



Universidade de Aveiro

2015

Departamento de Eletrónica, Telecomunicações e
Informática

Henrique Manuel de
Almeida Moreira

Plataforma Online de Criação e Gestão de Projetos ITED



Henrique Manuel de
Almeida Moreira

Plataforma Online de Criação e Gestão de Projetos ITED

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas de Informação, realizada sob a orientação científica do Doutor Cláudio Jorge Vieira Teixeira, Equiparado a Investigador Auxiliar e do Doutor Joaquim Manuel Henriques de Sousa Pinto, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho à minha família pelo apoio incondicional em todas as áreas da minha vida.

o júri

Presidente

Professor Doutor Joaquim Arnaldo Carvalho Martins

Professor Catedrático da Universidade de Aveiro

Professor Doutor Fernando Joaquim Lopes Moreira

Professor Associado do Departamento de Inovação, Ciência e Tecnologia da Universidade Portucalense

Professor Doutor Cláudio Jorge Vieira Teixeira

Equiparado a Investigador Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos

Dirijo os meus primeiros agradecimentos à minha família, nomeadamente aos meus pais, por todos os sacrifícios que tiveram que fazer para que me pudessem dar uma boa educação e a oportunidade de conseguir o nível de estudos que desejava e para que nunca nada me faltasse durante o meu crescimento, levando à conclusão deste capítulo tão importante da minha vida.

Aos Professores Joaquim de Sousa Pinto e Cláudio Jorge Vieira Teixeira por me terem dado a oportunidade de me estrear no mundo laboral e por demonstrarem interesse e disponibilidade no decorrer deste projeto.

E finalmente aos meus amigos e a todas as pessoas com quem fui formando laços ao longo da vida, que me ajudaram a formar e a crescer a diversos níveis.

A todos os que aqui referi mas também aqueles que não me referi e que tiveram um contributo na minha vida, um sincero obrigado.

palavras-chave

ITED, gestão, plantas, plataforma.

resumo

Esta dissertação tem como principal objetivo o desenvolvimento de uma aplicação *web* que permita a criação, gestão e edição de projetos ITED, fornecendo ferramentas de auxílio quer a empreiteiros, projetistas, instaladores de redes de comunicações, de diversas tecnologias, em edifícios, fornecendo ainda ferramentas de auxílio à orçamentação de produtos, com base em restrições da rede desenhada. O Project ITED pretende assim capacitar o utilizador com uma ferramenta prática, intuitiva e fácil de utilizar e principalmente sem restrições de plataforma de acesso ou localização.

keywords

ITED, management, blueprints, platform.

abstract

This thesis aims to develop a web application that allows the creation, management and editing of ITED projects, providing assistance tools either to contractors, designers or communication network installers, of various technologies, in buildings, still providing aid to product budgeting tools, based on the designed network restrictions. The Project ITED aims thus to empower the user with a practical, intuitive and easy to use tool and mostly without access platform or location restrictions.

Índice

1	Introdução	1
1.1	Enquadramento.....	1
1.2	Motivação.....	1
1.3	Objetivos	3
1.4	Metodologia	4
1.5	Organização da dissertação	5
2	Estado de Arte	7
2.1	ITED 3	7
2.2	CYPEITED	8
2.3	Televés Cast 60.....	13
2.4	Comparação entre aplicações	17
2.5	<i>User Interface Design</i>	17
3	Tecnologias utilizadas.....	21
3.1	Ferramentas Web.....	21
3.1.1	<i>Twig</i>	21
3.2	Ferramentas de trabalho vetorial	22
3.2.1	D3.js.....	22
3.2.2	<i>Dagre-d3</i>	23
3.2.3	<i>Svg-pan-zoom</i>	24
3.2.4	<i>Raphaël</i>	24
3.3	Ferramentas de desenvolvimento	25
3.3.1	<i>Symfony</i>	25
4	Conceção e Desenvolvimento	27
4.1	Levantamento de requisitos.....	27
4.2	Arquitetura.....	32

4.3	Implementação	34
4.3.1	Modelo de base de dados	34
4.3.2	<i>Symfony</i>	38
5	Protótipo	39
5.1	FrontOffice	39
5.2	Estúdio.....	44
5.2.1	Desenho do diagrama	44
5.2.2	Escolha do equipamento.....	56
5.2.3	Inventário e diagramas.....	57
5.3	BackOffice.....	61
5.3.1	Inserção de dados	63
6	Problemas e soluções.....	67
7	Conclusão	71
8	Trabalho futuro	73
9	Bibliografia	75
10	Anexos.....	79
10.1	Anexo A	79
10.1.1	Ferramentas Web.....	79
10.1.2	Ferramentas de desenvolvimento	81
10.2	Anexo B	83
10.3	Anexo C.....	95

Lista de Figuras

Figura 1 - tempo gasto por plataforma [4].....	3
Figura 2 – setor de tubagens do CYPEITED	8
Figura 3 - setor de instalações do CYPEITED	10
Figura 4 - setor de desenhos do CYPEITED.....	11
Figura 5 - configuração personalizada do software Televés Cast60	13
Figura 6 – Assistente ITED do software Televés Cast 60.....	15
Figura 7 - exemplo de uma infografia criada através da biblioteca D3.js	22
Figura 8 - exemplo de um grafo gerado pela biblioteca dagre-d3.....	23
Figura 9 – diagrama de casos de usos de gestão de projetos	28
Figura 10 - diagrama de casos de uso de gestão de plantas.....	28
Figura 11 - diagrama de casos de uso do estúdio	29
Figura 12 - arquitetura MCV do Symfony [19]	32
Figura 13 - Diagrama de sequência do processamento de uma Ajax Request	34
Figura 14 - diagrama de classes da aplicação.....	35
Figura 15 - vista geral da base de dados	36
Figura 16 - página de autenticação do utilizador	39
Figura 17 - processo de autenticação externa	40
Figura 18 - página de gestão de projetos	41
Figura 19 - página de gestão de um projeto	43
Figura 20 - página de mapa dos projetos de um utilizador.....	44
Figura 21 - exemplo da página do estúdio de uma planta horizontal	45
Figura 22 - interface de edição da ligação.....	50
Figura 23 - interface de criação de uma planta vertical nova	51
Figura 24 - exemplo da interface de estúdio de uma planta vertical	52
Figura 25 - janela de gestão de pisos numa planta vertical	54
Figura 26 - exemplo de desenho de um diagrama completo	55
Figura 27 - interface de escolha de equipamento	56
Figura 28 - exemplo de interface de um diagrama geral	60
Figura 29 - vista da lista de cabos.....	62
Figura 30 - vista da criação de cabos.....	62

Figura 31 - representação dos problemas de escala na planta (à direita: escala muito alta; à esquerda: escala muito baixa)	68
Figura 32 - painel de routing do SymfonyProfiler para a rota session/login	100

Lista de Tabelas

Tabela 1 - excerto de código de Twig.....	21
Tabela 2 - excerto de código de utilização da biblioteca Dagre-d3	23
Tabela 3 - excerto de código de utilização da biblioteca Svg-pan-zoom	24
Tabela 4 - excerto de código de uma página HTML.....	79
Tabela 5 - excerto de código de uma folha de estilos CSS	80
Tabela 6 - entidade Autenticação	83
Tabela 7 - entidade Perfil	83
Tabela 8 - entidade Utilizador	84
Tabela 9 - entidade Obra.....	85
Tabela 10 - entidade Tipo.....	85
Tabela 11 - entidade Planta	86
Tabela 12 - entidade Versão	86
Tabela 13 - entidade Piso	87
Tabela 14 - entidade TipoComponente.....	87
Tabela 15 - entidade TipoAplicacao	87
Tabela 16 - entidade ModeloAplicacao.....	88
Tabela 17 - entidade Nó	88
Tabela 18 - entidade Ligação.....	89
Tabela 19 - entidade Tubo	89
Tabela 20 - entidade TuboLigacao	90
Tabela 21 - entidade Cabo	90
Tabela 22 - entidade Qualidade	90
Tabela 23 - entidade CaboQualidade.....	91
Tabela 24 - entidade TuboLigacaoCaboQualidade.....	91
Tabela 25 - entidade Componente.....	92
Tabela 26 - entidade Subcomponente	93
Tabela 27 - entidade Material.....	93

Lista de Acrónimos

ANACOM	Autoridade Nacional de Comunicações
ATE	Armário de Telecomunicações de Edifício
ATI	Armário de Telecomunicações Individual
CAD	<i>Computer Aided Design</i>
CATV	<i>Community Antenna Television</i>
CC	Cabo Coaxial
CEMU	Caixa de Entrada de Moradia Unifamiliar
CP	Caixa de Passagem
CRUD	<i>Create Read Update Delete</i>
CSS	<i>Cascading Style Sheet</i>
CVM	Câmara de Visita Multioperador
DOM	<i>Document Object Model</i>
DWG	<i>Drawing</i>
FO	Fibra Ótica
HTML	<i>Hypertext Markup Language</i>
IEETA	Instituto de Engenharia Eletrónica e Informática de Aveiro
ITUR	Instalações de Telecomunicações em Urbanizações
IPTV	<i>Internet Protocol Television</i>
ITED	Infraestruturas de Telecomunicações em Edifícios
MVC	<i>Model-View-Controller</i>
NIF	Número de Identificação Fiscal
JSON	<i>JavaScript Object Notation</i>
PAT	Passagem Aérea de Topo
PC	Par de Cobre
PHP	<i>PHP: Hypertext Preprocessor</i>
QE	Quadro Elétrico
RegEx	<i>Regular Expression</i>
SGBD	Sistema de Gestão de Bases de Dados
SHA	<i>Secure Hash Algorithm</i>
S-MATV	<i>Single Master Antenna Television</i>
SQL	<i>Structured Query Language</i>

SVG	<i>Scalar Vector Graphics</i>
TT	Tomada Terminal
UML	<i>Unified Modeling Language</i>
VML	<i>Vector Markup Language</i>
XML	<i>eXtensible Markup Language</i>
ZAP	Zona de Acesso Privilegiado

1 Introdução

1.1 Enquadramento

A presente proposta de dissertação teve como origem o contacto de uma empresa de venda de quadros elétricos e produtos derivados com a Universidade de Aveiro, mais especificamente com o Instituto de Engenharia Eletrónica e Informática de Aveiro (IEETA) acerca da necessidade da construção de uma plataforma *online* personalizada, relacionada com a sua área de negócio, para disponibilização a outras entidades. Esta aplicação adotará o nome de “Project ITED”.

1.2 Motivação

“O desenvolvimento das actividades económicas e sociais, os enormes progressos tecnológicos verificados e as novas exigências decorrentes do ambiente concorrencial estabelecido em Portugal, impuseram a necessidade de formular novas regras para o projecto, instalação e gestão das Infra-estruturas de Telecomunicações em Edifícios.” [1]

Nascia assim a 1ª versão do Manual de ITED (Infraestruturas de Telecomunicações em Edifícios), atualmente na 3ª versão devido a mudanças no paradigma do setor do imobiliário.

O principal objetivo deste manual seria o de fornecer normas para instalação de infraestruturas de telecomunicações em edifícios com o intuito de facilitar, estruturar o trabalho e reduzir o custo a entidades desta área de negócio.

Este manual pretende também salvaguardar os interesses dos consumidores de telecomunicações. Isto passa por conseguir infraestruturas modernas, fiáveis e adaptadas aos serviços fornecidos pelos operadores públicos de comunicações eletrónicas.

Muitas vezes são os compradores de edifícios residenciais os mais desprotegidos, no que diz respeito à qualidade das infraestruturas de telecomunicações. Estatisticamente os edifícios só são vendidos após a sua conclusão, o que faz com que não existam possibilidades de escolha nem de mudança nas instalações já efetuadas.

Estas estatísticas dizem respeito a cerca de 80% dos edifícios construídos em Portugal [2], o que faz com que seja relevante a existência de um reforço na qualidade e na capacidade de adaptação às novas tecnologias de infraestruturas de telecomunicações.

Um dos grandes problemas na projeção destas infraestruturas passa por normalmente ser necessário desenhar os componentes manualmente, tipicamente por cima de uma planta desenhada num *software* CAD (*Computer Aided Design*), da ligação destes componentes e por fim, do cálculo de atenuação e de níveis de sinal, desde a origem da ligação até aos pontos terminais.

A atenuação é a quantidade de energia perdida pelo sinal através da sua propagação num cabo e o nível de sinal é a medida da quantidade de sinal [3], sendo esta afetada pela atenuação e outros fatores. Estes cálculos são necessários para assegurar a qualidade da instalação das redes de comunicação.

Este processo tipicamente ocupa muito tempo aos projetistas e aumenta de complexidade consoante o aumento do tamanho da planta sobre a qual se está a projetar.

Estas constantes mudanças no setor e constrangimentos de projeção de infraestruturas fazem com que as ferramentas existentes para este tipo de mercado vão ficando desatualizadas e obsoletas, levando a que haja a necessidade da criação de uma ferramenta que auxilie os projetistas de infraestruturas de comunicação a conceber infraestruturas adaptadas às restrições impostas na atualidade e que ajudem a agilizar este processo.

Estas ferramentas devem também contribuir para orientar o projetista de modo a evitar a introdução de erros nos desenhos das instalações e conseqüentemente, garantir uma instalação de maior qualidade.

Com as alterações no modo de utilização de dispositivos eletrónicos, nomeadamente em relação ao *boom* no mercado de dispositivos móveis (Figura 1) e com os planos de dados móveis a tornarem-se cada vez mais acessíveis, com maior capacidade oferecida e velocidade de acesso, e com um crescendo na sua adesão, surge assim também a necessidade de adaptar estas ferramentas às plataformas mais utilizadas na atualidade. Estas são, para além de dispositivos do tipo desktop, os *tablets* e os *smartphones* (Figura 1).

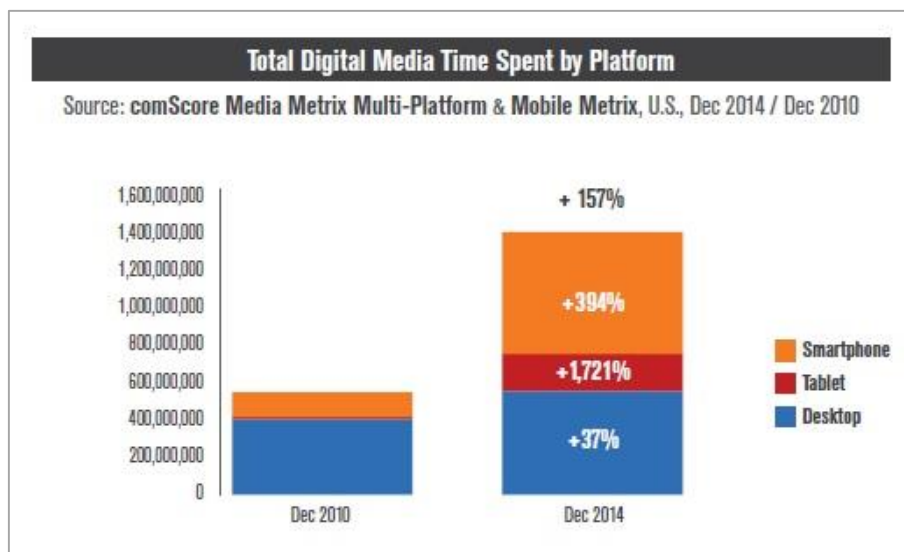


Figura 1 - tempo gasto por plataforma [4]

1.3 Objetivos

No que diz respeito a aplicações para desenho de diagramas ITED, já existem algumas no mercado. Umas estão adaptadas às normas ITED atuais e outras não. No entanto todas se tratam de aplicações para plataformas do tipo *desktop*.

De uma forma geral esta dissertação tem como objetivo a produção do protótipo de uma aplicação que permita o desenho de diagramas ITED e que ofereça funcionalidades de cálculo de atenuações e de níveis de sinal, adequando-se às normas atuais.

O cálculo de atenuações e de níveis de sinal é uma operação fundamental na projeção de diagramas ITED, de maneira a assegurar a qualidade das instalações. Este processo tipicamente envolve muitos parâmetros.

O protótipo a desenvolver tem como objetivo facilitar a tarefa de cálculo, procurando automatizar a inserção de alguns dos parâmetros e ao mesmo tempo, tentar manter um nível adequado de opções de personalização nos componentes que os afetam.

Nas aplicações já existentes existem duas abordagens na escolha de equipamento para instalação na infraestrutura de comunicações: a utilização de equipamento real com características predefinidas ou a utilização de equipamento genérico com personalização destas características.

Estes equipamentos, para além de alterarem os custos de uma instalação também afetam os valores dos cálculos abordados anteriormente.

De maneira a servir as necessidades tanto do projetista como da entidade para a qual a aplicação está a ser desenvolvida tem-se como objetivo a construção de um sistema de disponibilização de equipamentos reais. Pretende-se construir um sistema com duas vertentes: numa primeira fase o projetista introduziria os parâmetros necessários na instalação; numa segunda fase seria apresentada uma listagem de equipamento recomendado que cumprisse os parâmetros da instalação.

A vantagem está em a entidade para a qual a aplicação esta a ser desenvolvida poder sugerir a venda dos seus produtos e em o projetista poder aceder diretamente a uma listagem de equipamentos sem necessidade de consulta manual de fontes externas.

Para finalizar pretende-se implementar um mecanismo que produza um inventário baseado nas escolhas do projetista e diversos gráficos que ilustrem a infraestrutura projetada com os valores de atenuações e níveis de sinal calculados.

As aplicações disponíveis no mercado adequam-se tipicamente apenas a um tipo de entidade: o projetista. Nesta dissertação irá tentar-se produzir um protótipo que sirva as necessidades desta entidade (utilizador principal), assim como as necessidades de outros tipos, tais como os administradores do sistema, responsáveis entre outras tarefas, por manter atualizada a listagem de equipamento a sugerir.

De modo a acompanhar as tendências da construção de páginas web, influenciadas pelo crescimento do mercado dos dispositivos móveis, pretende-se ainda que o protótipo a desenvolver seja uma aplicação *web* e que apresente uma interface responsiva, isto é, que adapte a sua estrutura e conteúdo a apresentar ao tamanho do ecrã do dispositivo onde é acedida.

1.4 Metodologia

No decorrer desta dissertação ir-se-á proceder ao estudo e desenvolvimento de uma aplicação que permita libertar o utilizador de constrangimentos de acesso na utilização de uma ferramenta que possibilite o desenho de diagramas ITED e cálculo de atenuação e níveis de sinal nas ligações.

Em primeiro lugar irão ser estudadas ferramentas semelhantes, já existentes no mercado, de maneira a perceber que funcionalidades fazem sentido existir e que lacunas podem ser preenchidas. Seguidamente serão efetuadas diversas reuniões de *brainstorming* a fim de apresentar um conjunto de requisitos necessários. Será depois apresentada uma arquitetura que permita suportar os requisitos definidos. Por fim, será apresentado um protótipo que implemente a arquitetura proposta e que consiga resolver os problemas encontrados.

Irá existir um processo de iteração dos requisitos da aplicação ao longo de diversas reuniões, a fim de traçar um rumo ao projeto à medida que este for avançando.

Finalmente serão retiradas conclusões que sumariem a viabilidade do protótipo desenvolvido.

1.5 Organização da dissertação

Este documento encontra-se dividido em 9 capítulos, iniciando-se com este primeiro capítulo de enquadramento.

O segundo capítulo consiste numa análise do ambiente atual em que o projeto se insere e no estudo de diversas ferramentas que partilham características similares com o projeto a desenvolver.

No terceiro capítulo irão ser descritas e explicadas as tecnologias que serão utilizadas no desenvolvimento da aplicação.

O quarto capítulo apresenta um levantamento dos requisitos necessários para o desenvolvimento do sistema e de que maneira irá ser implementado.

O quinto capítulo apresenta o protótipo concebido tendo em conta os constrangimentos apresentados anteriormente, procurando ilustrar de forma exaustiva os componentes principais que o constituem.

No sexto capítulo irão ser apresentados e explicados os principais problemas encontrados no decorrer do desenvolvimento do projeto e de que forma foram abordados.

O sétimo capítulo é onde serão apresentadas as principais considerações finais, retiradas no decorrer do desenvolvimento deste projeto.

No oitavo capítulo, é onde serão sugeridas possíveis alterações que poderiam levar à implementação de uma versão melhorada e mais robusta do sistema a desenvolvido.

Por fim, será apresentada uma secção de anexos contendo informação mais detalhada sobre os componentes utilizados na conceção deste projeto.

2 Estado de Arte

De modo a poder traçar um rumo para um projeto novo, é necessário compreender o meio em que este está inserido e que conceitos lhe estão inerentes. Sendo assim, neste capítulo, irá ser feita uma análise das soluções atuais de *software* disponíveis no mercado, das suas funcionalidades e das lacunas que necessitam de ser colmatadas, tentando ainda compreender as obrigações na área das Infraestruturas de Telecomunicações em Edifícios. Como este projeto se trata de uma plataforma *online* irá ser também apresentada uma breve análise de boas práticas na área de *User Interface*.

2.1 ITED 3

“A defesa dos interesses dos consumidores de telecomunicações passa por infraestruturas modernas, fiáveis e adaptadas aos serviços fornecidos pelos operadores públicos de comunicações eletrónicas”. [2]

O Manual de Infraestruturas de Telecomunicações em Edifícios, atualmente na terceira versão, trata-se de um documento produzido pela ANACOM (Autoridade Nacional de Comunicações), sendo esta a autoridade reguladora nacional no âmbito das comunicações, que se destina a fornecer técnicas de instalação e especificações técnicas de equipamentos e materiais aplicadas aos espaços, redes de tubagem, redes de cabos e equipamentos associados, com o objetivo de auxiliar as entidades envolvidas no desempenho do seu trabalho.

São abordadas soluções técnicas, com base nos mínimos exigidos, orientadas às tecnologias de cabo coaxial, fibra ótica e par de cobre presentes nas instalações de edifícios.

Nas próximas secções irá ser apresentada uma análise das ferramentas já disponíveis no mercado, ligadas ao desenvolvimento de diagramas ITED, que apresentam características e funcionalidades semelhantes às listadas nos objetivos definidos. Nesta análise procurar-se-á identificar os seus pontos fortes e também descobrir lacunas a preencher, de maneira a justificar o desenvolvimento de um novo protótipo que complemente as funcionalidades em falta e seja uma mais-valia para o utilizador.

2.2 CYPEITED

O CYPEITED é um programa de elaboração de projetos de redes de telecomunicações de edifícios multifamiliares. Este programa rege-se sobre a terceira e atual versão das normas ITED. Uma consulta à página de programas da CYPE revelou que este *software* tem um custo de 380.00€ [5] e por esta razão não pôde por nós ser testado.

A informação de seguida apresentada foi retirada da apresentação de um *webminar* [6] realizado pela empresa Top Informática que tem como missão “Proporcionar às empresas, instituições e particulares do ramo da Engenharia e Construção produtos e conhecimentos que permitam a máxima rentabilização da sua atividade com a melhor qualidade.” [7].

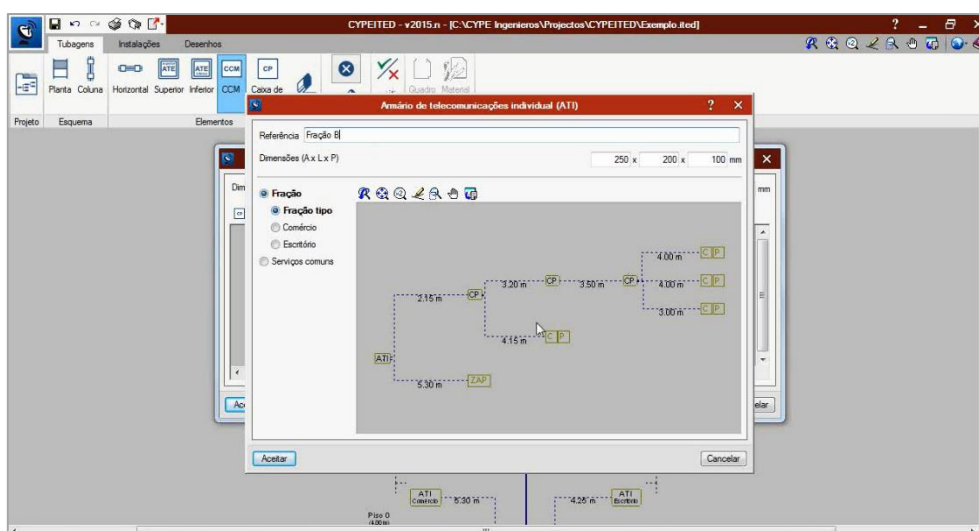


Figura 2 – setor de tubagens do CYPEITED

Este programa encontra-se dividido em três setores: Tubagens, Instalações e Desenhos.

O primeiro passo, na secção de Tubagens, após a criação de um novo projeto, é configurar o número de pisos e de colunas. Aqui define-se a altura entre pisos e quantos se encontram acima e abaixo da rasante (piso 0). Após a sua criação é possível editar individualmente a altura de cada piso e atribuir-lhe um nome de referência.

De seguida adicionam-se frações. Adicionar uma fração coloca imediatamente, na planta da fração, um componente ATI (Armário de Telecomunicações de Edifício) e permite de seguida a adição de CP (Caixas de Passagem) e de TT (Tomadas Terminais).

Ao adicionar uma CP é necessário definir de início as suas dimensões (Altura x Largura x Profundidade) e de seguida procede-se à ligação à ATI previamente instalada.

