



PAULO ANDRÉ BRANCO DIAS

Licenciado em Ciências e Engenharia Informática

EVOLUÇÃO DO MÓDULO DE ESTATÍSTICA DOS RECURSOS GEOLÓGICOS

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa
31 de Maio, 2023

EVOLUÇÃO DO MÓDULO DE ESTATÍSTICA DOS RECURSOS GEOLÓGICOS

PAULO ANDRÉ BRANCO DIAS

Licenciado em Ciências e Engenharia Informática

Orientador: Nuno Luís Raimundo Evangelista

Senior Consultant, Unipartner

Coorientador: Pedro Abílio Duarte de Medeiros

Associate Professor, NOVA University of Lisbon - School of Science and Technology

RESUMO

A instabilidade da solução atual do software disponibilizado visa à proposta para melhoria da mesma. Com a existência de versões mais atualizadas, torna-se mais viável e prática a comunicação com as entidades de suporte (Fabasoft e Microsoft) com a atualização da ferramenta. As versões desatualizadas deixam de ser compatíveis com ferramentas de produtividade e com os próprios *browsers*. Pode criar instabilidade na utilização da aplicação, isto devido a erros que surgem no sistema que suporta as bases de dados aplicacionais. Com o objetivo de melhorar e tirar partido do melhor que a ferramenta nos pode oferecer, pretende-se desenvolver um novo projeto na versão mais recente do Fabasoft.

A solução proposta endereça os riscos identificados através da implementação de um projeto de resultados rápidos e consistentes. Será um projeto alinhado com a estratégia em desenvolvimento pelo cliente. Será desenvolvido um novo módulo para a estatística dos recursos geológicos: *minas, pedreiras e águas*, uma vez que partilham bastantes semelhanças em termos de modelo conceptual. Também irá ser desenvolvido um módulo composto pelas instalações físicas de cada área e os respetivos processos estatísticos (inquérito único de minas e pedreiras e estatística de águas). É importante referir que outro dos módulos principais do projeto é a componente analítica que permitirá disponibilizar uma ferramenta de análise aos dados estatísticos abordados neste projeto.

Uma vez completada esta atualização, a interação com a ferramenta de gestão documental será muito mais simples e confiável, sendo feita a recolha de informação num novo portal de acesso e desenvolvido em tecnologia atual e aberta.

Palavras-chave: Fabasoft, Estatística, Recursos Geológicos, Atualização

ABSTRACT

The instability of the current solution of the software that is being used aims to a proposal to improve it. With the existence of more up-to-date versions, it becomes more feasible and practical to communicate with support entities (Fabasoft and Microsoft) by updating the tool. Outdated versions are no longer compatible with productivity tools and with *browsers*. It can create instability in the use of the application, due to errors that appear in the system that supports the application databases. In order to improve and take advantage of the best that this tool can offer us, we intend to develop a new project in the latest version of *Fabasoft eGov-Suite*.

The proposed solution addresses the identified risks through the implementation of a project with fast and consistent results. It will be a project aligned with the strategy that is being developed by the client. A new module will be developed for the statistics of geological resources: *mines*, *quarries* and *waters* since they are very similar in terms of the conceptual model and as they are, actually, hydrogeological resources. A module will also be developed comprising the physical facilities of each area and the respective statistical processes (single inquiry of mines and quarries and water statistics). It is important to mention that another of the project's main modules is the analytical component that will make available an tool to analyze the statistical data addressed in this project.

Once this update is completed, the interaction with the document management tool will be much simpler and more reliable, with the collection of information being carried out in a new access portal and developed using current and open technology.

Keywords: Fabasoft, Statistics, Geological resources, Update

ÍNDICE

Índice de Figuras	vi
1 Introdução	1
1.1 Relevância e Motivação	1
1.2 Objetivos	2
1.3 Contexto	3
1.4 Trabalho Esperado e Contribuições	4
1.5 Estrutura do Documento	5
2 Trabalho Relacionado e Enquadramento	7
2.1 Recursos Geológicos e a sua importância	7
2.1.1 Pedreiras	7
2.1.2 Minas	8
2.1.3 Recursos Hidrogeológicos	9
2.2 Modelo Conceptual	9
2.2.1 Processo de Negócio	10
2.2.2 Componente Analítica	11
2.3 Questões e riscos a abordar	11
2.4 Enquadramento	12
2.4.1 Arquitetura da Ferramenta	12
2.4.2 Tecnologia	14
2.5 Conclusões	16
3 Avaliação de alternativas	17
3.1 Alternativas de Ferramentas de Gestão Documental	17
3.1.1 EdocLink	17
3.1.2 EdocLink vs Fabasoft	18
3.1.3 SmartDocs	19
3.1.4 SmartDocs vs Fabasoft	20

3.1.5	GFIDocs	20
3.1.6	GFIDocs vs Fabasoft	21
3.1.7	Fabasoft: Porquê?	22
4	Arquitetura e Organização da Solução	23
4.1	Arquitetura da Solução	23
4.2	Conclusões	29
5	Descrição da Implementação	30
5.1	Abordagem da Solução	30
5.1.1	Instalações Físicas	30
5.1.2	Estatísticas dos Recursos Geológicos	31
5.1.3	Localizações, Regiões e Portal de Serviços	32
5.1.4	Importância da Hierarquia	33
5.2	Particularidades da Ferramenta	33
5.2.1	Desenvolvido na DSL <i>Object Model</i>	33
5.2.2	Desenvolvido na DSL <i>User Interface</i>	36
5.2.3	Desenvolvido na DSL <i>Use Case</i>	39
5.3	Gestão de acessos dos utilizadores criados	42
5.4	Conclusões	42
6	Avaliação da Aplicação Realizada	43
6.1	Portal de Serviços	43
6.2	<i>BackOffice</i>	44
6.3	Componente Analítica	54
6.4	Conclusões	54
7	Conclusões e Trabalho Futuro	56
7.1	Conclusão	56
7.2	Trabalho Futuro	57
	Bibliografia	59

