

**\*Título do Trabalho de Projecto  
Usando a Primeira Letra na Forma Maiúscula\***

|  |  |
| --- | --- |
|  | \*Fernando Pessoa\* |
|  | \*Ricardo Reis\* |

|  |  |
| --- | --- |
| Orientadores | \*Àlvaro de Campos\* |
|  | \*Alberto Caeiro, SoftCompany\* |

Relatório \*de progresso\* \*beta\* final\* realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2014/2015

\*Maio\* de 2015

**Instituto Superior de Engenharia de Lisboa**  
Licenciatura em Engenharia Informática e de Computadores

**\*Título do Trabalho de Projecto  
Usando a Primeira Letra na Forma Maiúscula\***

|  |  |
| --- | --- |
| \*75463\* | \*Fernando António Nogueira Pessoa\* |
| \*72453\* | \*Ricardo Manuel Augusto dos Santos Reis\* |
|  |  |

|  |  |
| --- | --- |
| Orientadores: | \*Álvaro José Silva Veiga de Campos\* |
|  | \*Alberto Joaquim Alves Caeiro, SoftCompany\* |

Relatório \*de progresso\* \*beta\* final\* realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2014/2015

\*Maio\* de 2015

# Resumo

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

Para a empresa a aplicação não só ira garantir uma administração simples de entrevistas, candidatos, projetos, cliente e vagas, como também uma forma de se introduzir para os candidatos, demonstrando o que mais valida nos mesmos e como estes mais facilmente se podem apresentar.

No outro lado ira oferecer ao candidato uma forma simples, interativa e apelativa de administra 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, existira uma aplicação Móvel que servira como um suporte para os candidatos.

Também durante o processo de desenvolvimento desta aplicação iremos colocar os nossos conhecimentos da arquitetura de desenvolvimento OutSystems em pratica.

Arquitetura cujo o conhecimento é cada vez mais requisitado no mercado informático e tecnológico Português, o que torna qualquer experiencia sobre a mesma indispensável para o futuro.

**Palavras-chave:** IView; OutSystems; Widgets; Colaboradores; Candidatos; Vagas; Entrevistas; Currículos;

# Abstract

Abstract text (1 page).

**Keywords:** sorted keyword list, delimited by ;.

**Índice**

[Resumo v](#_Toc512102614)

[Abstract vii](#_Toc512102615)

[Lista de Figuras xi](#_Toc512102616)

[Lista de Tabelas xiii](#_Toc512102617)

[1. Introdução 1](#_Toc512102618)

[1.1 Enquadramentos 1](#_Toc512102619)

[1.2 Objectivos 1](#_Toc512102620)

[1.3 Organização do documento 2](#_Toc512102621)

[2. Formulação do Problema 4](#_Toc512102622)

[2.1 Estado de arte 4](#_Toc512102623)

[2.2 Descrição do Projeto 7](#_Toc512102624)

[2.2.1 Componente Web 7](#_Toc512102625)

[2.2.2 Componente Móvel 8](#_Toc512102626)

[2.2.3 Requisitos Não Funcionais 8](#_Toc512102627)

[2.3 Descrição da plataforma, modelos de desenvolvimentos 8](#_Toc512102628)

[2.3.1 Code Generator 9](#_Toc512102629)

[2.3.2 Deployment Services 10](#_Toc512102630)

[2.3.3 Applications Services 10](#_Toc512102631)

[3 Solução Proposta - Grandes Ideias 17](#_Toc512102632)

[3.1 Layers da OutSytesms 17](#_Toc512102633)

[3.2 Base de dados 18](#_Toc512102635)

[3.2.1 Utilizadores 19](#_Toc512102636)

[3.2.2 Vagas 22](#_Toc512102637)

[3.2.3 Eventos 25](#_Toc512102638)

[3.3 Wireframes 26](#_Toc512102639)

[4 Avaliação Experimental 47](#_Toc512102640)

[4.1 Nome da primeira secção deste capítulo 47](#_Toc512102641)

[4.2 A segunda secção deste capítulo 47](#_Toc512102642)

[4.2.1 A primeira sub-secção desta secção 47](#_Toc512102643)

[4.2.2 A segunda sub-secção desta secção 47](#_Toc512102644)

[4.3 Análise de resultados 47](#_Toc512102645)

[5 Conclusões 49](#_Toc512102646)

[Referências 50](#_Toc512102647)

[A.1 Diagramas da Aplicação 51](#_Toc512102648)

[A.2 Modelos de dados 53](#_Toc512102649)

# 

# Lista de Figuras

[Figura 1 - Arquitectura OutSystems [5] 8](#_Toc512108127)

[Figura 2 - Componentes OutSystems, Code Generator [5] 9](#_Toc512108128)

[Figura 3 - Arquitectura OutSystems, Deployment Services [5] 10](#_Toc512108129)

[Figura 4 - Arquitetura OutSystems, Scheduler Service [5] 11](#_Toc512108130)

[Figura 5 - Arquitetura OutSystems, Log Service [5] 11](#_Toc512108131)

[Figura 6 - Estrutura 4 Layer Canvas 17](#_Toc512108132)

[Figura 7 - Aplicação da 4 Layers Canvas sobre a IView 18](#_Toc512108133)

[Figura 8 - Modelo EA, Utilizadoes 19](file:///C:\Users\Diogo\Desktop\LEIC\PS\relatorio\Relatorio_WORD_2010%20v0.1.docx#_Toc512108134)

[Figura 9 - Modelo EA, Vagas 22](file:///C:\Users\Diogo\Desktop\LEIC\PS\relatorio\Relatorio_WORD_2010%20v0.1.docx#_Toc512108135)

[Figura 10 - Modelo EA, Eventos 25](file:///C:\Users\Diogo\Desktop\LEIC\PS\relatorio\Relatorio_WORD_2010%20v0.1.docx#_Toc512108136)

[Figura 11 – Frames, Home Page 27](#_Toc512108137)

[Figura 12 - Frames, Header 27](#_Toc512108138)

[Figura 13 - Frame, MoreInfo 31](#_Toc512108139)

[Figura 14 - Frame, MyCurriculum 33](#_Toc512108140)

[Figura 15 - Frames, MyProfile 35](#_Toc512108141)

[Figura 16 - Frames, Forms 35](#_Toc512108142)

[Figura 17 - Frames, Applications 37](#_Toc512108143)

[Figura 18 - Frames, Vacancies 38](#_Toc512108144)

[Figura 19 - Frames, CheckCV 39](#_Toc512108145)

[Figura 20 - Frames, Events 40](#_Toc512108146)

[Figura 21 - Frames, Candidate 40](#_Toc512108147)

[Figura 22 - Frames, Event 41](#_Toc512108148)

[Figura 23 - Frames, Application 42](#_Toc512108149)

[Figura 24 - Frames, Vacancy 44](#_Toc512108150)

[Figura 25 - Frames, FormEditAdd 44](#_Toc512108151)

[Figura 26 - Frames VacancyAdd 46](#_Toc512108152)

# Lista de Tabelas

[Tabela 1 -Um exemplo de legenda de tabela. Prazos de entrega de Projecto e Seminário, **Erro! Marcador não definido.**](#_Toc416101908)

# Introdução

## Enquadramentos

Muitas empresas tecnológicas portuguesas tem um processo de entrevista bastante arcaico e limitado, não colocando os seus conhecimentos tecnológicos em utilização.

Normalmente este processo envolvem qualquer transferência de ficheiros importantes e algo que sensíveis, como (normalmente, mas não limitados) a currículos e dossiers de competência[[1]](#footnote-1), por correio eletrónico.

Neste processo qualquer marcação de entrevista é realizada por telemóvel, o que em si não é um problema notável, com a única informação estabelecida sobre a entrevista sendo disponível também por transferências por correio eletrónico.

O que garante uma dependência notável pela parte do candidato ao colaborador, se o segundo se esquecer ou não informa de forma correta o candidato o mesmo poderá ter dificuldades para participar na entrevista.

Uma das raras exceções que não se limita a este processo, seria a consultoria Mind Source, que fornece, o que chamam, “Portal de Emprego”[1]. Este portal disponibiliza a capacidade de guardar ficheiros e realizar candidaturas a cargos disponíveis, mas pouco mais.

Também existe a aplicação “My Profile” [2] desenvolvido pela empresa de recursos humanos Randstad. Esta aplicação disponibiliza um formato simples e interativo de desenvolver o currículo e dossier de capacidades, mas nada mais.

Atendendo as limitações consideradas, nasceu o projeto a realizar, a aplicação IView, que tem como objetivo expandir as funcionalidades das duas aplicações referidas, como também incluir funcionalidades para a empresa que a utilize.

No desenvolvimento da aplicação será utilizado como base uma empresa de tecnologia fictícia a que chamamos PS Tec, servindo de enquadramento para outras possíveis empresas.

Atendendo as limitações consideradas, nasceu o projeto a realizar, a aplicação IView, que tem como objetivo expandir as funcionalidades do Portal de Emprego da Mind Source.

No desenvolvimento da aplicação será utilizado como base uma empresa de tecnologia fictícia a que chamamos PS Tec, servindo de enquadramento para outras possíveis empresas.

## Objectivos

O projeto de IView foca-se no desenvolvimento de duas componentes, uma web e uma aplicação móvel, que trabalham em conjunto para garantir uma comunicação entre candidato e entrevistador de uma forma simples e acessível.

Comunicação que poderá ter inicio no momento em que uma nova vaga para um cargo na empresa é criada, até a momento final da contratação.

A componente *Web*, componente central, onde serão desenvolvidas grande parte das funcionalidades, irá garantir acessos de leitura e escrita à base de dados onde será guardado, entre outros dados:

* Currículos, dossiês de competências e disponibilidades[[2]](#footnote-2) 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,

Já a aplicação móvel servirá como apoio e irá incluir alguns dos acessos da componente *Web*.

## 1.3 Organização do documento

O restante relatório tem a seguinte composição de capítulos, secções e subsecções:

* Introdução – introdução do relatório,
  + Enquadramentos – uma introdução a ideia do projeto, como também algumas inspirações pessoais,
  + Objetivos – os objetivos gerais e principais a desenvolver no projeto.
* Formulação do Problema – alguma informação sobre da aplicação IView,
  + Descrição do Projeto – os componentes que compõem a aplicação e as suas funcionalidades,
    - Componente *Web* – o componente que correrá na internet,
    - Componente Móvel - o componente que correrá em dispositivos moveis,
    - Requisitos Não Funcionais – alguns requisitos que serão considerados no desenvolvimento, mas que não serão garantidamente incluídos.
  + Descrição da plataforma, modelos de desenvolvimentos – estabelecimento de algumas características essenciais da OutSytems, plataforma de desenvolvimento do projeto, incluindo alguns componentes essenciais desta arquitetura,
    - Code Generator – componente responsável por gerir código,
    - Deployment Services - componente responsável por implementar as aplicações geradas num servidor de aplicações,
    - Applications Services – responsáveis por administrar a execução de *tasks* agendadas em lote e fornecem serviços de registro assíncrono para armazenar eventos como erros, auditorias e métricas de desempenho,
    - Três Layers de Desenvolvimento – os três layers da aplicação OutSystems Studio que permite o desenvolvimento das aplicações
    - Web LifeCycle – ciclo de vida das paginas web desenvolvidas pela OutSytems,
    - Mobile LifeCycle– ciclo de vida de ecrãs moveis desenvolvidas pela OutSytems,
    - Sincronismo– o problema de sincronização de dados locais e da base de dados existentes nas aplicações móveis como também os padrões sincronismo existentes para resolver tal problema.
* Solução Proposta - Grandes Ideias – como a aplicação é estruturada e executa as funcionalidades e requisitos já estabelecidos.
  + Layers da OutSytesms – breve descrição da 4 Layers Canvas, estrutura da OutSystems, e como esta é aplicada a IVIew,
  + Base de dados – descrição e demonstração das entidades que compõem a base de dados da aplicação dividida em três partes, para simplificar a apresentação,
    - Utilizadores – entidades sobre os utilizadores da aplicação, incluindo informação de candidatos e colaboradores,
    - Vagas – entidades sobre vagas, aplicações e as estruturas das duas,
    - Eventos – entidades sobre os eventos e entrevistas.
  + Wireframes – alguns exemplos de páginas, demonstrando a estrutura regular e a utilização das widgets, incluindo as razoes da utilização.