Ao adicionar uma TT define-se de início as tecnologias que esta suporta. Entre as opções encontra-se Cabo Coaxial, Par de Cobre e Fibra Ótica, podendo ainda definir-se a tomada como uma ZAP (Zona de Acesso Privilegiado).

Uma ZAP trata-se do local de instalação de uma tomada terminal onde chegam dois cabos de pares de cobre, dois cabos coaxiais e onde é reservado espaço para duas tomadas terminais de fibra ótica [3].

Após isto, de maneira semelhante às caixas de passagem, liga-se a tomada ao ATI ou a uma CP.

As ligações efetuadas possuem um comprimento de tubagem predefinido, sem relação com a distância entre os componentes ligados no editor. Alterações ao comprimento da tubagem devem ser feitos de forma manual.

Efetuada a criação de frações procede-se então à introdução de ATE (Armários de Telecomunicações de Edifícios). É possível adicionar dois ATE: um superior e um inferior, uma vez que é possível obter comunicações por dois meios: cabo (inferior) e antena (superior). Ao adicionar um ATE faz-se a ligação até às frações previamente criadas através dos ATI, sendo possível colocar caixas de passagem entre os diversos componentes.

A edição de tubagens nesta fase é semelhante à edição de tubagens na fase de criação de frações.

Concluído o processo de definição dos ATE começa a fase de colocação das diversas frações e de ATE na coluna montante. É possível ligar frações através de caixas de coluna montante, que não são nada mais do que caixas de passagem instaladas na coluna do edifício. É ainda possível adicionar uma CVM (Câmara de Visita Multioperador) nesta fase.

Uma funcionalidade interessante é a de replicação de frações, útil quando diversos pisos seguem a mesma estrutura de instalação. Permite poupar tempo no desenho da instalação e ajuda a evitar a introdução de erros no desenho da instalação.

Por fim corre-se a ferramenta de cálculo que verifica se a tubagem instalada suporta a rede desenhada.

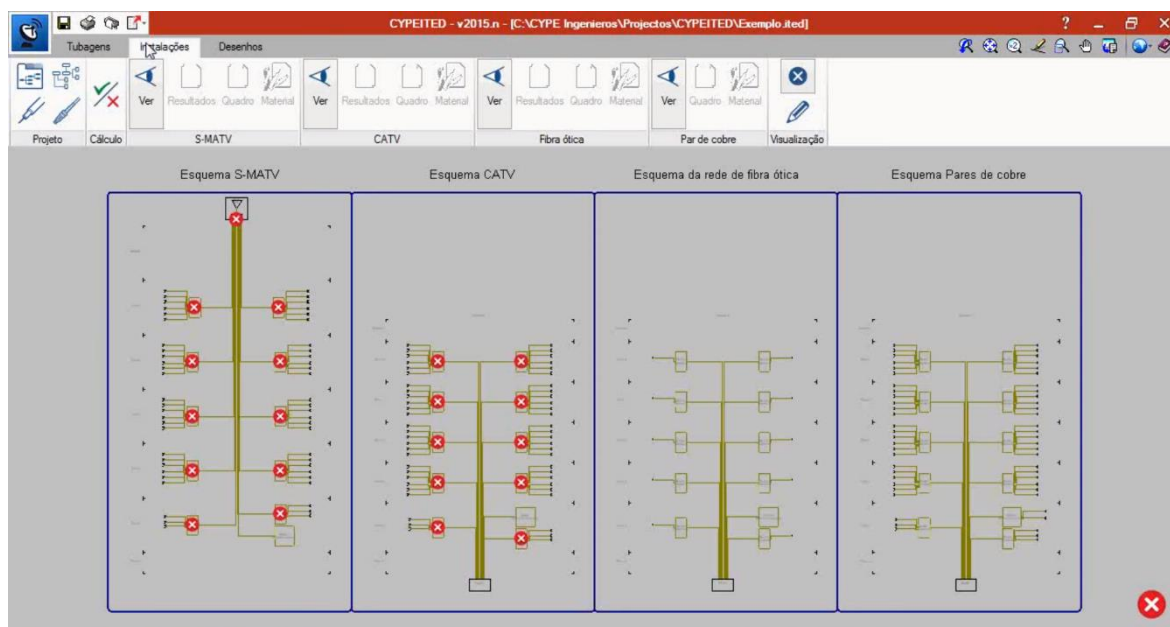


Figura 3 - setor de instalações do CYPEITED

Os próximos passos são realizados no segundo setor (Instalações). Aqui são apresentados quatro diagramas de instalações das tecnologias: S-MATV (*Single Master Antenna Television*), CATV (*Community Antenna Television*), Fibra Ótica e Pares de Cobre.

O diagrama de S-MATV representa a instalação das tecnologias que permitem a receção e distribuição de canais de TV/FM [8] [9] em edifícios, através da receção de sinal via antena.

O diagrama de CATV representa a instalação das tecnologias que permitem a receção e distribuição de canais de TV via cabo [3].

O diagrama de fibra ótica representa a instalação da tecnologia que fornece o serviço de ligação à *Internet* [3].

O diagrama de Par de Cobre representa instalação das redes que suportam os serviços de voz, dados, internet e IPTV (*Internet Protocol Television*) [10].

No esquema de S-MATV é possível editar os valores de ganhos e atenuações nos amplificadores e repartidores e o número de saídas neste último.

No esquema de CATV é possível definir no distribuidor de cada fração, mais uma vez, o número de saídas e ainda o valor da perda de sinal por inserção.

Esta restrição faz com que seja necessário executar um comando de verificação no final de configurados os valores indicados. O comando apresenta depois o resultado dos valores dos ganhos e das atenuações nas tomadas terminais para redes S-MATV e CATV e ainda das perdas de sinal na rede de Fibra Ótica.

Finalmente é gerado um inventário com a dimensão dos cabos e com os valores das tomadas terminais referidos anteriormente. É também apresentada uma secção onde são ilustrados os desenhos das redes com os valores introduzidos no seu desenho.

Terminada esta fase passa-se à secção de desenhos (Figura 4).

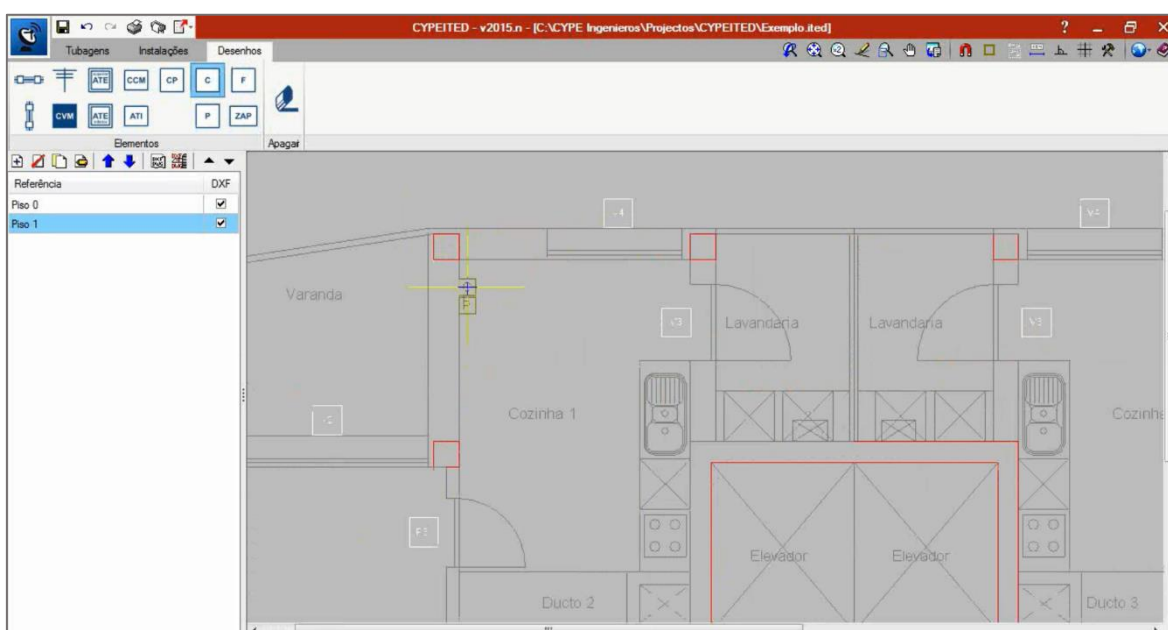


Figura 4 - setor de desenhos do CYPEITED

Aqui é possível importar ficheiros DWG (*Drawing*) e desenhar livremente diagramas das redes sobre eles. No entanto é necessário inserir manualmente os componentes, as tubagens e respetivos comprimentos.

Pontos fortes

Apesar de não ser possível inserir vários tubos na ligação entre dois componentes existe uma ferramenta de verificação de tubagem. Esta ferramenta verifica se as tubagens inseridas possuem diâmetros adequados à quantidade de cabos que por elas passam. Esta funcionalidade ajuda a mitigar erros que futuramente teriam que ser corrigidos, na fase de instalação da infraestrutura, auxiliando na redução de custos.

Em edifícios de construção vertical, em muitos casos, as frações adotam a mesma estrutura da instalação de andar para andar, tornando o processo de desenho das infraestruturas repetitivo. A ferramenta de duplicação de frações vem combater este problema ajudando, tanto a reduzir o tempo de desenho dos diagramas, como a reduzir a probabilidade de introdução de erros pelo projetista.

Pontos fracos

O método de introdução de tubagem oferecido promove a introdução de erros por parte do utilizador, sem que haja um mecanismo de verificação dos dados introduzidos.

Outro problema encontrado tem a ver com o diâmetro dos tubos que é possível introduzir numa ligação. Numa instalação real deve ser possível passar diferentes tubos numa ligação entre dois componentes. Isto é feito para separar tecnologias ou para reservar espaço para instalação futura de novas ligações.

O programa só permite a introdução de um diâmetro de tubo o que pode introduzir problemas na instalação da tecnologia na infraestrutura. Além disso, não é oferecida a possibilidade de especificar a altura a que os componentes são instalados na parede do edifício, o que faz com que o comprimento da tubagem possa ser mal calculado.

Para cada fração é necessário editar manualmente o valor das saídas no repartidor, uma vez que este não configura os valores automaticamente com base no número de tomadas de cada fração.

Na fase de desenho livre encontra-se um problema. O desenho efetuado nesta secção não possui nenhuma relação com os desenhos efetuados na primeira secção do programa. Além disso, os valores de comprimento das ligações são introduzidos manualmente o que, mais uma vez,

permite a introdução de erros. Como se trata de um desenho livre é também possível desenhar um diagrama completamente diferente do desenho projetado na secção de Tubagens.

Tudo isto faz com que os desenhos produzidos sirvam apenas de referência para os instaladores e que não tenham mais nenhuma utilidade prática.

2.3 Televés Cast 60

O *Televés Cast 60*, como o nome indica, é um *software* disponibilizado pela empresa *Televés*, centrada no *design*, desenvolvimento e fabricação de equipamentos para distribuição de serviços de telecomunicações. Trata-se de um *software* de acesso gratuito, mediante registo prévio por *email*.

Deparamo-nos de imediato com um entrave na utilidade desta aplicação. Esta foi concebida segundo as normas ITED da 2ª versão, sendo esta de uma versão desatualizada. Ainda assim consideramos que a aplicação merece destaque, pelo que irá ser de seguida apresentada.

O primeiro passo na utilização da aplicação é a configuração personalizada. Neste passo é possível definir uma série de parâmetros a serem utilizados por defeito.

A configuração do sistema está dividida em 5 categorias. “Visualização”, “Tipo de Rede”, “Assistente ITED”, “Cor de Texto Geral” e “Cor de Texto dos Amplificadores” (Figura 5).

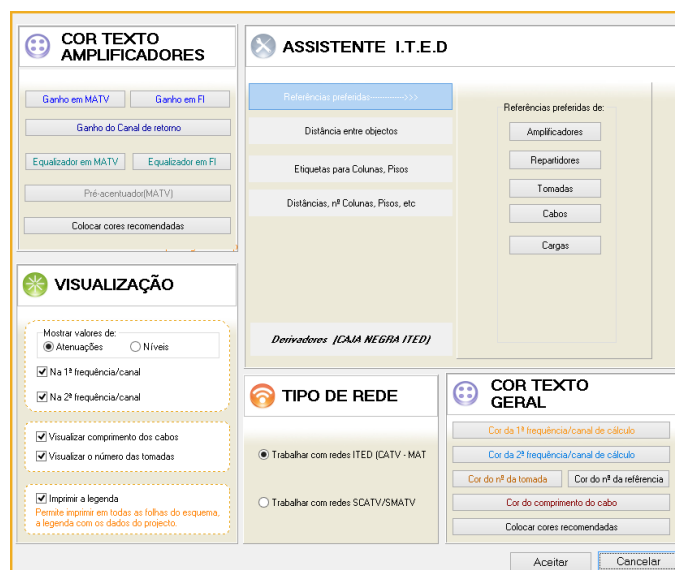


Figura 5 - configuração personalizada do software Televés Cast60

Em cada uma das categorias é possível configurar os seguintes parâmetros:

- **Visualização:**
 - Mostrar valores de atenuação / níveis de sinal.
 - Visualizar o número das tomadas.
 - Visualizar o comprimento dos cabos.
- **Tipo de rede:** Trabalhar com redes “ITED (CATV – MATV)” / “SCATV/SMATV”.
- **Assistente ITED:**
 - Referências preferidas: selecionar material por defeito para Amplificadores, Repartidores, Tomadas, Cabos e Cargas.
 - Distância entre objetos:
 - Distância vertical entre: plantas, frações, objetos e tomadas.
 - Distância horizontal entre: derivadores, derivador, objetos e pisos.
 - Etiquetas para colunas, pisos:
 - Modificar a cor das etiquetas para colunas montantes, pisos, frações e tomadas.
 - Modificar o texto das etiquetas para portais, plantas, frações e restantes tomadas.
 - Distancias, nº. Colunas, Pisos, etc.:
 - Definir os valores para:
 - Número de colunas montantes.
 - Número de pisos.
 - Número de frações.
 - Tipologia da fração.
 - Distância entre caixa de coluna e ATI.
 - Distância entre pisos.
 - Distância entre ATI e tomadas.

Nas outras duas categorias é apenas possível configurar a cor do texto de diferentes etiquetas.

Configurado o sistema passa-se ao próximo passo “Assistente ITED”. Aqui define-se inicialmente o número de colunas, o número de pisos e a distância do ATE até ao primeiro elemento da baixante.

É então gerado um diagrama utilizando as configurações introduzidas e passa-se para um ecrã onde é possível efetuar edições, entre outras operações (Figura 6).

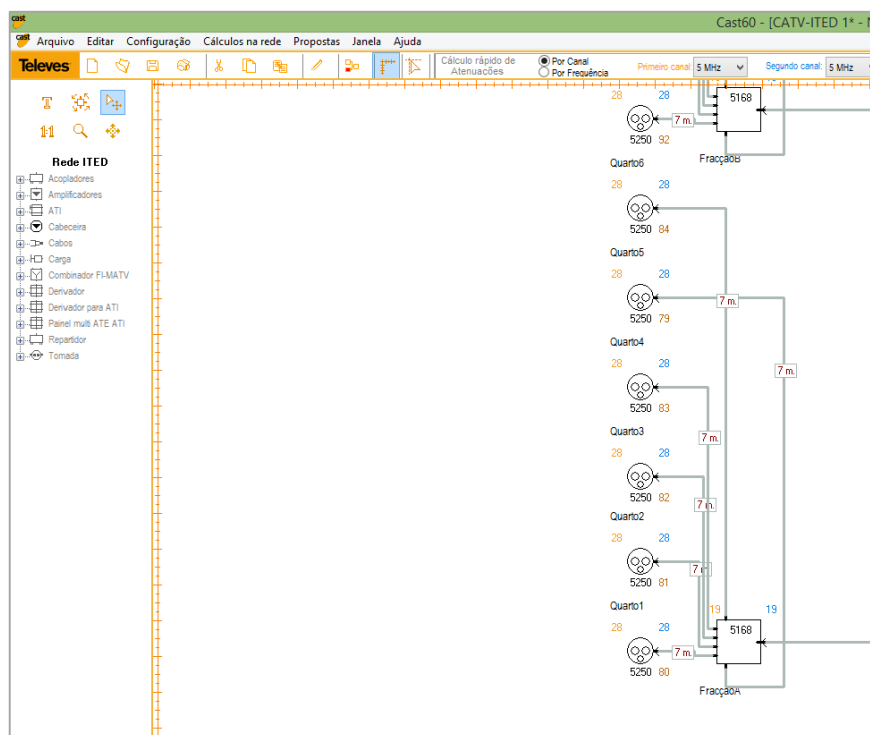


Figura 6 – Assistente ITED do software Televés Cast 60

Neste ecrã é possível executar ações como inserir novos componentes, arrastar e editar as propriedades dos já existentes, consultar uma tabela de cálculo de atenuações e de níveis de sinal nas tomadas e, ainda, gerar uma proposta da instalação para o cliente.

Pontos fortes

O elevado nível de personalização das *labels* e dos equipamentos a trabalhar por defeito, facilita bastante a utilização da aplicação e a perceção do utilizador sobre os componentes instalados.

A instalação de componentes utiliza como fonte equipamentos com referências reais ao invés de utilizar equipamentos genéricos com propriedades personalizáveis. Esta funcionalidade

facilita a orçamentação do projeto e ajuda a compreender as limitações dos equipamentos no desenho de infraestruturas.

Pontos fracos

Ao testar a aplicação foram encontrados alguns problemas, principalmente a nível de interface e interações com o utilizador.

Alguns elementos da barra de ferramentas superior têm problemas de renderização e é impossível arrastar algumas janelas. Outro problema encontrado foi a falta da possibilidade de retroceder e refazer ações de edição por parte do utilizador.

No entanto, as maiores restrições da aplicação não são tanto a nível da interface e das interações mas mais a nível das funcionalidades disponibilizadas.

Apesar de ser possível adicionar tipos de cabos diferentes através das suas referências, é difícil de visualizar essa informação e é impossível de compreender, de forma imediata, o panorama geral das tecnologias de cabos instaladas.

Além disso a instalação de tubagem não está contemplada, algo que é obrigatório nas normas ITED da terceira versão.

Outra funcionalidade em falta é a possibilidade de desenhar o diagrama sobre uma planta, de forma a poder posicionar os componentes nos respetivos lugares e a calcular de forma automática o comprimento das ligações efetuadas.

Todos estes fatores, em conjunto, limitam severamente o desenho de diagramas de redes ITED e irão inevitavelmente fazer com que sejam introduzidos erros pelo utilizador, comprometendo os valores de atenuação e de níveis de sinal nas tomadas e consequentemente a qualidade da instalação.

2.4 Comparação entre aplicações

A tabela seguinte pretende apresentar uma comparação sumarizada dos pontos fortes e fracos entre o *software* analisado previamente.

Funcionalidade/Aplicação	CYPEITED	Televés Cast 60
Versão atual de ITED	Sim	Não
Utilização gratuita	Não	Sim
Desenhar diretamente sobre uma planta	Não	Não
Calcular atenuações	Sim	Sim
Calcular níveis de sinal	Sim	Sim
Inserir tubagens	Sim	Não
Inserir cabos discriminadamente	Não	Sim
Gerar um inventário	Sim	Sim
Aplicação <i>web</i>	Não	Não

De destacar que se tratam ambas de aplicações *desktop* e como tal não satisfazem os requisitos estabelecidos.

2.5 *User Interface Design*

O que é o *User Interface Design*? Pode-se definir-se *User Interface Design* como sendo uma área de *design* onde ocorrem interações entre o Humano e o programa de computador. O objetivo principal desta área é o de garantir que, as informações visuais transmitidas ao utilizador o irão ajudar a tomar melhores decisões, dando-lhe controlo total sobre as funcionalidades do programa [11].

Sendo assim, uma interface deve tornar a interação intuitiva, eficiente e ter uma componente estética apelativa, de modo a produzir o *output* desejado, minimizando o número de *outputs* não desejados pelo utilizador.

Tendo adquirido então estes conceitos, o que é que nos leva a concluir que a área de *User Interface Design* se torna importante no desenvolvimento de uma aplicação *web*?

Há alguns anos as aplicações eram desenvolvidas tendo em conta *layouts* para máquinas do tipo *Desktop*. Nos dias que correm este paradigma tem vindo a alterar-se, com a introdução no mercado de outros tipos de dispositivos (*smartphones, tablets*) mais compactos e móveis e com uma variedade alargada de sistemas operativos e de navegadores de *internet*.

Isto levou o conceito de *User Interface Design* a ter que se adaptar. Quando se desenvolve uma interface, agora é necessário considerar que *layout* aplicar, que detalhes e informação apresentar e que ferramentas e tipo de interação se oferece ao utilizador, tendo em conta o dispositivo que este utiliza para aceder à aplicação.

De modo a construir uma boa interface existe um determinado conjunto de boas práticas que deve ser seguido [12], tais como:

- **Apelo visual:** deve partir de técnicas como alinhar e fornecer contraste entre elementos do ecrã, agrupar informação e usar cores e gráficos de forma simples e eficiente.
- **Clareza:** a interface deve ser visualmente e linguisticamente bem definida.
- **Compatibilidade:** deve ser garantida compatibilidade entre o utilizador, a tarefa que ele pretende realizar e a plataforma de acesso.
- **Compreensibilidade:** o sistema deve ser facilmente compreendido, respondendo a questões como: para o que olhar? O que fazer? Quando o fazer? Onde fazer? Porquê fazer? Como o fazer?
- **Configurabilidade:** permitir personalização e configuração de definições.
- **Consistência:** o sistema deve parecer, agir e operar da mesma maneira de forma global.
- **Controlo:** o utilizador deve controlar a interação. As ações devem partir de pedidos explícitos do utilizador, devem ser executadas rapidamente e ter a capacidade de serem interrompidas. Os meios de executar ações devem ainda estar ao alcance das capacidades do utilizador.
- **Eficiência:** minimizar movimentos dos olhos e mãos para controlar ações. O sistema deve antecipar as necessidades do utilizador.

- **Familiaridade:** usar linguagem e conceitos familiares ao utilizador e metáforas visuais sempre que possível.
- **Flexibilidade:** o sistema deve ser flexível em relação aos conhecimentos, capacidade e experiência do utilizador.
- **Previsibilidade:** a interface deve ser construída de modo a que o utilizador consiga antecipar a progressão natural de cada tarefa.
- **Recuperabilidade:** o sistema deve permitir que ações sejam reversíveis e que o utilizador nunca perca o trabalho desenvolvido.
- **Responsividade:** o sistema deve fornecer *feedback* (visual, textual ou auditório) ao utilizador para cada ação que execute.
- **Simplicidade:** implementar uma interface o mais simples possível mantendo as funcionalidades desejadas.
- **Tolerância:** o sistema deve tolerar e prevenir erros introduzidos pelo utilizador. Deve proteger-se contra erros catastróficos e, quando ocorrer um erro exibir mensagens construtivas.
- **Transparência:** o sistema deve permitir ao utilizador concentra-se na sua tarefa sem necessitar de se preocupar com a sua mecânica da interface.

3 Tecnologias utilizadas

Neste capítulo irão ser apresentadas as principais tecnologias e ferramentas que foram utilizadas no desenvolvimento deste projeto, procurando explicar brevemente as suas funcionalidades assim como a sua utilização e, em casos particulares, uma justificação para a sua escolha, face a outras alternativas.

Face à listagem de tecnologias ser extensa, apenas serão apresentadas de seguida as menos “comuns”, sendo que as restantes serão apresentadas na secção de anexo 10.1.

3.1 Ferramentas Web

3.1.1 *Twig*

Twig é um motor de *templates* para páginas PHP (*PHP: Hypertext Processor*). É rápido pois é compilado em código PHP otimizado. É flexível pois permite a criação de *tags* próprias. Um *template* é um ficheiro de texto simples que permite gerar código de diversas linguagens. O *template* contém *tags* que controlam a lógica e possui expressões ou variáveis que são substituídas por valores durante a compilação. Será utilizado na aplicação para gerar os *layouts* das páginas *web* [13].

Um exemplo de código que representa a criação de um *layout* de uma página *HTML* pode ser visto abaixo na Tabela 1.

```
{% extends "layout.html" %}

{% block content %}
    Content of the page...
{% endblock %}
```

Tabela 1 - excerto de código de *Twig*

3.2 Ferramentas de trabalho vetorial

3.2.1 D3.js

Data-Driven Documents (D3) é uma biblioteca *JavaScript* que permite gerar visualizações dinâmicas de informação em ambiente *web*, fazendo uso principalmente de *HTML* (*Hypertext Markup Language*), *CSS* (*Cascading Style Sheet*) e *SVG* (*Scalable Vector Graphics*) na sua criação [14].

O princípio base do seu funcionamento está na associação de elementos *DOM* (*Document Object Model*) das páginas *HTML* a dados.

Para além dos tipos de gráficos convencionais (barras, linhas e circular) oferece um vasto leque de opções de visualização como *TreeMaps*, Diagramas de *Voronoi*, Dendrogramas, entre outros (Figura 7), e dá ao utilizador o poder de manipular, apresentar e criar interações com informação através de mecanismos de seleção de elementos *DOM*/dados, animações baseadas em transições e de associação de dados a elementos *DOM*.

Será utilizado na aplicação para apresentar e fornecer mecanismos de interação na página de desenho dos diagramas *ITED*.