ÍNDICE DE FIGURAS

1.1	Arquitetura da Solução	4
2.1	Modelo Conceptual Atual	10
2.2	Processo de Negócio Atual	10
2.3	Componente Analítica	11
2.4	Arquitetura Fabasoft	12
2.5	Esboço da Arquitetura Final Fabasoft	14
4.1	<i>Mockup</i> que demonstra como escolher o processo a criar	24
4.2	<i>Mockup</i> que demonstra o processo escolhido para criar	25
4.3	Criação de um objeto em Fabasoft	25
4.4	Componente de BI	26
4.5	Migração de Dados - Exportação	27
4.6	Migração de Dados - Importação	28
5.1	Exemplo da hierarquia seguida para cada tipo de Instalação Física	31
6.1	Todos os componentes desenvolvidos no âmbito do módulo de <i>BackOffice</i>	45
6.2	Classes com permissão de criação de objetos para utilizadores com o perfil das <i>Águas</i>	46
6.3	Classes disponibilizadas para a criação de objetos para utilizadores com o perfil das <i>Águas</i>	47
6.4	Classes com permissão de procura de objetos para utilizadores com o perfil das <i>Águas</i>	48
6.5	Classes disponibilizadas para a procura de objetos para utilizadores com o perfil das <i>Águas</i>	49
6.6	Formulário correspondente à Estatística das <i>Águas</i>	50
6.7	Exemplo da seleção de valor de uma propriedade do tipo enumerado	51
6.8	Exemplo de um agregado com múltiplos valores	51
6.9	Número de Inquéritos Únicos de <i>Minas</i> existentes na versão anterior	52

6.10	Número de Inquéritos Únicos de <i>Minas</i> existentes na nova versão após migração de dados	52
6.11	Dados referentes a um quadro de um Inquérito Único das Minas específico na versão anterior	53
6.12	Dados referentes a um quadro de um Inquérito Único das Minas específico na nova versão	53

ÍNDICE DE LISTAGENS

5.1	Criação da classe ClassIFAguaMineral	34
5.2	Criação de uma struct	34
5.3	Criação de uma propriedade do tipo composto	34
5.4	Criação de uma propriedade do tipo enumerado	35
5.5	Associação de propriedades	35
5.6	Criação de página de um formulário	36
5.7	Adição de uma página a um determinado formulário	38
5.8	Associação do formulário à classe	38
5.9	Adição de um separador estendendo um formulário	39
5.10	Desenvolvimento de um automatismo	40
5.11	Caminho para chegar a uma determinada propriedade	41
5.12	Associação do automatismo à propriedade	42

INTRODUÇÃO

Neste capítulo, a motivação e o contexto por trás desta dissertação são apresentados. O problema a ser abordado é identificado e um conjunto de objetivos é definido a fim de se desenvolver uma solução. Por fim, são descritas as contribuições esperadas para esta tese, bem como a estrutura do documento. De frisar que este projeto será desenvolvido num ambiente empresarial e em conformidade com muitos requisitos requeridos pelo cliente com o qual se está a trabalhar.

1.1 Relevância e Motivação

A Fabasoft disponibiliza diferentes ferramentas de *software* e a ferramenta utilizada neste projeto serve para a desmaterialização de processos e para a gestão documental. Na realidade, vai ser feita uma atualização de todo o ambiente de gestão documental fornecido por esta ferramenta, pois essa é a principal razão da realização deste projeto. O nosso cliente tem dados a tratar, referentes a diferentes módulos. É necessário tratar do módulo de estatística dos recursos geológicos e hidrogeológicos e, para isso, é preciso que a ferramenta utilizada seja de confiança, sustentável e otimizada. Todo o tratamento destes dados já acontecia com a versão anterior desta mesma ferramenta, mas tal como todo o tipo de *software*, este requer atualizações e interoperabilidade com outros *softwares*/aplicações que sejam utilizados no âmbito deste módulo de estatística. Como estes dados são de grande importância para o País, a fiabilidade dos mesmos tem de ser muito elevada. O objetivo é dar continuidade ao acesso que se pode ter a esses dados, otimizando o meio para chegar a esse fim.

Como já foi referido, já havia toda uma estrutura por detrás deste módulo. Um dos primeiros objetivos é entender essa mesma estrutura, perceber toda a interoperabilidade do projeto e definir o que devia ou não ser atualizado/refeito. É desafiante reestruturar toda a arquitetura que está por detrás de uma ferramenta utilizada e perceber quais são os pontos com pouco suporte e que necessitam de uma alteração. Para isso, é necessário definir qual irá ser a nova estrutura do projeto, definir novas hierarquias e alocação de certas propriedades. Feito isto, é necessário exportar todos os dados que já existiam

na versão anterior. Depois de tratados, importar na nova versão. Isto para não haver perda de dados de todos os anos anteriores, pois não queremos que o acesso a esses dados desapareça, o que torna este processo muito sensível. Uma vez assegurada toda a migração destes dados que constam na versão anterior para a nova versão, é aí que entra a ligação com a ferramenta de Power BI, ao exportar os dados de novo para que a equipa encarregue pela componente de BI (*Business Intelligence*) consiga também proceder à atualização dessa mesma componente. Muito sucitamente, é muito importante tratar da atualização desta ferramenta pois, como todos sabemos, o mundo da tecnologia está numa ascensão constante e a interoperabilidade entre todas as partes constituintes deste projeto têm de ser atualizadas consoante essa evolução.