# Formulação do Problema

Estamos no início do novo capítulo. Aqui podemos colocar algum texto introdutório e de resumo do conteúdo do capítulo. Por exemplo, a secção 2.1 trata aspetos referentes às citações de bibliografia. Na secção 2.2 apresenta-se um exemplo de enumeração de conteúdos. O uso de tabelas é exemplificado na secção 2.3. Nas secções 2.4 e 2.5 abordam-se expressões matemáticas e o uso de figuras de grandes dimensões.

# Estado de arte

Em si um considerável numero de empresas, pelo menos as da área das tecnologias, utilizam uma aplicação semelhante a IView. Semelhante, mas com um notável numero de limitações que a IView não inclui.

A mais notável é a interação entre candidato e colaborador, uma comunicação que não existe nas aplicações referidas. Estas aplicações são, normalmente, para a empresa gerir candidatos, os seus currículos e a partilha dos mesmos com clientes.

Ao não incluir qualquer tipo de comunicação entre as duas partes, adiciona-se dificuldade ao candidato de realizar qualquer alteração ao seu currículo e dossiê de capacidades.

Sendo que depois da alteração é necessário partilhar ambos com um colaborador por email, o que não só não garante que os ficheiros atualizados sejam utilizados em próximas partilhas com clientes, como não garante que estes ficheiros sejam considerados em qualquer situação.

Continuando da limitação anterior, estas aplicações funcionam unicamente com dossiês de capacidades em formato PDF. Assim não só dificulta a atualização do dossiê, como não é incluindo qualquer mecanismo que filtra informação desnecessária e/ou repetida e não indica que tipos de tecnologias a empresa procura, ou pelo menos considera importantes.

Regularmente as bases do dossiê, não expressam certa informação que pode ser importante para o candidato preencher o mesmo. Dois exemplos simples seriam:

* Como um candidato demonstra que tem experiencia com uma tecnologia que não chega a um ano. Ou mesmo quando está entre um ano e o ano seguinte, deve-se arredondar para cima ou para baixo?

Regularmente é requisitado que o candidato expresse a sua capacidade em níveis, mas não é estabelecido o que cada nível implica.Mesmo que esta falta de informação não pareça importante, deve-se considerar que um candidato pode ser alguém que esteja a entrar agora no mercado de trabalho e ainda não saiba como tratar esta situação.

Estas aplicações também não incluem qualquer tipo de funcionalidade de marcação de entrevistas, com algumas utilizando o Google Calendar. As que não utilizam qualquer mecanismo permitem que qualquer falha de comunicação entre as duas partes interessadas leve a que a entrevista não ocorra, ou que não ocorra como desejado pelas duas partes.

Os que utilizam o Google Calendar conseguem ultrapassar esta limitação, mas o candidato fica dependente, por completo, de receber um convite, que pode causar problemas.

Um exemplo que ocorreu a um dos alunos a realizar este projeto, Diogo Aires, demonstra este problema. Sendo que tinha sido estabelecido uma entrevista, mas o convite nunca tinha chegado ao próprio, só depois de conseguir entrar em contacto com o colaborador com quem tinha sido estabelecido a entrevista, foi encontrado o erro, o convite tinha sido enviado para o email de outro individuo.

Durante o processo de investigação foram encontradas duas aplicações que permitem a comunicação entre candidato e colaborador.

A primeira é a anteriormente referida “Portal de Emprego” [1] estabelecido pela Mind Source. Esta permitia aos candidatos duas funcionalidades principais:

* Gerir ficheiros, currículos e dossiês, assim quebrando a necessidade de comunicar com um colaborador cada vez que se deseja realizar uma atualização de um dos ficheiros
* Verificar vagas disponíveis e aplicar-se às mesmas

Mas estas funcionalidades têm limitações importantes, num lado não indica que ficheiros o candidato deve incluir e qual 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

Outra aplicação é uma aplicação de gerir perfil produzida pela Randstad [2]. Esta aplicação não inclui qualquer capacidade de verificar vagas e aplicações, estando limitado unicamente a construir e gerir o perfil do candidato.

Esta aplicação permite um desenvolvimento do currículo e dossiê de capacidades muito mais fácil e de fazer alterações sobre o mesmo.

Mas sendo que Randstad não é uma empresa na área das tecnologias, por isso o perfil inclui um extenso raio de informações que torna a formação do perfil um pouco mais complexo.

Uma vertente do dossiê de capacidades para a área da informática que a Randstad não inclui na sua aplicação é a lista de aplicações desenvolvidas. Uma vertente importante que demonstra numa forma prática alguma experiencia pela parte do candidato e que tipo de responsabilidades tem habito tomar.

Como referido anteriormente esta aplicação não inclui qualquer tipo de informação sobre vagas e aplicações o que torna o iniciar o processo de entrevista algo só possível pela parte da empresa ou utilizando outras fontes para descobrir as vagas.

Apesar destas limitações a aplicação da Randstad foi parte da inspiração gráfica para o projeto, sendo que, apesar de não o podermos provar, acreditamos que esta aplicação foi produzida utilizando a OutSystems.

Existem ainda duas outras aplicações que estão algo relacionadas com a IView, mas ao contrário das duas que já referimos estas não são limitadas a uma só empresa e servem mais para partilhar informação sobre perfis e vagas na área.

Primeiro temos a rede social LinkedIn [3], a mesma permite gerir um perfil, semelhante a aplicação da Randstad, mas numa forma menos interativa. Fora disso a 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 perfil para vairas empresas.

Quase num completo oposto, temos a aplicação ITJobs [4], que em si fornece uma extensa fonte de informação sobre empregos, formações e eventos na área de tecnologias e informática.

Mas não inclui qualquer possibilidade de gerir perfis, sendo que nem permite o estabelecimento de um utilizador registrado, e qualquer processo de entrevista e suas informações são feitas pela empresa da vaga e não são acessíveis pela ITJobs.

Ao inicio desta secção foi estabelecido que aplicações semelhantes a IView, mesmo que limitadas, são bastante comuns em empresas na área de informática e tecnologias.

Tal facto demonstra como uma aplicação do tipo IView pode ser verificada como útil e desejada para a esta área.

Por exemplo, para este exemplo utilizaremos de novo a experiencia pessoal de um dos autores do projeto, Diogo Aires, em algumas entrevistas na área o autor referiu e estabeleceu a aplicação com o entrevistador.

Foi a partir destas entrevistas que dois fatores foram estabelecidos:

* Um extenso numero de empresas da área de informática e tecnologias incluem o tipo de aplicação da IView,
* Normalmente a possibilidade de candidatos conseguirem interagir na aplicação é uma funcionalidade que não foi considera na criação da aplicação, mas que a mesma era desejável,
  + Normalmente entrevistadores referem que no futuro a empresa para que trabalham deverá iniciar um projeto interno para estabelecer esta funcionalidade,

# Descrição do Projeto

A aplicação IView está dividida em dois componentes principais, tendo cada componente um conjunto individual de requisitos funcionais, de acordo com o seu objetivo específico.

## Componente Web

Esta vertente do projeto, foca-se na disponibilização de diferentes funcionalidades aos dois elementos participantes na aplicação: candidatos e empresa.

INCLUIR AQUI OS CASES - INICIO

Também incluirá, 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.

Os colaboradores da empresa, que participem no processo de entrevista, serão 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 e os respetivos estados das candidaturas;
* Verificar candidaturas a novos cargos, permitindo promover um candidato para o próximo passo do processo ou terminar o mesmo, sendo o candidato informado de tal automaticamente, por email;
* Marcar novas entrevistas e verificar entrevistas já marcadas;
* Guardar entrevistas, cuja informação poderá ser utilizada em futuras considerações.

Candidatos a novos cargos na empresa serão capazes de:

* Gerir o seu currículo e dossiê de capacidades, com limitações e indicações estabelecidas pela empresa;
  + O dossiê poderá incluir informação como: experiencia 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

INCLUIR AQUI OS CASES - FIM

## Componente Móvel

A aplicação móvel tem como objetivo ser uma fonte de informação conveniente e *lightweight* para Candidatos que utilizam IView, para este fim foram mantidas as funcionalidades determinadas estritamente necessárias e nada mais.

Neste componente os candidatos serão 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 respetivas informações.

A falta de peso deve-se, em parte, devido 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 restinguidos a informações acessíveis unicamente a um candidato, a aplicação desenvolvida pode tomar partidos dos padrões de sincronização mais simples.

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

## 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

## Descrição da plataforma, modelos de desenvolvimentos

Para o desenvolvimento das duas aplicações será utilizada a plataforma de desenvolvimento OutSystems, cuja arquitetura encontra-se ilustrada na Figura 1.

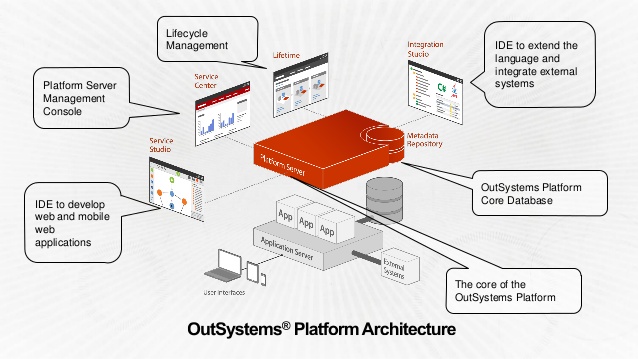


Figura - Arquitetura OutSystems [5]

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 componente *web* quer na componente *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, que iremos verificar nas próximas subsecções.

## Code Generator

Este componente é responsável por ler a interface gráfica utilizada para desenvolver a aplicação e a partir da mesma e desenvolve os componentes nativos para correr no server da aplicação. Uma simples representação deste componente pode ser verificada na Figura 2.

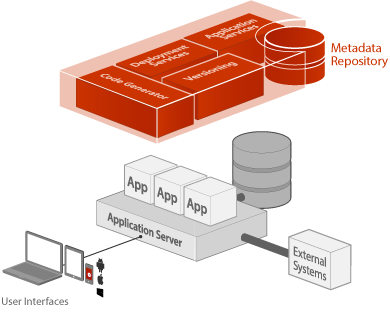


Figura - Componentes OutSystems, Code Generator [5]

Este componente tem a responsabilidade de:

* executar uma análise profunda da interface gráfica, cuja informação é utilizada para verificar dependências externas e aplicar otimizações a aplicação,
* verificar otimizações possíveis sobre o código,
  + Por exemplo, se uma aplicação incluir uma consulta a base de dados, mas o resultado nunca é utilizado, então a mesma não é incluída no código final,
* gerar *proxies* de integração para isolar uma aplicação de alterações realizadas sobre serviços externos,
* gerar código nativo para todas as camadas na pilha de aplicativos: do SQL para interagir com o base de dados e os serviços de lógica de negócios de *back-end* para os componentes JavaScript no *front-end*,
* agrupar as capacidades de gerir de sessão, autenticação e configurações para a aplicação,
* adicionar recursos de registro e monitoramento a aplicação,
* gerir os scripts de construção para atualizar o modelo de base de dados e implementar a aplicação.

## Deployment Services

Neste componente da arquitetura da OutSystems, 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.

A Figura 3, que se segue, demonstra uma simples representação deste componente.

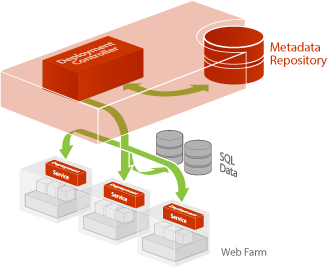


Figura 3 - Arquitetura OutSystems, Deployment Services [5]

Existem dois elementos de implementação principais neste componente:

* O serviço de implantação, que implanta a aplicação .NET num servidor *front-end* específico,
* O controlador de implantação, que coordena a implantação em todos os servidores *front-end*.

## Applications Services

A plataforma de desenvolvimento da OutSystems fornece servidores que administram as aplicações em tempo de execução.

Para tal inclui dois servidores separados:

* O **Scheduler Service, que administra a execução de *threads* planeadas,**
  + - Este servidor pode ser verificado na Figura 4,
  + O Log Service, responsável por gerir erros, auditorias e o desempenho de uma aplicação,
    - A Figura 5 demonstra uma representação deste servidor.

