista – é possível verificar a opção;
 - O passo não é entrevista – permite demonstrar informação geral do candidato;

- O passo é contracto – o candidato é contratado;
- O passo não é contracto – o processo pode ser continuado para o próximo passo.
- Estado Finished, o passo já terminou, nesse caso existem duas possibilidades:
 - O passo é entrevista – é possível verificar a opção;
 - O passo não é entrevista – permite demonstrar informação geral do candidato.

Deve ser notado que a aplicação pode ser terminada, utilizando o Botão “End” no canto superior direito.

As vezes estes tipos de páginas demonstram informação bastante extensa, então para facilitar a visualização e organização da página são utilizadas Tabs para dividir blocos de informação, um exemplo da utilização das Tabs nestas páginas pode ser verificado na Figura 33, Figura 34 e Figura 35 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.

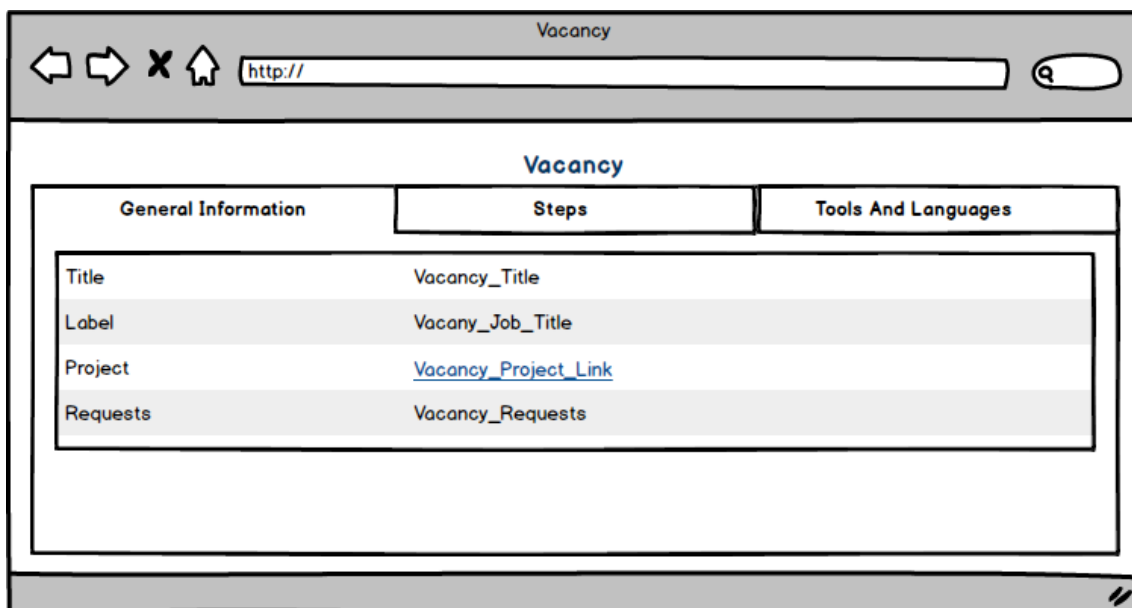


Figura 33 - Frames, Vacancy General Information Tab.