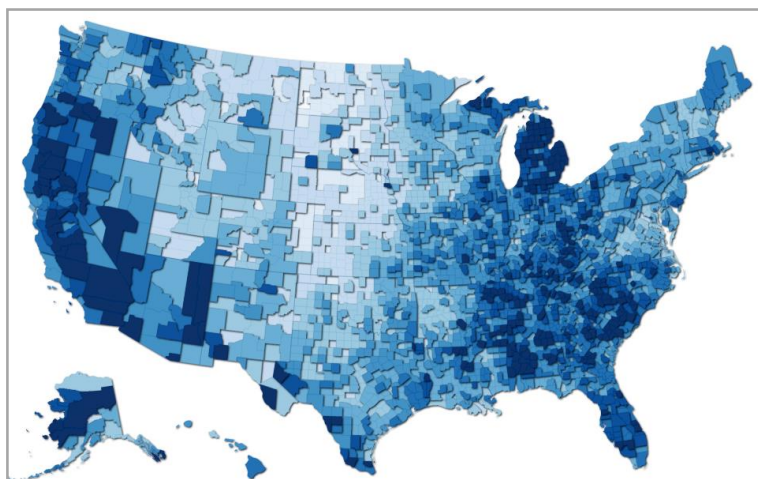


Figura 7 - exemplo de uma infografia criada através da biblioteca *D3.js*

3.2.2 *Dagre-d3*

Dagre-d3 é uma biblioteca *JavaScript* que serve de *front-end* para a biblioteca *Dagre* que, por sua vez, auxilia o processo de criação de gráficos hierárquicos direcionais, recorrendo à biblioteca *D3.js* como mecanismo de renderização dos gráficos em páginas HTML [15]. Será utilizado para gerar diagramas na aplicação.

Um exemplo da utilização desta biblioteca para gerar um gráfico direcional pode ser observado abaixo na Tabela 2.

```
// create a new directed graph
var g = new dagreD3.graphlib.Graph().setGraph({});

g.setNode("1", { label: "node 1" });
g.setNode("2", { label: "node 2" });
g.setNode("3", { label: "node 3" });
g.setNode("4", { label: "node 4" });
g.setEdge("1", "2", {arrowhead: "normal", label: "edge 1"});
g.setEdge("2", "3", {arrowhead: "normal", label: "edge 2"});
g.setEdge("2", "4", {arrowhead: "normal", label: "edge 3"});

// create the renderer
var render = new dagreD3.render();

// run the renderer. this is what draws the final graph.
var svg = d3.select("svg"), inner = svg.select("g");
render(inner, g);
```

Tabela 2 - excerto de código de utilização da biblioteca *Dagre-d3*

O excerto de código apresentado acima gera um grafo com a seguinte aparência:

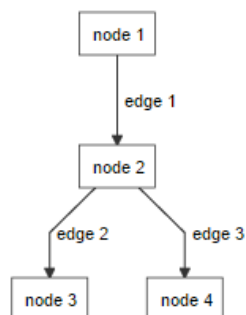


Figura 8 - exemplo de um grafo gerado pela biblioteca *dagre-d3*

3.2.3 *Svg-pan-zoom*

Svg-pan-zoom é uma biblioteca *JavaScript* que fornece funcionalidades de arrastamento e *zoom* em elementos *SVG*, através de eventos de rato ou eventos definidos pelo utilizador [16]. Oferece ainda um leque de opções de personalização da ferramenta.

Será utilizado para fornecer ações de *zoom* e arrastamento na página de desenho do diagrama ITED.

Um exemplo da utilização desta biblioteca para implementação dos mecanismos de arrastamento e de *zoom* pode ser visualizado abaixo na Tabela 3.

```
var panZoom = svgPanZoom(element, {  
  panEnabled: true,  
  zoomEnabled: true,  
  dblClickZoomEnabled: false,  
  controlIconsEnabled: true,  
  center: true,  
  fit: true,  
  minZoom: 0,  
  maxZoom: 9999  
});
```

Tabela 3 - excerto de código de utilização da biblioteca *Svg-pan-zoom*

3.2.4 *Raphaël*

Raphaël é uma biblioteca *JavaScript* que pretende simplificar a manipulação de vetores em ambiente *web* [17]. Utiliza *SVG* e *Vector Markup Language* (*VML*) como tecnologias base na criação de gráficos.

Será utilizado para prever a largura de texto *SVG* na página de desenho do diagrama ITED.

3.3 Ferramentas de desenvolvimento

3.3.1 *Symfony*

Symfony trata-se de uma *framework*, de uso gratuito, orientada à *web*, mais especificamente: uma *framework* para desenvolver aplicações MVC (*Model-View-Controller*) em ambiente *web* PHP [18]. O seu objetivo principal é agilizar o desenvolvimento e manutenção de aplicações *web* fornecendo ferramentas, divididas em componentes e dando controlo total ao utilizador sobre as suas configurações. O mecanismo de *templating* do *Symfony* é o *Twig*, previamente abordado.

4 Conceção e Desenvolvimento

Neste capítulo irão ser apresentados os passos percorridos, desde a idealização até à conceção de um protótipo funcional.

4.1 Levantamento de requisitos

Antes de se proceder ao desenvolvimento de qualquer implementação do sistema foi necessário efetuar uma análise dos requisitos que seriam necessários na aplicação para utilização das entidades interessadas.

Numa fase inicial debateu-se que seria interessante desenvolver uma aplicação *web* onde fosse simplesmente possível, sobre o desenho de uma planta de uma infraestrutura, desenhar ligações entre caixas que representassem os diversos tipos de componentes de uma instalação ITED.

Foram-se efetuando diversas sessões de *brainstorming* e de levantamento de requisitos, com elementos da empresa que solicitou o desenvolvimento da aplicação, e chegou-se à conclusão que seria interessante desenvolver um sistema um pouco mais complexo.

Em primeiro lugar, como a aplicação se destinaria maioritariamente a projetistas de infraestruturas de redes de comunicação, seria necessário desenvolver um mecanismo em que fosse possível organizar plantas por projeto e outro onde fosse possível efetuar desenhos de diagramas ITED sobre as referidas plantas.

Uma visualização da utilização deste sistema pode ser observada abaixo na Figura 9.

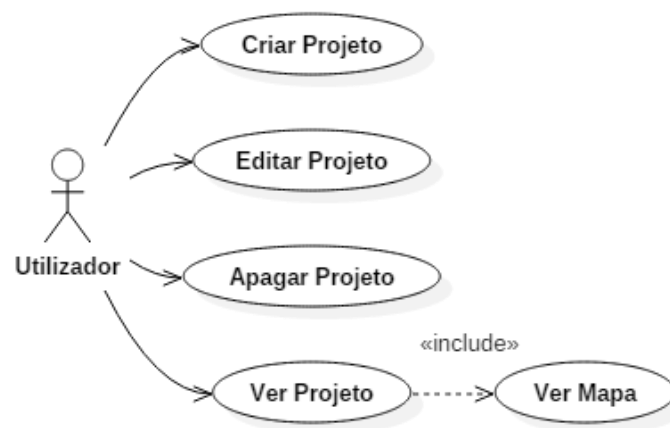


Figura 9 – diagrama de casos de usos de gestão de projetos

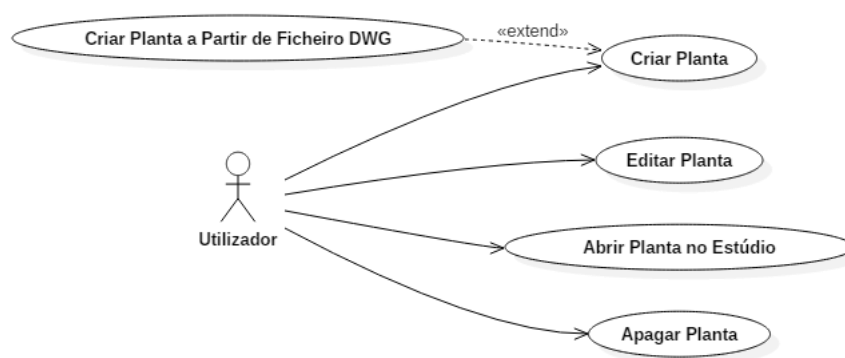


Figura 10 - diagrama de casos de uso de gestão de plantas

De maneira a organizar plantas seria então fornecido um mecanismo onde fosse possível a criação, edição e remoção de projetos. Neles deveria ser ainda possível a criação, edição e remoção de plantas.

Ao criar uma planta seria necessário implementar um sistema que recebesse um ficheiro DWG, contendo o desenho da planta, e que a convertesse para um formato consumível e manipulável por *browsers*, neste caso o formato SVG.

Deveria ser ainda possível ao utilizador visualizar um mapa com marcadores que indicassem a localização dos projetos em que ele está a trabalhar ou já trabalhou.

No entanto, o foco principal da aplicação passaria pela construção de um “estúdio” que fornecesse ferramentas para o desenho dos ditos diagramas ITED. Neste estúdio deveria ser possível inserir componentes, efetuar ligações entre eles e, nas ligações, inserir tubagem e cabos na tubagem.

Uma visualização da utilização deste sistema pode ser observada abaixo na Figura 11.

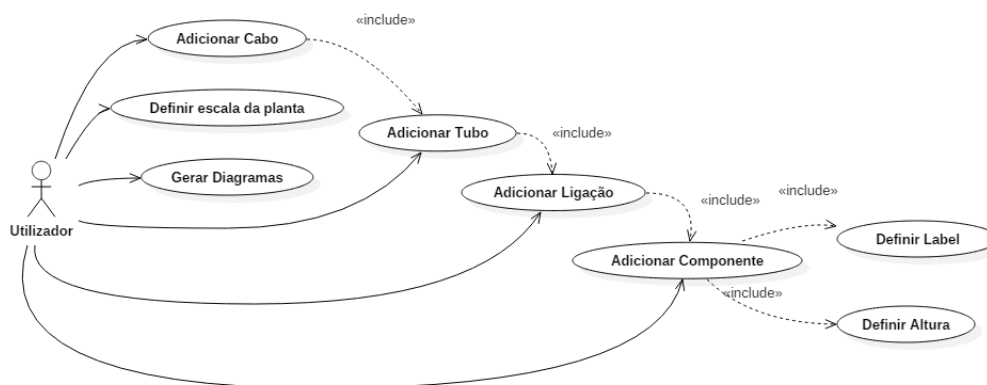


Figura 11 - diagrama de casos de uso do estúdio

Ao debater ideias chegou-se à conclusão que, da listagem de todos os tipos de equipamentos presentes no manual de ITED 3 [3] (versão atual e pela qual nos devemos reger), só seria interessante disponibilizar alguns: os módulos principais de uma instalação. Os restantes equipamentos seriam subcomponentes destes e a sua adição à instalação seria feita numa fase posterior.

Da listagem de tipos de componentes principais fazem então parte:

- **Armário de Telecomunicações de Edifício (ATE):** ponto de distribuição instalado na rede coletiva, ou na rede individual de edifícios não residenciais, onde se encontram alojados os repartidores gerais, que recebe os cabos provenientes dos operadores, ou das Instalações de Telecomunicações em Urbanizações (ITUR), distribuindo-os pelo edifício [3].
- **Armário de Telecomunicações Individual (ATI):** ponto de distribuição instalado na rede individual, onde se encontram alojados os repartidores de cliente, que permite a interligação entre redes, permitindo a gestão das telecomunicações individuais [3].

- **Caixa de Entrada de Moradia Unifamiliar (CEMU):** caixa de acesso restrito, instalada entre a CVM e o ATI, numa moradia unifamiliar. A CEMU é de instalação opcional e tem como função abrigar os dispositivos que terminam os cabos provenientes do ATI, ou como caixa de entrada, facilitando a manobra futura de cabos de operador [3].
- **Caixa de Passagem (CP):** caixa destinada a facilitar o enfiamento de cabos [3].
- **Câmara de Visita Multioperador (CVM):** compartimento de acesso aos troços de tubagem subterrâneos, geralmente no exterior dos edifícios, para seu uso exclusivo, através do qual é possível instalar, retirar e ligar cabos e proceder a trabalhos de manutenção [3].
- **Quadro Elétrico (QE):** quadro elétrico de um ATE ou de um ATI.
- **Passagem Aérea de Topo (PAT):** tubagem que permite a passagem de cabos para ligação às antenas, instaladas normalmente no topo do edifício [3].
- **Tomada de Telecomunicações/Terminal (TT):** extremo da rede individual de cliente onde se prevê a ligação de um equipamento, por via do respetivo chicote.

Como extra deveria ser possível inserir um componente de “curva” para representar curvas nas tubagens.

Tipicamente no desenho de diagramas, é adotada uma vista superior da planta. Isto faz com que se trabalhe sobre um ambiente de duas dimensões em que o eixo da altura é ignorado. Para evitar esta situação, e futuros problemas na instalação, foi previsto na aplicação que os componentes deveriam ter uma propriedade “altura” onde fosse especificada a distância vertical a que serão instalados.

Para alguns tipos de componentes, nomeadamente para o ATI, é ainda necessário ter uma opção onde seja possível definir o nível de sinal de entrada para que seja possível o cálculo de níveis de sinal nas tomadas terminais.

Uma vez que podem ser colocados múltiplos componentes do mesmo tipo, por desenho, surgiu a necessidade de os identificar univocamente. Sendo assim foram avançadas duas soluções: a primeira foi atribuir automaticamente um número a cada componente (por tipo de componente e ordem de colocação). A segunda opção foi dar a possibilidade ao utilizador de atribuir uma *label* ao componente, que melhor o identifique.

Em relação às ligações, o seu comprimento deveria ser calculado automaticamente com base na posição dos componentes que a compõem e com base na escala da planta. Além disso deveria existir uma restrição que impedisse o utilizador de criar ciclos fechados de ligações.

No que diz respeito às tubagens, deveria ser possível inserir várias, de diversos diâmetros (20, 25, 32, 40, 60, 63, 90 e 100 mm), por ligação.

Os cabos adotariam o mesmo princípio das tubagens, devendo ser possível passar vários cabos, de diversas tecnologias, em cada tubo. As tecnologias de cabos que deveriam ser disponibilizadas seriam as tecnologias de Cabo Coaxial, de Fibra Ótica e de Par de Cobre.

O produto final do desenho dos diagramas seria a sua representação num esquema, com os níveis de sinal e de atenuação nas tomadas terminais.

Os cálculos das atenuações e de níveis de sinal, para além de outros fatores, dependem das atenuações dos equipamentos instalados. Deste modo, e de maneira a servir os propósitos de dois tipos de entidades (empresa e projetista) surgiu a necessidade da criação de uma ferramenta de sugestão de equipamentos (da empresa) ao projetista, a serem instalados. Esta sugestão de equipamento ocorreria entre a fase de desenho e a fase onde são gerados os diagramas.

De maneira a representar a coluna montante do edifício (caso este tenha uma) existe a necessidade de criar plantas verticais. Para representar a coluna montante do edifício deve ser possível associar as plantas horizontais do projeto a diversos pisos da planta vertical. Isto deveria ser feito através da ligação dos ATI das plantas horizontais aos ATE da planta vertical.

Efetuada estas ligações irá ser possível calcular corretamente as atenuações desde os extremos das ligações superiores e inferiores do edifício até às tomadas terminais das plantas horizontais.

Passando a outro aspeto da aplicação, deveria ainda ser possível à empresa editar o conteúdo apresentado na aplicação *web*, em especial a lista de equipamento a disponibilizar.

Sendo assim ficou definida a necessidade da criação de um *BackOffice* acessível apenas através de uma conta com privilégios de administração. Ficou também estabelecido que um administrador não poderia ver nem editar as plantas criadas por outros utilizadores.

No que diz respeito às contas de administração, para além da existência de uma conta única de *Super Administrador* deveria ser possível a criação de múltiplas contas de Administrador. O *Super Administrador* seria o único com privilégios para gerir contas de utilizador.

A empresa para que a aplicação está a ser desenvolvida possui um centro de aplicações que é acessível através de contas de utilizador criadas na sua página *web*. De maneira a reutilizar estas contas e a fornecer um mecanismo de autenticação genérico para outras entidades surgiu a

necessidade de desenvolver um meio de autenticação externo para validar as contas de utilizador e fornecer acesso à aplicação.

4.2 Arquitetura

Na presente dissertação pretende-se criar uma plataforma acessível em qualquer altura, a partir de qualquer localização, por vários utilizadores. A arquitetura do sistema deve por isso suportar estes requisitos.

Se um projetista visitar uma obra deve poder, mediante uma ligação à internet, obter facilmente os desenhos da instalação e dos diagramas, a fim de poder avaliar a coerência entre o diagrama que foi projetado e o que está a ser instalado, de maneira a poder efetuar alterações se necessário. Sendo assim, torna-se um requisito da aplicação o acesso em tempo real aos dados armazenados.

De maneira a proporcionar esta funcionalidade, a arquitetura proposta é uma arquitetura do tipo *Model View Controller* (MVC) conforme esquematizada na Figura 12.

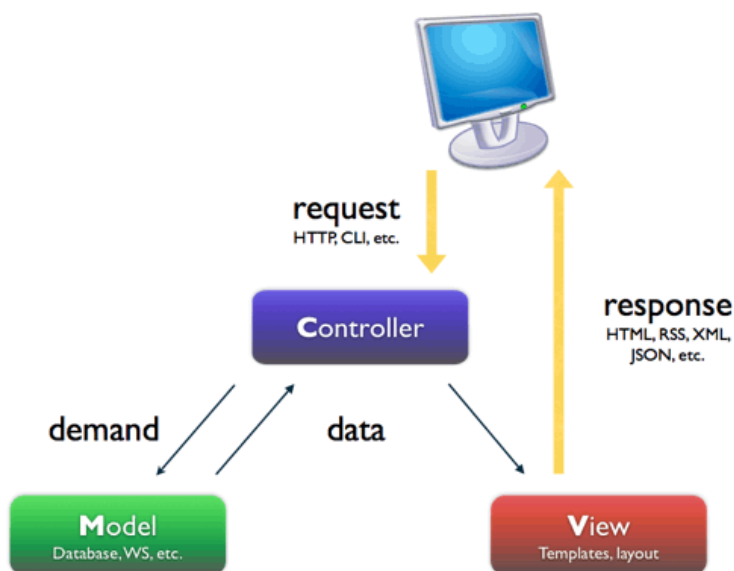


Figura 12 - arquitetura MCV do Symfony [19]

A camada *Model* define toda a lógica de negócio, incluindo a base de dados. A camada *View* é onde o utilizador efetua todas as interações com a aplicação.

A camada *Controller* é onde é executado código que efetua pedidos à camada *Model* e que passa os dados resultantes para a camada *View* para renderização.

Podemos dividir ainda a arquitetura em duas camadas, sendo elas a camada *front-end* e a camada *back-end*.

A primeira diz respeito à aplicação, como ela é vista pelo utilizador, no seu dispositivo. Trata-se portanto dos mecanismos que possibilitam ao utilizador realizar um conjunto de ações de interação com o sistema.

Já o *back-end* diz respeito às camadas que armazenam informação relevante e que são responsáveis pelos processos de consulta e de transporte de informação até à camada de apresentação *front-end*, mediante pedido pelo utilizador.

O Sistema de Gestão de Bases de Dados (SGBD) escolhido, para armazenar todos os dados da aplicação, foi uma base de dados relacional *MySQL* que utiliza, como o nome indica, a linguagem SQL (*Structured Query Language*).

Para consulta de informação, será criada uma série de serviços através dos controladores do *Symfony*, que permitem consultas com parâmetros através de um *HTTP Request* devolvendo a informação sobre a forma de um *HTTP Response*.

O tipo de respostas possíveis é variado, podendo tratar-se de uma página *web*, XML (*eXtensible Markup Language*), objeto JSON (*JavaScript Object Notation*) entre outros. Isto faz com que seja possível a apresentação de páginas de forma tradicional e também que seja possível efetuar pedidos *Ajax*, sobre a forma de um *WebService*, para consulta de informação dinâmica, sem ser preciso recarregar a página.

O processo de comunicação através de um pedido *Ajax* é efetuado da seguinte forma:

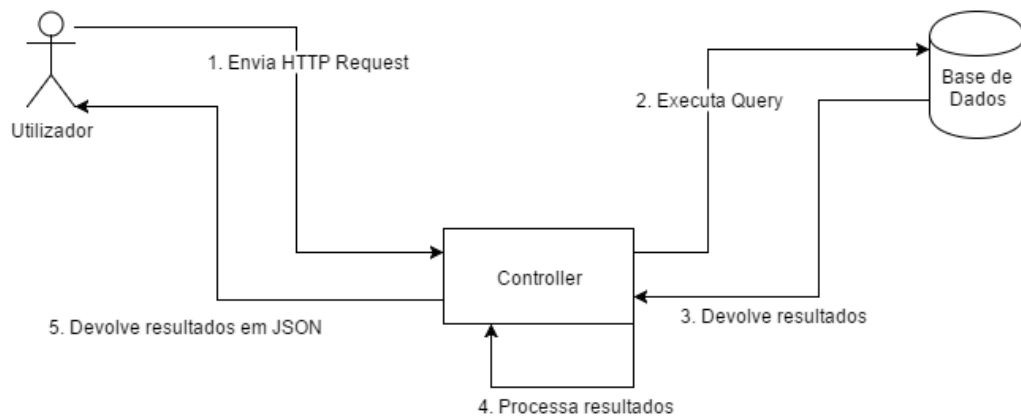


Figura 13 - Diagrama de sequência do processamento de uma Ajax Request

4.3 Implementação

De seguida irão ser explicados os mecanismos utilizados e o modo como foi implementada a arquitetura do sistema apresentada no capítulo anterior.

4.3.1 Modelo de base de dados

Seguindo os bons princípios do *design* de bases de dados como a normalização (processo de organização de tabelas e colunas de modo a evitar redundância de dados) entre outros, a construção da base de dados do sistema foi executada tendo em conta a informação que se pretende armazenar ao invés de pensar a estrutura de informação, de modo a simplesmente satisfazer os requisitos da aplicação.

Para definir inicialmente a estrutura dos dados a armazenar foi desenvolvido um diagrama UML (*Unified Modeling Language*) de classes que pode ser visualizado de seguida (Figura 14).

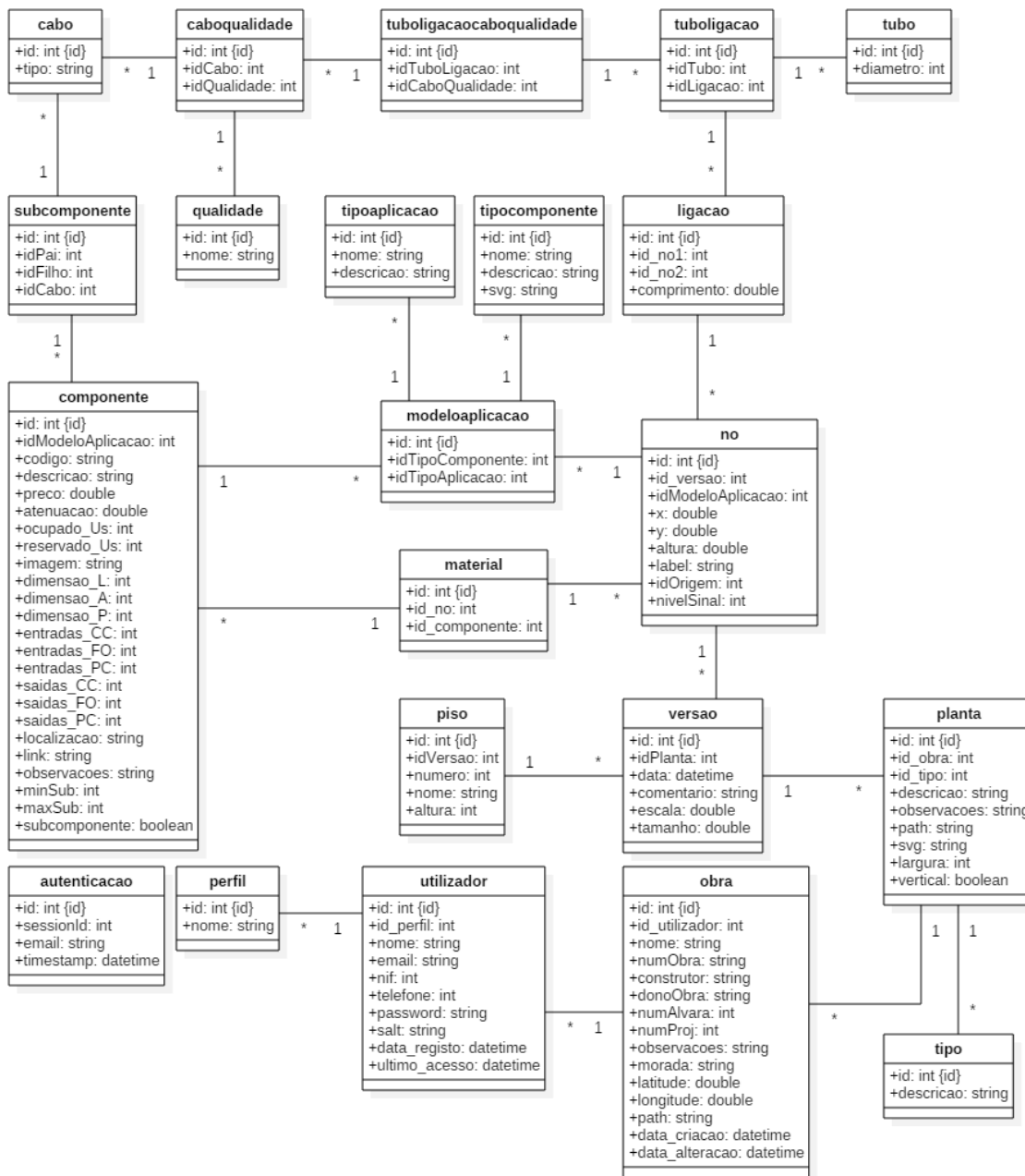


Figura 14 - diagrama de classes da aplicação

Tendo o diagrama de classes como base foi então construída a base de dados da aplicação, tendo-se chegado ao seguinte resultando, apresentado no diagrama abaixo (Figura 15).