1.2 Objetivos

O principal objetivo deste projeto, como se pode perceber pelo que foi escrito anteriormente, é o de atualizar/desenvolver toda a ferramenta de gestão documental e atualizar o modelo analítico que irá ser utilizado para as estatísticas dos recursos geológicos. Assim, podemos dividir os objetivos em três categorias:

- **Recolha de informação** - A criação de um novo portal de acesso, desenvolvido em tecnologia atual;
- **Validação e processamento:**
 - Atualizar a ferramenta de gestão documental para a sua versão mais recente, assegurando que o modelo conceptual utilizado anteriormente é mantido e é compatível com as mais recentes e variadas ferramentas de produção;
 - Integração com o novo portal de acesso;
 - Desenvolvimento consciente para que seja mantida a escalabilidade com processos futuros;
 - Migração dos dados para garantir a preservação digital.
- **Instrumentos estatísticos** - Atualização do modelo analítico que suporta os instrumentos estatísticos dos recursos geológicos.

Após a conclusão desta atualização, o objetivo passa por familiarizar o cliente com o que foi desenvolvido, dando-lhes a formação necessária de forma a que o uso da plataforma seja mais acessível. Para esse fim, pretendemos ensinar o cliente a entrar no novo portal, efetuando o *login*, a criar novos utiiladores, criar novos processos e entre outros passos que sejam importantes para o bom uso da ferramenta. Concluído este processo, o cliente estará apto a utilizar o produto, preenchendo os formulários e contribuir para que sejam construídas novas estatísticas dos recursos geológicos.

1.3 Contexto

O projeto irá ser desenvolvido numa linguagem disponibilizada pela **Fabasoft: *app.ducx***. A ferramenta utilizada será o **Fabasoft egov-Suite**, que é um produto **Fabasoft** e irá ser utilizado pelo cliente para interagir com todas interfaces de utilizador desenvolvidas, mais propriamente os formulários. Este produto é utilizado para a desmaterialização de processos e como uma ferramenta de gestão documental. A desmaterialização é o processo de transformar toda a informação que seja física em informação digital. Com a desmaterialização de processos, o contacto com os clientes é agora feito através de aplicações, portais ou até por e-mail. Com isto, queremos concluir que toda a troca de informação é feita digitalmente. Ao optar por este caminho inovador de lidar com processos, o uso de papel é muito reduzido, o que também aponta para o contributo de um ambiente mais sustentável, para além de que é muito mais fácil de organizar e aceder a simples processos, documentos ou que quer que se esteja à procura.

Com a disponibilização da versão mais recente deste *software* e estando ciente do estado do *software* que o cliente utiliza, foi-lhes então proposta uma atualização do sistema. Sabendo que as estatísticas dos recursos geológicos são um tópico muito importante e requerem uma certa eficiência e consistência pela ferramenta que está a ser utilizada, temos de atualizar o sistema para a versão mais recente de forma a manter a sua sustentabilidade e interoperabilidade entre todos os sistemas dos quais este *software* dependa. Por exemplo, podemos olhar para os casos do **Internet Explorer/Microsoft Edge**, que foram sendo atualizados à medida que os anos passaram e, com isso, algumas funcionalidades que o Fabasoft oferecia já não estão disponíveis ou a funcionar. Alguns dados podem ser perdidos e esse é um ponto muito preocupante.

Como o título desta dissertação indica, o objetivo do produto final é atualizar o módulo de estatística dos recursos geológicos, portanto toda a arquitetura irá ser construída cuidadosamente e de forma a que se consiga manter a componente principal como um todo, isto porque está tudo ligado e organizado.

1.4 Trabalho Esperado e Contribuições

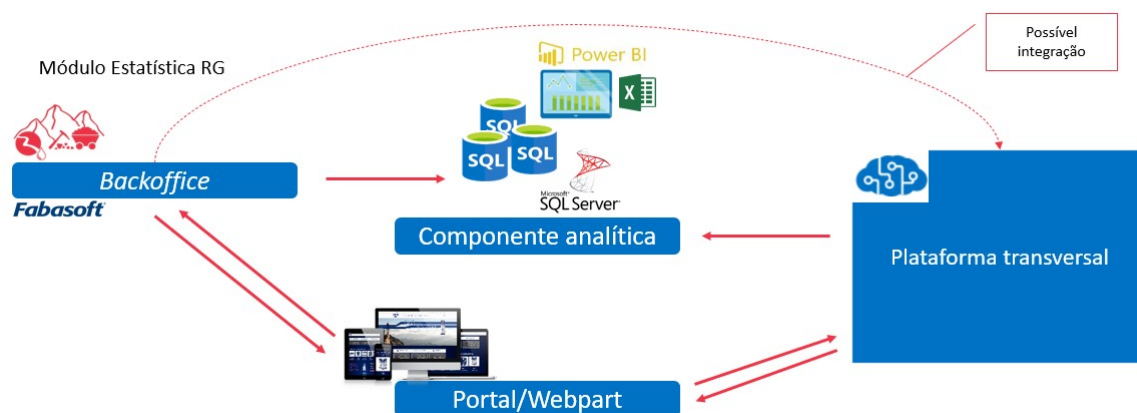


Figura 1.1: Arquitetura da Solução

Na fase de conclusão desta dissertação, o trabalho e contribuições esperadas são:

- A criação de uma componente nova e atualizada das estatísticas dos recursos geológicos. Tem de se definir uma estratégia de implementação, isto é: definir que objetivos são prioritários e devem estar desenvolvidos em primeiro lugar. Para isso, terá de se criar uma nova arquitetura, baseada na versão anterior. Serão criados vários componentes de *software* e será seguida uma determinada hierarquia. Uma vez terminada a definição de componentes que serão criadas, que classes irão conter, que propriedades irão pertencer às suas classes e estar definida toda a arquitetura que terá de ser respeitada ao longo deste projeto, passamos à implementação desses mesmos componentes. Como todas as ferramentas, a **Fabasoft egov-Suite** tem determinadas regras a seguir e toda a lógica em volta da implementação dos componentes utilizando esta ferramenta será explicada em detalhe mais à frente. Terminada a implementação, terão sempre de ser efetuados alguns testes para ver se tudo está a funcionar como é suposto. Isto tudo faz parte da componente de *BackOffice*;
- É necessário haver a integração dos componentes implementados com a outra grande componente deste projeto: a componente de *Business Intelligence*, que será tratada pela equipa de BI. Para a equipa de BI conseguir tratar dessa componente, é necessário haver toda uma migração dos dados já existentes. Para essa migração acontecer, são necessários vários passos:
 - Primeiro têm de ser exportados todos os dados que digam respeito às estatísticas dos recursos geológicos, dados esses que estão todos armazenados na versão desatualizada da ferramenta de gestão documental;
 - Uma vez seguida toda a lógica de exportação, é necessário tratar dos dados e importá-los na nova versão já implementada;

- Por fim, tem de se verificar a veracidade dos dados e a qualidade da importação, uma vez que é um processo complexo e requer muita atenção.
- É esperado o desenvolvimento de um portal novo e atualizado para que possa ser feita a integração com toda a atualização do sistema.
- Apesar de competir à equipa de BI todo o tratamento de dados que irão ser disponibilizados posteriormente para as estatísticas, é importante frisar este ponto como trabalho esperado deste projeto, pois toda a componente de *BackOffice* foi modelada tendo em vista o objetivo final de construir a componente de BI. A disponibilização dos dados para que possam construir *datasets* e possa haver interação com os mesmos - isto consoante o que se queira aceder - depende sempre da migração que terá que ser feita anteriormente.

Como resultado final deste projeto, são esperados diagramas interativos referentes à estatística dos recursos geológicos de forma que seja mais fácil aceder a determinados dados e conjuntos de dados disponibilizados que transmitam informação relevante.

1.5 Estrutura do Documento

This document is structured into 3 different chapters:

- **Introdução** - O primeiro capítulo é composto pela motivação para o desenvolvimento da dissertação, dando uma contextualização do projeto e que objetivos são definidos. No fim, estará o trabalho esperado.
- **Trabalho Relacionado e Enquadramento** - No segundo capítulo está uma pequena introdução aos recursos geológicos abordados no decorrer deste projeto. De seguida está uma demonstração do modelo conceptual da versão anterior, este que tem de ser mantido. São abordadas as questões e riscos da versão anterior e dado algum enquadramento da ferramenta que foi utilizada para o desenvolvimento desta dissertação.
- **Avaliação de Alternativas** - No terceiro capítulo estão enumeradas algumas alternativas à ferramenta que foi utilizada e a descrição detalhada de algumas das funcionalidades que são importantes ter em conta. No final, tem a razão por detrás da escolha da ferramenta que foi utilizada.
- **Arquitetura e Organização da Solução** - Neste capítulo está a arquitetura construída, tal como a razão por detrás de algumas escolhas que levaram a essa mesma arquitetura final.
- **Descrição da Implementação** - No quinto capítulo é onde está feita toda a descrição do que foi desenvolvido no decorrer desta dissertação. Que abordagem foi tida

nos diferentes componentes desenvolvidos, que particularidades estão presentes na ferramenta utilizada e explicação das mesmas.

- **Avaliação da Aplicação Realizada** - Neste penúltimo capítulo estão descritos que testes eram requeridos por cada componente do projeto, tal como a demonstração dos resultados dos mesmos que tenham ido ao encontro do que foi acordado com o cliente.
- **Conclusões e Trabalho Futuro** - O sétimo e último capítulo deste documento descreve as conclusões relativamente ao que foi desenvolvido no decorrer da dissertação e uma reflexão acerca do trabalho futuro, onde algumas possibilidades de desenvolvimento podem ser discutidas para projetos futuros.

TRABALHO RELACIONADO E ENQUADRAMENTO

2.1 Recursos Geológicos e a sua importância

Portugal tem um grande potencial no que toca à mineração e, conjugando-o com a dependência da União Europeia de algumas matérias-primas existentes no nosso país, surge uma oportunidade através da promoção da atividade de investigação e aproveitamento dos recursos geológicos, não só para valorizar os recursos endógenos nacionais como também os serviços, distribuição e comercialização de produtos, geração de empregos e impactos positivos na sociedade, induzidos local e territorialmente. Podemos concluir que é muito importante ter conhecimento dos recursos geológicos que Portugal possui, para que se possam fazer estatísticas de forma a aproveitar o que estes realmente podem oferecer.

2.1.1 Pedreiras

Uma pedreira é um conjunto formado por qualquer massa mineral que seja um objeto de licenciamento para as instalações necessárias à sua lavra, área de extração e zonas de defesa, por jazidas de massas minerais extraídas, estéreis e retiradas, bem como para os seus anexos.