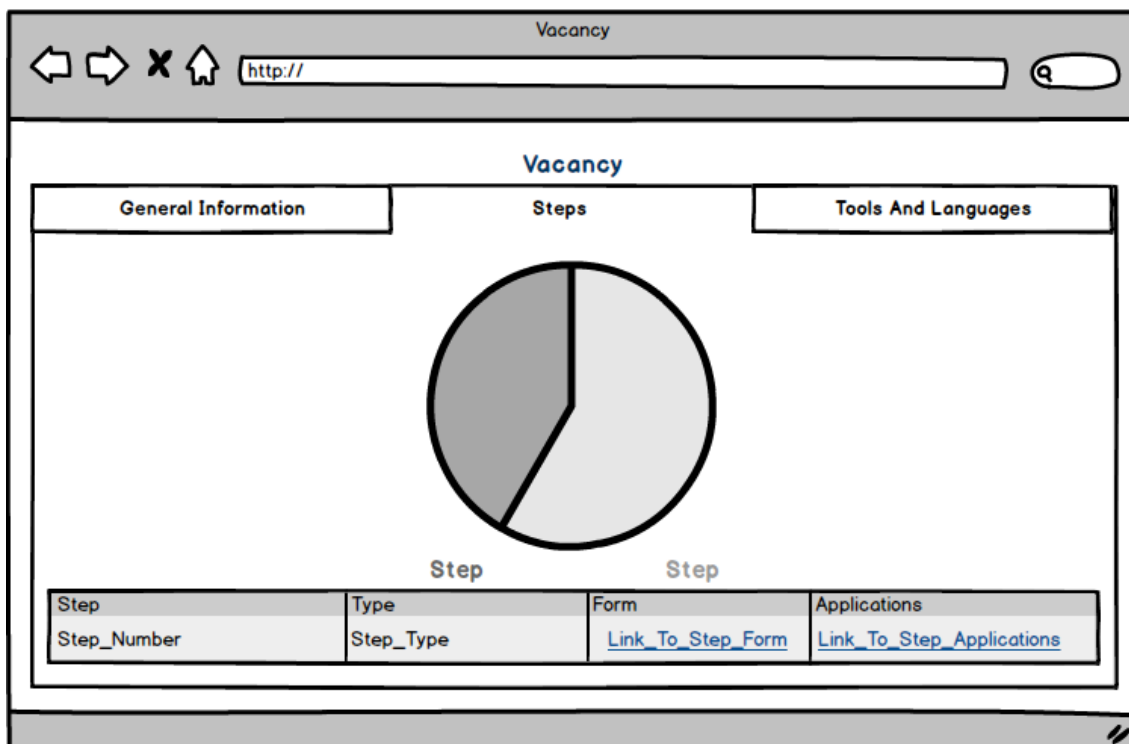


Figura 34 - Frames, Vacancy Steps Tab.

Figura 35 - Frames, Vacancy Tools and Languages Tab.

O único outro fator a notar da página da Figura 34 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, mas deve ser notado para esta.

Por ultimo deve ser verificado páginas que servem para introduzir novas instancias das entidades da base de dados, um exemplo bastante simples destas páginas é a verificada na Figura 36, que demonstra a página que permite introduzir um formulário.

Figura 36 - Frames, FormEditAdd.

A página da Figura 36, demonstra os dois *widgets* mais comuns a páginas para adicionar instancias, o primeiro sendo o Edit Record, este *widget* é normalmente utilizado quando se quer introduzir a informação que compõe a instancia a introduzir, o outro *widget* regular é o Edit Table, que é utilizado regularmente para introduzir outras instancias sobre uma entidade relacionada com a instancia central a adicionar. No exemplo da Figura 36, o Editable Table serve para introduzir as questões (FormQuestions) que compõem o formulário. Outros dois elementos regulares destas páginas são os botões “Create” e “Cancel”, com o primeiro servindo para criar a instancia formulada na página e o segundo para cancelar a criação da instancia.

Em algumas situações uma página simples não será suficiente para estruturar uma instancia a adicionar a base de dados, nesses casos uma de duas soluções são utilizadas, a utilização de Tabs ou de PopUps, o exemplo de uma PopUp utilizada neste tipo de paginas pode ser verificada na Figura 37.

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

Figura 37 - Frames, VacancyAdd Form Chosse PopUp.

The screenshot shows a web browser window titled "VacancyAdd" with a search bar. The main content area is titled "Vacancy Add" and features a "Create" button. Below this is a tabbed interface with four tabs: "Main", "Steps", "Tools", and "Languages". The "Main" tab is active, displaying a form with the following fields:

Title	Vacancy_Title_Input		
Number of People	Vacancy_Number_Of_People_Input		
Possition	Vacancy_Possition_Chosse		
Requests	Vacancy_Requests_Input		
Project	Vacancy_Project_Chosse	Chosse Project	

Figura 38 - Frames, Vacacny Add Main Tab.