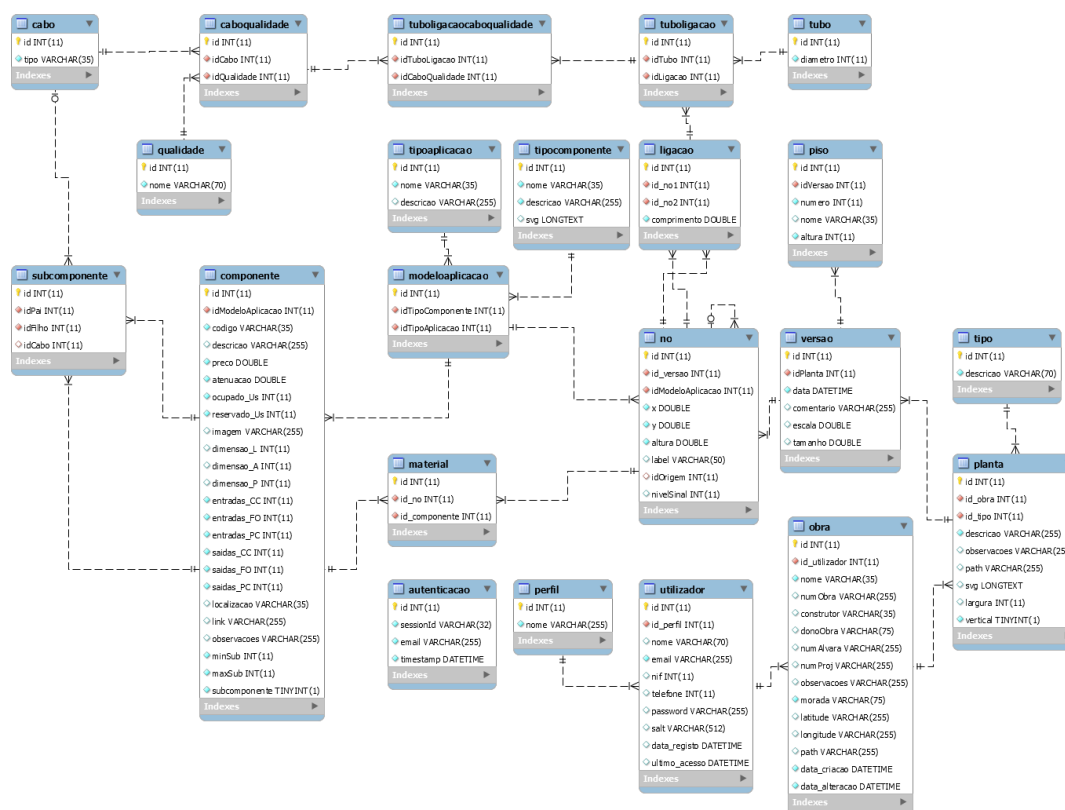


Figura 15 - vista geral da base de dados

A partir da vista geral da base de dados apresentada acima, podemos agrupar as tabelas em diversos setores.

As tabelas “perfil”, “utilizador” e “autenticação” dizem respeito às contas dos utilizadores. O perfil indica que tipo de função o utilizador vai desempenhar no sistema, atribuindo ou negando acesso a diferentes áreas da aplicação. A tabela utilizador armazena as credenciais de acesso e dados úteis do indivíduo/entidade. A tabela autenticação armazena informação temporária que permite ao sistema confirmar que o acesso a uma conta de utilizador é feito pelo próprio.

As tabelas “obra”, “planta” e “tipo” armazenam informação relativa aos projetos. Um projeto/obra contém informações relativas ao projeto e ao edifício físico em questão. Armazena ainda coordenadas geográficas da localização do edifício que serão posteriormente disponibilizadas num mapa e oferece a possibilidade de armazenar uma fotografia do projeto.

Cada obra pode possuir ainda um conjunto de plantas de diversos tipos. A tabela “planta” por sua vez armazena o desenho da planta num formato SVG que poderá posteriormente adotar diversas versões. Além disso, no caso de se tratar de uma planta vertical, é possível armazenar

informação relativa aos andares, nomeadamente número, altura e nome, informações armazenadas na tabela “pisso”.

As tabelas mencionadas de seguida retêm informação relativa à infraestrutura mencionada acima, sendo elas as tabelas “cabo”, “qualidade”, “caboqualidade”, “tubo”, “ligação”, “tuboligacao”, “tuboligacaocaboqualidade”, “tipoaplicacao”, “tipocomponente”, “modeloaplicacao” e “no”.

Passando ao mais específico, a tabela “tipocomponente” é onde irão ser armazenados os dados relativamente aos tipos de nós, que irão ser instalados no desenho da planta. A tabela “tipoaplicacao” armazena os dados relativos aos métodos de aplicação de um nó (mais especificamente para nós do tipo ATE e ATI). Por fim, a tabela “modeloaplicacao” relaciona o tipo de componente com o tipo de aplicação disponível.

O modelo de aplicação é depois associado à tabela “no” que armazena as coordenadas x e y de um nó, em determinada versão, assim como a altura, opcionalmente uma *label* descritiva e um id de origem que é usado para ligar dois nós em casos específicos que abordaremos posteriormente.

Entre dois nós pode existir uma ligação, representada na tabela com o mesmo nome (ligacao), que armazena ainda a distância entre eles.

Em cada ligação podem ser passados tubos. A tabela “tubo” é onde serão armazenadas as categorias de tubos disponíveis, definidas pelo seu diâmetro. E para finalizar a tabela “tuboligacao” faz uma associação entre a tabela “tubo” e “ligação”, indicando para cada ligação que tubos contém e de que categoria.

Em cada tubo é ainda possível passar vários cabos. A tabela “cabo” é onde irá ser armazenada informação relativa à tecnologia da infraestrutura a ser instalada (cabo coaxial, fibra ótica ou par de cobre). A tabela “qualidade” é onde serão armazenados os diferentes níveis de qualidade dos cabos e a tabela “caboqualidade” é onde é feita a relação entre as duas tabelas referidas anteriormente, associando um tipo de cabo a uma ou mais qualidades. Finalmente é na tabela “tuboligacaocaboqualidade” que será armazenada informação que indica que cabos e com que qualidade, é que passam num tubo inserido numa ligação entre dois nós, em determinada versão de uma planta.

Além da informação da infraestrutura, a base de dados dá ainda suporte ao armazenamento de informação relativa aos materiais a serem aplicados numa planta. Essa informação é armazenada nas tabelas “no”, “modeloaplicacao”, “material”, “componente” e “subcomponente”.

A tabela “componente” é onde serão armazenados os dados relativamente aos equipamentos disponíveis para compra. Cada componente possui um modelo de aplicação, assim como os nós instalados. A tabela “subcomponente” foi criada para indicar que componentes são na verdade subcomponentes de outro componente “pai” e que possuem, opcionalmente, uma limitação de instalação a um tipo de cabo (tecnologia). Por fim, a tabela “material” associa a cada nó, um componente e uma lista de subcomponentes.

4.3.2 *Symfony*

A *framework Symfony* foi essencial na conceção e desenvolvimento deste projeto uma vez que permite de forma fácil e personalizável o desenvolvimento de aplicações e de modelos de dados.

Devido à extensão das funcionalidades disponibilizadas por esta *framework*, encontra-se disponível na secção de Anexo C, uma descrição de algumas das funcionalidades por ela fornecidas e algumas configurações que foram implementadas no desenvolvimento da aplicação.

5 Protótipo

A construção de um protótipo trata-se de uma fase essencial no desenvolvimento deste projeto. É nele que vão ser implementadas as funcionalidades projetadas nas fases anteriores, tendo como objetivo a produção de uma ferramenta útil e prática que possa ser testada por diversas entidades.

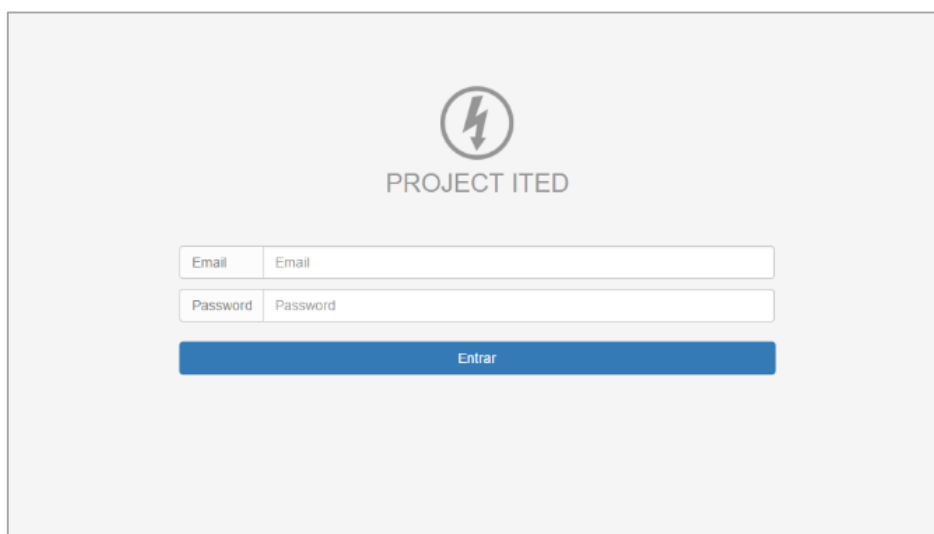
Este protótipo está dividido em três setores distintos, de modo a adaptar-se às necessidades dos diversos tipos de utilizador e a três necessidades distintas.

Estes setores tratam-se do *FrontOffice*, do Estúdio e do *BackOffice* e serão apresentados de seguida.

5.1 FrontOffice

O *FrontOffice* apresenta a interface *Web* de interação com o utilizador dito “comum”. Dele fazem parte as interfaces de autenticação do utilizador, as interfaces de gestão das obras e das plantas e ainda a interface de edição da conta de utilizador.

Ao entrar no sistema, o utilizador depara-se com a página de autenticação ilustrada na Figura 16 apresentada abaixo:



A imagem mostra a interface de autenticação do sistema. No topo, há um ícone de um relâmpago dentro de um círculo, seguido pelo texto "PROJECT ITED". Abaixo disso, há dois campos de entrada: o primeiro é rotulado "Email" e o segundo "Password". Ambos os campos são retangulares e brancos. Abaixo dos campos, há um botão azul com o texto "Entrar" em branco.

Figura 16 - página de autenticação do utilizador

Para se autenticar, o utilizador deve introduzir o seu *email* e senha nos campos indicados e clicar no botão entrar para submeter os dados do formulário e autenticar-se no sistema.

A empresa para que a aplicação está a ser desenvolvida possui um centro de aplicações onde já existem utilizadores registados. De maneira a reutilizar estas contas (e possivelmente contas de utilizador de outras fontes) foi desenvolvido um sistema de autenticação externo e genérico, desacoplado da aplicação em si.

O processo de autenticação pode ser visualizado abaixo na Figura 17:

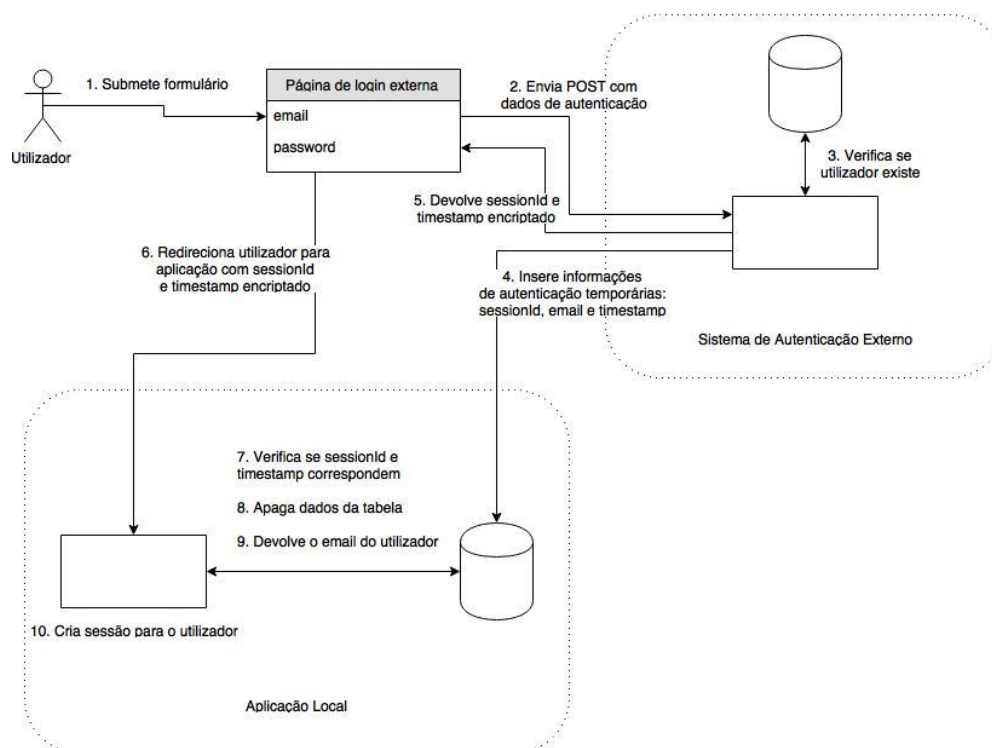


Figura 17 - processo de autenticação externa

Em primeiro lugar o utilizador envia os dados de autenticação (*email* e *password*) pelo formulário da página de autenticação externa.

O sistema depois envia através de um método POST estes dados para o sistema de autenticação externo.

Este sistema por sua vez faz uma consulta à base de dados da empresa que verifica se o utilizador existe. Caso não exista o utilizador é redirecionado de volta à página de autenticação externa para corrigir erros nas credenciais.

Caso o utilizador exista, o sistema envia, através de um método POST, as informações de autenticação temporárias (*sessionId*, *email* e *timestamp*) para a aplicação local que as armazena na tabela “Autenticacao”. Após executar esta ação redireciona o utilizador para o sistema local com os parâmetros *sessionId* e o *timestamp*, encriptando este último.

De seguida o sistema local confirma se ambos os *sessionId* correspondem e verifica se o *timestamp* recebido corresponde à encriptação do *timestamp* armazenado na base de dados local.

Caso os dados correspondam, é iniciada uma sessão utilizando a conta de *email* armazenada na tabela “Autenticação” e são apagados os dados da tabela correspondentes à tentativa de autenticação.

Além deste mecanismo de autenticação, o sistema suporta também a criação de contas, localmente através do *BackOffice*.

Dependendo do tipo de utilizador, após autenticação, este será redirecionado para uma página conforme os seus privilégios. Como estamos a abordar as interfaces do utilizador comum este seria redirecionado para a página de gestão de projetos, conforme apresentado na figura abaixo:

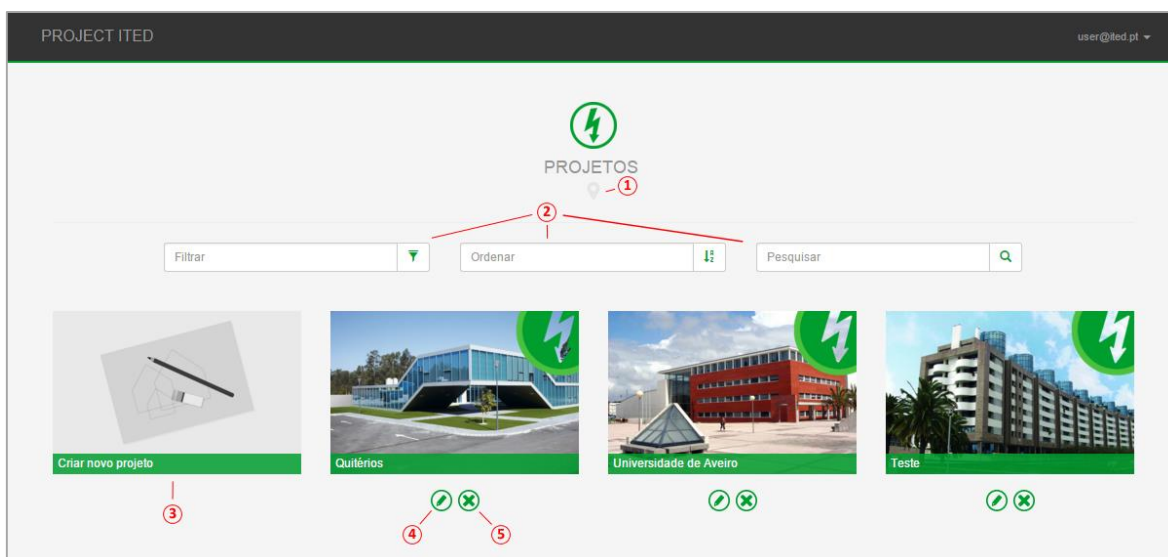


Figura 18 - página de gestão de projetos

Nesta página são portanto apresentados os projetos, que um utilizador criou, e são fornecidas ferramentas para a gestão e edição do seu conteúdo, assim como a criação de novos projetos.

Partindo de uma análise da página de cima para baixo temos como elemento número 1 o botão de um marcador de mapa. Ao clicar neste botão, o utilizador é redirecionado para uma página, onde é apresentado um mapa, com a localização dos projetos a que o utilizador tem acesso, permitindo obter uma imagem visual das áreas onde existe maior densidade de projetos e, possivelmente, de maiores oportunidades de negócio.

Continuando a análise da interface da página atual, são apresentadas, nos elementos assinalados a 2, três caixas de texto onde é possível filtrar os projetos existentes tanto por nome, como por tipo de projeto assim como efetuar a sua ordenação, facilitando o acesso aos projetos, por parte do utilizador.

De seguida são apresentadas caixas que representam os projetos. A primeira caixa (elemento número 3) representa sempre a ação de criar um novo projeto, que leva o utilizador a uma página onde é apresentado um formulário e as ferramentas necessárias para a sua criação.

As caixas seguintes representam os projetos já existentes. Um clique na imagem leva o utilizador à página de gestão do projeto, onde é apresentada informação do projeto e uma interface de gestão das plantas que nele estão contidas.

Um clique no elemento número 4 leva o utilizador à página de edição dos dados do projeto e, finalmente, um clique no elemento número 5 permite ao utilizador eliminar o projeto e todas as plantas associadas, mediante confirmação.

Carregando na imagem da obra passa-se então à página de gestão do projeto onde pode ser visualizada informação relativa ao projeto e uma listagem de plantas (Figura 19).

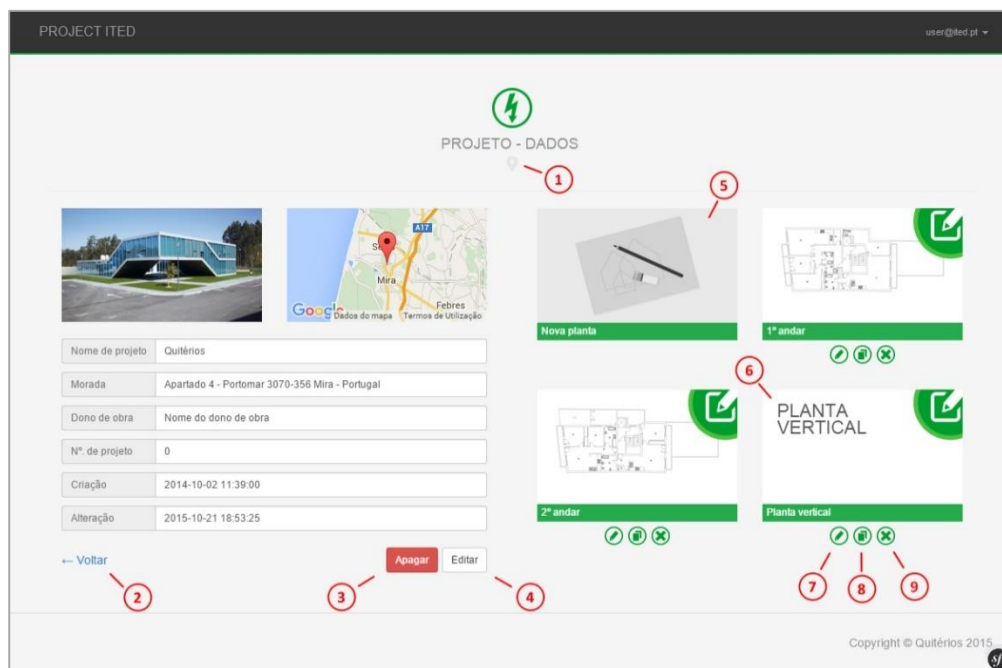


Figura 19 - página de gestão de um projeto

Existem algumas semelhanças face à página anterior em relação aos projetos.

Nesta página é possível: aceder ao mapa de projetos (1), voltar à página anterior (2), apagar o projeto e todas as suas plantas (3) e editar as informações do projeto (4).

Em relação às ações relacionadas com as plantas, é possível: criar novas plantas (5), editar as já existentes (7), duplicar plantas (8) e apagar plantas (9).

Ainda, como ação principal, é possível aceder ao estúdio (6), que se trata do ponto central desta aplicação e que será abordado em maior detalhe mais à frente.

Como já foi referido anteriormente é possível navegar, através da página de gestão dos projetos, para uma página dedicada a demonstrar visualmente a localização dos projetos do utilizador, representados por marcadores colocados sobre um mapa Google (1). Ao colocar o rato por cima do marcador é indicado o nome do projeto, como demonstrado (a vermelho) na Figura 20 apresentada abaixo:

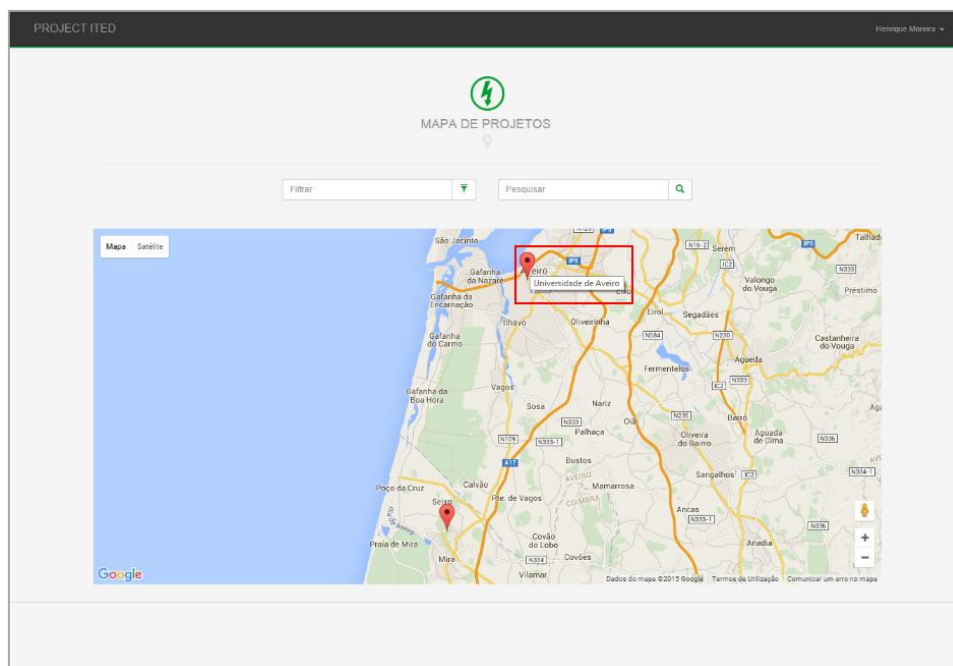


Figura 20 - página de mapa dos projetos de um utilizador

5.2 Estúdio

Como já foi referido anteriormente o estúdio trata-se do ponto central desta aplicação. É portanto de um editor de projetos ITED onde é possível trabalhar sobre uma planta *Computer Aided Design* (CAD), previamente carregada, ou, em casos especiais (como no caso de planta verticais), de construir a própria planta com características personalizáveis.

O desenho de um projeto ITED no estúdio encontra-se ele próprio dividido em três fases distintas sendo elas: o “Desenho do diagrama”, “Escolha do equipamento” e “Inventário e diagramas” que irão ser abordadas de seguida.

5.2.1 Desenho do diagrama

Como o próprio nome indica, a primeira fase da criação do projeto é onde se irá desenhar o diagrama, composto pelos componentes de uma instalação ITED e pelas tubagens e cabos que os ligam.

Dependendo do tipo de planta sobre o qual se está a desenhar, a interface do estúdio pode adotar duas variantes, a variante de uma planta horizontal e a variante de uma planta vertical. Cada uma tem um propósito específico.

Planta horizontal

A planta horizontal pode representar o diagrama de pisos individuais, caso se trate de um edifício, ou de secções ou totalidade de uma infraestrutura, caso se trate de uma infraestrutura com um piso apenas.