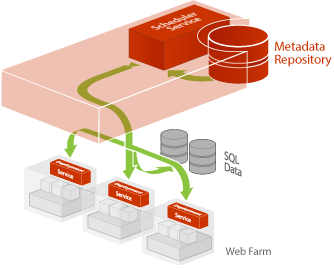


Figura 4 - Arquitetura OutSystems, Scheduler Service [5]

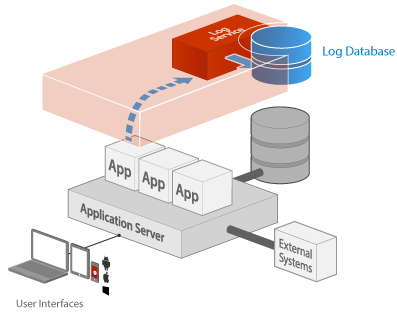


Figura - Arquitetura OutSystems, Log Service [5]

A plataforma OutSystems, pelo Scbeduler Service, suporta a execução de tarefas em lote agendadas nas aplicações. Em tempo de execução, o serviço de agendamento recupera o próximo trabalho pendente da base de dados de *metadados* e executa-o.

As tarefas em lote são projetadas para escalabilidade horizontal, de modo que são distribuídas em todos os servidores *front-end*.

No Log Service, as aplicações são instrumentadas automaticamente para criar *logs* de erro, auditoria e desempenho.

Em tempo de execução, essas métricas são capturadas de forma assíncrona, para que o desempenho das aplicações não seja comprometido enquanto eles criam um fluxo constante de métricas.

## Três Layers de Desenvolvimento

A OutSystems Service Studio, onde é possível desenvolver tanto aplicações *web* como moveis, divide o desenvolvimento em três *layers*, cada uma dividida num conjunto de componentes.

Primeiro temos a Interface Layer, verificado na Figura 6 (móvel), que se centra na interface gráfica da aplicação e que é composto pelos seguintes componentes:

* UI Flows – onde é incluído tanto os ecrãs que compõem a interface gráfica da aplicação desenvolvida, como também padrões utilizados,
  + No desenvolvimento de aplicações *Web* uma página pode incluir ações logicas, como Preparation e Screen Action, enquanto no desenvolvimento de aplicações Móvel é possível incluir *Fetches* (sobre a Base de Dados, Dados Locais ou de outras fontes) e Client Action,
* 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,*

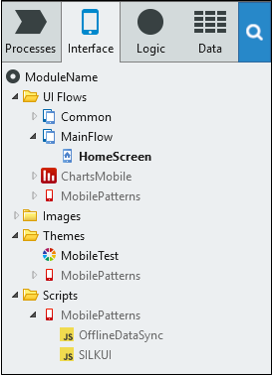


Figura - Interface Layer

Depois temos a Logic Layer, responsável pelo desenvolvimento de ações que podem correr durante o processo da aplicação. A Figura 7 demonstra esta *Layer* (móvel), onde se pode verificar que o *layer* inclui:

* Client Actions – ações que ocorrem no lado do cliente,
* Server Actions – ações que ocorrem no lado do servidor, existe unicamente no desenvolvimento móvel,
* Integrations – permite configurar integrações que a aplicação pode utilizar,
  + Permite integrações em servidores SOAP, REST e SAP,
* Roles – onde é possível incluir papeis de segurança,
  + Inclui sempre os papeis *Anonymous*, utilizadores não registrados, e *Registered*, utilizadores registrados,
* Exceptions – onde é incluído as exempções desenvolvidas para a aplicação.

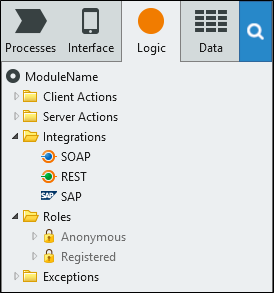


Figura - Logic Layer

Por ultimo temos a Data Layer, verificada na Figura 8Figura 1 (móvel), 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 informação,
  + Inclui Database, onde os dados *online* são guardados em estruturas SQL e no caso de aplicações moveis, Local Storage, dados *offline*.
* 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 móvel,
* Site Properties – propriedades do site,
* Resources – recursos disponíveis na aplicação.

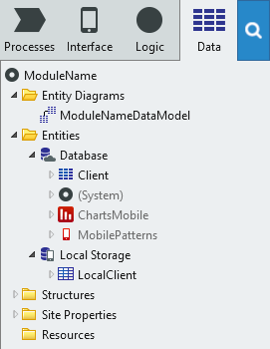


Figura – Data Layer

## Web LifeCycle

O ciclo de vida de uma página Web desenvolvida é normalmente composto por três passos, com opcional:

* Preparação – onde é obtido e gerido dados demonstrados na página, este passo é opcional sendo que se a informação da página for completamente estática (e não obtida da base de dados), então não é necessária preparação. Este passo ocorre completamente pela ação Preparation,
* Composição – a estrutura da página e os seus elementos HTML são geridos. Também é obtido 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 móvel poder ser visto como também divido nos três passos do ciclo de vida verificado na ultima subsecção, este ciclo é algo mais complexo, sendo que não é incluído 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,*
  + Pode ser utilizado para inicializar o ecrã, definindo valores *default*,
* OnReady – ação ocorre quando ecrã está pronto, mas antes do *render* de o mesmo começar,
  + Permite manipular o DOM do ecrã,
* OnRender – ação para correr quando o ecrã termina o seu *render*, ou cada vez que um elemento do ecrã muda,
  + Utilizado para atualizar componentes,
* OnDestroy – ação ocorre antes de um ecrã ser destruído,
  + Este *handler* permite aplicar lógica a destruição de ecrãs.

## Sincronismo

Desenvolvimento sobre a plataforma móvel 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 OutSytems fornece uma *framework* de sincronismo, alguns dos exemplos de sincronismo possíveis são:

* [Read-Only Data](https://success.outsystems.com/Documentation/10/Developing_an_Application/Use_Data/Offline/Offline_Data_Sync_Patterns/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](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, modo *offline*, mas que em principio, instancias individuais irão ser alteradas por um único utilizador, 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 móvel 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 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 a um candidato e por isso não existirá concorrência.

# Solução Proposta - Grandes Ideias

O foco deste capitulo será a proposta, em forma geral, do projeto gerado, referindo a estrutura da aplicação, que segue a arquitetura 4 Layers Canvas estabelecida pela OutSytems, a base de dados da aplicação, os *wireframes* que demonstram alguns exemplos de páginas que compõem o componente web e o por ultimo alguma lógica para cumprir certas funcionalidades importantes de notar.

## Layers da OutSytesms

## Como qualquer aplicação desenvolvida na arquitetura OutSystems,

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 demonstram uma breve representação e descrição desta estrutura, descrevendo cada uma das camadas.

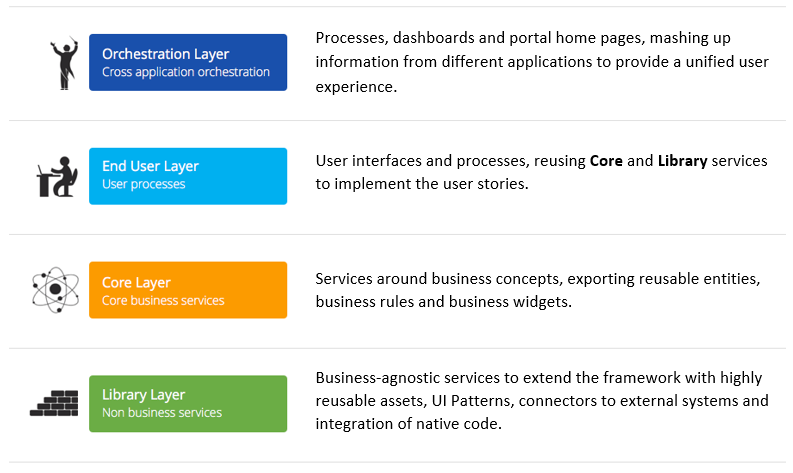


Figura - Estrutura 4 Layer Canvas

Em termos da arquitetura da IView os quatro *layers* serão:

* 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 móvel, o projeto utiliza a Web\_UI e App\_UI,
* Core Layer – esta *layer* será composta por quatro módulos:
  + IViewCore – onde é desenvolvido a base de dados e algumas estruturas partilhadas,
  + IViewWeb – funcionalidades de utilizadores não registados, colaboradores e candidatos da componente We,
  + IViewMobileCore – base de dados local,
  + IViewMobile – capacidade dos candidatos na mobile.
* Library Layer – para este projeto utilizamos como dependências:
  + Liverpool Template,
  + Silk UI,
  + FileSystem,
  + Html2PdfConverte.

A aplicação da 4 Layers Canvas sobre a aplicação pode ser verificada na Figura 10.

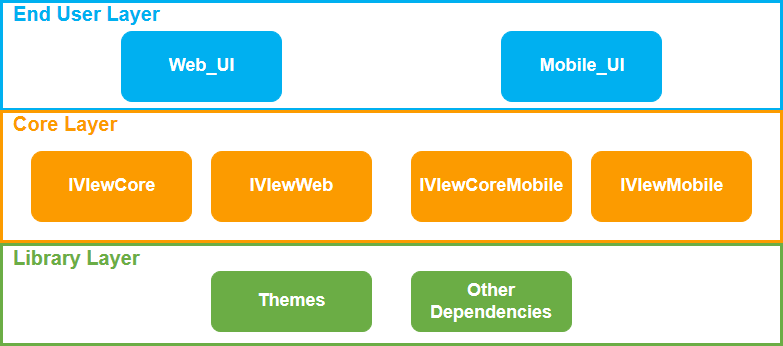


Figura - Aplicação da 4 Layers Canvas sobre a IView

No momento ainda estamos a considerar a possibilidade de dividir o módulo IViewWeb em três módulos mais pequenos, sendo os mesmos:

* IView – funcionalidades de utilizadores não registados, e partilhada entre outros módulos,
* IViewCandidates – capacidades dos candidatos,
* IViewEmployees – capacidades dos colaboradores.

## 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 é verificado as entidades que incluem informação dos dois utilizadores da aplicação, Candidato e Colaborador,
* Vagas, as entidades sobre as vagas, projetos e aplicações a vagas,
* Eventos, todas as entidades relacionadas com eventos e entrevistas.

Deve ser notado que o modelo é composto por tabelas dinâmicas e tabelas estáticas, com o segundo tipo de tabelas servindo para limitar valores importantes a decisões. Nas figuras a mostrar as tabelas dinâmicas são azuis, enquanto que as estáticas são vermelhas.

### 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.

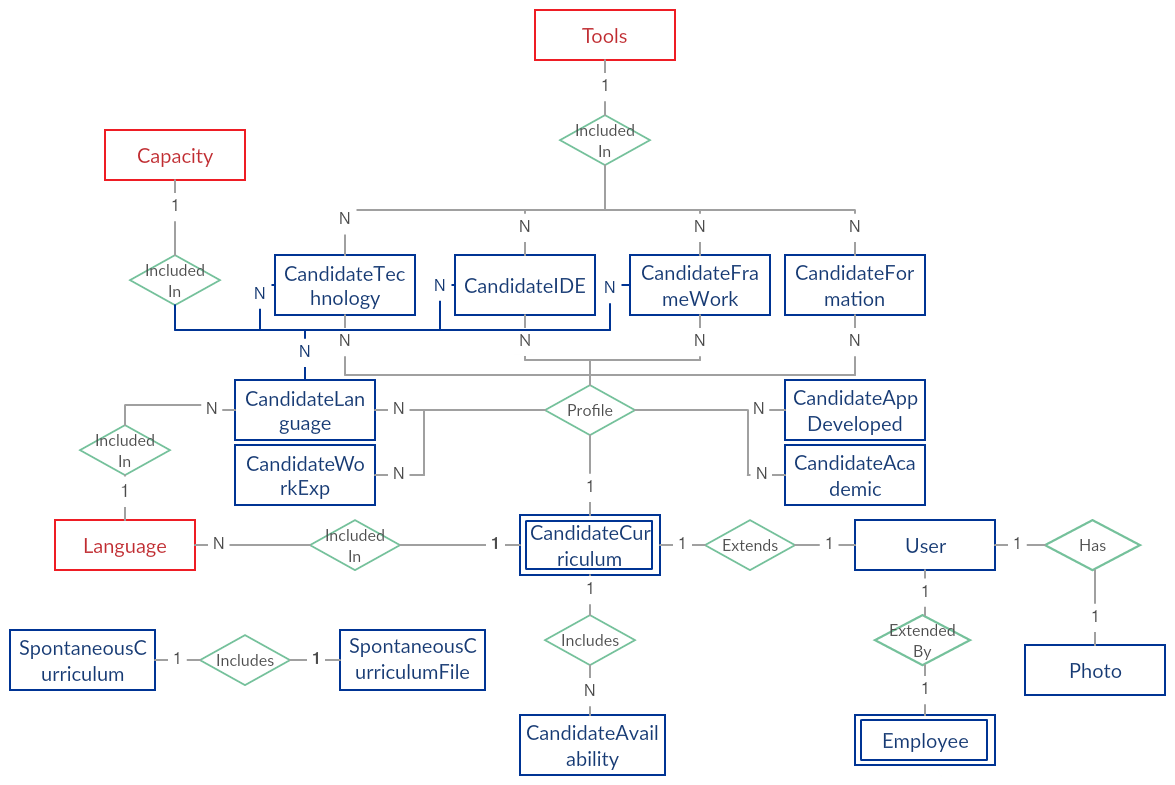
 A Figura 11 demonstra o modelo Entidade-Associação desta componente do modelo.