The screenshot shows the "Steps" tab of the "Vacancy Add" interface. It contains a "Step Type" dropdown menu with "Step_Type_Chosse" selected, a "Form" dropdown menu with "Step_Form_Chosse" selected, and a "Chosse Form" button. Below these are three radio buttons: "Add Step", "Save", and "Delete".

Figura 39 - Frames, Vacacny Add Sterps Tab.

The screenshot shows the "Tools" tab of the "Vacancy Add" interface. It features a "Tools" header and a checkbox labeled "Tool_Chosse".

Figura 40 - Frames, Vacacny Add Tools Tab.

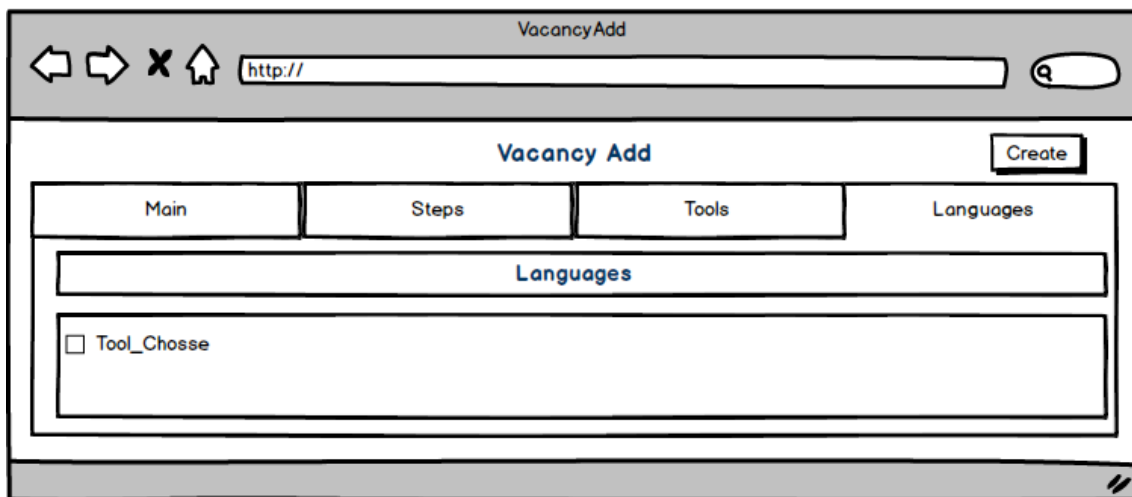


Figura 41 - Frames, Vacacncy Add Languages Tab.