A interface do estúdio de uma planta horizontal caracteriza-se por ter já incluída uma planta, carregada previamente pelo utilizador. O funcionamento da aplicação do estúdio é principalmente à base de ações do tipo *drag & drop* de modo a facilitar a interação entre a aplicação e o utilizador.

Um exemplo da interface para uma planta horizontal pode ser visualizado abaixo conforme apresentado na Figura 21:

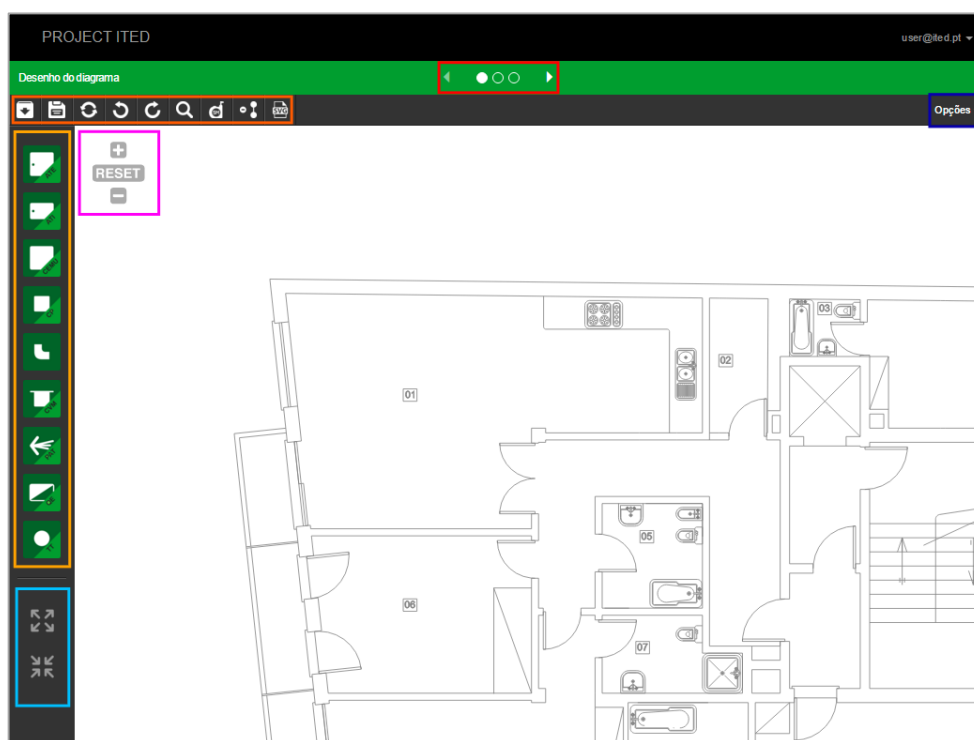


Figura 21 - exemplo da página do estúdio de uma planta horizontal

Como se pode visualizar na imagem apresentada, as ferramentas da aplicação encontram-se agrupadas em certas categorias:

- Na caixa a vermelho estão representadas as ferramentas de navegação no estúdio, com indicação da secção em que o utilizador se encontra.
- Na caixa a laranja estão as ferramentas de operações comuns na planta.
- Na caixa a amarelo estão apresentados os tipos de componentes que é possível instalar na planta.
- Na caixa a azul-claro encontram-se ferramentas cuja utilização varia conforme a planta ser vertical ou não.
- Na caixa a rosa encontram-se ferramentas de controlo do *zoom* da planta.
- Finalmente, na caixa a azul-escuro, encontram-se apresentadas opções de visualização na aplicação, mediante interação com um menu *dropdown*.

Partindo da análise das ferramentas representadas na caixa a laranja, temos as seguintes (da esquerda para a direita):

- **Carregar** abre uma janela que permite ao utilizador visualizar a lista de versões dos diagramas da planta, tendo como informação visível a data e hora de criação da versão, um comentário (opcional) e tem ainda as ações de carregar para o estúdio a versão escolhida e a opção de a eliminar.
- **Gravar** abre uma janela onde o utilizar pode, opcionalmente, inserir um comentário e gravar a versão atual do diagrama da planta.
- **Limpar** elimina todos os elementos inseridos pelo utilizador, apresentando apenas a planta sobre a qual se está a trabalhar.
- **Anular** retrocede as ações do utilizador, no diagrama da planta, uma vez (atalho *Ctrl+z*).
- **Repetir** avança as ações do utilizador, no diagrama da planta, uma vez (atalho *Ctrl+y*).
- **Inspecionar diagrama** percorre a estrutura do diagrama desenhado à procura de falhas, relatando o resultado do estado do projeto numa janela. Algumas das verificações efetuadas são:
 - Recomendar instalar uma CP quando um cabo tem mais de 15m.
 - Verificar se o comprimento de um cabo entre um ATE e um ATI tem menos de 90m.

- Verificar se todos os nós estão ligados.
- Verificar a instalação de uma PAT.
- Verificar se existem ciclos nas ligações.
- **Ferramenta de escala** permite redimensionar a escala da planta, uma vez que esta informação não vem incluída no ficheiro CAD que foi previamente enviado pelo utilizador. Esta ferramenta apresenta, no ecrã, uma barra ligada por dois pontos. Clicar na barra abre uma janela onde é possível configurar a distância que ela representa. Arrastar a barra reposiciona-a. Arrastar cada um dos pontos permite alterar o seu tamanho (mantendo a distância que ela representa) o que faz com que, tendo uma medida de referência na planta (ex. uma porta) se possa ajustar a escala da planta de acordo com o tamanho da barra representado no ecrã e a distância que o utilizador configurou.
- **Remover nós isolados** encontra componentes que não tenham ligação a outros e remove-os do diagrama. Esta funcionalidade pode ser útil quando existe um nó que o utilizador não encontre no diagrama.
- **Exportar diagrama** exporta o diagrama desenhado para um ficheiro SVG a ser armazenado no dispositivo do utilizador para: consulta *offline*, reimportação para *Autocad*, ou simplesmente para impressão.

Ainda na mesma barra de ferramentas (a azul-escuro) encontra-se um menu do tipo *dropdown* que fornece opções de visualização do diagrama tais como alternar a vista entre cabos e entre tubagem e ainda ativar ou desativar a numeração de nós no diagrama.

Já a barra lateral esquerda, fornece ferramentas de outro cariz. Aqui irão ser apresentadas ferramentas cuja utilização varia com o tipo de diagrama que se está a desenhar.

O primeiro conjunto de ícones (a amarelo) representa os diversos tipos de componentes disponíveis para instalação em relação ao tipo de planta em que se está a desenhar. Para adicionar um componente ao diagrama basta arrastá-lo da barra lateral esquerda para a posição desejada na planta.

Um clique com o botão direito do rato no ícone permite definir uma escolha de equipamento, por defeito, associada ao tipo de componente a instalar. A fase de escolha de equipamento irá ser abordada mais à frente neste documento.

O segundo conjunto de ícones (a azul claro) contém duas ferramentas que permitem aumentar, ou diminuir, o tamanho da planta, em relação ao tamanho dos ícones dos componentes apresentados no ecrã, uma vez que estes têm um tamanho predefinido.

E para terminar a análise do funcionamento dos botões no estúdio, para a fase de desenho do diagrama, é ainda fornecido um conjunto de botões (a cor rosa) que permite ajustar ou reiniciar, o valor de *zoom* da visualização da planta.

Além do conjunto de botões apresentados existem ainda outros mecanismos, não explícitos, que são necessários para desenhar um diagrama de uma planta de ITED. Para melhor compreensão do funcionamento destes mecanismos explicar-se-á o processo de desenho de um diagrama desde o início.

Após criar uma planta nova o utilizador depara-se com um projeto limpo e com a planta enviada como plano de fundo, conforme demonstrado anteriormente (Figura 21). Para começar a trabalhar sobre a planta deve-se passar pelos seguintes passos:

1. **Ajustar o tamanho da planta** em relação ao tamanho dos nós, utilizando os botões de “Aumentar planta” ou “Reduzir planta”, uma vez que os ficheiros DWG carecem de informações relativas à escala. Deste modo, o tamanho dos nós em relação às dimensões da planta fica em conformidade.
2. **Ajustar a escala da planta**, recorrendo à “Ferramenta de escala”. Em primeiro lugar define-se a dimensão que a ferramenta representa e em segundo lugar ajusta-se a barra a uma parte de planta que se conheça as medidas. Deste modo tanto a escala da planta, como a espessura do traço da planta fica em conformidade.
3. **Arrastar os nós para a planta**. Se necessário pode-se ajustar a sua posição posteriormente utilizando a combinação “*Ctrl* + arrastar”. Neste passo é possível também definir, para todos os nós, a altura a que se encontra e definir uma *label* que ajude a melhor identificar o nó. Para nós do tipo ATE ou ATI é ainda possível definir o método de aplicação, sendo que para ATE existem as opções de bastidor ou caixa vazia e para ATI existem as opções de *rack* ou caixa vazia. Os passos descritos, após a colocação do nó na planta, podem ser efetuados em qualquer altura.
4. **Ligar os nós** clicando com o botão esquerdo do rato e arrastando a partir de um nó, até outro nó. Ao executar esta ação é mostrada uma linha que tem como origem o primeiro nó e como destino a posição do ponteiro do rato, até se largar o botão

esquerdo do rato com o ponteiro por cima de outro nó. A animação da linha é feita de maneira a fornecer *feedback* visual da ação ao utilizador. Largar o botão esquerdo do rato com ponteiro fora de um nó, ou sobre o nó de origem, cancela a ligação. Se for efetuada uma ligação em nós que já estejam ligados ou se a ligação criar um *loop* no circuito é exibido um alerta ao utilizador e cancelada a ligação que este pretendia efetuar.

5. **Adicionar tubos** clicando primeiro na ligação. Ao fazê-lo é mostrada uma janela que apresenta a lista de tubos na ligação e a quantidade de cabos por tipo. Os cabos podem ser do tipo Cabo Coaxial (CC), Fibra Ótica (FO) ou Par de Cobre (PC). Para adicionar um tubo novo basta selecionar o diâmetro a partir da lista de diâmetros disponíveis, clicando de seguida no botão “Guardar”. Ao realizar esta ação é também possível definir o número de tubos novos que se pretende adicionar, para o diâmetro selecionado.
6. **Adicionar cabos** a partir da janela anterior. Para adicionar um cabo clica-se no botão de editar do tubo. Feito isto, à semelhança da lista de tubos, é apresentada uma lista com os cabos que passam dentro do tubo, apresentando o seu tipo e qualidade. A qualidade trata-se de uma referência do cabo. O processo de adicionar um cabo novo é semelhante ao de adicionar um tubo: seleciona-se o tipo de cabo; ao fazê-lo a lista das qualidades do cabo para o tipo escolhido é atualizada; seleciona-se a qualidade e quantidade escolhida e clica-se no botão “Guardar”.
7. **Outras ações:**
 - a. **Eliminar um cabo:** para eliminar um cabo clica-se na ligação onde este passa. Esta ação abre a janela de “Propriedades da ligação”. Aqui clica-se no botão “Editar” do tubo correspondente e por fim clica-se no botão de “Apagar” do cabo.
 - b. **Eliminar um tubo:** para eliminar um tubo, à semelhança de eliminar um cabo, clica-se na ligação. Na janela aberta clica-se no botão “Apagar” do tubo correspondente e, por fim, clica-se no botão “Confirmar” que é mostrado ao executar a última ação. Eliminar um tubo elimina também todos os cabos que nele estavam inseridos.
 - c. **Eliminar uma ligação:** para eliminar uma ligação basta clicar nela e carregar no botão “delete”. Eliminar uma ligação elimina todos os tubos e cabos inseridos nela.

- d. **Eliminar um nó:** para eliminar um nó basta clicar nele, ação que desencadeia o aparecimento de uma janela, e carrega-se no botão “delete”. À semelhança de eliminar uma ligação, eliminar um nó implica eliminar também todas as ligações de que ele faça parte e, consecutivamente, todos os tubos e cabos inseridos nessas ligações.
- e. **Selecionar equipamento por defeito:** esta ação é opcional e está desenhada para facilitar a seleção de equipamento numa fase posterior, por parte do utilizador. Para selecionar um equipamento por defeito para determinado tipo de componente clica-se com o botão direito do rato no ícone do tipo de componente correspondente. Esta ação desencadeia o aparecimento de uma janela que mostra uma lista dos equipamentos disponíveis para esse tipo de componente. Seleciona-se o pretendido e, por fim, clica-se no botão “Guardar”.
- f. **Inspecionar diagrama:** esta ação também é opcional mas é recomendável e deve ser executada como o último passo antes do utilizador passar à fase de escolha de equipamento, uma vez que permite verificar no diagrama desenhado uma lista de possíveis erros ou de situações em que possam ser sugeridas melhorias. Para executar esta ação basta clicar no botão “Inspecionar diagrama” situado na barra de ferramentas superior.

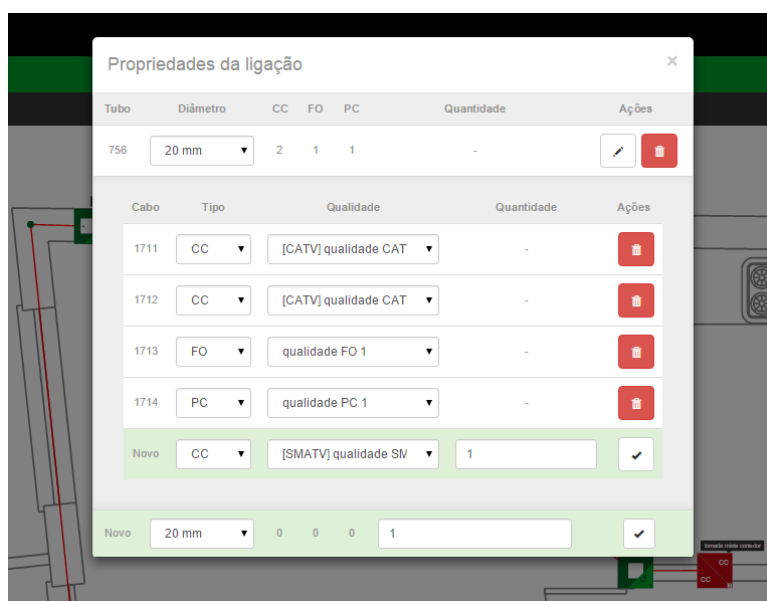
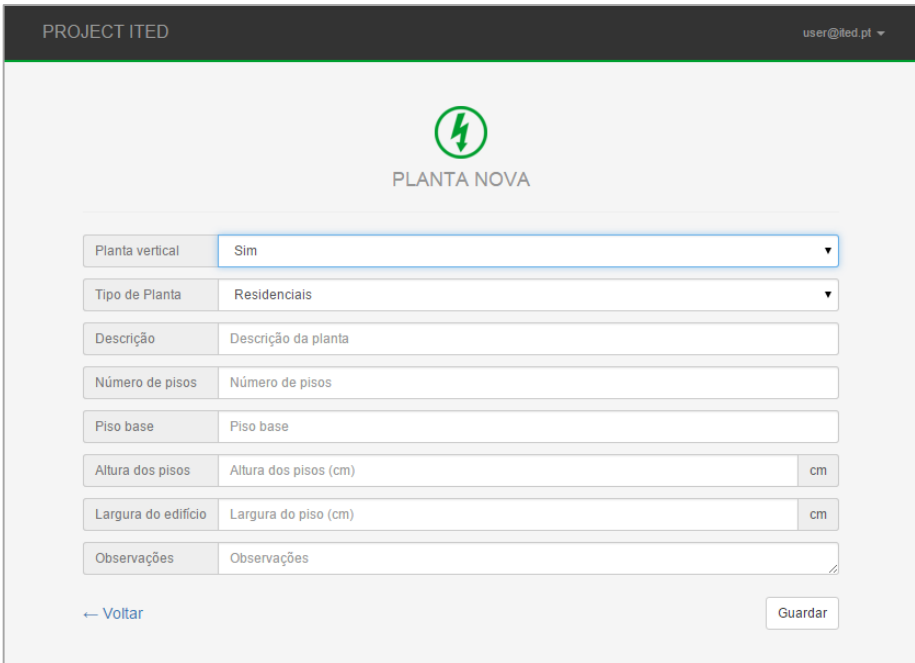


Figura 22 - interface de edição da ligação

Planta vertical

O processo de desenho de uma planta do tipo vertical é, em muitos aspetos, semelhante ao processo de desenho de uma planta horizontal. No entanto existem algumas funcionalidades diferenciadoras que merecem destaque.

Algumas dessas diferenças podem ser logo notadas durante o primeiro passo, no processo de criação de uma planta nova. Os campos do formulário apresentado são diferentes, em relação aos campos apresentados para uma planta horizontal, como pode ser visualizado abaixo (Figura 23):




PROJECT ITED		user@ited.pt
 PLANTA NOVA		
Planta vertical	Sim	
Tipo de Planta	Residenciais	
Descrição	Descrição da planta	
Número de pisos	Número de pisos	
Piso base	Piso base	
Altura dos pisos	Altura dos pisos (cm) cm	
Largura do edifício	Largura do piso (cm) cm	
Observações	Observações	
← Voltar	<input type="button" value="Guardar"/>	

Figura 23 - interface de criação de uma planta vertical nova

Como pode ser verificado o campo de envio de um ficheiro DWG é removido, sendo substituído por quatro campos novos: “Número de pisos”, “Piso base”, “Altura dos pisos” e “Largura do edifício”. Os três primeiros podem ser editados dentro do estúdio, conforme será explicado mais adiante, mas a largura do edifício deve ser editada na página de edição da planta. O campo “Número de pisos” refere-se ao número total de pisos que o edifício terá, contando também com os pisos abaixo da rasante (piso zero). O campo “Piso base” refere-se ao piso mais baixo do edifício e, em conjunto com o campo anterior, irá atribuir a numeração automaticamente a cada piso. O

campo “Altura dos pisos” refere-se a uma medida genérica da altura dos pisos mas, mais tarde, cada piso poderá ter uma altura personalizada. Finalmente o campo “Largura do edifício” refere-se à largura máxima do edifício.

Após guardar estas informações e do utilizador aceder ao estúdio, depara-se com a seguinte interface (conforme os campos introduzidos anteriormente), como representado na Figura 24:

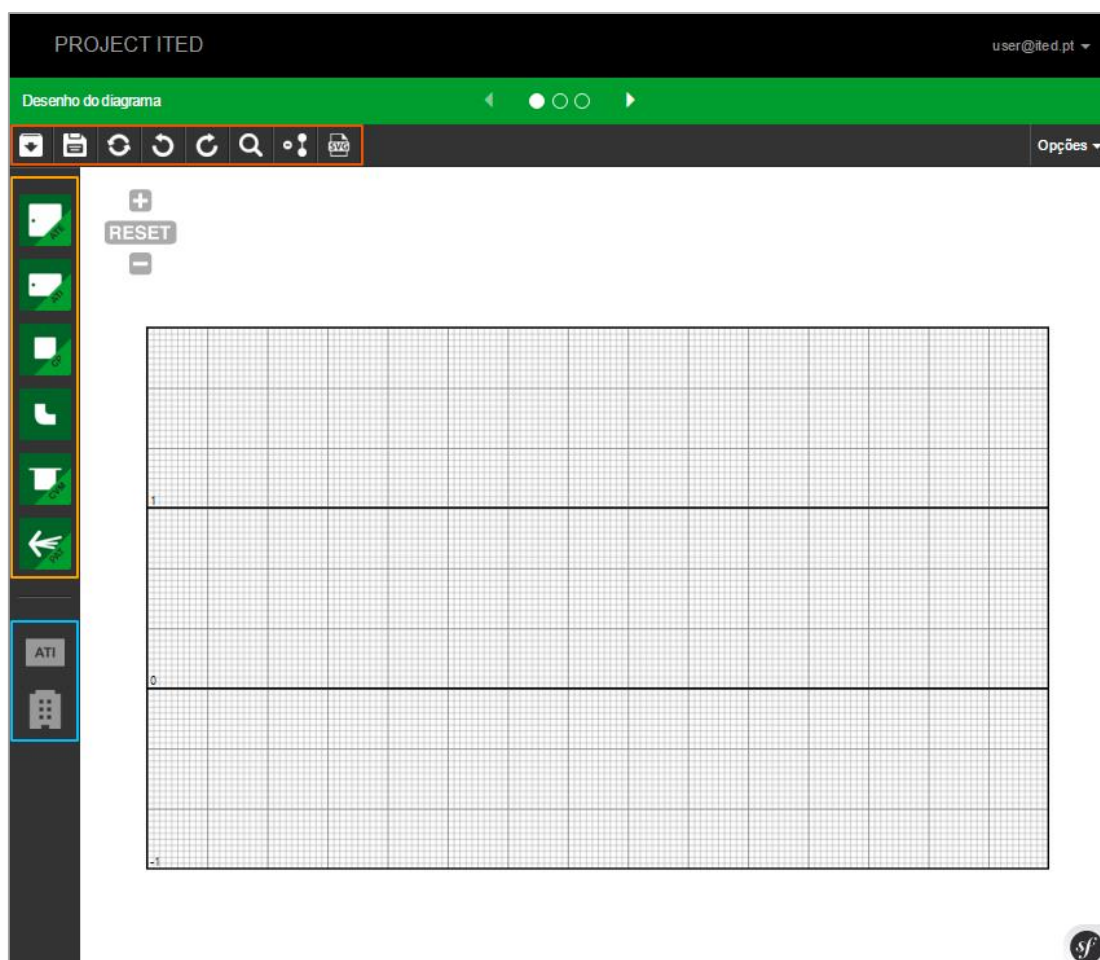


Figura 24 - exemplo da interface de estúdio de uma planta vertical

Como se pode verificar, em comparação à interface de estúdio apresentada, anteriormente na Figura 21, existem algumas diferenças na planta e nas funcionalidades apresentadas.

Comparando a barra de tarefas superior (a laranja) constata-se que a funcionalidade de “Ferramenta de escala” foi removida. Isto acontece porque, devido ao facto de ser o utilizador a

construir o desenho da planta de fundo e de serem conhecidas as dimensões do edifício, a planta poder ser desenhada à escala de 1:1 no editor.

Devido à mesma razão e, passando agora à análise das diferenças na barra de ferramentas lateral (a azul-claro), as funcionalidades de “Aumentar planta” e “Reduzir planta” foram também removidas tendo sido substituídas pelas ferramentas: “Carregar ATI de todas as plantas” e “Gerir pisos”.

A primeira irá permitir fazer a ligação das plantas individuais à coluna do edifício, através das ATI das plantas individuais.

A segunda irá permitir a edição da estrutura vertical do edifício, oferecendo funcionalidades de adição e remoção de pisos e suas características (altura e descrição).

De notar que alguns tipos de componentes na barra lateral (a amarelo), nomeadamente CEMU, QE e TT foram também removidos para este tipo de planta, uma vez que não deverão ser usados na construção deste tipo de diagrama.

Como já foi referido anteriormente, uma planta vertical ou de coluna montante, serve para efetuar ligações entre os componentes ATI das diversas plantas horizontais e para fazer ligação aos pontos de entrada das telecomunicações do edifício, quer superiores tais como através de uma PAT quer inferiores através da CVM. A ligação às ATI é um requisito pois é necessário injetar nelas a atenuação e nível de sinal das ligações que as precedem para que os cálculos finais destas variáveis sejam corretos.

Passando agora então a uma análise das ferramentas acrescentadas a este tipo de planta, tem-se as seguintes:

- **Carregar ATI de todas as plantas** como o próprio nome indica, insere nesta planta os nós ATI que estão desenhados nas outras plantas horizontais deste projeto. São carregadas informações do nó e ainda o número de entradas e saídas de cabos nesse nó, na respetiva planta horizontal. A adição de *labels* nos nós nas plantas horizontais é recomendável pois, nesta situação e não só, permite que estes sejam facilmente identificados, auxiliando no seu posicionamento e correta ligação aos nós do atual diagrama vertical.
- **Gerir pisos** apresenta uma janela de aspeto semelhante à interface de “Propriedades da ligação” abordada anteriormente na secção da planta horizontal.

No entanto, esta apresenta uma listagem de pisos no edifício, exibindo para cada piso o seu número, um nome (opcionalmente atribuído pelo utilizador) e a altura. À exceção do número do piso, os outros dois valores são editáveis, clicando no botão “Editar”, efetuando as alterações necessárias e finalmente, clicando em “Guardar”. Para adicionar um piso introduz-se o nome (opcional) e a altura e clica-se na seta correspondente: seta para cima adiciona o andar no topo; seta para baixo adiciona o andar no fundo. A interface descrita pode ser visualizada mais abaixo na Figura 25.

- **Grelha** é uma opção localizada no menu do tipo *dropdown* na barra de ferramentas superior, à direita em “Opções” que permite ativar ou desativar a visualização da grelha no fundo da planta. A grelha serve para auxiliar o utilizador a colocar os nós na posição correta. Cada quadrado grande da grelha corresponde a um metro e cada quadrado pequeno corresponde a 100cm. Esta opção está ativa por defeito.