Figura - Modelo EA, Utilizadores

As entidades da Figura 11 são:

* User, uma entidade padrão da OutSytems, representa os utilizadores registrados na aplicação,
  + Nota: 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.
* 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 registrados 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,
  + Nota: 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,
  + Nota 2: inclui uma referencia a linguagem nativa do utilizador.
* CandidateAvailability, instancias desta entidade representam a disponibilidade que o candidato tem para entrevistas, associando a um dia de semana uma janela de tempo,
  + Nota: um candidato pode ter mais que uma CandidateAvailability para o mesmo dia da semana.
* 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.

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.

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.

### 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.

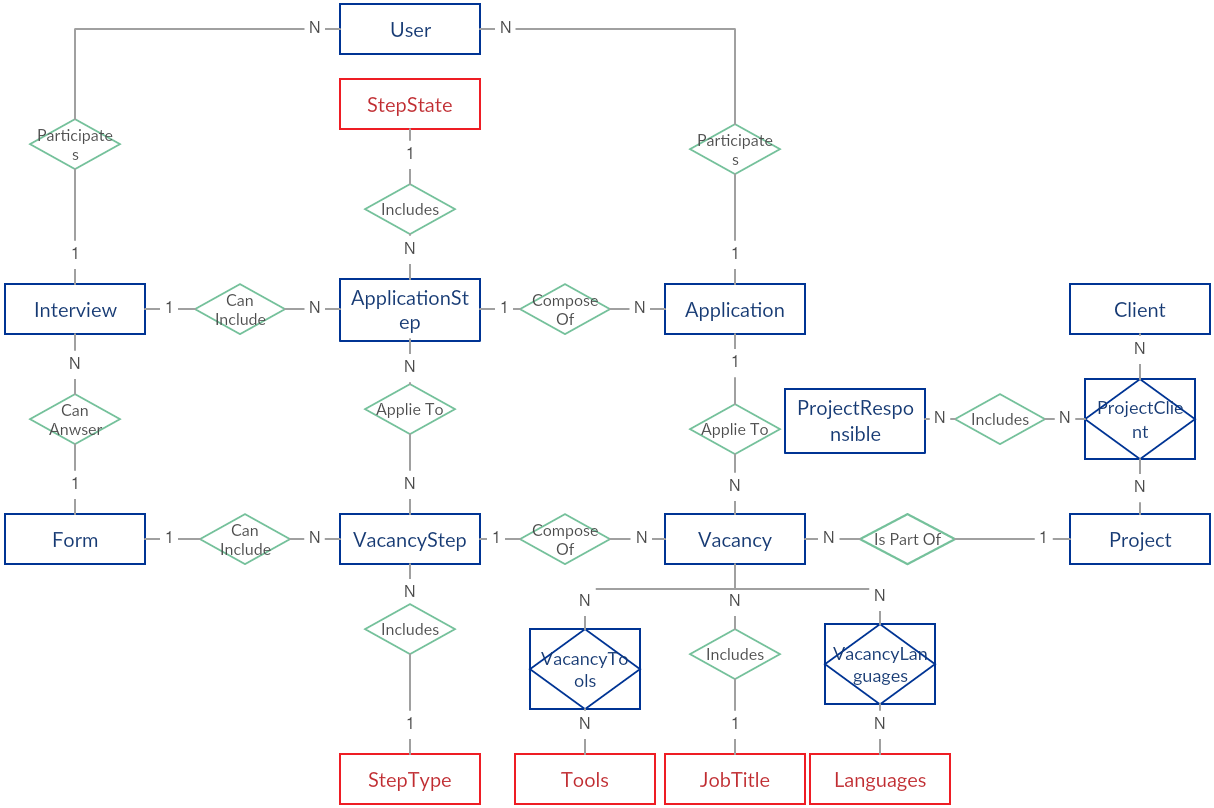
 Tais entidades podem ser verificadas na Figura 12 que se segue.

Figura - Modelo EA, Vagas

As entidades incluídas na Figura 12 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,
  + Nota: um projeto não tem de incluir um cliente, podendo ser completamente interno.
* 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,
  + Nota: cada vaga faz reverencia a uma instancia de JobTitle, para representar o cargo que a vaga procura ocupar.
* 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,
  + Nota: 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,
    - Technicial 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,
  + Nota: 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 duma 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 sé 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

### Eventos

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

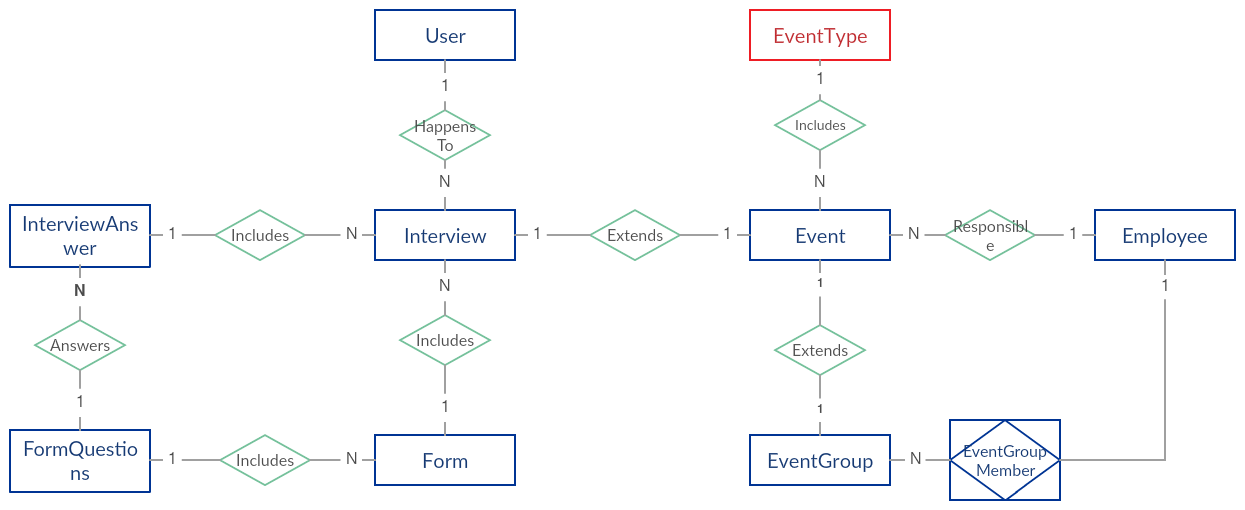
Estas entidades podem ser verificadas na Figura 13 que se segue.

Figura - Modelo EA, Eventos

Na Figura 13 pode ser verificado 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,
  + Nota: 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,
  + Nota: uma entrevista pode ser estabelecida sem um formulário, nesse caso a entrevista é informal.
* 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.

Outro fator a notar é a existência de uma dependência algo que circular na entidade InterviewAnwser, sendo que a mesma refere uma Inteview 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 InterviewAnwser tem com FormQuestion), e por outro lado, garantir que entrevistas não iniciadas (e por isso sem nehuma InterviewAnwser associada) possam obter o seu formulário mais facilmente.

## Wireframes

Para esta secção será verificado algumas *frames* para páginas importantes, com algumas sendo únicas na aplicação e outros exemplos que demonstram o formato habitual de várias páginas.

Também serão 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 da páginas, como se pode ver na Figura 14, onde é verificada a página inicial da aplicação, acompanhada pela Figura 15 onde é demonstrado os dois possíveis menus de utilizador.

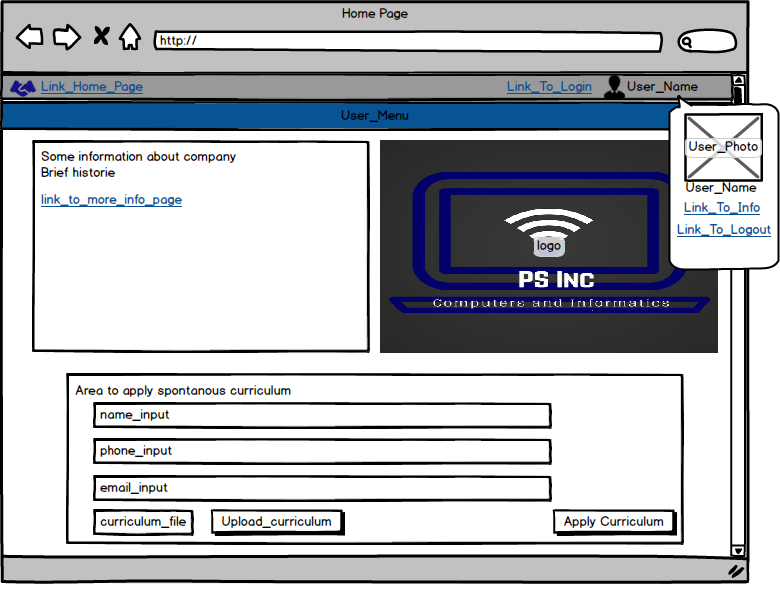


Figura – Frames, Home Page

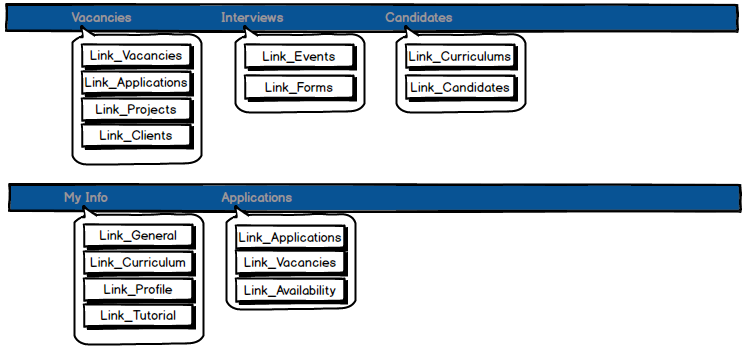


Figura - Frames, Header

Começando com o *header* o mesmo inclui sempre um *link* para a Home Page (Figura 14) a esquerda, mais a 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 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.

Originalmente os menus eram mais simples, mas também menus eficazes, sendo que nas primeiras versões eram compostos unicamente por botões que serviam de *links*.

Tal estrutura levantou um problema muito cedo e notável, o espaço ocupado por tantos botões. Usando os botões só para incluir os *links* já estabelecidos, cada menu ocuparia três vezes o espaço vertical atualmente ocupado pelo menu.

Em vez disso os menus, incluindo o menu do *header*, são incluídos num *rich widget* da OutSytems chamado Ballon.

Este *widget* permite esconder os *links*, até um elemento da página ser pressionado, sendo tal elemento o texto ou *container* onde o texto se encontra. Assim o menu existe, mas só aparece quando necessário.

Uma possibilidade considerada para estes menus foi a utilização do *rich widget* Accordion, que também permite fazer algo semelhante. Mas devido algumas limitações visuais, o Ballon foi escolhido sobre o Accordion.

O menu de colaboradores tem *links* para as seguintes páginas:

* Vacancies, vagas já estabelecidas, permite introduzir uma nova vaga,
* Applications, lista de aplicações de candidatos a vagas já estabelecidas,
* Projects, projetos iniciados, permite iniciar um novo projeto,
  + Nota: é possível criar vagas a partir de um projeto.
* Clients, clientes da PS Tec, permite adicionar um cliente,
  + Nota: é possível criar projetos a partir de um cliente.
* Events, demonstra os eventos em que o utilizador participa, permite introduzir um novo evento,
* Forms, lista de formulários existentes, permite adicionar e alterar formulários,
* Curriculums, demonstra as candidaturas espontâneas que ainda não foram recusadas ou aceitadas,
* Candidates, lista dos candidatos já existentes,
  + Nota: cada candidato demonstrado permite ver as informações gerais, o currículo, dossiê de capacidades, aplicações e até entrevista já realizadas.