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

4 Avaliação Experimental

5 Conclusões

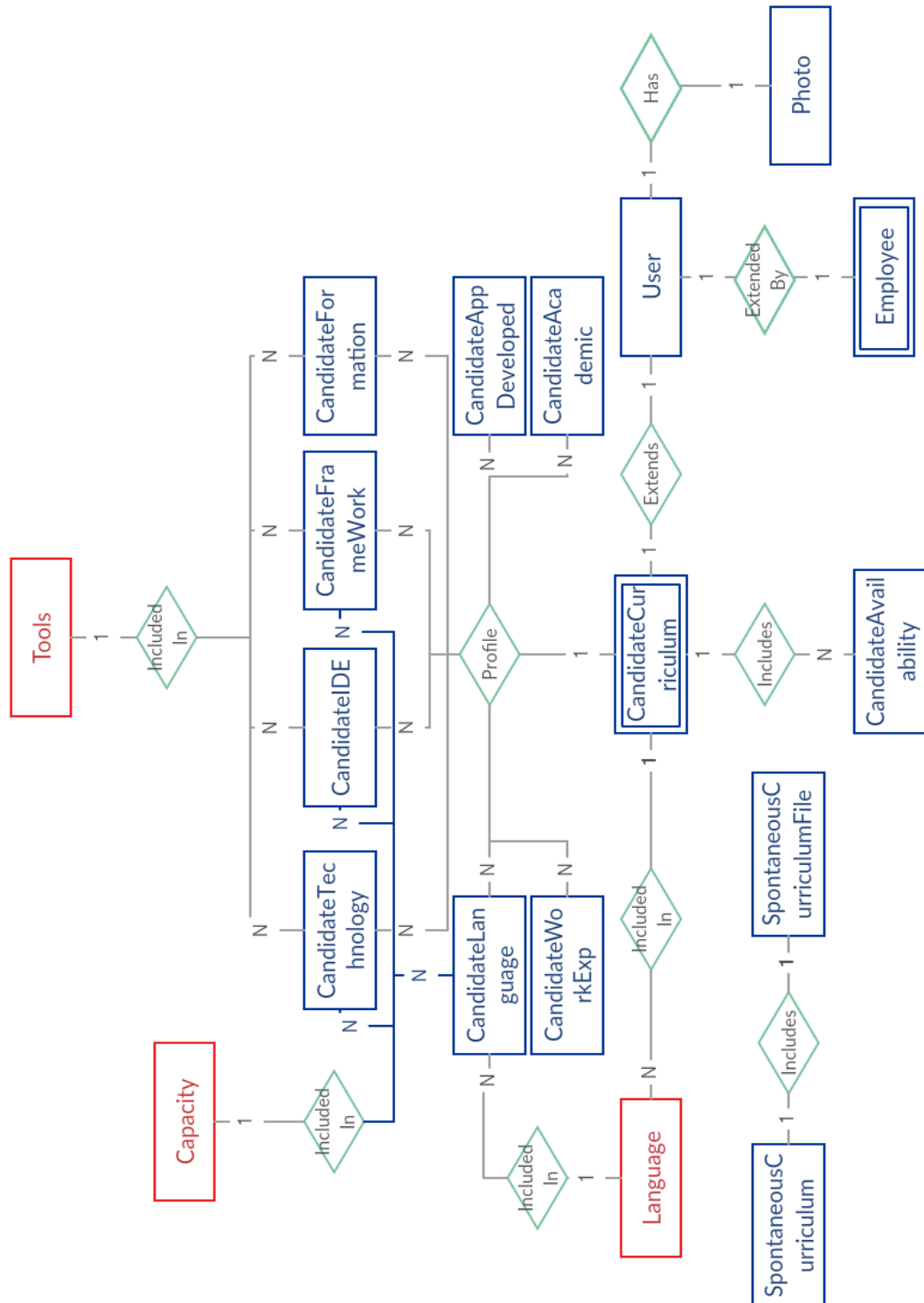
Referências

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

A.1 Diagramas da Aplicação

A.2 Modelos de dados

Deve ser notado que que nos modelos que se seguem, as entidades estáticas estão a vermelho, enquanto que entidades dinâmicas a azul.



Modelos De Dados 1 - Utilizadores

As entidades no modelo anterior são:

- User, uma entidade padrão da Outsystems, representa os utilizadores registados na aplicação;
- Photo, todos os utilizadores podem incluir uma fotografia, que é representada por esta entidade;
- Capacity, as instancias desta entidade demonstram os cinco níveis de capacidades reconhecidas pela aplicação
- Tools, todas as linguagens tecnológicas, frameworks e IDEs que a aplicação considera importantes e que os candidatos podem demonstra a sua capacidade do dossiê de capacidades,
- Languages, idiomas reconhecidos pela aplicação,
- SpontaneousCurriculum, como estabelecido anteriormente uma capacidade de utilizadores não registados é a capacidade de realizar uma candidatura espontânea. Esta candidatura envolve fornecer informação pessoal, nome, telemóvel e email e um currículo. Destas informações as primeiras três são guardadas em instancias desta entidade,
- SpontaneousCurriculumFile, o currículo em si da candidatura espontânea é guardado nesta entidade, que refere a SpontaneousCurriculum,
- Employee, as instancias desta entidade adicionam informação ao User, tal informação sendo relacionada com a posição e funções do utilizador na PS Tec,
- CandidateCurriculum, uma das entidades que adiciona informação a utilizadores candidatos, representando o currículo do mesmo,
- CandidateAvailability, instancias 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 a capacidade que um candidato tem com uma linguagem de tecnologia, reconhecida pela aplicação,
- CandidateFramework, uma entidade que faz parte do dossiê de capacidades de candidatos, representa a capacidade que um candidato tem com uma *framework*, reconhecida pela aplicação,
- CandidateIDE, uma entidade que faz parte do dossiê de capacidades de candidatos, representa a capacidade 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 curso académicos que o candidato considere importante de notar,
- `CandidateAppDeveloped`, uma entidade que faz parte do dossiê de capacidades de candidatos, representa aplicações desenvolvidas pelo candidato que o candidato considere importante de notar.