As pedreiras são divididas em 4 classes, que estão ordenadas por ordem decrescente pelos impactos que podem causar.

Assim sendo, a classificação de cada pedreira segue os seguintes critérios:

- **Classe 1:** São aquelas com uma área igual ou superior a 25 hectares.
- **Classe 2:** as pedreiras subterrâneas ou mistas e as que sendo a céu aberto possuam:
 - Área inferior a 25 hectares;
 - Profundidade de escavações superior a 10 metros;
 - Produção superior a 150 000 t/ano;

- Número de trabalhadores superior a 15;
- Utilização de mais de 2 000 kg de explosivos, por ano, no método de desmonte.
- **Classe 3:** as pedreiras a céu aberto que não excedam nenhum dos seguintes limites:
 - Área - 5 hectares;
 - Profundidade de escavações - 10 metros;
 - Produção - 150 000 t/ano;
 - Número de trabalhadores - 15;
 - Utilização de explosivos até 2 000 kg.
- **Classe 4:** As pedreiras de calçada e de laje se enquadradas na definição e limites das Pedreiras de classe 3.[6]

As pedreiras podem, por vezes, ser muito desvalorizadas dado que nem todos conhecem as suas potencialidades nem o que estas têm a oferecer. A verdade é que falamos de locais essenciais à extração de minerais e Portugal é um dos países mais ricos nos mesmos. São essenciais em duas frentes, nomeadamente a extração de materiais de construção e até mesmo minerais decorativos. Ainda que não seja comum referirmo-nos às pedreiras como minas, a verdade é que, de certa forma, o são. É verdade que não têm a mesma profundidade, nem tão pouco se estendem no subsolo, mas isso não remove o "estatuto" de mina a uma pedreira. Falamos, por isso, de minas a céu aberto, facilmente identificáveis pelo seu tamanho. Claro que essa aparência não é regra, até porque existem exemplos de pedreiras que se encontram entre montanhas e que, por isso, não são tão visíveis quanto isso - é o caso da pedreira da Serra dos Candeeiros, em Rio Maior, por exemplo. Na realidade, Portugal tem um potencial mineiro e geológico ímpar na União Europeia, que lhe permite levar a cabo diversas investigações e enriquecer, assim, a atividade de extração de minerais e pedras de locais como jazidas e pedreiras. A procura tem sido crescente, o que possibilita, não só, a movimentação da economia nesse setor, como também o aumento de postos de trabalho. As vendas a fornecedores que procurem certo tipo de minerais e pedras são igualmente beneficiadas desta prática que continua tão ativa em Portugal.[7]

2.1.2 Minas

A mina é uma escavação cujo o objetivo é obter recursos úteis para o Homem dessa mesma escavação; Recursos como minerais, combustível, água etc. Os túneis subdivisórios feitos (ou naturais) por mineiros dentro de minas subterrâneas são conhecidos como galerias.

Pese embora Portugal só tenha hoje três minas metálicas em produção (Neves-Corvo, Aljustrel e Panasqueira), importa registar que em 2011 estas foram responsáveis por 57% (460 M€) do valor das exportações portuguesas em recursos minerais, que em termos globais atingiram 807 M€. Os minerais para a construção, onde se incluem as rochas

ornamentais, foram responsáveis por 37,5% (303 M€), enquanto os minerais industriais só representaram 5,5% (42 M€). Uma referência especial ao grande potencial de crescimento das exportações em rochas ornamentais e minerais industriais, desde que implementadas políticas de apoio ao setor que contemplem as áreas dos transportes ferroviários, dos portos e da comercialização agregada. Tratando-se de recursos endógenos, com uma incorporação de importações diminuta, caso estes projetos mineiros venham a tornar-se uma realidade, as perspectivas económicas para Portugal podem ser consideradas otimistas, não só para as futuras exportações, mas também como motor do desenvolvimento regional. A exemplo do que acontece com as atuais minas em produção, quase todos os projetos em fase experimental situam-se no interior do País, em zonas deprimidas económica e socialmente, podendo por isso contribuir para que essas regiões fixem populações e se dinamizem economicamente.[1]

2.1.3 Recursos Hidrogeológicos

O objecto da indústria de engarrafamento de águas minerais naturais e de águas de nascente é a gestão de recursos hidrogeológicos. As águas minerais naturais pertencem ao domínio público do Estado, são aproveitadas em regime de concessão e, caso tenham comprovadamente determinadas propriedades terapêuticas, podem ser utilizadas em balneários termais e prescritas por médicos para tratamento de certo tipo de doenças. Algumas destas águas podem também ser utilizadas no engarrafamento. As águas de nascente, sendo igualmente recursos geológicos sujeitos a exigentes normas de licenciamento e protecção, pertencem ao domínio privado e são aproveitadas em regime de licenciamento. [8]

2.2 Modelo Conceptual

O modelo conceptual atual pode ser entendido pela figura seguinte (figura 2.1).

Cada etiqueta pertencente ao círculo é uma classe/processo de negócio. Os processos de negócio tão todos criados a partir da sua base: Instalação física. A solução atual tem informação desmaterializada com mais de 15 anos:

- Informação acerca de aproximadamente 2700 pedreiras (ativas, inativas ou até suspensas);
- Informação sobre aproximadamente 280 minas;
- Mais de 10000 inquéritos que dizem respeito às pedreiras;
- Mais de 10000 inquéritos que dizem respeito às minas.