Já o menu de candidatos tem links para as páginas:

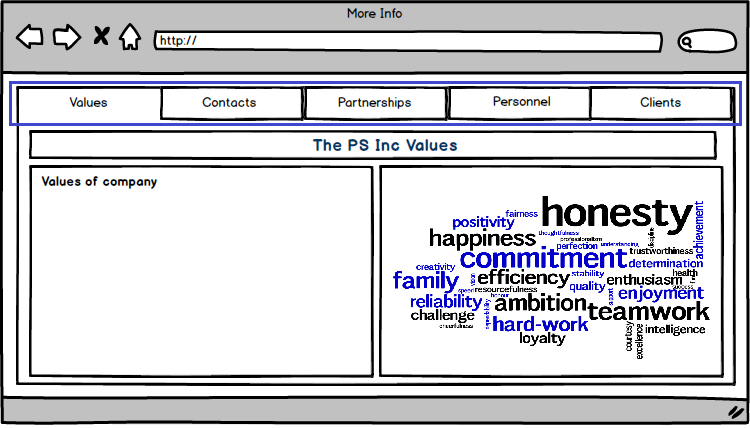
* General, informação geral do candidato,
* Curriculum, currículo do candidato, permite alterar o mesmo,
* Profile, dossiê de capacidades do candidato, permite alterar o mesmo,
* Tutorial, uma página de tutorial que explica algumas características do currículo e dossiê,
* Applications, aplicações a vagas existentes que o candidato esta aplicado,
* Vacancies, vagas disponíveis que o candidato ainda não se aplicou,
  + Nota: a página permite uma pesquisa inteligente de vagas.
* Availability, com esta página o candidato pode demonstrar a sua disponibilidade para entrevista.

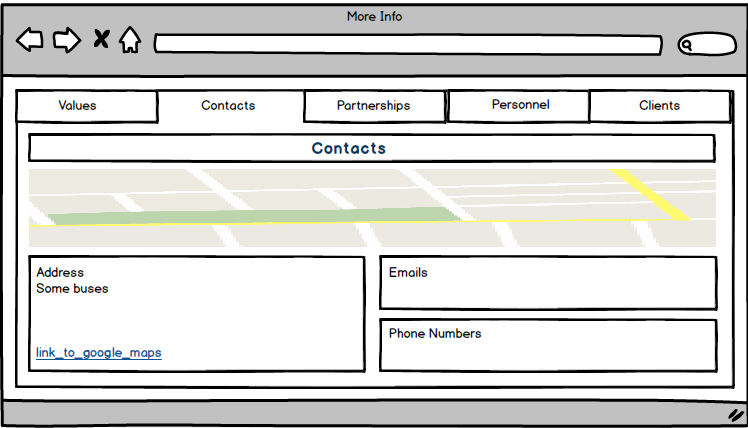
A páginapágina principal, como se pode verificar na Figura 14, inclui o logotipo da PS Tec, como também um simples resumo da mesma, onde é incluindo um *link* para página de informações.

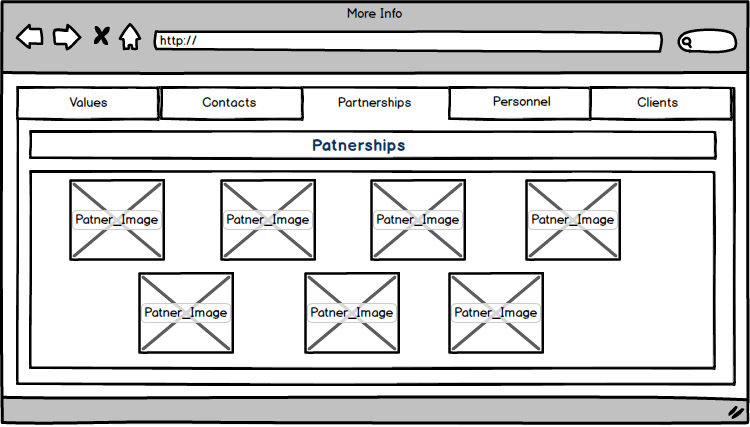
Também é nesta páginapágina onde é possível realizar a candidatura espontânea, pela utilização de três *inputs* (um texto, outro email e outro *phone*) onde o candidato pode introduzir algumas informações gerais, mais um *upload* *input* onde o candidato pode introduzir o seu currículo. Estes dados são todos fornecidos à aplicação por um só botão.

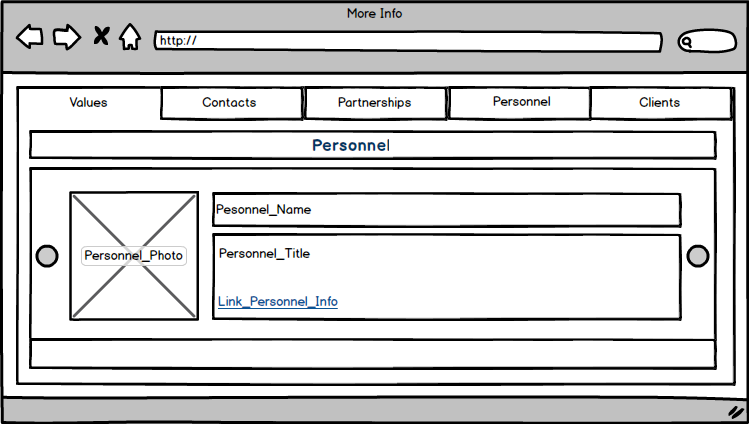
Na Figura 16 é possível verificar a página de informação geral da PS Tec que introduz dois elementos visuais fundamentais à aplicação, mais outro menos utilizável.

Note-se que a figura é composta por cinco elementos, mas todos são uma só página, com cada elemento demonstrando uma parte diferente da página.









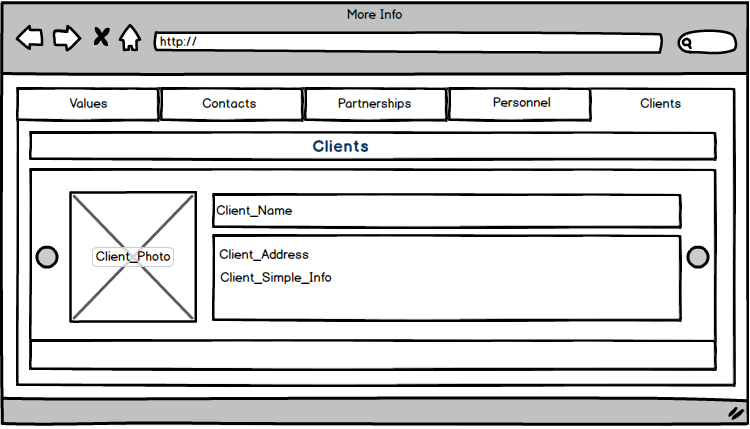


Figura - Frame, MoreInfo

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, mas especificamente no inicio da figura com borda azul à volta.

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 dividas em *tabs*.

Originalmente, antes de se descobrir a existência deste rich widget, foram consideradas duas possibilidades, ambas com limitações notáveis.

A primeira era incluir nas páginas com muita informação um menu extra, no topo da página, com vários botões que demonstram a parte da página desejada. Depois considerou-se passar esse menu para um ponto mais abaixo e usar *ballon* para esconder os botões.

Ambas possibilidades envolviam algum código extra da parte da aplicação, para esconder a parte das páginas que não se quer visualizar era necessário utilizar a característica Visible, associando a mesma a uma variável que era alterada com o pressionar dos botões já referidos.

Esta alteração seria sempre acompanhada por um *ajax refresh* que escondia as partes da página não desejadas e demonstra a parte da página desejada.

Então estas formas de visualização envolviam sempre incluir uma variável extra a página, como também uma Screen Action, elementos desnecessários com a utilização da 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 demonstra informação que possa ocupar mais que uma linha ou inclua pelo menos uma imagem.

No caso da página da Figura 16, esta *widget* é utilizada com a *rich widget* Caroussel para demonstrar os clientes e colaboradores num formato simples.

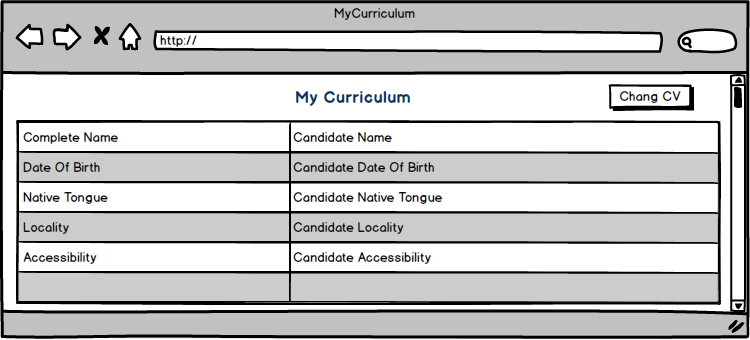
E cada um dos colaboradores e clientes demonstrados são representados por uma instancia do *rich widget* CardLeftImage, que permite demonstra uma figura a esquerda e informação geral a direita. Este *rich widget* é utilizado varias vezes, principalmente em listas de Users e clientes.

A partir deste ponto iremos verificar páginas não gerais mais especificas a candidatos ou colaboradores, começando com candidatos.

Estas páginas só podem ser acedidas por utilizadores que possuem um dos dois papeis já referidos, IViewCandidate e IViewEmployee. Esta restrição é estabelecida na página na sua característica de acessos por Roles.

Deve ser notado que outro Role que tem acesso a todas as páginas é o Role UserManager.

Na secção **Base de dados** foi estabelecido a existência de um currículo que um candidato pode verificar é alterar, estas duas capacidades são possíveis na página da Figura 17.



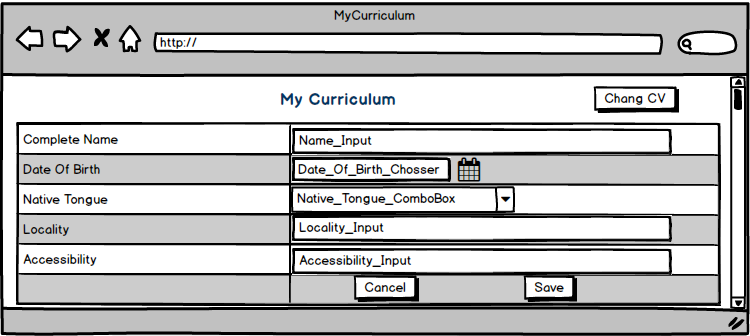


Figura - Frame, MyCurriculum

Como se pode notar a Figura 17 é composta por mais que um elemento, com o primeiro elemento demonstrando a página quando a mesma serve para só demonstrar o currículo e o segundo elemento quando a página demonstra como a alteração ao currículo é realizada.

Também deve ser notado que a frame só demonstra alguns elementos do currículo, não todos.

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 às vezes separados. O formato apesar de simples combina visualmente com qualquer página e por isso tem uma utilidade regular quando se quer demonstra ou introduzir informação extensa.

Nesta página são utilizados dois *inputs* que irão aparecer algumas vezes, primeiro é o Combo Box, verificado a frente de “Date Of Birth” no segundo elemento da Figura 17.

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 17, 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 widget* 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 **Base de dados**, um candidato consegue controlar o seu dossiê de capacidades que é composto de várias entidades associadas ao colaborador. A página verificada na Figura 18, demonstra esta página que introduz alguns elementos notáveis e regularmente utilizados na aplicação.