Deve ser notado que `User` na verdade representa todos os utilizadores reconhecidos na cloud onde corre as aplicações desenvolvidas, para restringir o acesso a só utilizadores estabelecidos na aplicação é utilizado dois roles, `IViewCandidate` e `IViewEmployee`. Estes dois também são utilizados para diferenciar os utilizadores que são candidatos e os que são colaboradores.

Um fator a notar é a utilização de `Photo`, sendo que originalmente considerou-se outra possibilidade para a fotografia. Tal era utilizar um campo na entidade `Employee` e `CandidateCurriculum`, as únicas duas entidades obrigatórias dos utilizadores da `IView`.

Tal ideia levantou o problema que em certas situações a fotografia de utilizador é utilizada sem saber que tipo de utilizador está a utilizar a aplicação. Como por exemplo a fotografia é sempre demonstrada no *header* da página.

Também existe o facto que a fotografia não vai sempre ser necessária, mas devido ao tipo, a mesma pode ocupar um certo espaço, que quando agrupando em grandes números, podem atrasar qualquer *aggregate*. Mas com a fotografia separada, tal tempo não acontece, exceto quando realmente necessário.

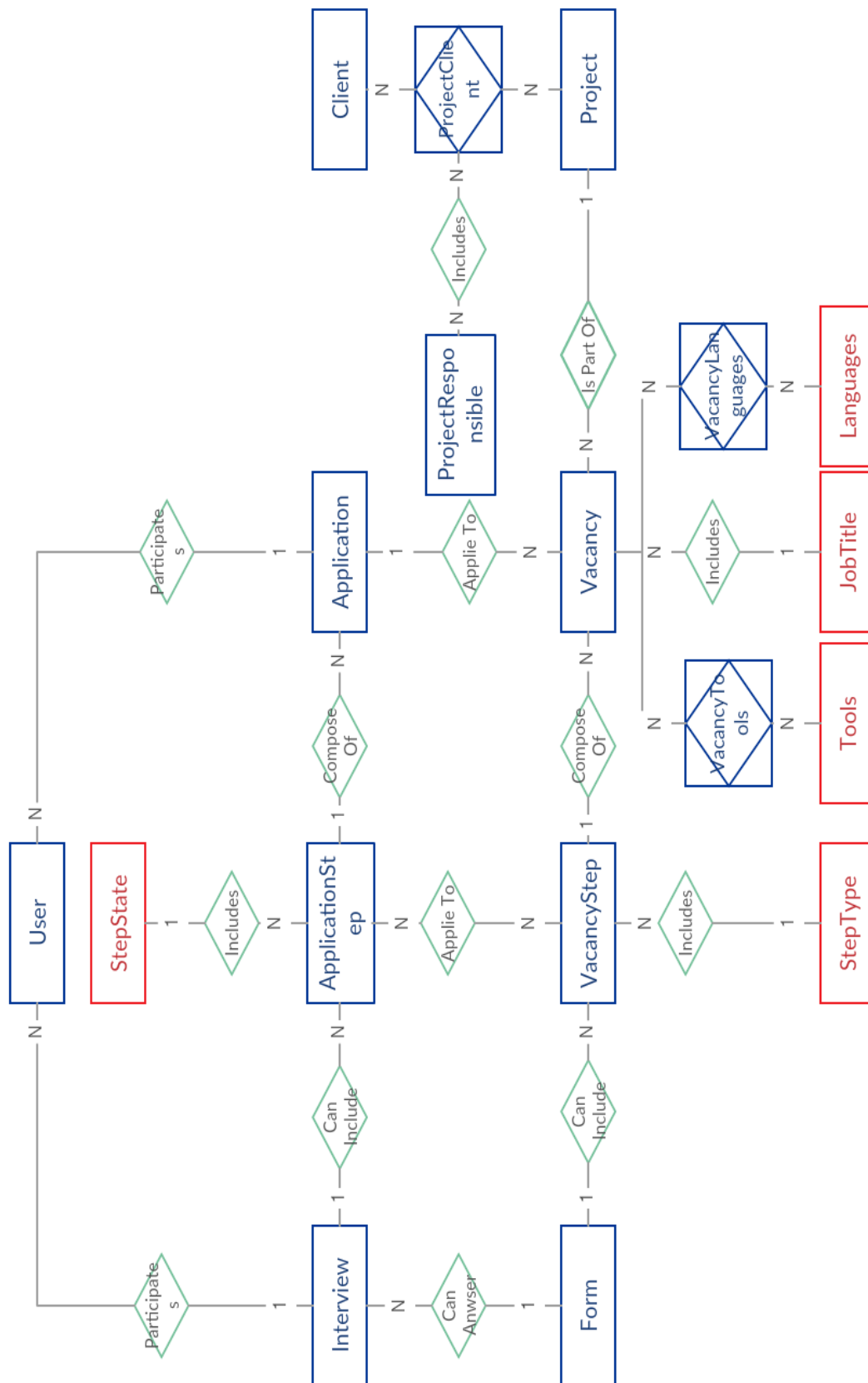
Este mesmo custo adicional é a razão da existência da divisão entre `SpontaneousCurriculum` e `SpontaneousCurriculumFile`, sendo que o segundo incluí um ficheiro pdf de tamanho variável.