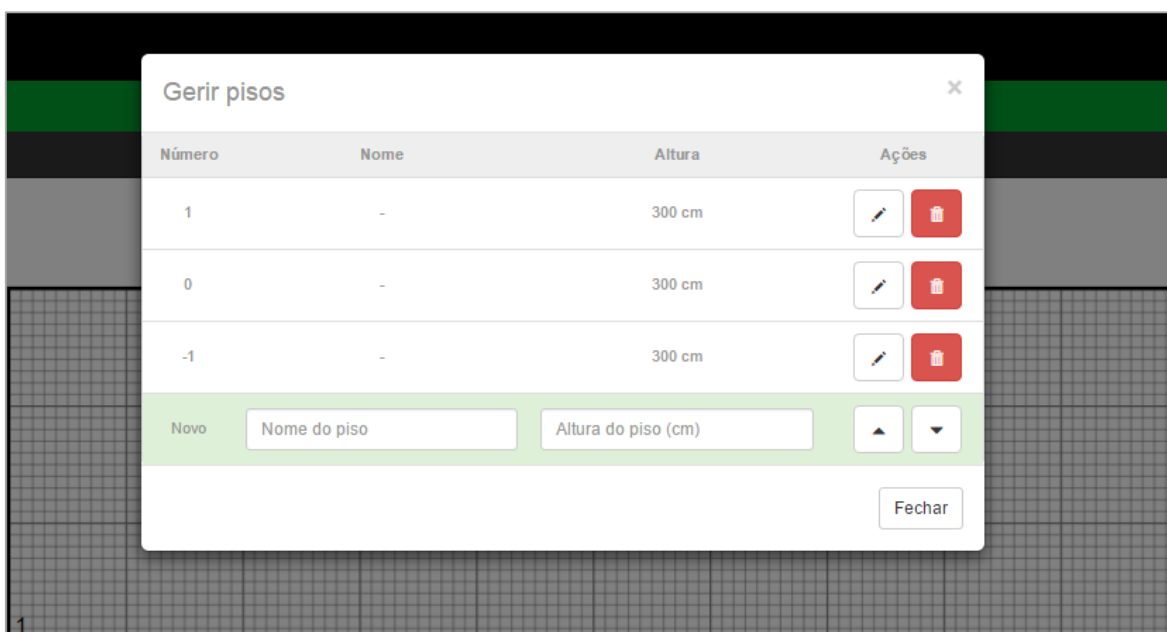


Figura 25 - janela de gestão de pisos numa planta vertical

Em relação aos passos recomendados no desenho do diagrama, o processo é fundamentalmente semelhante ao processo de desenho de uma planta horizontal, tirando o facto de que não é necessário ajustar as dimensões da planta.

Como primeiro passo é recomendável definir bem a estrutura vertical do edifício através do gestor de pisos e, como segundo passo, deve-se carregar os nós ATI das outras plantas.

A partir deste ponto o desenho do diagrama segue os mesmos padrões do desenho do diagrama de uma planta horizontal.

Para finalizar...

Estando o desenho efetuado, para qualquer tipo de planta, chega agora a altura de avançar para o próximo passo. Este trata-se da escolha do equipamento para os componentes desenhados no diagrama.

Para avançar para este próximo passo basta carregar na seta direita situada no módulo central da barra superior colorida. O círculo branco representa a etapa em que o utilizador se encontra e as setas permitem navegar entre etapas. Um exemplo do aspeto do desenho de um diagrama concluído pode ser visualizado abaixo na Figura 26:

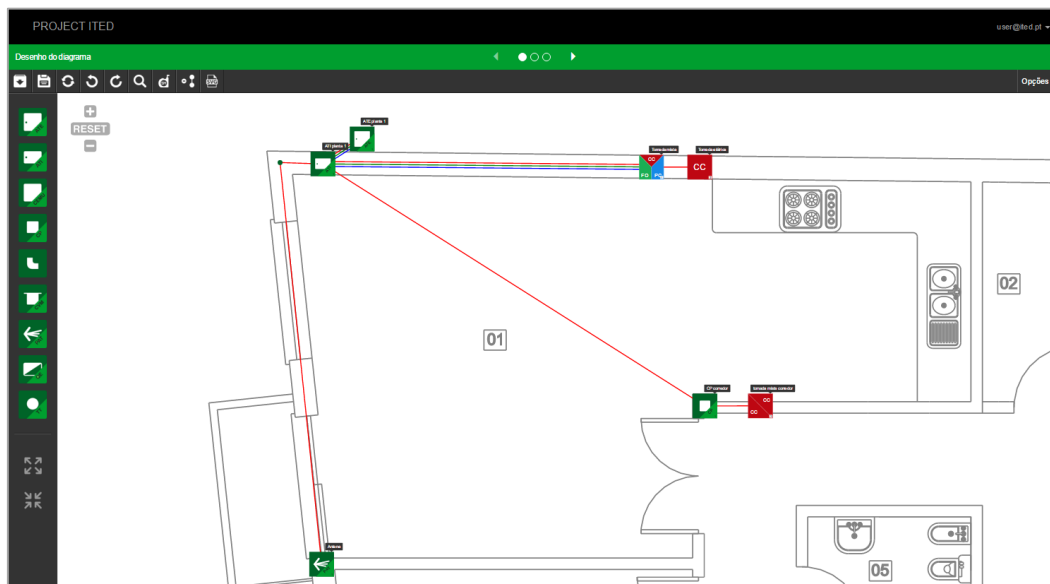


Figura 26 - exemplo de desenho de um diagrama completo

5.2.2 Escolha do equipamento

Nesta secção iremos abordar a segunda etapa da criação de um projeto ITED que se trata da seleção de material. Mas porque é que esta fase é necessária e quem é que ela satisfaz?

Ao desenhar um diagrama ITED têm-se como objetivo, para além de obter um esquema geral das ligações, obter esquemas da infraestrutura de comunicações por tecnologia (CC, FO, PC e Tubagem). Estes esquemas, para certas tecnologias, devem apresentar os valores de atenuação e de nível de sinal, nos extremos das ligações, que são as tomadas terminais.

Os fatores que influenciam estes valores passam pelo comprimento das ligações e dos equipamentos que são instalados entre elas.

Sendo assim, esta funcionalidade pretende satisfazer duas entidades: a empresa para a qual a aplicação está a se desenvolvida e que a irá disponibilizar e os projetistas que irão usufruir da aplicação. A empresa tira benefício pois pode recomendar os seus produtos para instalação, lucrando com as vendas. Os projetistas tiram benefício pois têm imediatamente à sua disposição uma lista de produtos que podem adequar às necessidades da sua instalação, ajustando-os a obter os valores de atenuação e de níveis de sinal realistas.

Tendo então o desenho do diagrama efetuado passa-se à fase de escolha de equipamento, que apresenta a seguinte interface (Figura 27):

PROJECT ITED		Escolha de equipamento												Henrique Moreira
Código	Descrição	Aplicação	Atenuação	Espago (oir)	CC (e/s)	FO (e/s)	PC (e/s)	Preço	Dimensões (LxAxP)	Localização	Observações	Selecionar		
ATE nº.1	ATE planta 1	CP TEST...	CP TEST...	Default	1 dB	0/0 U	5/5	5/5	5/5	10.5 €	0x0x0 cm	nenhuma	Nulla eu...	<input checked="" type="checkbox"/>
ATI nº.1	ATI planta 1	CP TEST...	CP TEST...	Default	1 dB	0/0 U	10/10	10/10	10/10	10.5 €	0x0x0 cm	nenhuma	nenhuma	<input checked="" type="checkbox"/>
TT nº.1	Tomada mista													
TT nº.2	Tomada elétrica													
CP nº.1	CP corredor													
TT nº.3	tomada mista corredor													

Figura 27 - interface de escolha de equipamento

Na coluna esquerda é apresentada a lista de nós. É apresentado em cada um, primeiro o tipo de componente (ATE, ATI, TT, entre outros), depois o número do nó (por tipo de componente) e ainda, para facilitar a identificação do nó, a *label* atribuída pelo utilizador.

Para dar *feedback* ao utilizador, sobre o estado de seleção do equipamento, os nós que não tenham um equipamento selecionado são apresentados a vermelho. Selecionar o nó altera o conteúdo da coluna direita da interface.

Na coluna direita é apresentada, até ao meio vertical da página, a lista de equipamento disponível para instalação, tendo em conta os constrangimentos definidos no desenho, nomeadamente o número de entradas e saídas de cabos por tecnologia que devem ser suportados pelo equipamento.

Selecionar um equipamento apresenta abaixo da lista, todas as suas características e ainda uma interface para adicionar módulos/subcomponentes ao equipamento selecionado. Esta funcionalidade é particularmente interessante caso o utilizador necessite de personalizar o equipamento a ser instalado quando uma *bundle* de equipamento, disponibilizada pela empresa, não serve as necessidades da instalação.

Tendo o desenho do diagrama feito e o equipamento selecionado é possível agora gerar os diagramas das instalações, por tecnologia, e gerar um inventário dos equipamentos a instalar no projeto.

Mais uma vez avança-se para a próxima fase navegando na barra superior colorida.

5.2.3 Inventário e diagramas

Esta fase do projeto é apenas para consulta por parte do utilizador e para retificação do esquema desenhado e equipamento selecionado.

Aqui são apresentadas ao utilizador seis visualizações, sendo elas: “Diagrama geral”, “Diagrama de Tubagem”, “Diagrama de Cabos Coaxiais”, “Diagrama de Fibra Ótica”, “Diagrama de Par de Cobre” e “Inventário”.

Aos diagramas de cabos coaxiais, fibra ótica e par de cobre iremo-nos referir, por vezes, como diagramas por tecnologia.

Os diagramas apresentados são hierárquicos, de modo a representar a orientação das ligações dos cabos aos componentes.

Em relação às hierarquias, uma CVM e PAT têm prevalência sobre uma ATE e uma CEMU. Uma ATE e uma CEMU têm prevalência sobre uma ATI. Uma ATI tem prevalência sobre os restantes componentes.

No diagrama geral é apresentada informação relativa às ligações de todos os nós, exceto dos nós de curva, pois estes são apenas nós de auxílio ao correto desenhar do diagrama.

As informações apresentadas são o comprimento da ligação e o número de cabos que por ela passa (por tecnologia).

Em todos os nós é também apresentada a *label* que melhor ajuda a identificar unicamente cada nó.

Nas tomadas terminais são ainda apresentadas as atenuações para as frequências de 47MHz, 862MHz, 950MHz e 2150MHz e ainda os níveis de sinal. Estes cálculos são feitos na infraestrutura de cabos coaxiais e a fórmula de cálculo é a seguinte:

$$A_{LP} = A_{\text{cabo}} + A_{DR} + n \times A_C + A_{TT}$$

As variáveis representam a seguinte informação:

- A_{LP} = atenuação da ligação permanente (dB)
- A_{cabo} = atenuação do cabo em função do comprimento (dB)
- A_{DR} = atenuação dos dispositivos de repartição, ou derivação, se aplicável (dB)
- n = número de conetores considerados
- A_C = atenuação por conetor (dB)
- A_{TT} = atenuação da tomada terminal, se aplicável (dB)

Em caso de inexistência de valores reais, deve considerar-se:

$$A_C = 0.0001 \times f_{\text{MHz}}$$

A variável representa a seguinte informação:

- f_{MHz} = frequência limite dos sistemas (47 MHz, 862 MHz, 950 MHz ou 2150 MHz)

Tendo os valores das atenuações é então possível calcular os níveis de sinal. O cálculo do nível de sinal é efetuado da seguinte maneira:

$$S_{TT} = S_{ATI} - A_{TT}$$

As variáveis representam a seguinte informação:

- S_{TT} = nível de sinal na tomada terminal (dB μ V)
- S_{ATI} = nível de sinal no ATI (dB μ V)
- A_{TT} = atenuação na tomada terminal (dB)

O ícone apresentado nas tomadas terminais é relativo às tecnologias que a tomada possui, tal como é representado na 1ª fase, de desenho do diagrama.

Por fim, nas caixas de passagem, são ainda apresentadas as suas dimensões (largura x altura x profundidade).

Passando ao diagrama de tubagem, este apresenta um *layout* semelhante ao diagrama geral mas as informações apresentadas variam. Em vez de serem indicados os cabos que passam nas ligações, é indicado o número de tubos, por diâmetro. A informação do comprimento da ligação mantém-se e a informação relativa à atenuação nas tomadas não é apresentada.

Passando aos diagramas por tecnologia (CC, FO e PC) existem algumas alterações, face aos diagramas descritos anteriormente. Nestes diagramas, para além dos nós de curva, são também removidos os nós de caixas de passagem.

As tomadas terminais são ainda apresentadas todas ao mesmo nível, algo que não acontecia anteriormente caso uma tomada terminal servisse também como caixa de passagem para outra tomada. Voltam aqui também a ser apresentadas informações como a atenuação e níveis de sinal e são acrescentadas informações novas.

Nas ligações é agora apresentada a quantidade de cabos e a sua referência. Nos nós ATI, caso tenha sido adicionado um repartidor na fase de escolha de equipamento, é apresentada a referência do repartidor e é ainda apresentado o valor de nível de sinal neste componente.

Nas tomadas terminais, do mesmo modo, é apresentada a sua referência e o ícone refere-se ao ícone da tecnologia do diagrama apresentado.

No ecrãs dos diagramas é ainda possível ir para modo de ecrã inteiro para melhorar a visualização e é ainda possível converter o diagrama para um ficheiro SVG e descarregá-lo para futura consulta ou impressão.

Um exemplo da interface de diagramas, neste caso específico do diagrama geral, pode ser visualizado abaixo na Figura 28:

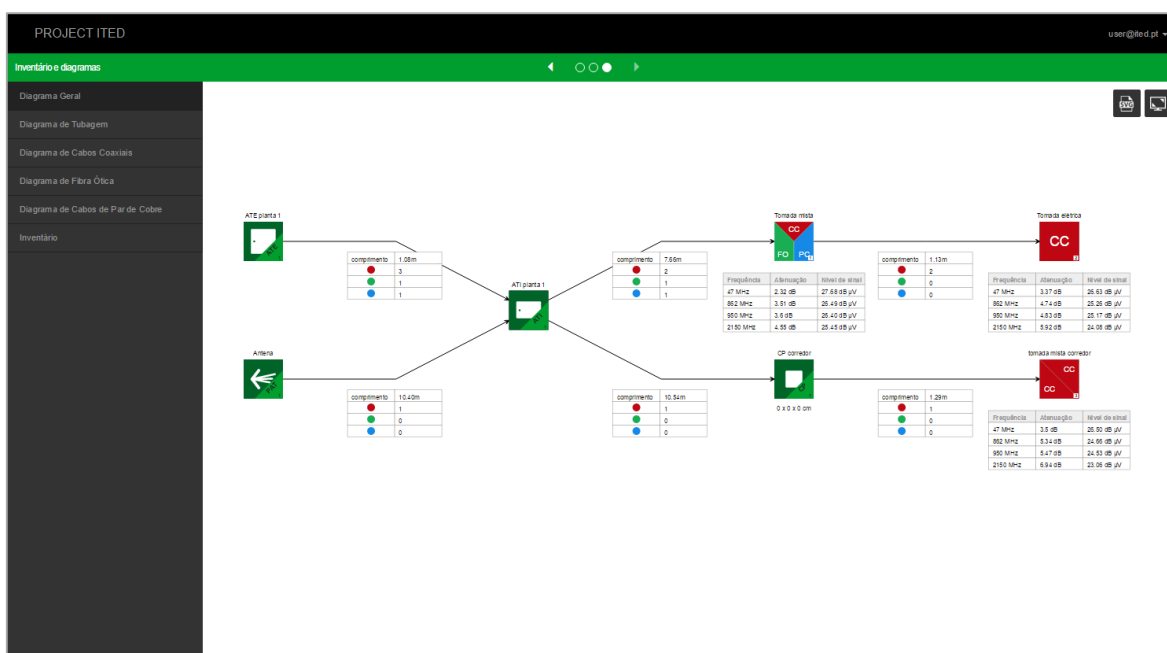


Figura 28 - exemplo de interface de um diagrama geral

Por fim, na última opção, é possível consultar o inventário. Este encontra-se dividido em três segmentos, sendo estes “Tubagem”, “Cablagem” e “Equipamento”.

Na secção de tubagem é discriminada informação relativa ao diâmetro e comprimento total de tubagem para esse diâmetro.

Na secção de cablagem é discriminada informação relativa ao tipo de cabo, referência do cabo e comprimento total de cabo para essa referência.

Na secção de equipamento é discriminada informação relativa a todo o equipamento selecionado na fase anterior, sendo ele todos os equipamentos e módulos/subcomponentes neles instalados.

Por fim é calculado e apresentado o preço total do equipamento.

5.3 BackOffice

Os dados apresentados anteriormente, em ilustrações do sistema, foram introduzidos pelo utilizador, neste caso pelo utilizador do tipo Administrador ou *Super Administrador*. Estes são os responsáveis pela gestão do conteúdo apresentado.

Sendo assim foi necessária a implementação de um sistema de *BackOffice*, não acessível ao utilizador sem privilégios de administração, que oferece funcionalidades *Create Read Update Delete* (CRUD).

O mecanismo de criação, leitura, edição e remoção de dados é coerente entre as diversas páginas, variando apenas os campos apresentados. Sendo assim serão apresentados mais abaixo as vistas referentes às ações descritas, apenas para uma das entidades.

Além disso, as vistas tentam manter coerência em relação ao *design*, o que pode ser observado, por exemplo, comparando as interfaces do *BackOffice* com a interface das fases dois e três do estúdio. Deste modo, a navegação entre páginas é efetuada de modo semelhante à navegação nas interfaces referidas anteriormente.

Como é indicado na seção 10.3.1.5, na restrição de acesso a rotas, as entidades sobre as quais os administradores têm acesso são “Cabo”, “Cabo/Qualidade”, “Componente”, “Modelo de Aplicação”, “Qualidade”, “Subcomponente”, “Tipo de Aplicação”, “Tipo de Componente”, “Tipo de Planta”, “Tubo” e “Utilizador”.

Cada entidade possui três vistas, sendo estas a vista de lista, de criação e de edição. A título exemplificativo serão apresentadas as vistas de lista (Figura 29), onde podem ser consultados todos os registos de uma entidade, e a vista de criação (Figura 30), onde podem ser inseridos novos registos, ambas da entidade “Cabo”.

A vista de edição é, em termos visuais e de interação, muito semelhante à interface de criação, pelo que não carece apresentação.

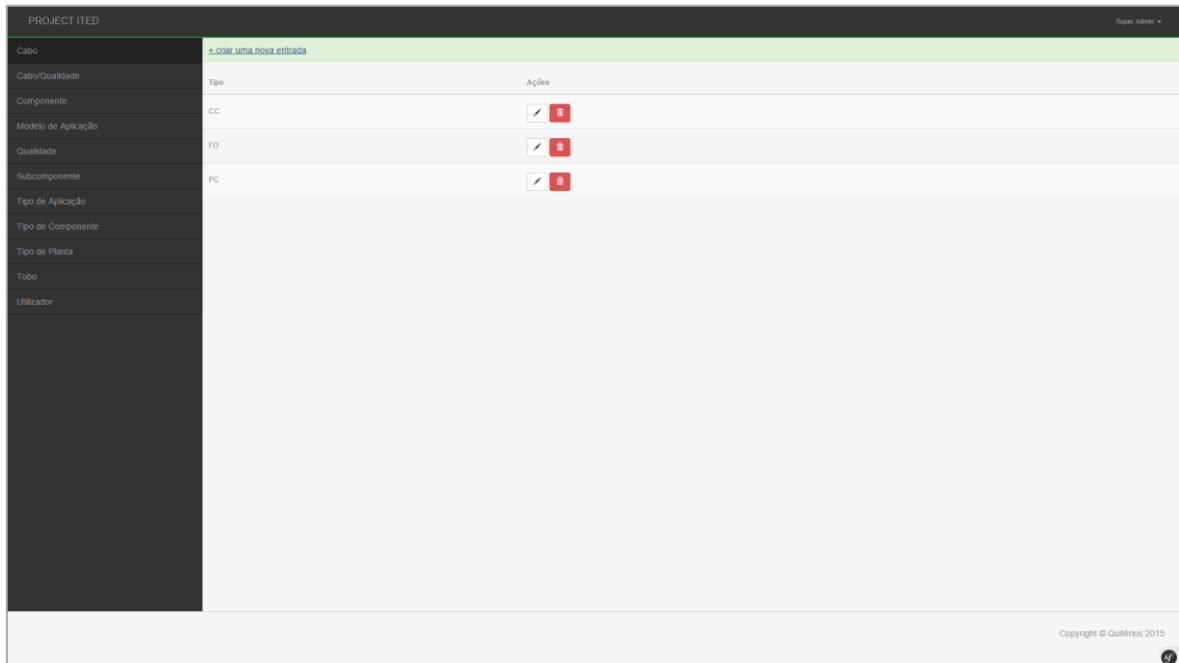


Figura 29 - vista da lista de cabos

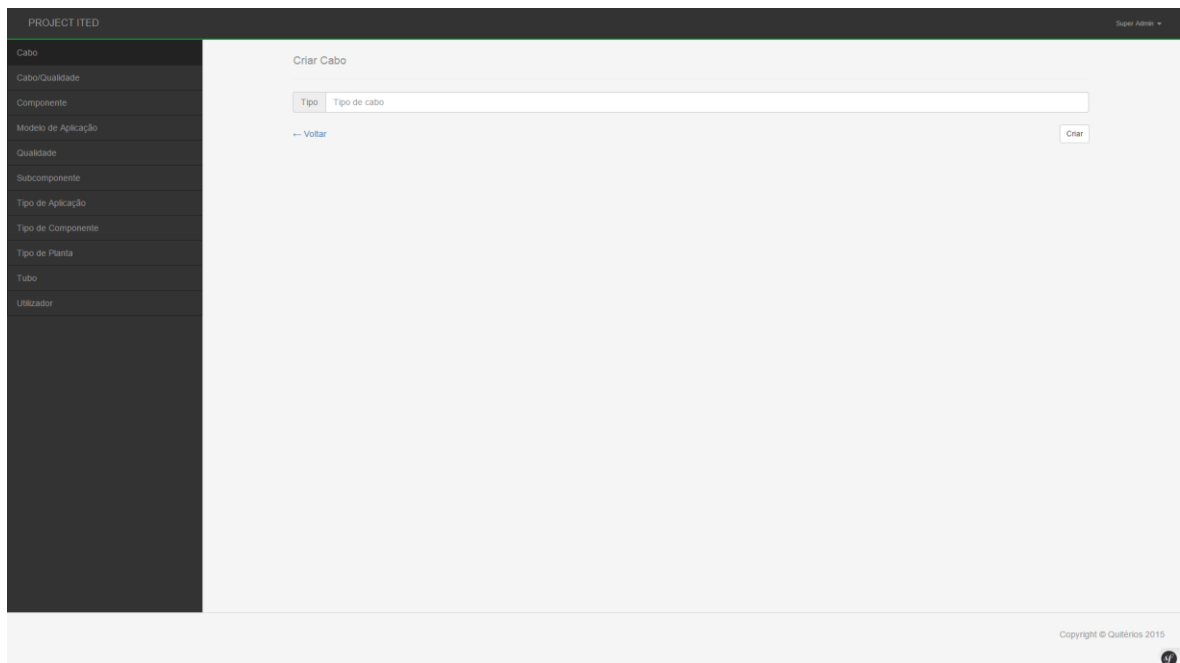


Figura 30 - vista da criação de cabos

5.3.1 Inserção de dados

A criação das vistas foi feita por entidades do *Symfony*. Deste modo, para entidades relacionadas é necessário inserir informação seguindo uma sequência. Para melhor utilização do *BackOffice* proceder-se-á descrição destes processos.

Adicionar um tipo de planta

Adicionar um tipo de planta envolve apenas criar uma nova entidade “Tipo” com os seguintes campos obrigatórios:

- Descrição

Adicionar um cabo

Adicionar um cabo implica relacionar uma tecnologia (tipo de cabo) com uma qualidade (referência). Deste modo, em primeiro lugar, adiciona-se uma tecnologia, caso esta não exista.

Nota: esta aplicação usa tecnologias previamente definidas, sendo estas Cabo de Cobre (CC), Fibra Ótica (FO) e Par de Cobre (PC). Estas tecnologias não devem ser modificadas, uma vez que o estúdio possui funcionalidades que delas dependem. De qualquer modo, o sistema foi desenhado para poder ser acrescentado conteúdo e, portanto, esta funcionalidade permanece disponível.

Para adicionar uma tecnologia cria-se uma nova entidade “Cabo” com os seguintes campos obrigatórios:

- Tipo

O segundo passo é adicionar uma referência (qualidade). Para isto cria-se uma nova entidade “Qualidade” com os seguintes campos obrigatórios:

- Nome

Por fim relacionam-se as duas entidades criadas, criando uma terceira chamada de “CaboQualidade”. O sistema foi desenhado desta maneira de modo a permitir associar a mesma

referência a tecnologias diferentes, caso surja um cabo que o suporte. Para criar a entidade referida é necessário inserir os seguintes campos obrigatórios:

- Cabo
- Qualidade

Adicionar um tubo

Para adicionar um tubo basta criar uma nova entidade “Tubo” com os seguintes campos obrigatórios:

- Diâmetro

Adicionar um tipo de componente

Adicionar um tipo de componente (tipo de nó) envolve alguns passos. É necessário definir o tipo de componente, o tipo de aplicação e, finalmente, criar um modelo de aplicação. O sistema foi desenhado desta maneira pois um tipo de componente pode-se adequar a diversos tipos de aplicação e um tipo de aplicação pode servir para vários tipos de componentes.