Figura 2.1: Modelo Conceptual Atual

2.2.1 Processo de Negócio

Este modelo conceptual será também utilizado para a nova solução, onde as Instalações Físicas continuam a ser a peça chave e central dos processos de negócio.

Para o fornecimento de estatísticas, os dados são retirados pelos exploradores/proprietários das próprias minas e pedreiras e têm um processo empresarial associado à apresentação desses dados, o que está representado na seguinte figura:

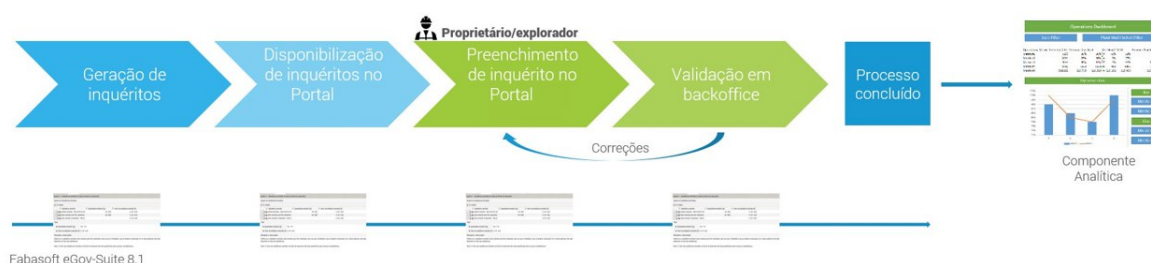


Figura 2.2: Processo de Negócio Atual

O processo de negócio pode ser percebido ao explicar cada um dos seus passos, que são aqueles mencionados na figura 2.2:

- **Geração de inquéritos** - Associação automática à sua instalação/entidade física. Por ano é gerado um **Inquérito Único** por pedreira/mina baseado também em dados anteriores. Em 2020 foram gerados mais de 1300 inquéritos de pedreiras e mais de 200 inquéritos de minas.
- **Disponibilização de inquéritos no portal** - Cada proprietário/explorador tem um portal específico, a possibilidade de pesquisar o histórico de inquéritos e a possibilidade de descarregar documentos de inquéritos entregues.

- **Preenchimento de inquérito no portal** - Acesso por autenticação ao portal. Conclusão do inquérito que já está preenchido pelo cliente com base em valores dos últimos anos.
- **Validação em *back office*** - Validação do inquérito enviado. Possibilidade de devolução para preenchimento, caso seja necessária uma correção. Após a aprovação, os documentos sobre o preenchimento dos formulários e sua disponibilidade no portal são gerados.

2.2.2 Componente Analítica

Após todo este processo ser concluído, chega a componente analítica composta pelos três seguintes passos:



Figura 2.3: Componente Analítica

- **Exportação de dados** - Após validação, os valores aprovados são exportados para uma base de dados externa à aplicação.
- **Processamento de dados** - Os dados exportados são processados de acordo com regras definidas pelos serviços para escrever relatórios.
- **Disponibilização no portal** - É dado acesso aos dados processados através do portal designado para o cliente, juntamente com outras áreas da organização do cliente.

Todos esses dados foram armazenados e tratados com o SQL Server 2008, na versão anterior. Após esses processos serem concluídos, as estatísticas estão prontas para serem observadas e analisadas para fazer algumas conclusões ou estudos que o cliente precise.

2.3 Questões e riscos a abordar

A solução atual é desenvolvida na **Fabasoft eGov-Suite**, versão 8.1. Essa versão é muito antiga e, conseqüentemente, pode causar muitos problemas e é arriscado confiar nessa mesma versão. Então, há uma lista de problemas pelos quais nenhum sistema deve passar:

- **Versões obsoletas** - Como é tão antiga, o suporte que se podia dar já não é possível, pois agora eles têm versões mais recentes para lidar.
- **Falta de compatibilidade** - A falta de compatibilidade com as ferramentas de produtividade e *browsers* traz alguns problemas:

- Problemas ao usar o portal, por exemplo, ao preencher os formulários;
 - Obriga a instalação de *plugins* nas áreas de trabalho dos utilizadores do portal e a utilização de *browsers* antigos para que ainda seja possível usar o software.
- **Instabilidade** - Problemas foram identificados no sistema que suporta as bases de dados, o que causa instabilidade ao sistema;
 - **Indisponibilidade do serviço de aplicação** - Dificuldades ao aceder a relatórios estatísticos devido à indisponibilidade do serviço;
 - **Dados desatualizados** - Causado devido a problemas com o processamento de dados após uma extração;
 - **Perda de dados** - Dados com mais de 10 anos de informação podem ser perdidos e não são recuperáveis.

2.4 Enquadramento

2.4.1 Arquitetura da Ferramenta

Todos os sistemas têm uma arquitetura montada, em que a solução é baseada e segue os princípios dessa mesma arquitetura. A ferramenta utilizada para desenvolver todo o sistema de gestão documental deste módulo de estatísticas dos recursos geológicos não foge à regra.