Ao contrario das outras entidades que adicionam informação, esta é de 1 por 1 (um `User` só pode ser referido uma vez por `CandidateCurriculum`), e a aplicação impõe esta relação como obrigatória,

A inclusão da referencia a `Languages` pelo `CandidateCurriculum`, representa a linguagem nativa do utilizador.

A entidade Capacity não só é utilizada para estabelecer a capacidade de candidatos com varias vertentes, como também para estabelecer informações desses mesmos níveis, que possam estabelecer os candidatos da escolha do nível.

Um candidato pode ter mais que uma CandidateAvailability para o mesmo dia da semana.



Modelos De Dados 2 – Vagas

No modelo 2 é possível verificar as seguintes entidades:

- Client, nesta entidade é possível incluir informações dos clientes da PS Tec, que podem iniciar projetos com a PS Tec,
- Project, projetos a serem realizados pela PS Tec, por estes projetos é possível estabelecer vagas para candidatos,
- ProjectClient, com esta entidade é possível associar vários clientes a um projeto, e claro um cliente a vários projetos,
- ProjectResponsible, cada cliente 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 instancias desta entidade representam tal responsável e por isso são associados diretamente a ProjectClient,
- JobTitle, um cargo geral e regular na área da tecnologia,
 - Nota: inclui desenvolvedor, analisador, consultor, designer e manager.
- Vacancy, as vagas a preencher em si, podem ou não ser associadas a projetos e por isso podem ser utilizadas para representar vagas internas ou para projeto para clientes,
- VacancyTool, com esta entidade é possível associar varias ferramentas reconhecidas pela aplicação a uma vaga, ferramentas cujo conhecimento pode ser visto como essencial para preencher a vaga,
- VacancyLanguage, cada instancia 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 instancias desta entidade,
- StepType, tipo de passo do processo de entrevista, serve principalmente para saber como a aplicação deve ser tratada no passo em que ocorre, existem cinco possíveis passos:
 - Consideration, passo inicial de qualquer processo (existe sempre por omissão), demonstra que o candidato esta a ser considerado para o processo,
 - General Interview, uma entrevista geral para estabelecer as motivações do candidato. Apesar desta poder aparecer em vários passos a aplicação considera entrevistas gerais que já ocorreram no passado para preencher passos futuros, assim um candidato só terá de realizar uma entrevista geral,