O primeiro passo trata-se então de criar um tipo de componente. Isto é feito criando uma nova entidade “TipoComponente” com os seguintes campos obrigatórios:

- Nome
- Descrição

O segundo passo envolve criar um ou mais tipos de aplicação. Para isto cria-se o número de entidades “TipoAplicacao” desejadas com os seguintes campos obrigatórios:

- Nome

Por fim criam-se as relações necessárias através da criação de entidades “ModeloAplicacao” com os seguintes campos obrigatórios:

- Tipo de Componente
- Tipo de Aplicação

Adicionar um equipamento

Para adicionar um equipamento (componente) basta criar uma entidade “Componente” nova com os seguintes campos obrigatórios:

- Modelo de Aplicação
- Código
- Unidades Ocupadas
- Unidades Reservadas
- Entradas CC
- Entradas FO
- Entradas PC
- Saídas CC
- Saídas FO
- Saídas PC
- Preço
- Atenuação
- É subcomponente?
- Mín. subcomponentes
- Máx. subcomponentes

Adicionar um subcomponente

Para adicionar um subcomponente é necessário ter previamente um componente criado, efetuando os passos referidos anteriormente e que tenha no campo “É subcomponente?” o valor “Não” indicando que é um componente “Pai”. Após a criação do componente cria-se uma nova entidade “Subcomponente” com os seguintes campos obrigatórios:

- Componente Pai
- Componente Filho

Nota: a entidade possui o campo Cabo uma vez que o sistema foi desenhado para futuramente poder impor restrições no equipamento a determinada tecnologia (tipo de cabo).

Adicionar um utilizador

Para adicionar um utilizador basta criar uma nova entidade do tipo utilizador com os seguintes campos obrigatórios:

- Perfil
- Nome
- *Email*
- NIF (Número de Identificação Fiscal)
- *Password*

Nota: para adicionar um utilizador novo é necessário ter privilégios de *Super Administrador*. Além disso a *password* tem que ser inserida pelo *Super Administrador* devendo ser o utilizador a alterá-la na primeira utilização. Como trabalho futuro seria conveniente gerar automaticamente uma *password* e enviá-la para o *email* introduzido.

Fica assim concluída a apresentação do *BackOffice* e, conseqüentemente, do protótipo implementado.

6 Problemas e soluções

Durante o desenvolvimento deste projeto foram encontrados alguns obstáculos, que necessitaram de especial atenção para serem ultrapassados. Neste capítulo irão ser apresentados alguns desses problemas e que soluções foram adotadas na sua resolução.

Um dos problemas iniciais com que nos deparámos foi com a conversão das plantas CAD com formato DWG, para um formato que pudesse ser utilizado e manipulado em ambiente *web*. Após alguma pesquisa foram encontradas diversas soluções. Nenhuma oferecia uma licença de utilização gratuita. De entre as opções analisadas a escolhida foi o “Total CAD Converter” [20] da empresa “CoolUtils” pois, para além de ter dos preços mais competitivos no mercado e de o *software* funcionar como pretendido, esta oferece ainda a possibilidade de execução do programa através da linha de comandos. Esta funcionalidade faz com que seja possível a sua execução através de uma página *web*.

Outro dos problemas relacionados com a utilização de ficheiros DWG é que estes não possuem informação relativa à escala das plantas. Esta informação é bastante importante pois é através dela e das coordenadas x e y dos componentes que é possível calcular o comprimento das ligações e das tubagens e cabos que nelas estão inseridas. O comprimento das ligações, por sua vez, irá afetar os cálculos das atenuações e níveis de sinal nas tomadas terminais.

Além disso, ao converter uma planta, a largura do traço é definida com um valor padrão. Ao carregar plantas demasiado pequenas para a largura do traço predefinido, deixa de haver a perceção dos detalhes. O contrário, quando as plantas são muito grandes para a largura do traço predefinido, faz com que, ao visualizar a planta com o *zoom* correto, o traço pareça invisível.

Uma demonstração deste problema pode ser visualizada abaixo na Figura 31.

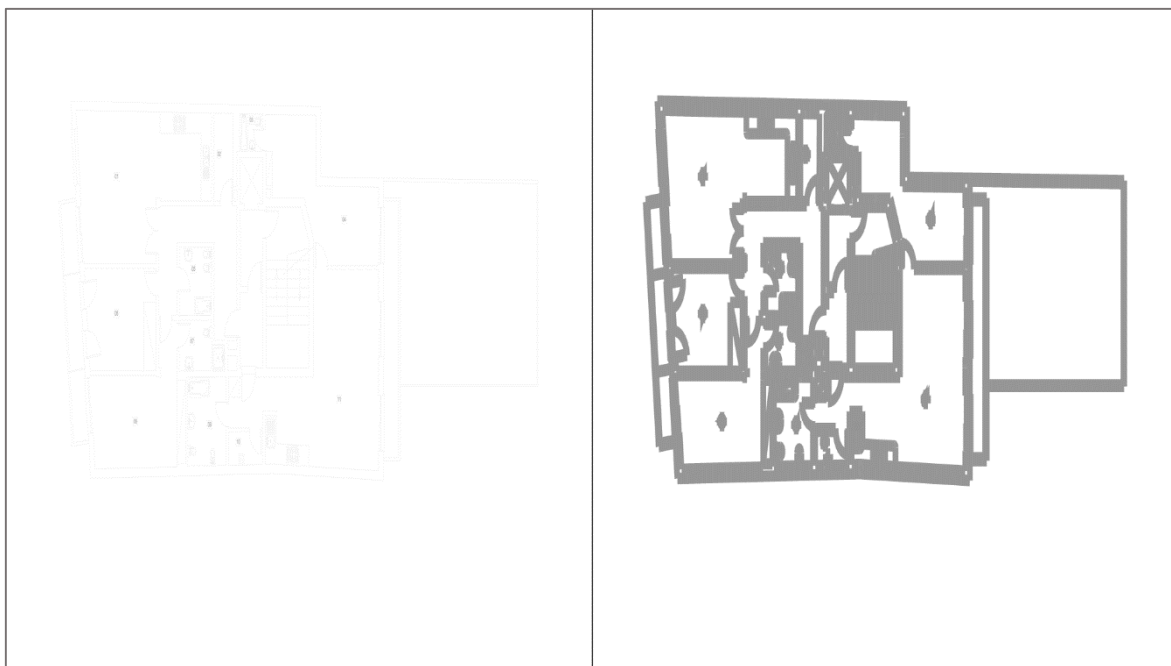


Figura 31 - representação dos problemas de escala na planta (à direita: escala muito alta; à esquerda: escala muito baixa)

Deste modo foi necessária a criação de um mecanismo que pudesse, após o envio da planta determinar a sua escala, criando depois uma série de mecanismos que corrigissem quer a parte dos cálculos, quer a parte de visualização da planta. A solução encontrada foi a criação de uma ferramenta que se assemelha a uma fita métrica mas em que se possa editar o valor da distância que ela representa. Sendo assim, conhecendo uma medida de um objeto na planta, dimensiona-se a fita ao comprimento desse objeto e atribui-se o valor da distância que ele representa. A possibilidade de variar o valor do comprimento da fita introduz mais flexibilidade facilitando o processo de definição da escala.

Um dos maiores obstáculos encontrados no desenvolvimento do protótipo foi desenvolver código que fosse compatível com a maioria dos *browsers* da atualidade, nomeadamente garantir compatibilidade com o Internet Explorer. A principal área da aplicação onde nos foram encontrados desafios foi no desenvolvimento do estúdio, uma vez que é a parte da aplicação onde são corridos mais *scripts*, de diversas bibliotecas.

O principal problema do Internet Explorer está na utilização da biblioteca D3.js para renderizar código SVG carregado a partir da base dados, recorrendo à *tag* “.html()” no objeto onde deveria ser colocado o conteúdo.

Para ultrapassar este obstáculo foi desenvolvida uma visualização ligeiramente diferente. Ao invés de carregar os ícones dos nós, foram renderizados retângulos com o nome do tipo de componente. Apesar de se perder um pouco das dicas visuais transmitidas ao utilizador garantiu-se assim o seu funcionamento.

Nas tomadas terminais deixam de ser mostradas automaticamente as tecnologias que a tomada possui, mas é possível visualizar esta informação colocando o cursor do rato por cima da ligação anterior à tomada.

O mecanismo desenvolvido funciona para todos os *browsers* que não suportem o carregamento dos ícones pois a visualização alternativa é renderizada numa camada abaixo do ícone, ficando visível apenas quando o ícone não o é.

7 Conclusão

Após a finalização da apresentação da arquitetura e do protótipo desenvolvidos surge agora um momento de reflexão em que devem ser retiradas ilações, procurando sumarizar os aspetos conseguidos e os que ficaram por conseguir.

O principal objetivo desta dissertação passou pelo desenvolvimento do protótipo de uma aplicação *web* em que fosse possível a criação e edição de projetos ITED.

Para que a concretização deste projeto fosse possível foi necessária a idealização e desenvolvimento de uma arquitetura que desse suporte à implementação do sistema. Nesta arquitetura houve a necessidade de utilizar um mecanismo de armazenamento e de consulta de dados da aplicação, de um motor de *templating* de maneira a apresentar interfaces de interação com o utilizador e ainda de uma infraestrutura que interligasse estes dois componentes de maneira a apresentar as informações necessárias ao utilizador.

O sistema desenvolvido para dar resposta a estas necessidades tem por base componentes do *Symfony* que fornecem mecanismos altamente configuráveis e de fácil utilização, e ainda de uma base de dados *MySQL* como a que foi descrita.

O protótipo desenvolvido fornece os mecanismos para a criação de projetos ITED. Isto é possível de verificar na secção do estúdio da aplicação onde são fornecidas ferramentas para o desenho de diagramas de infraestruturas de comunicação das diversas tecnologias, que passam pelo desenho dos componentes, interligando-os com tubagens e cabos.

Este protótipo resolveu também alguns problemas encontrados em outras aplicações disponíveis como, por exemplo, o facto de não ser possível trabalhar diretamente por cima de uma planta, de modo a desenhar os diagramas necessários.

No protótipo desenvolvido, para além de isto ser possível, devido à ferramenta de escala implementada não é necessário definir o comprimento das ligações.

Foi também implementado um sistema que permite a seleção de equipamentos reais para instalação no projeto. Isto faz com que os cálculos finais apresentados nos diagramas das tecnologias tenham maior precisão, favorecendo tanto a entidade que os comercializa, assim como o instalador que não necessita de procurar soluções fora da aplicação.

Apesar dos diagramas das tecnologias apresentarem informações relevantes, existe ainda algum trabalho a desenvolver para que possam apresentar todo o conjunto de informações desejadas.

Muitas das funcionalidades que se pretendia implementar foram conseguidas mas, como se pôde verificar, existem alterações a aplicar que ficaram para segundo plano. A necessidade de alterar certos aspetos teve origem na evolução da aplicação ao longo do tempo e da constante mudança dos requisitos o que faz com que, uma solução apresentada para um problema atual possa não servir na perfeição para requisitos futuros.

Como acabámos de demonstrar existe sempre espaço para alterações e introdução de novas funcionalidades, ainda mais visto que este é um projeto que continuará em desenvolvimento.

Não obstante, o objetivo principal de fornecer um mecanismo de desenho de diagramas ITED foi, até à data, desenvolvido com sucesso.

8 Trabalho futuro

Como os requisitos da aplicação foram sofrendo mudanças ao longo do tempo, derivado das iterações propostas pelo cliente e por mudanças em certos aspetos da aplicação, existem determinadas decisões que, se fossem conhecidos os requisitos atuais da aplicação, teriam sido tomadas de maneira diferente. Muitas vezes, com a necessidade de iterar funcionalidades na aplicação estas mudanças foram caindo para segundo plano.

Um exemplo concreto deste problema diz respeito aos gráficos. Apesar de a maior parte das funcionalidades estar já integrada, ficou para abordagem posterior o cálculo de perdas de sinal nas tomadas terminais, nomeadamente para a tecnologia de fibra ótica, uma vez que este não foi um requisito inicial.

Atualmente cada tentativa de autenticação fica registada na base de dados e só é removida caso o utilizador passe por todos os passos sem que ocorra nenhum erro. Existir algum erro no processo de autenticação após a inserção de dados na tabela de autenticação significa que eles lá irão persistir indefinidamente. Seria interessante aplicar um mecanismo de limpeza desta tabela periodicamente.

Outro ponto que seria interessante de desenvolver seria a criação de um sistema de edição de estúdio *offline*. Seria necessário criar um mecanismo que permitisse o armazenamento de uma versão da base de dados localmente, de modo a poder aceder aos dados sem ligação à rede. Quando o utilizador se voltasse a ligar à rede, existiria um mecanismo de sincronização que verificasse a integridade dos dados inseridos e que atualizasse os registos mediante validação.

Esta funcionalidade seria de interesse uma vez que, apesar de já existirem planos de dados móveis bastante acessíveis, nem toda a gente opta por aderir. Além disso podem existir áreas de fraca ou até inexistente cobertura de rede.

Outro ponto relacionado com a utilização de dispositivos móveis é a capacidade de edição dos nós no estúdio a partir de um dispositivo móvel. Atualmente a interação com componentes do estúdio não suporta as ações realizadas por toque e arrastamento num *smartphone* ou *tablet*. Isto deve-se ao facto de, o evento em *JavaScript*, para este tipo de gestos, não ser compatível com as ações de interação tradicionais de teclado e rato.

Será necessário implementar uma solução que sirva de ponte entre as interações, tornando a utilização do estúdio em dispositivos móveis uma realidade.

Apesar disso, as funções de navegação no estúdio e no resto da aplicação estão funcionais e as interfaces adaptam-se corretamente aos diversos ecrãs.

9 Bibliografia

- [1] ANACOM, “ANACOM - Manual ITED (1ª edição - Julho de 2004),” 05 07 2004. [Online]. Available: <http://www.anacom.pt/render.jsp?contentId=718561>. [Acedido em 29 09 2015].
- [2] ANACOM, “ANACOM - Manual ITUR (2.ª edição - Novembro de 2014),” 26 Novembro 2014. [Online]. Available: <http://www.anacom.pt/render.jsp?contentId=1340703>. [Acedido em 29 09 2015].
- [3] ANACOM, “ANACOM - Manual ITED (3.ª edição - Setembro de 2014),” 08 09 2014. [Online]. Available: <http://www.anacom.pt/render.jsp?contentId=1327353>. [Acedido em 29 09 2015].
- [4] A. L. Adam Lella, “2015 U.S. Digital Future in Focus - comScore, Inc,” 26 Março 2015. [Online]. Available: <http://www.comscore.com/Insights/Presentations-and-Whitepapers/2015/2015-US-Digital-Future-in-Focus>. [Acedido em Outubro 2015].
- [5] CYPE, “CYPE. Software para engenharia e construção,” CYPE, [Online]. Available: <http://programas.cype.pt/>. [Acedido em Outubro 2015].
- [6] Top, “SERVIÇOS - VÍDEOS DE WEBINARS - Top Informática – CYPE em Angola, Cabo Verde, Moçambique e Portugal. Software Estruturas, REH, Acústica, Redes, Orçamentação, Sustentabilidade, etc.,” EDIGMA, [Online]. Available: <http://www.topinformatica.pt/index.php?cat=13&item=72672>. [Acedido em Outubro 2015].
- [7] Top Informática, “- A TOP - Top Informática – CYPE em Angola, Cabo Verde, Moçambique e Portugal. Software Estruturas, REH, Acústica, Redes, Orçamentação, Sustentabilidade, etc.,” EDIGMA, [Online]. Available: <http://www.topinformatica.pt/index.php?cat=7&hrq=104>. [Acedido em Outubro 2015].
- [8] European Telecommunications Standards Institute, “Digital Video Broadcasting (DVB); Satellite Master Antenna Television (SMATV) distribution systems,” Agosto 1997. [Online]. Available:

http://www.etsi.org/deliver/etsi_en/300400_300499/300473/01.01.02_60/en_300473v010102p.pdf. [Acedido em Outubro 2015].

- [9] M. Long, Digital satellite TV handbook, Winter Beach, Fla: Mark Long Enterprises, 1998.
- [10] L. Pizarro, “Tecnologias de pares de cobre,” 1 Julho 2009. [Online]. Available: http://ave.dee.isep.ipp.pt/~see/workshop/images/ISEP_04.pdf. [Acedido em Outubro 2015].
- [11] B. Myers, S. E. Hudson e R. Pausch, “Past, Present, and Future of User,” Nova Iorque, 2000.
- [12] W. O. Galitz, The Essential Guide to User Interface Design - Second Edition, New York, NY: John Wiley & Sons, Inc., 2002.
- [13] SensioLabs, “The Twig Book,” 1 November 2015. [Online]. Available: <http://twig.sensiolabs.org/pdf/Twig.pdf>. [Acedido em November 2015].
- [14] M. Bostock, “D3.js - Data-Driven Documents,” 2015. [Online]. Available: <https://github.com/mbostock/d3/wiki>. [Acedido em Outubro 2015].
- [15] C. Pettitt, “cpettitt/dagre-d3 · GitHub,” 2013. [Online]. Available: <https://github.com/cpettitt/dagre-d3>. [Acedido em Outubro 2015].
- [16] A. Leofreddi, “ariutta/svg-pan-zoom · GitHub,” 2009. [Online]. Available: <https://github.com/ariutta/svg-pan-zoom>. [Acedido em Outubro 2015].
- [17] D. Baranovskiy, “Raphaël—JavaScript Library,” 2008. [Online]. Available: <http://raphaeljs.com/>. [Acedido em Outubro 2015].
- [18] SensioLabs, “What is Symfony,” 2014. [Online]. Available: <https://symfony.com/what-is-symfony>. [Acedido em Outubro 2015].
- [19] SensioLabs, “Symfony - Day 4: The Controller and the View,” SensioLabs, [Online]. Available: http://symfony.com/legacy/doc/jobeeet/1_2/en/04?orm=Propel. [Acedido em Outubro 2015].
- [20] “Convert CAD to PDF or Images Smart,” CoolUtils, 27 Outubro 2015. [Online]. Available: <http://www.coolutils.com/TotalCADConverter>. [Acedido em Outubro 2015].

- [21] W3Schools, "HTML5 Introduction," [Online]. Available: http://www.w3schools.com/html/html5_intro.asp. [Acedido em Outubro 2015].
- [22] W3Schools, "CSS Tutorial," [Online]. Available: <http://www.w3schools.com/css/default.asp>. [Acedido em Outubro 2015].
- [23] W3Schools, "JavaScript Tutorial," [Online]. Available: <http://www.w3schools.com/js/default.asp>. [Acedido em Outubro 2015].
- [24] jQuery Foundation, "jQuery," [Online]. Available: <https://jquery.com/>. [Acedido em Outubro 2015].
- [25] M. Otto e J. Thornton, "Bootstrap · The world's most popular mobile-first and responsive front-end framework.," 19 Agosto 2011. [Online]. Available: <http://getbootstrap.com/>. [Acedido em Outubro 2015].
- [26] R. Lerdorf, "PHP: Hypertext Preprocessor," 1994. [Online]. Available: <https://secure.php.net/>. [Acedido em Outubro 2015].
- [27] SensioLabs, "Databases and Doctrine (The Symfony Book)," SensioLabs, [Online]. Available: <http://symfony.com/doc/current/book/doctrine.html>. [Acedido em Outubro 2015].
- [28] SensioLabs, "Controller (The Symfony Book)," SensioLabs, [Online]. Available: <http://symfony.com/doc/current/book/controller.html>. [Acedido em Outubro 2015].
- [29] SensioLabs, "Forms (The Symfony Book)," SensioLabs, [Online]. Available: <http://symfony.com/doc/current/book/forms.html>. [Acedido em Outubro 2015].
- [30] SensioLabs, "Form Types Reference (The Symfony Reference)," SensioLabs, [Online]. Available: <http://symfony.com/doc/current/reference/forms/types.html>. [Acedido em Outubro 2015].
- [31] SensioLabs, "The View (The Symfony Quick Tour current)," SensioLabs, [Online]. Available: http://symfony.com/doc/current/quick_tour/the_view.html. [Acedido em Outubro 2015].

- [32] Federal Information Processing Standards, "SECURE HASH STANDARD," Março 2012. [Online]. Available: <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>. [Acedido em Outubro 2015].
- [33] X. W. a. H. Yu, "How to Break MD5 and Other Hash Functions," [Online]. Available: <http://merlot.usc.edu/csac-f06/papers/Wang05a.pdf>. [Acedido em Outubro 2015].
- [34] P. Patel, J. Patel e P. Virparia, "A Cryptography Application using Salt Hash," 6 Junho 2013. [Online]. Available: <http://www.ijaiem.org/Volume2Issue6/IJAIEM-2013-06-22-060.pdf>. [Acedido em Outubro 2015].

10 Anexos

10.1 Anexo A

Nesta secção serão apresentadas as tecnologias mais comuns utilizadas no desenvolvimento deste projeto.

10.1.1 Ferramentas Web

HTML

HyperText Markup Language 5 (HTML 5) consiste numa linguagem estruturada de programação composta por *tags*, interpretadas por *browsers*, utilizada para a produção de páginas *web*. O número 5 representa a versão atual da linguagem [21].

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

Tabela 4 - excerto de código de uma página HTML

CSS 3

Cascading Style Sheets 3 (CSS 3) é uma linguagem de definição de estilos a aplicar em páginas HTML, permitindo facilmente editar o aspeto, *layout*, criar animações, entre outras operações. O número 3 representa a versão atual da linguagem e é compatível com versões anteriores [22].

Uma página HTML com a mesma estrutura, mas com ficheiros de estilos diferentes, pode apresentar interfaces completamente distintas.

Os estilos CSS podem ser aplicados numa página de três formas distintas, apresentadas por ordem de prioridade:

- Colocado o código diretamente no elemento.
- Colocado o código no cabeçalho da página.
- Carregando os estilos a partir de um ficheiro.

```
body {  
  background-color: #d0e4fe;  
}
```

Tabela 5 - excerto de código de uma folha de estilos CSS

JavaScript

JavaScript é uma linguagem de programação *web*, orientada a objetos, interpretada, que contém uma abordagem de programação multi-paradigma. Inicialmente foi concebido para utilização *client-side*, fornecendo mecanismos de controlo do *browser* e do *Document Object Model* (DOM). Atualmente, para além da utilização *client-side* existem também aplicações onde o código é interpretado do lado do servidor (ex. NodeJs) [23].

jQuery

jQuery é uma biblioteca *JavaScript* que foi desenvolvida para simplificar a criação de *scripts*, facilitando operações de navegação na página HTML, o controlo do *Document Object Model* (DOM), a gestão de eventos, criação de animações, entre outras operações [24].

Bootstrap

Bootstrap é uma *framework front-end* que fornece um conjunto de ferramentas para a desenvolver páginas *web* flexíveis e responsivas de forma rápida e fácil.

Vem equipado com um conjunto de *templates* HTML e CSS para componentes como tipografia, formulários, botões, tabelas, navegação entre outros e é compatível com a maioria dos *browsers* atuais [25].

10.1.2 Ferramentas de desenvolvimento

PHP

PHP é uma ferramenta de *scripting* grátis, orientada à *web*. O código PHP é sempre interpretado do lado do servidor e só posteriormente apresentado do lado do cliente. A linguagem PHP apresenta um paradigma de programação orientado a objetos, possui portabilidade (capacidade de correr em qualquer plataforma) e ainda tipagem dinâmica (não é necessário declarar o tipo de dados das variáveis) [26].

A escolha de PHP face a outras linguagens de *scripting* para aplicações *web*, tal como o ASP.NET, deve-se ao facto de se ter recorrido à *framework Symfony* no desenvolvimento do protótipo, tendo este sido desenvolvido para PHP e em esta linguagem de programação fornecer todas as funcionalidades desejadas.

10.2 Anexo B

Nesta secção irão ser descritas mais pormenorizadamente as tabelas que compõem o modelo de dados da aplicação que oferece suporte à arquitetura do sistema.

Autenticação

A entidade autenticação pretende armazenar informação temporária, relativa às tentativas de autenticação do utilizador, de maneira a garantir que os pedidos de *login* são feitos pelo próprio.

Atributo	Descrição
<u>id</u>	Identificador único da autenticação
sessionId	Token de tentativa de autenticação
email	<i>Email</i> do utilizador
timestamp	<i>Timestamp</i> da tentativa de autenticação

Tabela 6 - entidade Autenticação

Perfil

A entidade perfil pretende armazenar os diferentes níveis de privilégios de utilizador.

Atributo	Descrição
<u>id</u>	Identificador único do perfil
nome	Nome do perfil

Tabela 7 - entidade Perfil

A entidade perfil tem uma relação de um-para-muitos com a entidade utilizador.

Utilizador

A entidade utilizador pretende armazenar as informações de início de sessão e dados adicionais do utilizador.

Atributo	Descrição
<u>id</u>	Identificador único do utilizador
idPerfil	Identificador do perfil de utilizador
nome	Primeiro e último nome do utilizador
email	Endereço de <i>email</i> do utilizador
nif	Número de Identificação Fiscal do utilizador
telefone	Número de telefone do utilizador
password	Senha do utilizador
salt	<i>Token</i> de autenticação da <i>password</i>
dataRegisto	Data de registo do utilizador
ultimoAcesso	Data de último acesso à aplicação, pelo utilizador

Tabela 8 - entidade Utilizador

A entidade utilizador tem uma relação de um-para-muitos com a entidade obra e muitos-para-um com a entidade perfil.

Obra

A entidade obra pretende armazenar os diferentes projetos do utilizador.