Abaixo está a arquitetura que é disponibilizada a todos aqueles que pretendam utilizar a ferramenta:

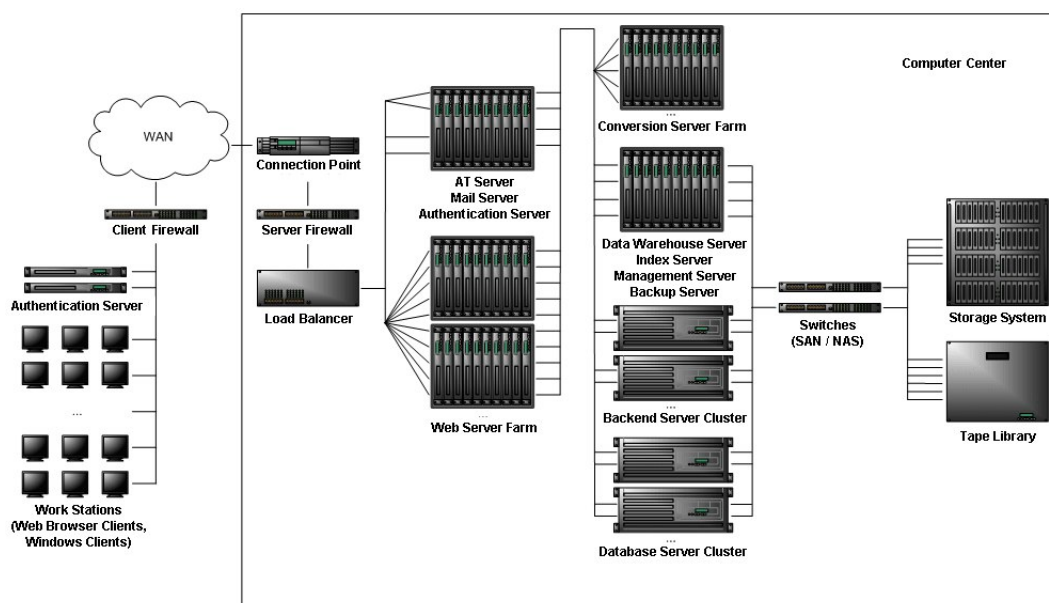


Figura 2.4: Arquitetura Fabasoft

Esta arquitetura, para este projeto em específico, tem adaptações. Nem todos os *hardwares* que constam na arquitetura são utilizados e outros são utilizados com outra abordagem que faça mais sentido para o objetivo final deste projeto. Como a própria figura indica, há um fluxo de como é feita a interação entre o utilizador e a ferramenta e todas as funcionalidades que esta oferece. Através das *Work Stations*, que representam os postos de trabalho de cada utilizador, acedemos à ferramenta *Fabasoft eGov-Suite*, onde o utilizador irá interagir com a interface desenvolvida. Não está a ser utilizado nenhum *Authentication Center*. Em vez disso, o processo de autenticação passa pelo uso do serviço *Azure Active Directory* (*Azure AD*). Este é um serviço de gestão de identidade e acesso fornecido pela **Microsoft** como parte do seu conjunto de serviços em *Cloud*. Este serviço baseado em *Cloud* permite às organizações gerir identidades de utilizadores e acessos a recursos em diferentes aplicações, incluindo aplicações da Microsoft, aplicações de terceiros e aplicações personalizadas. A *Azure AD* é uma solução de autenticação e autorização centralizada que ajuda as organizações a controlar o acesso dos seus utilizadores e a proteger informações confidenciais. Algumas das funcionalidades disponibilizadas incluem a gestão de contas de utilizadores, autenticação multifator, gestão de grupos e permissões de acesso a aplicações e recursos. Esta solução permite também a integração com outras soluções de gestão de identidade e acesso, como o *Active Directory on-premises* e outras soluções de gestão de identidade de terceiros. Resumidamente, a *Azure AD* é uma solução em *Cloud* que ajuda as organizações a manter a segurança dos seus recursos digitais e a proteger informações confidenciais através de uma gestão centralizada de identidades e acessos. Para aceder ao ambiente de gestão documental, é necessária autenticação, tal como para aceder às *virtual machines*. Voltando à arquitetura, o *Load Balancer* também não é utilizado. Este serve para equilibrar a carga de trabalho entre servidores em ambientes de grande procura, ou seja, que seja utilizado por muitos utilizadores ao mesmo tempo. Assim, permite alocar recursos de forma uniforme e aumenta a disponibilidade dos serviços. Uma vez que, ao momento, apenas 10 utilizadores interagem com a ferramenta de gestão documental, não se justifica a utilização do *Load Balancer*. Apenas um servidor é utilizado para executar o *front-end* e esse servidor está a correr numa *virtual machine*. Passando agora para a parte de *back-end*, seguiu-se a mesma lógica que no *front-end*. Existe apenas um servidor para correr o *back-end* e esse servidor está a correr numa *virtual machine*. Nesse mesmo servidor está a correr o servidor de conversão. Este servidor de conversão serve para converter todos os tipos de ficheiros em *PDF*, desde documentos *Word*, documentos *Excel*, imagens, e.t.c. Todos esses ficheiros, que provêm de determinados processos, são guardados em disco. Quanto à Base de Dados que suporta a solução, está a ser usado o *Azure SQL Database* e não um servidor físico (*Database Server Cluster*) como está representado na figura 2.4. Contextualizando, o *Azure SQL Database* é um serviço *Cloud*, também oferecido pela **Microsoft** como parte de serviços *Cloud Azure*. Este serviço, que foi escolhido pelo cliente, ajuda a economizar quaisquer custos uma vez que deixa de haver necessidade de se investir em *hardware* e *software* para ter um servidor físico onde a Base de Dados esteja a correr. O *Azure SQL Database* é uma solução totalmente gerida, que oferece funcionalidades como cópia

de segurança e recuperação em caso de perda de dados, monitorização e optimização de desempenho, escalabilidade automática e atualizações de segurança constantes. Permite que as organizações criem bases de dados relacionais baseadas no Microsoft SQL Server que podem ser acedidas e geridas através da Internet. [4]