- Technical Interview, entrevista técnica em que o candidato devera demonstra a sua capacidade com as tecnologias importantes para a vaga a que aplica,
 - Orientation, orientação fornecida para entrevista a acontecer com o cliente,
 - Client Interview, entrevista com o cliente do projeto, a que a vaga esta associada,
 - Contract, passo final de qualquer processo (é incluindo por omissão), representa a contratação em si do candidato para preencher a vaga.
- 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,
 - Application, quando uma vaga é criada qualquer candidato da aplicação pode-se aplicar à mesma, tal aplicação a vaga é representada por uma instancia desta entidade,
 - ApplicationStep, como uma vaga, Vacancy, pode ser composta por vários passos, uma aplicação a uma vaga ocorrerá em vários passos, tais passos são representados por esta entidade,
 - StepState, esta entidade estabelece os três estados que uma ApplicationStep pode assumir, tais passos são:
 - ToSchedule, o evento associado ao passo ainda não foi marcado,
 - Scheduled, o passo já inclui um evento associado que ainda não foi realizado,
 - ToProcess, estado que representa que o passo já terminou e que no momento a situação do candidato esta a ser considerada, a aplicação pode ser terminada ou continuada (para o próximo passo),
 - Interview, uma entrevista que tenha ocorrido/ ainda para ocorrer com um utilizador, que refere. Possivelmente devido a um passo numa aplicação que também refere. Também refere um Form, o formulário utilizado da entrevista.

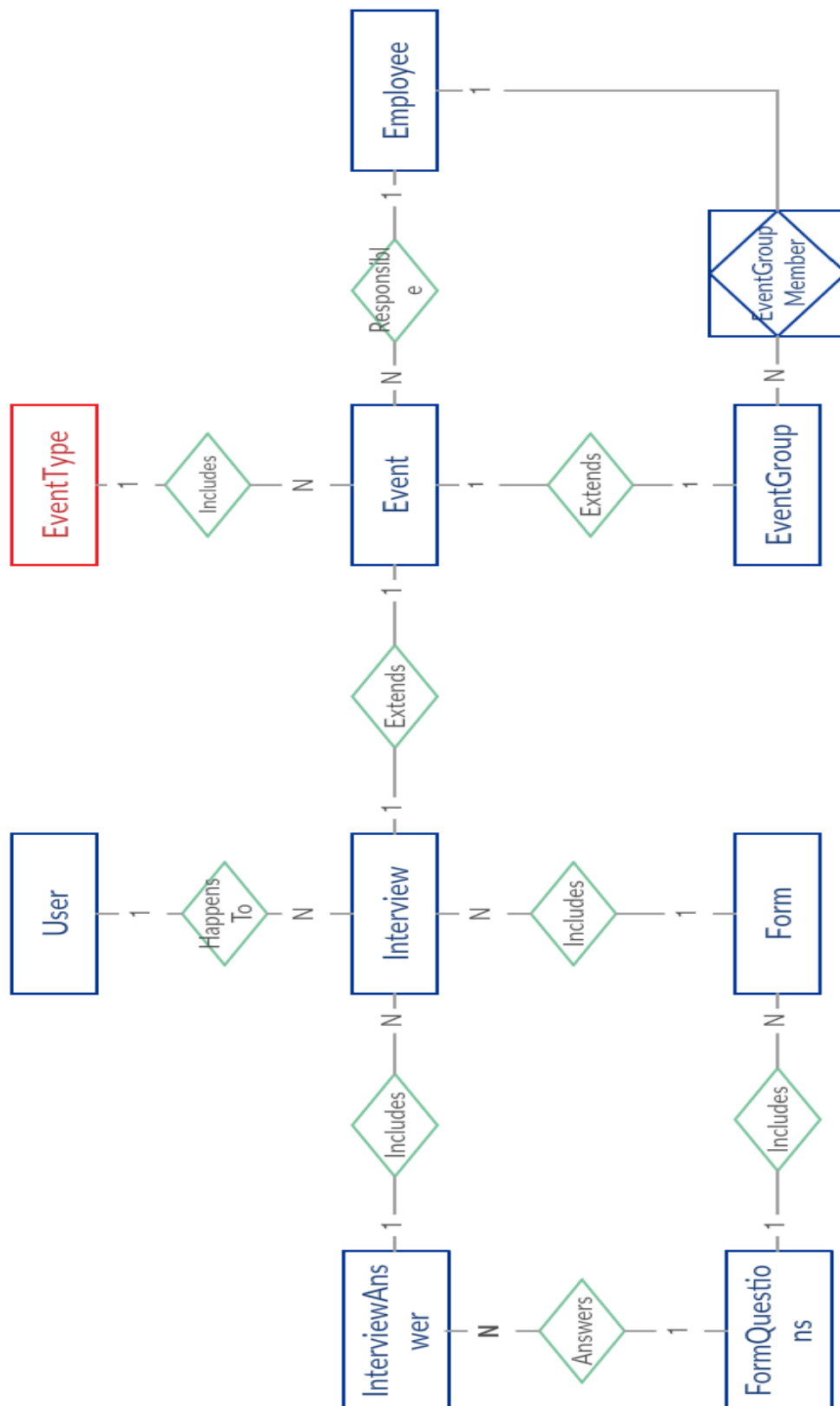
Como foi verificado é associado a Vacancy várias outras entidades de informação, JobTitle, VacancyTool e VacancyLanguage, estas entidades não só demonstram informações essenciais da vaga, capacidades necessárias, como também serve para pesquisas de vagas pelos candidatos.