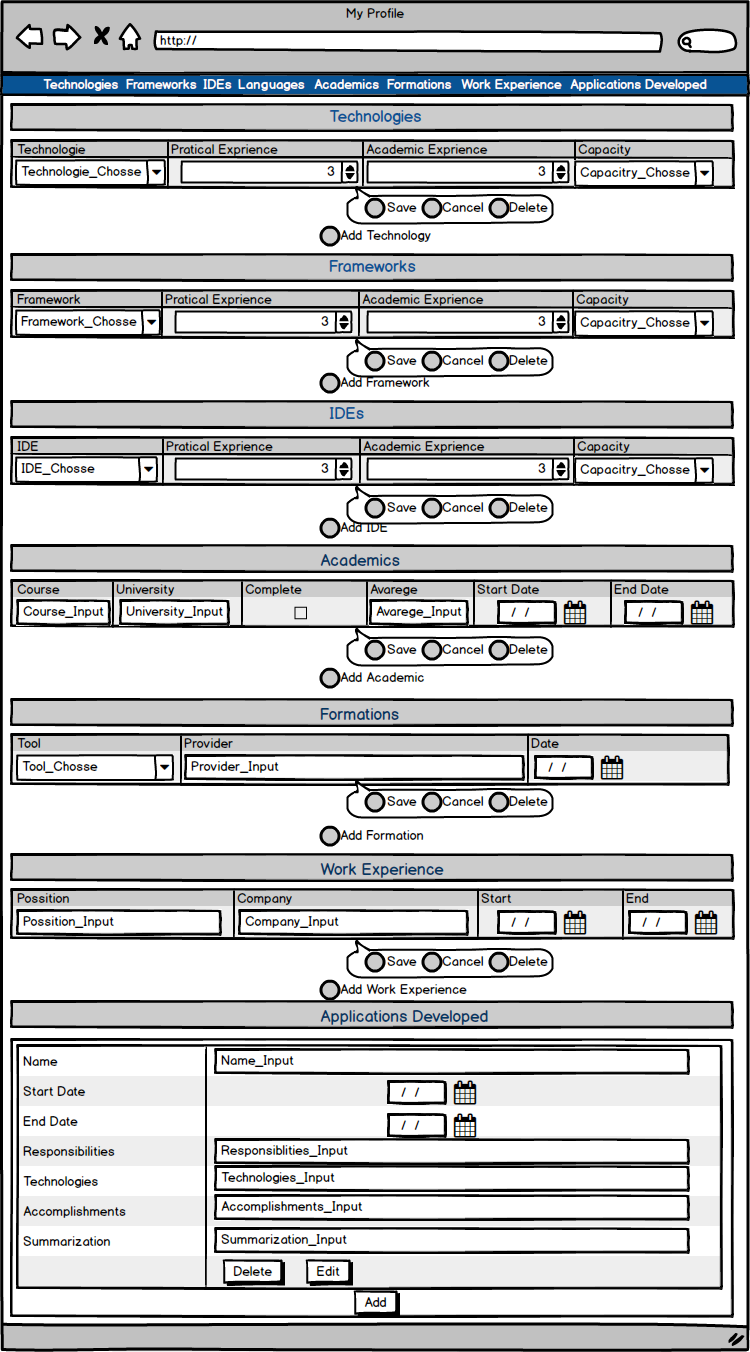


Figura - Frames, MyProfile

O primeiro elemento a notar é a Editable Tabel, 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 outro 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 Tabel 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 Editor, um *rich widget* que permite criar uma página tipo *popup* a partir de uma página já estabelecida na aplicação.

No caso da MyProfile o PopUp esta relacionado com o botão “Add”, verificado no final da página, este realiza um *navigate* para a página AddAppPopUp, que foi produzida para ser utilizada como um *popup*.

Normalmente PopUps não são utilizados para introduzir novos valores a uma entidade, mas sim para facilitar a escolha de instancias duma tabela.

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 a página demonstrada na Figura 19, que demonstra todas as instancias da entidade Form.

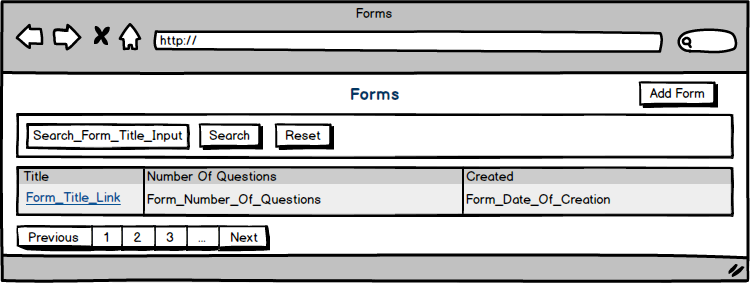


Figura - Frames, Forms

Esta página apesar de simples mostra os componentes mais comuns das páginas que demonstram as instancia duma entidade.

A primeira é a utilização da Table Records para demonstrar as instancias em si. Esta *widget* é semelhante a List Records, ambas demonstram instancias duma lista, mas ao contrario da List a Tabel demonstra a informação no formato de tabela. Esta *widget* serve principalmente para demonstrar instancias 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 a página onde a instancia pode ser demonstrada numa forma mais extensa, onde também é normalmente possível alterar ou destruir a instancia.

Esta tabela normalmente é formada para demonstrar um conjunto limitado de instancias 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.

Deve ser notado que a List\_Navigation não inclui em si a capacidade de alterar a tabela, para tal é necessário associar a mesma uma Screen Action. Esta deve obter a informação da tabela, normalmente realizando uma Resfesh Data sobre o *aggreate* que obteve a informação da tabela, seguindo de um Ajax Refresh sobre a tabela em si.

Esta forma de navegação não foi a primeira utilizada, sendo que originalmente foi utilizado dois botões um para obter a página seguinte e outra a página anterior.

Estes botões envolviam um quanto código extra da parte da aplicação, primeiro era utilizado quatro variáveis, o *current* que representava a página da lista verificada, nRows o numero de elemento por cada página da lista, needsNext e needsPrev que demonstram a necessidade do botão Next e o botão Prev.

As duas ultimas variáveis seriam utilizadas para definir as características de visibilidade e habilitar dos botões.

Para verificar a necessidade do Prev era bastante simples, sendo que desde que *current* fosse maior que zero então havia pelo menos uma página anterior a mostrar.

Já a necessidade de Next era mais complexo, primeiro a informação para a listagem teria de ocorrer com um SQLQuery em vez de um Aggregate (como é feito atualmente), e nesta *query* era obtida nRows mais 1 instancia da entidade.

Se o resultado dessa *query* for um numero de instancias igual a nRows mais 1 então é necessário o Next, para no mínimo obter a ultima instancia.

Com a utilização do List\_Navigation estas variáveis são desnecessárias e o *query* pode ser substituído por um Aggregate, que em si é otimizado, dai ter sido escolhida sobre o sistema dos botões Prev e Next.

Outro fator desta escolha é o facto que o List\_Navigation permite vários botões numerados, não sendo restinguido aos botões Next e Previous, que também inclui.

Outro elemento bastante comum nas páginas de instancias 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*. O outro botão permite restabelecer a tabela, com o valor de pesquisa *null*.

Por ultimo temos o botão de adição, no caso da Figura 19, o botão “Add Form”, este botão não é utilizado sempre neste tipo de páginas, mas aparece em grande parte. O botão serve sempre para navegar para uma página onde é possível adicionar um novo elemento.

Existem algumas exceções nas páginas de listagem das instancias, algumas introduzindo um formato alternativo a pesquisa, outros demonstram a informação numa forma bastante diferente e por últimos outros introduzem mais ações possíveis por cada instancia da entidade listada.

Um exemplo dessas exceções pode ser verificado na Figura 20, onde é verificada a página de listagem de aplicações a vagas existentes.

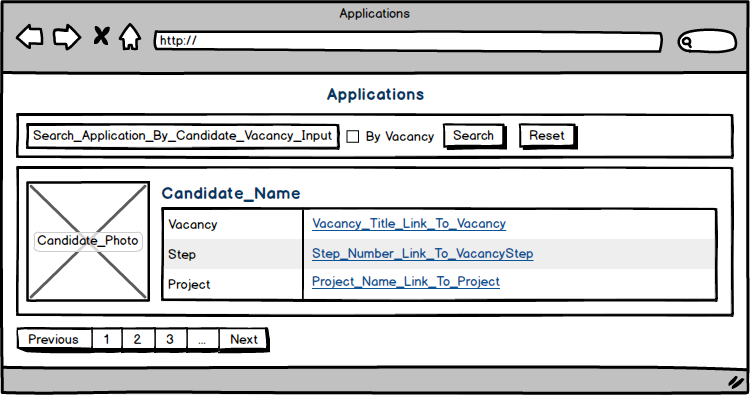


Figura - Frames, Applications

Como se pode verificar as figuras Figura 19 e Figura 20 tem três diferenças notáveis. Primeiro não existe o 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 20.

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.

A primeira *widget* forma uma estrutura da informação, incluindo uma fotografia do candidato da aplicação e a segunda a informação geral do candidato.

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 21.

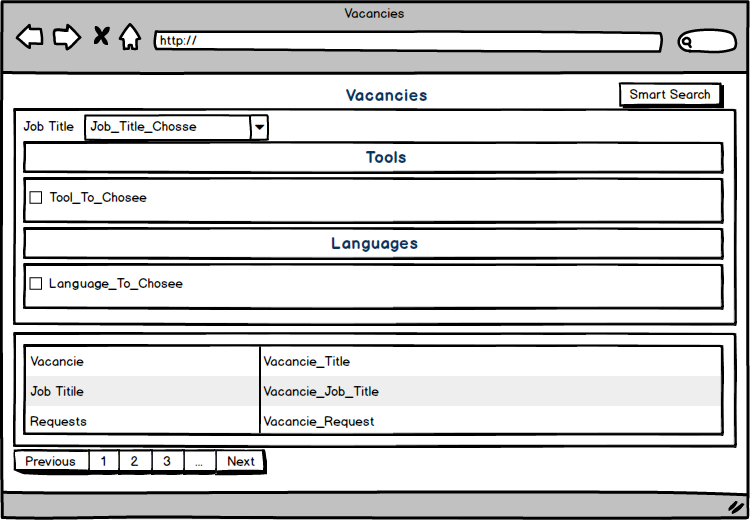


Figura - Frames, Vacancies

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.

Uma exceção que não envolve o formulário de pesquisa e adiciona capacidades a cada instancia da entidade pode ser verificada na Figura 22, que demonstra a página que lista as candidaturas espontâneas que ainda não foram tratadas.

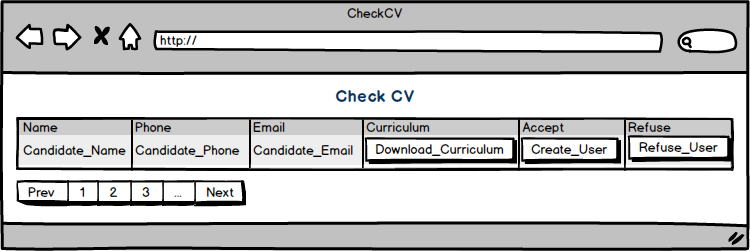


Figura - Frames, CheckCV

Como se pode ver cada instancia da tabela da Figura 22, 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.

A ultima exceção a notar é a página de listagem de eventos que pode ser verificada na Figura 23.

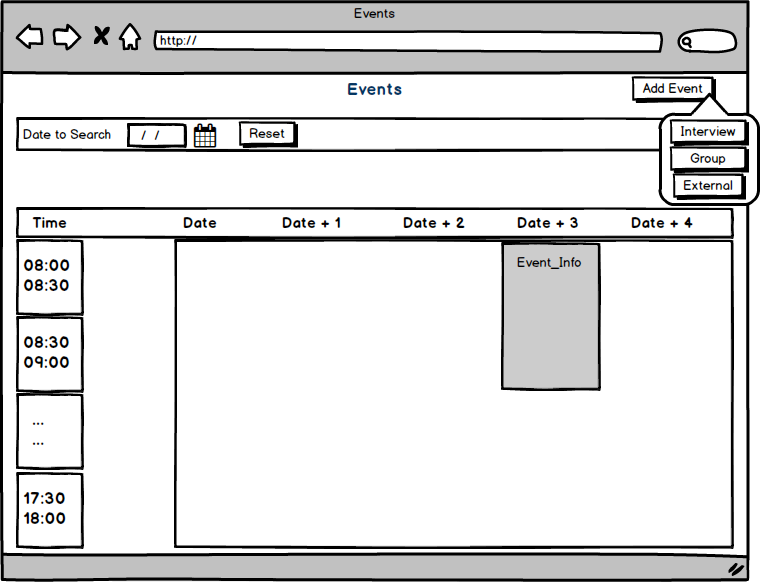


Figura - Frames, Events

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 Calandar 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 inicio e o fim do evento, na data correta. Algo que é possível, não com a utilização de uma Tabel Records, mas uma List Records onde cada elemento listado resulta num bloco ocupado ou livre.

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 24, que demonstra a página de um candidato.