Atributo	Descrição
<u>id</u>	Identificador único da obra
idUtilizador	Identificador único do utilizador
nome	Nome do projeto
numObra	Número da obra
construtor	Nome do construtor da obra
donoObra	Proprietário da obra
numAlvara	Número do alvará
numProj	Número do projeto
observacoes	Observações do projeto
morada	Morada da obra
latitude	Coordenadas de latitude da obra
longitude	Coordenadas de longitude da obra

path	Caminho para o ficheiro de imagem da obra
dataCriacao	Data de criação do projeto
dataAlteracao	Data de alteração do projeto

Tabela 9 - entidade Obra

A entidade obra tem uma relação de um-para-muitos com a entidade planta e muitos-para-um com a entidade utilizador.

Tipo

A entidade tipo pretende armazenar os diferentes tipos de plantas.

Atributo	Descrição
<u>id</u>	Identificador único do tipo de planta
descricao	Nome do tipo de planta

Tabela 10 - entidade Tipo

A entidade tipo tem uma relação de um-para-muitos com a entidade planta.

Planta

A entidade planta pretende armazenar informação das diferentes plantas de uma obra.

Atributo	Descrição
<u>id</u>	Identificador único da planta
idObra	Identificador único do projeto em que a planta está
idTipo	Identificador único do tipo de planta
descricao	Descrição da planta
observacoes	Observações relacionadas com a planta
path	Caminho para o ficheiro DWG
svg	Código do desenho da planta em SVG

largura	Largura da planta (Planta Vertical)
vertical	Booleano que indica se a planta é vertical

Tabela 11 - entidade Planta

A entidade planta tem uma relação de um-para-muitos com a entidade versao e uma relação de muitos-para-um com a entidade tipo e com a entidade obra.

Versão

A entidade versao pretende armazenar informação das diferentes versões de plantas.

Atributo	Descrição
<u>id</u>	Identificador único da versão
idPlanta	Identificador único da planta
data	Data de criação da versão
comentario	Comentário introduzido pelo utilizador
escala	Escala da planta
tamanho	Tamanho da planta face aos nós

Tabela 12 - entidade Versão

A entidade versao tem uma relação de um-para-muitos com a entidade no e piso, e uma relação muitos-para-um com a entidade planta.

Piso

A entidade piso pretende armazenar informação dos diferentes pisos de uma planta vertical.

Atributo	Descrição
<u>id</u>	Identificador único do piso
idVersao	Identificador único da versão da planta
numero	Número do piso
nome	Nome do piso

altura	Altura do piso
---------------	----------------

Tabela 13 - entidade Piso

A entidade piso tem uma relação de muitos-para-um com a entidade versao.

TipoComponente

A entidade tipocomponente pretende armazenar os diferentes tipos de componentes a instalar.

Atributo	Descrição
<u>id</u>	Identificador único do tipo de componente
nome	Nome do tipo de componente
descrição	Descrição do tipo de componente
svg	Código SVG de representação do ícone do tipo de componente

Tabela 14 - entidade TipoComponente

A entidade tipocomponente tem uma relação de um-para-muitos com a entidade modeloaplicacao.

TipoAplicacao

A entidade tipoaplicacao pretende armazenar os diferentes tipos de aplicação para tipos de componente.

Atributo	Descrição
<u>id</u>	Identificador único do tipo de aplicação
nome	Nome do tipo de aplicação
descrição	Descrição do tipo de aplicação

Tabela 15 - entidade TipoAplicacao

A entidade tipoaplicacao tem uma relação de um-para-muitos com a entidade modeloaplicacao.

ModeloAplicacao

A entidade modeloaplicacao pretende mapear os diferentes tipos de componente com tipos de aplicação dos componentes.

Atributo	Descrição
id	Identificador único do modelo de aplicação
idTipoComponente	Identificador do tipo de componente
idTipoAplicacao	Identificador único do tipo de aplicação

Tabela 16 - entidade ModeloAplicacao

A entidade modeloaplicacao tem uma relação de um-para-muitos com a entidade no e com a entidade componente e tem uma relação de muitos-para-um com a entidade tipocomponente e com a entidade tipoaplicacao.

No

A entidade no pretende armazenar os diferentes nós de componentes inseridos no desenho de uma versão da planta.

Atributo	Descrição
id	Identificador único do nó
idVersao	Identificador único da versão
idModeloAplicacao	Identificador único do modelo de aplicação do nó
x	Coordenadas x do nó
y	Coordenadas y do nó
altura	Altura de instalação do nó
label	<i>Label</i> identificadora do nó
idOrigem	Identificador único do nó de origem (ATI Planta Vertical)
nivelSinal	Nível de sinal no nó

Tabela 17 - entidade Nó

A entidade no tem uma relação de um-para-muitos com a entidade ligacao e com a entidade material, tem uma relação de muitos-para-um com a entidade modeloaplicacao, com a entidade versao e com a própria entidade no.

Ligação

A entidade ligação pretende armazenar as diferentes ligações entre nós.

Atributo	Descrição
<u>id</u>	Identificador único da ligação
idNo1	Identificador único no primeiro nó
idNo2	Identificador único do segundo nó
comprimento	Comprimento à escala entre os dois nós

Tabela 18 - entidade Ligação

A entidade ligacao tem uma relação de um-para-muitos com a entidade tuboligacao e uma relação de muitos-para-um com a entidade no.

Tubo

A entidade tubo pretende armazenar os diferentes tipos de tubos.

Atributo	Descrição
<u>id</u>	Identificador único do tubo
diametro	Diâmetro do tubo

Tabela 19 - entidade Tubo

A entidade tubo tem uma relação de um-para-muitos com a entidade tuboligacao.

TuboLigacao

A entidade tuboligacao pretende mapear as ligações com um tipo de tubo.

Atributo	Descrição
<u>id</u>	Identificador único da entidade
idTubo	Identificador único do tipo de tubo
idLigacao	Identificador único da ligação

Tabela 20 - entidade TuboLigacao

A entidade tuboligacao tem uma relação de um-para-muitos com a entidade tuboligacaocaboqualidade e uma relação de muitos-para-um com a entidade tubo e com a entidade ligacao.

Cabo

A entidade cabo pretende armazenar os diferentes tipos de cabos para instalação.

Atributo	Descrição
<u>id</u>	Identificador único do tipo de cabo
tipo	Nome do tipo de cabo

Tabela 21 - entidade Cabo

A entidade cabo tem uma relação de um-para-muitos com a entidade caboqualidade e com a entidade subcomponente.

Qualidade

A entidade qualidade tem como objetivo armazenar as diferentes qualidades de cabos.

Atributo	Descrição
<u>id</u>	Identificador único da qualidade de cabo
nome	Nome da qualidade de cabo

Tabela 22 - entidade Qualidade

A entidade qualidade tem uma relação de um-para-muitos com a entidade caboqualidade.

CaboQualidade

A entidade caboqualidade tem como objetivo mapear uma qualidade de cabo a diversos tipos de cabos.

Atributo	Descrição
<u>id</u>	Identificador único da entidade
idCabo	Identificador único do tipo de cabo
idQualidade	Identificador único da qualidade de cabo

Tabela 23 - entidade CaboQualidade

A entidade caboqualidade tem uma relação de um-para-muitos com a entidade tuboligacaocaboqualidade e uma relação de muitos-para-um com a entidade qualidade e com a entidade cabo.

TuboLigacaoCaboQualidade

A entidade tuboligacaocaboqualidade pretende armazenar os cabos e sua qualidade que passam em determinado tubo inserido numa ligação.

Atributo	Descrição
<u>id</u>	Identificador único da entidade
idTuboLigacao	Identificador único da entidade TuboLigacao
idCaboQualidade	Identificador único da entidade CaboQualidade

Tabela 24 - entidade TuboLigacaoCaboQualidade

A entidade tuboligacaocaboqualidade tem uma relação de muitos-para-um com a entidade tuboligacao e com a entidade caboqualidade.

Componente

A entidade componente pretende armazenar os diversos equipamentos que podem ser instalados num projeto.

Atributo	Descrição
id	Identificador único do componente
idModeloAplicacao	Identificador único do modelo de aplicação
codigo	Código de referência do equipamento
descricao	Descrição do equipamento
preco	Preço do equipamento
atenuacao	Atenuação que o equipamento produz
ocupadoUs	Número de unidade ocupadas pelo equipamento
reservadoUs	Número de unidades reservadas no equipamento
Imagem	Caminho para o ficheiro de imagem do equipamento
dimensaoL	Largura do equipamento
dimensaoA	Altura do equipamento
dimensaoP	Profundidade do equipamento
entradasCC	Número de entradas de cabo coaxial que o equipamento suporta
entradasFO	Número de entradas de fibra ótica que o equipamento suporta
entradasPC	Número de entradas de par de cobre que o equipamento suporta
saidasCC	Número de saídas de cabo coaxial que o equipamento suporta
saidasFO	Número de saídas de fibra ótica que o equipamento suporta
saidasPC	Número de saídas de par de cobre que o equipamento suporta
localizacao	Localização da instalação do equipamento (interior/exterior)
link	Ligação ao endereço do equipamento no <i>site</i> da empresa
observacoes	Observações do equipamento
minSub	Mínimo de módulos a instalar no equipamento, caso este não seja um subcomponente
maxSub	Máximo de módulos a instalar no equipamento, caso este não seja um subcomponente
subcomponente	Booleano que indica se é subcomponente

Tabela 25 - entidade Componente

A entidade componente tem uma relação de um-para-muitos com a entidade subcomponente e com a entidade material e uma relação de muitos-para-um com a entidade modeloaplicacao.

Subcomponente

A entidade subcomponente pretende definir que equipamentos podem ser utilizados como módulos num componente “pai” e, opcionalmente, limitá-lo à instalação em determinado tipo de tecnologia (tipo de cabo).

Atributo	Descrição
<u>id</u>	Identificador único do subcomponente
idPai	Identificador único do componente Pai
idFilho	Identificador único do componente Filho
idCabo	Identificador único do tipo de cabo

Tabela 26 - entidade Subcomponente

A entidade subcomponente tem uma relação de muitos-para-um com a entidade cabo e com a entidade componente.

Material

A entidade material pretende armazenar os equipamentos seleccionados para instalação numa versão de uma planta.

Atributo	Descrição
<u>id</u>	Identificador único do material
idNo	Identificador único do nó
idComponente	Identificador único do equipamento

Tabela 27 - entidade Material

A entidade material tem uma relação de muitos-para-um com a entidade componente e com a entidade no.

10.3 Anexo C

Nesta secção irão ser abordadas diversas funcionalidades e configurações do *Symfony* que irão dar suporte à criação da aplicação, desde a construção da base de dados, à criação de *templates* e formulários e ainda alguns mecanismos de segurança.

10.3.1.1 Doctrine

De maneira a facilitar a criação de mecanismos de persistência e de leitura de dados da base de dados, o *Symfony* fornece um componente chamado *Doctrine* [27]. Este componente é utilizado para mapear objetos numa base de dados relacional, como a que irá ser utilizada neste projeto.

Para criar a base de dados onde serão armazenados os dados da aplicação é primeiro necessário configurar os parâmetros de ligação. A configuração é feita no ficheiro “app/config/parameters.yml”.

De seguida segue-se o exemplo de uma configuração utilizada:

```
parameters:
  database_driver: pdo_mysql
  database_host: 127.0.0.1
  database_port: 3306
  database_name: ited
  database_user: root
  database_password:
```

Após definidos os parâmetros basta executar um comando na linha de comandos para construir a base de dados:

- `php app/console doctrine:database:create`

O próximo passo na criação de uma base de dados passaria pela construção das tabelas. Para isto devem ser definidas *Entities*.

As *Entities* tratam-se de classes PHP onde são definidos os campos e o seu tipo de dados, de maneira a que o *Symfony* as possa mapear em tabelas da base de dados. Isto faz com que seja possível persistir e ler objetos inteiros na base de dados.

Tendo feito isto executam-se os comandos que permitem construir as tabelas na base de dados e os métodos nas classes PHP que permitem ler ou persistir informação.

- `php app/console doctrine:generate:entities IEETAITEDBundle`
- `php app/console doctrine:schema:update`

Nota: “IEETAITEDBundle” foi o nome atribuído à *bundle* do projeto criado.

10.3.1.2 *Controllers*

Os *Controllers* (controladores) no *Symfony* tratam-se de ações que podem ser chamadas a receber uma *HTTP Request*, executar uma ação intermédia e por fim, devolver uma *HTTP Response*. O tipo de resposta devolvida pode ser uma página HTML, um documento XML, um *array* JSON, uma imagem, entre outros [28].

Pelo meio, entre receber a *request* e devolver a *response* pode-se executar uma série de operações tais como executar operações na base de dados, definir sessões de utilizador, entre outros, mas o objetivo final do controlador é sempre o mesmo: devolver um *HTTP Response*.

10.3.1.3 *Forms*

De forma a disponibilizar formulários para inserção ou edição de dados nas páginas *web*, o *Symfony* possui um componente, denominado de *Form* (Formulário) [29]. Este componente agiliza a criação e personalização de formulários e é disponibilizado sobre a forma de uma biblioteca *standalone*, podendo ser utilizada fora de projetos *Symfony*.

Este componente permite, entre outras funcionalidades, definir o tipo de dados dos campos, opções disponíveis (em campos do tipo *select*), atributos nos *inputs*, definir a *label*, tornar o campo obrigatório, desativá-lo, entre outras opções.

Permite também a definição de tipos de dados, fora do âmbito do tipo de dados para *inputs* de páginas HTML (*texto*, *number*, entre outros), fornecendo ainda ferramentas de apresentação e validação para esses tipo de dados (ex. *birthday*, *money*, *percent*, entre outros). Estas funcionalidades são disponibilizadas através do componente chamado *FormType* [30].

Neste projeto foram seguidas normas de boas práticas (sugeridas na documentação do *Symfony*) criando classes separadas para cada formulário ligado a uma das entidades definidas anteriormente, ao invés de as criar diretamente nos *Controllers*.

Adotar esta política faz com que seja possível reutilizar os formulários, adaptando-os a diferentes cenários (criação, edição, entre outros...).

A ligação de um formulário a uma entidade facilita a criação e validação dos dados introduzidos nos *inputs*. Ao fazer isto é ainda possível introduzir campos adicionais que não pertençam à entidade mapeada (ex. confirmar senha).

A definição de campos num formulário é feito através da função “add” que recebe como parâmetros: nome do campo, tipo de dados, opções.

Um exemplo da adição de um campo a um formulário pode ser observado abaixo:

```
->add('tipo', 'text', array(
    'required' => true,
    'label' => 'Tipo',
    'attr' => array(
        'class' => 'form-control',
        'placeholder' => 'Tipo de cabo',
        'maxlength' => 35
    )
));
```

A renderização dos formulários é feita através de *templates* TWIG, produzidos no componente *View* do *Symfony* apresentado de seguida.

10.3.1.4 Views

View (Vista) [31] é o componente responsável por renderizar o conteúdo gerado a partir de uma ação executada num *Controller* (Controlador).

Por sua vez, nas vistas são criados *templates* utilizando TWIG.

Através destes *templates* é possível estruturar e reutilizar informação e *layouts* de páginas *web*. Por exemplo, uma barra de menu é algo que terá a mesma estrutura em diversas páginas, mas cujo conteúdo poderá variar, consoante o tipo de utilizador autenticado no sistema. Desta forma é possível criar um *layout* para a seção do menu e, através de funções condicionais disponíveis em

TWIG, alterar o seu conteúdo. Este *layout* seria depois integrado no *layout* das páginas construídas nas vistas.

Como foi referido anteriormente a vista renderiza conteúdo gerado num controlador. Isto é particularmente útil para apresentar informação de entidades obtida na base de dados e, como já foi também mencionado anteriormente, para renderizar os formulários.

Na renderização de formulários existem diversas opções. É possível renderizar todo o formulário através de três comandos:

- **form_start(form)**: renderiza o início do formulário e define o *encode type* correto quando existem campos de ficheiros. Recebe como parâmetro o objeto do formulário enviado pelo controlador.
- **form_widget(form)**: renderiza todos os campos do formulário (*label*, *input* e mensagem de erro na validação). Recebe como parâmetro o objeto do formulário enviado pelo controlador.
- **form_end(form)**: renderiza o resto dos campos que ainda não o foram e fecha o formulário. Recebe como parâmetro o objeto do formulário enviado pelo controlador.

É ainda possível ter controlo sobre os campos a apresentar e a maneira como eles são apresentados individualmente.

10.3.1.5 Security

De forma a garantir a segurança e integridade dos dados na aplicação, foram implementados uma série de mecanismos de segurança, através do módulo de segurança do *Symfony*, que irão ser abordados de seguida. Este conjunto de mecanismos estão centrados à volta das contas de utilizador.

Um dos primeiros passos a efetuar trata-se da implementação de um método de autenticação dos utilizadores. Esta autenticação irá ser feita recorrendo à entidade “Utilizador”. O processo de autenticação é efetuado faseadamente em diversas partes da aplicação. Neste processo são definidas três rotas.

A primeira é a rota que leva o utilizador à página que contém o formulário de autenticação. Após submissão dos dados do formulário, o sistema é redirecionado para uma segunda página de verificação dos dados enviados. Caso sejam introduzidos dados válidos o utilizador está autenticado e prossegue na navegação, para uma página que depende do seu tipo de conta.

Caso a autenticação falhe o utilizador é então redirecionado para uma terceira rota onde serão apresentadas as razões que levaram a esta falha. No caso concreto desta aplicação, caso a autenticação falhe o utilizador é redirecionado de novo para a página de autenticação, de modo a poder corrigir os erros nos dados introduzidos.

Como medida adicional de segurança dos dados do utilizador, a sua senha é encriptada, antes de ser armazenada, recorrendo a um *Secure Hash Algorithm* (SHA) de 512 bits (SHA-512) que utiliza como parâmetro de entrada um *salt* gerado através de um *Message-Digest Algorithm 5* (MD5).

SHA é um conjunto de funções de *hashing*, utilizadas para encriptar dados [32]. As funções de *hashing* são operações matemáticas aplicadas em dados que resultam num valor final, denominado de *hash*. Ao comparar o valor de um *hash* previamente conhecido com o valor do *hash* calculado, é possível determinar a integridade dos dados. A versão do algoritmo de SHA utilizada foi a de 512 bits, sendo esta a versão mais difícil de quebrar devido à sua dimensão e ao poder de processamento necessário para o fazer.

MD5, à semelhança do SHA, é também um algoritmo de *hash* mas unidirecional de 128 bits [33]. Ser unidirecional significa que o *hash* não pode voltar a ser convertido no texto original, o que faz com que a sua utilização seja ideal quando se pretende encriptar informação confidencial, como é o caso da senha do utilizador.

Salt trata-se fundamentalmente de um conjunto de dados gerados aleatoriamente, que serve como um parâmetro opcional numa função de *hashing* [34].

Sempre que é registado um utilizador novo, ou que um existente altere a sua senha, é gerado um novo *salt* e, recorrendo a ele, encriptada a nova senha.

Outro dos mecanismos implementados trata-se de um sistema de controlo de acesso a diversas funcionalidades do sistema. O *Symfony* integra um módulo de segurança onde podem ser definidos *roles*, que se tratam de forma simplista, de níveis de privilégios na aplicação que são atribuídos a diferentes utilizadores.

Os *roles* definidos foram o **IS_AUTHENTICATED_ANONYMOUSLY**, **ROLE_USER**, **ROLE_ADMIN** e **ROLE_SUPER_ADMIN**, tendo sido listados de forma crescente em relação ao nível de privilégios e a hierarquia entre eles foi definida no campo “*role_hierarchy*” no ficheiro de segurança localizado em “*app/config/security.yml*”.

Tipos de utilizador que se encontram acima na hierarquia têm também acesso às páginas a que tipos de utilizador de hierarquia inferior têm acesso.

Tendo definida uma hierarquia de privilégios em utilizadores, é possível definir no campo “*access_control*” a que páginas da aplicação é que um tipo de utilizador pode aceder. Este procedimento é feito escrevendo o caminho na aplicação, através de uma *Regular Expression* (RegEx), seguido do conjunto de *roles* que têm privilégio de acesso.

Uma destas regras pode ser visualizada no exemplo abaixo, que regula o acesso às páginas de autenticação do utilizador:

```
- { path: ^/session/, roles: IS_AUTHENTICATED_ANONYMOUSLY }
```

O processo de autorização de acesso é feito percorrendo a lista de rotas definidas no ficheiro de segurança referido anteriormente. Se o caminho da página corresponder a uma rota registada e o utilizador tiver um *role* válido, o acesso é garantido, caso contrário o acesso é negado.

O processo de verificação das permissões de acesso a rotas pode ser visualizado abaixo na Figura 32:

qualidade_create	/qualidade/	Path "/qualidade/" does not match
qualidade_new	/qualidade/new	Path "/qualidade/new" does not match
qualidade_edit	/qualidade/{id}/edit	Path "/qualidade/{id}/edit" does not match
qualidade_update	/qualidade/{id}	Path "/qualidade/{id}" does not match
qualidade_delete	/qualidade/{id}/delete	Path "/qualidade/{id}/delete" does not match
openlogin	/session/openlogin	Path "/session/openlogin" does not match
login_form	/session/login	Route matches!

Note: The above matching is based on the configuration for the current router which might differ from the configuration used while routing this request.

Figura 32 - painel de routing do SymfonyProfiler para a rota session/login

Para o **role IS_AUTHENTICATED_ANONYMOUSLY**, que se trata de um utilizador não autenticado, foi definido apenas um conjunto de páginas de acesso, sendo estas todas as páginas cujo caminho comece por “/session”, garantindo assim acesso à página que contém o formulário de autenticação do utilizador, às restantes páginas onde a validação das credenciais submetidas é efetuada e, finalmente, à página de terminar sessão.

Como tal foi definida a seguinte regra, já demonstrada anteriormente:

```
- { path: ^/session/, roles: IS AUTHENTICATED ANONYMOUSLY }
```

O tipo de utilizador **ROLE_USER** trata-se do utilizador autenticado, sem privilégios de administrador. Este tipo de utilizador, para além de ter acesso às páginas a que o utilizador do tipo **IS_AUTHENTICATED_ANONYMOUSLY** tem acesso, deve ter acesso aos seus projetos, às plantas dentro deles, e à página de edição de informações da conta. A página do estúdio está incluída dentro do conjunto de páginas das plantas.

Para esse efeito, foram definidas as seguintes regras:

```
- { path: ^/obra/, roles: [ROLE_USER] }  
- { path: ^/planta/, roles: [ROLE_USER] }  
- { path: ^/utilizador/[0-9]+/settings, roles: [ROLE_USER] }
```

No que diz respeito às contas de administrador, estas devem ter acesso a todas as funcionalidades das contas de utilizador, assim como acesso ao *BackOffice* para edição de conteúdo. Como existem diferenças de permissões entre um utilizador do tipo **ROLE_ADMIN** e de um **ROLE_SUPER_ADMIN**, foi necessário discriminar o conjunto de páginas a que o utilizador do tipo **ROLE_ADMIN** tem acesso, ao invés de simplesmente garantir acesso às páginas que ainda não tinham sido restringidas.

Sendo assim, foram definidas as seguintes regras:

```
- { path: ^/cabo/, roles: [ROLE_ADMIN] }  
- { path: ^/cabocomponente/, roles: [ROLE_ADMIN] }  
- { path: ^/caboqualidade/, roles: [ROLE_ADMIN] }  
- { path: ^/componente/, roles: [ROLE_ADMIN] }  
- { path: ^/modeloaplicacao/, roles: [ROLE_ADMIN] }  
- { path: ^/perfil/, roles: [ROLE_ADMIN] }  
- { path: ^/qualidade/, roles: [ROLE_ADMIN] }  
- { path: ^/subcomponente/, roles: [ROLE_ADMIN] }  
- { path: ^/tipo/, roles: [ROLE_ADMIN] }  
- { path: ^/tipoaplicacao/, roles: [ROLE_ADMIN] }  
- { path: ^/tipocomponente/, roles: [ROLE_ADMIN] }  
- { path: ^/tubo/, roles: [ROLE_ADMIN] }
```

```
- { path: ^/versao/, roles: [ROLE_ADMIN] }  
- { path: ^/$, roles: [ROLE_ADMIN] }
```

De notar que não foram criadas regras de controlo de acesso para páginas de edição de obras e plantas no *BackOffice*, a pedido da empresa para qual a aplicação está a ser desenvolvida.

A razão por detrás desta decisão tem que ver com o facto de não ser desejável que outras entidades, para além das que criaram o projeto, tivessem acesso ao conteúdo criado.

No que diz respeito ao utilizador do tipo **ROLE_SUPER_ADMIN**, a única diferença de privilégios que tem para um utilizador do tipo **ROLE_ADMIN** é a gestão das contas de utilizadores. Por exclusão de partes nas páginas de acesso controlado, garante-se o acesso à página de edição de utilizadores garantindo acesso ao resto das páginas para utilizadores do tipo **ROLE_SUPER_ADMIN**, através da regra apresentada abaixo:

```
- { path: ^/, roles: [ROLE_SUPER_ADMIN] }
```

Para além da verificação das rotas a que o utilizador acede, é ainda verificado, através da sessão armazenada durante a autenticação de um utilizador, se o conteúdo a ser apresentado corresponde ao utilizador que o está a tentar consultar.

Esta verificação é efetuada através dos *controllers* do *Symfony* antes de a página ser apresentada. Caso um utilizador tente aceder a uma página fora do seu âmbito de permissões, através do acesso a rotas ou a conteúdo proibido, é apresentada uma página de erro com o código “403 Forbidden” indicando que o acesso foi negado.