Na pesquisa o cargo de trabalho, JobTitle, é o fator principal de pesquisa, seguido das ferramentas, VacancyTool, e por ultimo linguagens, VacancyLanguage. Por isso vagas são escolhidas em função do cargo, destas só as vagas com, pelo menos, uma ferramenta escolhida, são utilizadas e destas só as vagas com pelo menos uma linguagem escolhida.

Deve ser notado que a entidade StepType inclui informação geral do passo que é utilizada para esclarecer o candidato do que pode esperar no passo em que participa atualmente.

Algo a notar é que uma entrevista refere um candidato e é referida por uma ApplicationStep. Apesar da segunda poder tornar a primeira redundante (sendo que só uma Interview é referida por uma ApplicationStep, a mesma refere uma Application que em si refere um User), a primeira é necessária.

Tal deve-se ao facto que uma ApplicationStep poder ser apagada da base de dados (pela sua destruição própria, pela destruição da Application ou pela destruição da Vacancy), mas uma Interview não deve ser, assim a mesma pode ser utilizada para outros processos de entrevista. Esta situação também ocorre com a referência ao Form, sendo que pela ApplicationStep é possível obter a VacancyStep e por essa o Form



Modelos De Dados 3 - Eventos

No modelo anterior é possível averiguar as seguintes entidades:

- **Event**, esta entidade representa os vários eventos que podem ocorrer que envolvam pelo menos um colaborador, que refere;

- EventType, por esta entidade, que é referida por Event, é possível estabelecer que tipo de evento, existe três tipos de eventos:
 - External, evento externos que envolve um só colaborador, mas que ocorre durante dia e horário do trabalho e por isso deve ser considerado;
 - Group, evento de grupo que envolve mais que um colaborador;
 - Interview, entrevista em si entre colaborar e candidato.
- 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 refere. Deve referir o formulário utilizado;
- FormQuestion, cada instancia 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 instancias desta entidade.

Deve ser notado que uma entrevista pode ser estabelecida sem um formulário, nesse caso a entrevista é informal.

Outro fator a notar é a existência de uma dependência algo que circular na entidade InterviewAnswer, sendo que a mesma refere uma Interview e uma FormQuestion, com as duas referindo Form.

Mas as duas referencias são necessárias para, por um lado, facilitar o agregamento da questão com a resposta (a referencia que InterviewAnswer tem com FormQuestion), e por outro lado, garantir que entrevistas não iniciadas (e por isso sem nenhuma InterviewAnswer associada) possam obter o seu formulário mais facilmente.