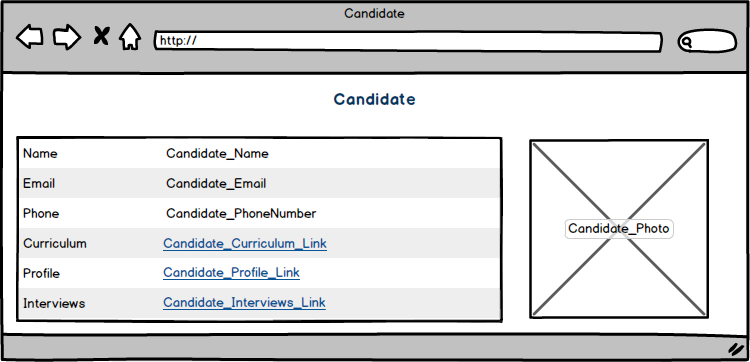


Figura - Frames, Candidate

Esta página, como muitas outras que demonstram uma instancia, utiliza o Show Record para demonstra 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 25, onde se pode verificar a página que demonstra alguma informação de um evento, mais os seus participantes.

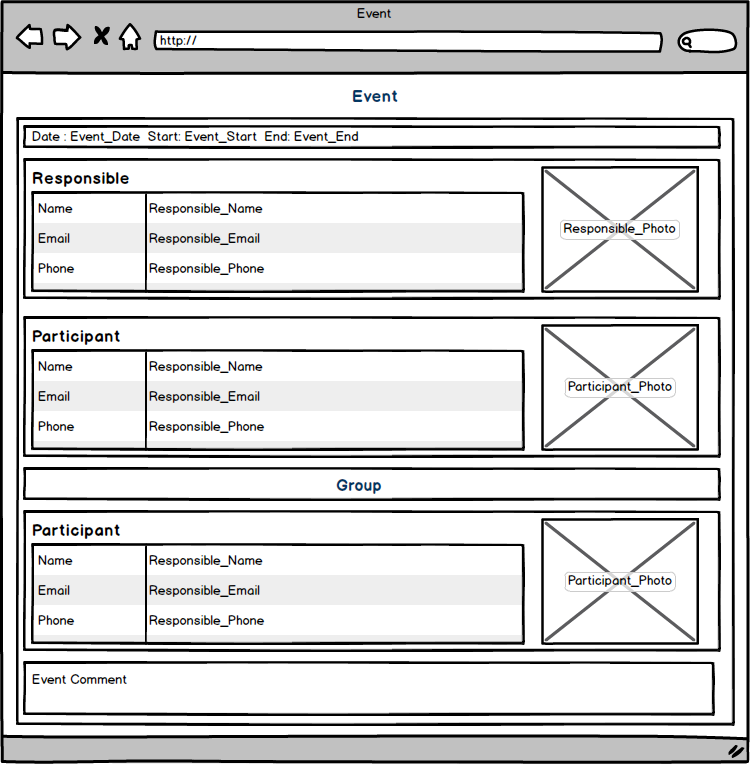


Figura - Frames, Event

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, o mesmo 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 esta numa List Record que lista todos os elementos de um grupo para um evento.

As vezes a listagem dos elementos pode ser mais complexa que a verificada na Figura 25, 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 26, 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.

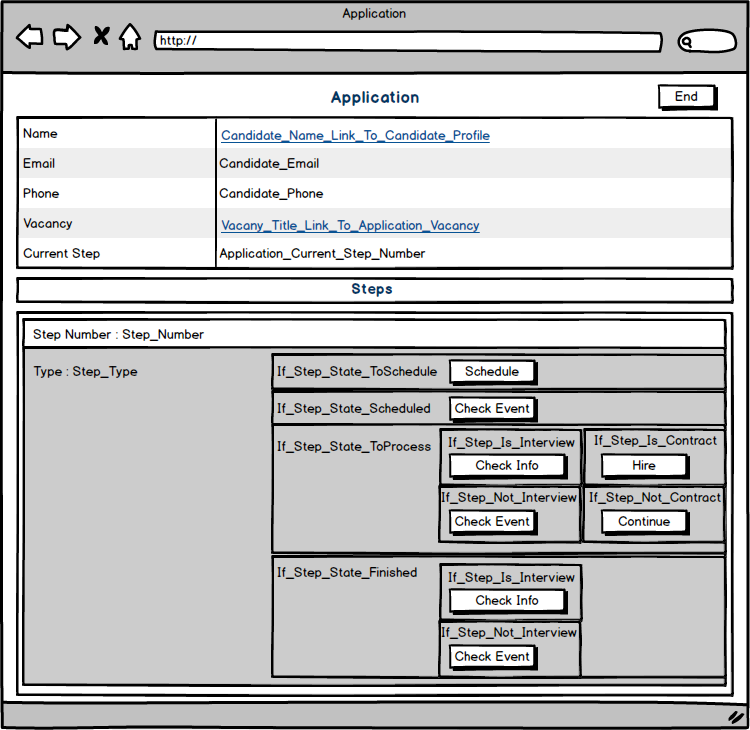


Figura - Frames, Application

Neste caso o valor demonstrado 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.

As vezes este tipo de páginas demonstram informação bastante extensa, então para facilitar a visualização e organização da página é utilizado Tabs para dividir blocos de informação.

Um exemplo da utilização das Tabs nestas páginas pode ser verificada na Figura 27, onde se pode verificar 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.

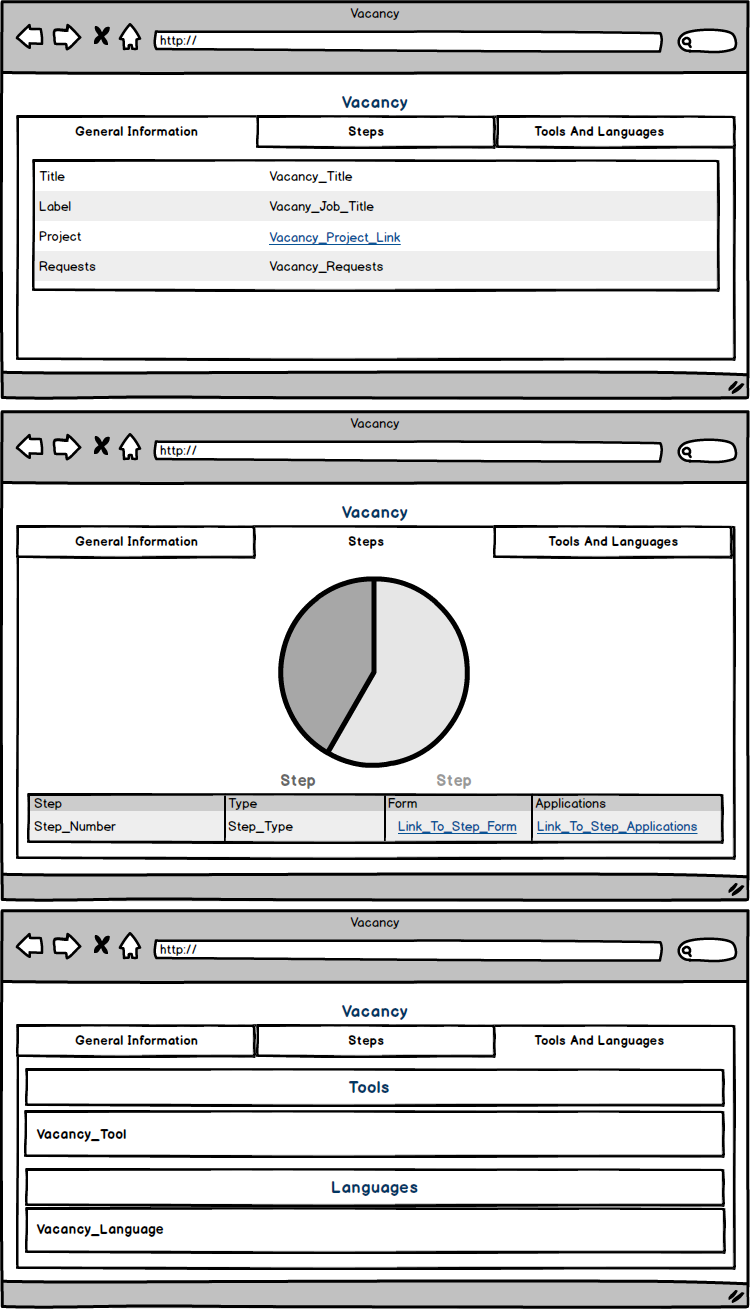


Figura - Frames, Vacancy

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, 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 28, que demonstra a página que permite introduzir um formulário.

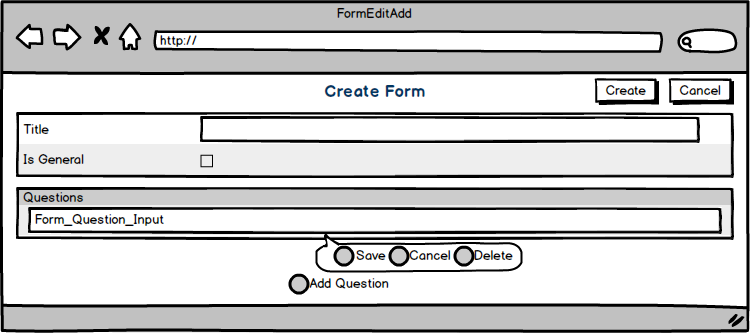


Figura - Frames, FormEditAdd

A página da Figura 28, 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. Para este cargo foi originalmente considerado o *widget* Form, que permitia o mesmo numa forma bastante simples, mas com uma visualização menus ampletiva e integrada ao resto da aplicação.

O outro *widget* regular é o Editabel Tabel, que é utilizado regularmente para introduzir outras instancias sobre uma entidade relacionada com a instancia central a adicionar. No exemplo da Figura 28, 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.

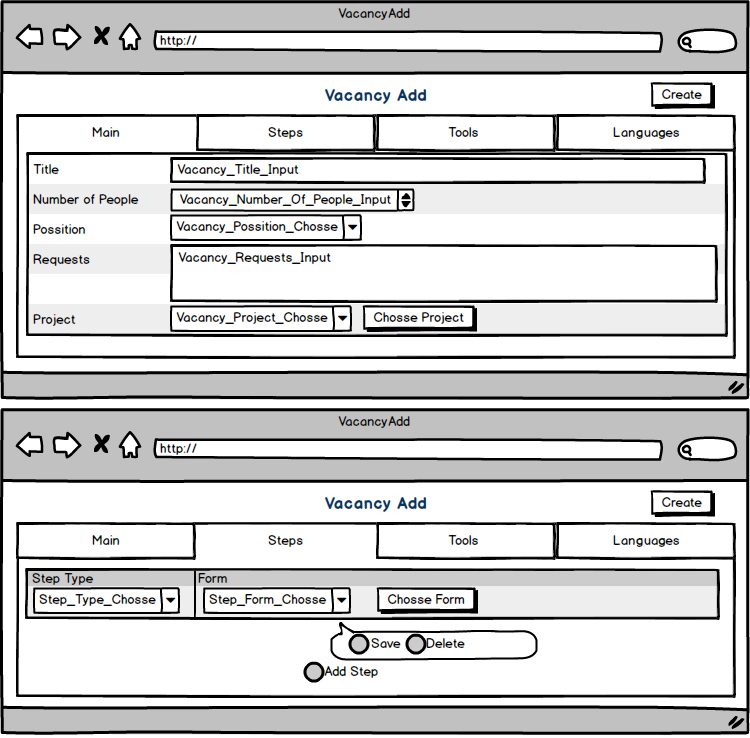
Deve ser notado que estas páginas partilham uma logica, a ideia que até o botão “Create” ser pressionado a instancia não é criada, nem as instancias relacionadas.

Esta lógica tem uma vantagem principal, ao não criar as instancias até ao momento desejado nunca são criadas instancias desnecessárias que são removidas de seguida. Também garante que se um utilizador iniciar o processo de criação, mas não o completar, saindo da página sem a utilização dos botões “Create” e “Cancel”, a instancia incompleta não ocupa espaço na base de dados.

Por outro lado esta lógica levantará algumas dificuldades no processo de formação das instancias a adicionar.

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 página assim é a representada na Figura 29 onde se pode verificar as varias Tabs individualmente, mais a PopUp que é utilizada na página, no final.



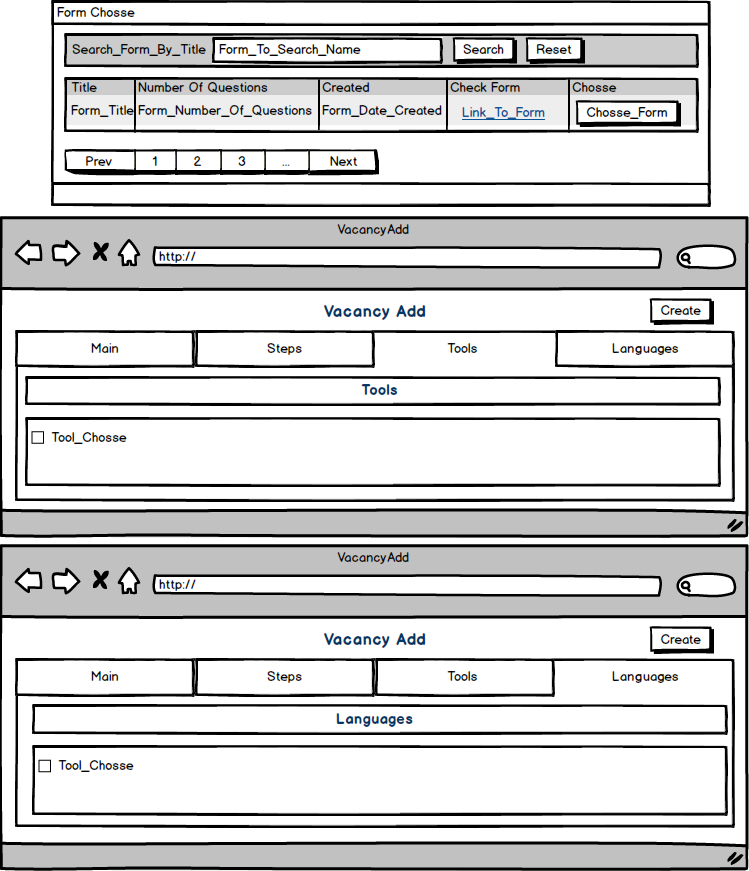


Figura - Frames VacancyAdd

Como se pode verificar na Figura 29, a adição de uma vaga divide-se entre quatro Tabs diferentes, com cada permitindo estruturar algo de diferente, mas importante, da vaga em si.

Já o PopUp serve para demonstra os Forms existentes que podem ser escolhidos para o passo da vaga a adicionar. Esta utilização de PopUp ocorre frequentemente em situações em que é necessário escolher uma instancia duma entidade referida por uma instancia a adicionar.

# Avaliação Experimental

A avaliação da nossa solução é apresentada neste capítulo. Aqui mostramos

como as nossas grandes ideias funcionaram

Exemplo de indentação do segundo parágrafo.

## 4.1 Nome da primeira secção deste capítulo

Texto da secção.

Continuação do texto noutro parágrafo.

## 4.2 A segunda secção deste capítulo

Na segunda secção deste capítulo, vamos abordar o enquadramento, o contexto e as funcionalidades.

### 4.2.1 A primeira sub-secção desta secção

As sub-secções são úteis para mostrar determinados conteúdos de forma

organizada. Contudo, o seu uso excessivo também não contribui para a facilidade

de leitura do documento[[3]](#footnote-3).

### 4.2.2 A segunda sub-secção desta secção

Esta é a segunda sub-secção desta secção, a qual termina aqui.

## 4.3 Análise de resultados

A análise de resultados segue aqui, nos próximos parágrafos de forma detalhada.

A data limite de entrega da versão final em 19 de Setembro de 2015 tem subjacente a inscrição em época especial. Não se verificando esta situação, a data limite de entrega é em 25 de Julho de 2015. O júri de cada projecto é constituído por indicação do respectivo orientador, até 25 de Maio de 2015. A avaliação da versão beta será realizada até 29 de Junho de 2015.

A discussão da versão final do projecto é pública e inclui até 30 minutos de apresentação e demonstração seguidos de discussão até ao máximo de 120 minutos de duração total da prova (40 e 150 minutos, respectivamente, quando o grupo tem três estudantes, e, 20 e 90 minutos quando o trabalho é individual). O júri da prova é proposto pelo orientador do projecto, tem pelo menos três elementos e inclui o arguente, o orientador e um docente de Projecto e Seminário (podendo este, em caso de impedimento, delegar num docente da área departamental). As deliberações do júri são tomadas por maioria simples.

A melhoria de classificação, se realizada no semestre de Inverno seguinte, terá por base trabalho adicional e discussão oral avaliados pelo mesmo júri. Quando realizada noutro semestre, envolve a realização de novo projecto.

A coordenação global dos projectos e dos seminários é feita pelos docentes de Projecto e Seminário, de acordo com as orientações definidas pela comissão coordenadora do curso. No sítio desta unidade curricular, é mantida a informação relevante, incluindo prazos, calendário dos seminários, estado dos projectos e estudantes e orientadores envolvidos. No final de cada ano lectivo, o regente elabora e apresenta à comissão coordenadora do curso um relatório sucinto sobre o funcionamento da unidade curricular. Em cada ano lectivo são identificados os melhores projectos para promover a sua divulgação pública.

# Conclusões

Neste trabalho tratou-se o problema. Foi formulada a solução que assenta nos princípios de boas práticas aprendidos ao longo do curso.

A solução obtida atingiu resultados satisfatórios.

# Referências

|  |  |  |
| --- | --- | --- |
| [1] |  | Mind Source, “Portal de Emprego,” Mind Source, [Online]. Available: https://emprego.mindsource.pt/. [Acedido em Abril 2018]. |
| [2] |  | Randstad, “My Profile,” Randstad, [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

Estamos no início do anexo 1. Nalguns casos, é conveniente colocar anexos de forma a complementar os resultados. Por vezes, em casos excepcionais devido à sua dimensão, as figuras têm que ser apresentadas de forma a ocupar toda a página, na forma de paisagem (*landscape*). Podemos fazer isso da forma que se apresenta na figura 3.

# A.2 Modelos de dados

Estamos no início do anexo 2.

O *relatório* é um resumo do projecto global. Apenas como referência, é expectável cerca de 30 a 40 páginas A4 não devendo exceder 50 páginas.. A estrutura deve ser discutida e aceite pelo orientador. Os capítulos apresentados devem ter, em geral, a seguinte organização:

**Capítulo 1** — Introdução

Enquadramento do trabalho, metas, objectivos e especificações do projecto e resumo da solução. Concluir com a descrição breve dos restantes capítulos.

**Capítulo 2** — Formulação do problema

Introdução dos conhecimentos necessários para entendimento do trabalho, estabelecimento de terminologia e descrição detalhada do problema e do seu contexto. Síntese de abordagens anteriores do problema, caso existam, indicando as razões porque são insatisfatórias.

**Capítulo 3** — Grande ideia 1

Este capítulo pode ser subdividido em secções, designadamente:

1. Introdução: descrição da abordagem do problema e da metodologia utilizada; identificação das tarefas;
2. Elenco das características / Análise do problema: especificações, constrições, ferramenta de análise, etc*.*
3. Projecto: modelo para resolução do problema;
4. Implementação: a implementação do modelo como sistema computacional; descrição concisa do *hardware* e do *software*; dificuldades e contradições encontradas e sua resolução;
5. Avaliação: testes realizados e resultados experimentais (quando possível, o objectivo, a montagem e o método usado devem ser brevemente descritos); análise crítica dos resultados.

…

**Capítulo k+2** — Grande ideia k

**Capítulo k+3** — Conclusões

Recapitulação do trabalho desenvolvido. Referir claramente as observações e conclusões importantes. Discussão de ideias e recomendações para trabalho futuro.

**Referências**

**Anexos**

Os anexos devem incluir as partes importantes do *dossiê do projecto*. O seu conteúdo depende da natureza do projecto, mas, em geral, pode incluir: listagem de programas, resultados de testes de *software*, exemplos de ecrãs de interface com o utilizador, esquemas dos circuitos, listagem de componentes, *data sheets* críticos, resultados de testes de *hardware*, desenhos mecânicos, análise económica, etc. (quando realizável, o *relatório* deve ser acompanhado da cópia do código, bibliotecas, etc. em suporte electrónico).

**Mais algumas dicas…**

O júri para avaliação do projecto final de curso apreciará o projecto, a sua demonstração e o respectivo relatório final (valorizando a escrita enquanto forma de divulgação de conhecimento). O relatório, depois de aceite e discutido, ficará disponível na biblioteca do departamento, para consulta.

O relatório deve ser auto-suficiente, isto é, o professor ou qualquer aluno finalista deve ficar apto a perceber o trabalho que realizou sem ter de ir à biblioteca ler os artigos originais.

Não escreva para o orientador, conhecedor de todo o detalhe, ou para um principiante. Tente escrever para uma audiência constituída por estudantes finalistas. Mantenha em mente o nível de conhecimentos do leitor a que se dirige. O relatório será uma base de trabalho para estudantes em circunstâncias semelhantes. Não dificulte o trabalho do leitor nem o faça estúpido (obviamente, …). Também é impossível ser totalmente claro. Evite afirmações dogmáticas (exemplo: "O *software* é a parte mais importante do computador.").

O relatório técnico não é uma história: usualmente não segue a cronologia das actividades realizadas. Também não é um romance (atenção à adjectivação). O relatório é um documento formal, feito para descrever os aspectos importantes do trabalho realizado.

Não tente descrever a função de cada componente, por exemplo a frase "O circuito IC2 e os componentes a ele associados formam um amplificador inversor …" é adequada. Contudo, descreva detalhadamente a função de cada componente ou circuito invulgar ou crítico.

As ilustrações (figuras, tabelas, gráficos e exemplos) são auxiliares preciosos para a explicação, mas envolvem muito trabalho. As figuras e as tabelas devem ser legíveis, instrutivas, legendadas e ter título. Os exemplos devem ser suficientemente detalhados para ilustrar o conceito.

O texto deve, pelo menos, ser analisado por um corrector ortográfico: os erros de ortografia são inadmissíveis.

Recomenda-se a leitura de alguns artigos e ou livros bem escritos para adquirir sensibilidade para a arte de escrever. Os artigos premiados em conferência são, normalmente, bons exemplos de escrita.

A escrita do relatório demora sempre mais tempo do que o inicialmente previsto.

**Resumo**

No essencial, a ideia é que tem algo para vender e o “Resumo” é a montra: a mensagem deve ser suficientemente clara e encorajar o cliente a entrar — se ele não a perceber passará ao lado. O resumo inclui: a motivação para o trabalho, como o fez e os resultados principais. Devem ser evitados chavões e palavras longas, as referências são proibidas e não deve utilizar acrónimos. Tenha em conta que o leitor será influenciado quer pela informação contida no resumo quer pela maneira como este está escrito. Não há desculpas para frases curtas ou desligadas, erros de gramática ou erros de sintaxe.

Não é fácil escrever um bom resumo.

**Introdução**

Procure dar resposta às seguintes questões: qual é o problema? porque é importante? o que é que outros já fizeram? quais as ideias base da solução apresentada? como está organizado o resto do relatório?

**Formulação do problema**

Defina o problema. Introduza a terminologia. Discuta as propriedades básicas.

**Solução do problema**

Enumere as coisas que fez e que considere importantes. Não seja modesto mas também não exagere.

A correcta avaliação do projecto é um aspecto crítico.

**Conclusões**

Procure dar resposta às seguintes questões: quais, se for caso disso, as lições aprendidas? o que, se algo, foi explicado? em que medida os objectivos foram atingidos? se existe algo que agora faria de forma diferente? quais as vantagens e desvantagens do trabalho realizado face a exemplos da literatura? que ideias para trabalho futuro?

**Referências**

A ideia subjacente à referência é que esta poupa papel e que o leitor poderá obter o documento em qualquer biblioteca científica razoável. Assim, é critério essencial referir revistas disponíveis em bibliotecas de instituições de ensino superior ou outras instituições profissionais. Em geral, não é razoável a referência a actas de conferências, dado que estas raramente estão acessíveis em bibliotecas pelo que, para todos os efeitos, estão indisponíveis. As referências a “Comunicações Privadas” são inaceitáveis. A informação dada deve ser sempre suficientemente detalhada por forma a que o leitor possa adquirir a publicação ou consultá-la numa biblioteca. Referências a dissertações de doutoramento ou outras devem indicar a instituição e o seu endereço. Sendo a referência essencial para o trabalho, no caso desta ser difícil de obter, dever-se-á incluir no documento, ou em apêndice, os seus pontos essenciais.

Cite uma referência sempre que está a incluir algo que não é contribuição sua ou quer indicar um conjunto de referências que o leitor pode consultar, mas cujo conteúdo não pode ser descrito adequadamente no relatório.

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. Este é um exemplo de nota de rodapé. Devem ser usadas com moderação, pois retiram a atenção ao texto principal. [↑](#footnote-ref-3)