Data Warehouse Server, Index Server, Management Server e Backup Server são tudo opções que dependem da decisão da equipa de *Business Intelligence* e não está diretamente relacionado com a arquitetura do que foi desenvolvido por mim. Também não constam na solução os *Switches, Storage System e Tape Library*. Assim, a arquitetura final, tendo em conta todos os ajustes descritos e a não utilização de alguns componentes, é a seguinte:

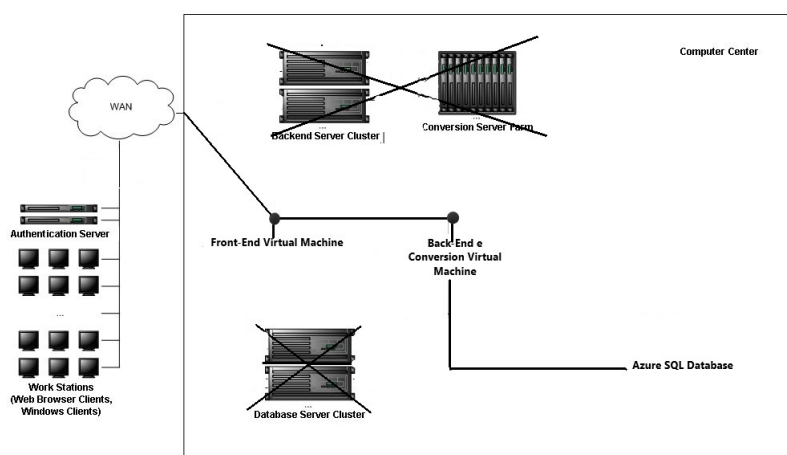


Figura 2.5: Esboço da Arquitetura Final Fabasoft

2.4.2 Tecnologia

Os componentes de *software* da nova solução foram desenvolvidos com o **Fabasoft *app.ducx***. O **Fabasoft *app.ducx*** é uma plataforma de desenvolvimento, orientada a casos de uso para aplicações empresariais compostas. É baseada em **Fabasoft Folio**, **Fabasoft Folio Cloud** e **Fabasoft eGov-Suite**. Com *domain specific languages* (DSL), o **Fabasoft *app.ducx*** é especialmente projetado para o desenvolvimento rápido de aplicações empresariais centradas em documentos e aplicações. O **Fabasoft *app.ducx*** oferece uma base poderosa e suporte de ferramentas. É baseado em conceitos de diferentes linguagens de programação, que são também familiares, como **JavaScript**, **C#** e **Java**.

Tudo pode ser integrado com o **Eclipse**, que é o ambiente de desenvolvimento integrado (IDE) usado para implementar todo o código [10].

Atualmente, o **Fabasoft *app.ducx*** contém oito DSLs distintas:

- **Object Model Language *app.ducx* (.*ducx-om*)** - O objetivo da *Object Model Language app.ducx* é definir o modelo de objetos persistente para um componente de software. Usando-a, é possível definir os elementos básicos que compõem o modelo de objetos:
 - classes de objetos;

- propriedades e campos;
 - tipos de enumeração;
 - tipos compostos;
 - extensões de classes e tipos de objeto existentes. [2]
- **Resource Language *app.ducx* (.*ducx-rs*)** - A *Resource Language app.ducx* da Fabasoft *app.ducx* permite definir recursos, como objetos de *string*, mensagens de erro e símbolos.
- **Use Case Language *app.ducx* (.*ducx-us*)** - O objetivo da *Use Case Language app.ducx* é definir e implementar casos de uso. Com a *Use Case Language app.ducx* é possível definir novos casos de uso e fornecer implementações de método para esses casos de uso. A linguagem de expressão ou aplicações virtuais podem ser usadas no *app.ducx* para implementar casos de uso. Alternativamente, o **Java** pode ser utilizado para a sua implementação.
- **Business Process Language *app.ducx* (.*ducx-bp*)** - A *Business Process Language app.ducx* permite definir o modelo de processo para o componente de software. O modelo do processo de negócio é composto por um ou mais processos que definem uma lista ordenada de atividades sequenciais ou paralelas e elementos que servem de controlo ao processo, como condições ou ciclos. Cada definição de cada atividade consiste numa lista de etapas/passos que devem ser realizados por um utilizador responsável por concluir uma determinada atividade.
- **User Interface Language *app.ducx* (.*ducx-ui*)** - A *User Interface Language* da Fabasoft *app.ducx* é usada para definir formulários, páginas de formulário, itens do menu e outros elementos de interface do utilizador para as classes de objeto do projeto.
- **Organizational Structure Language *app.ducx* (.*ducx-os*)** - Os elementos abstratos da estrutura organizacional necessários para a solução final são definidos com o uso da *Organizational Structure Language app.ducx*. O Fabasoft permite representar uma estrutura organizacional usando elementos estruturais abstratos e concretos. Esses elementos da estrutura organizacional podem ser usados para atribuir direitos de acesso e para definir participantes num fluxo de um determinado processo.
- **Customization Language *app.ducx* (.*ducx-cu*)** - O objetivo da *Customization Language app.ducx* é definir, personalizar e adaptar o componente de *software* a requisitos específicos do projeto ou da solução. Um bloco de personalização consiste em declarações de importação e pontos de personalização.
- **Expression Language *app.ducx* (.*ducx-xp*)** - Esta é a DSL que permite aceder ao modelo de objecto do Fabasoft e invocar casos de uso. As *Expression Language app.ducx* podem ser incorporadas em blocos de expressões e noutros elementos de linguagem das DSL do **Fabasoft *app.ducx*** [2].

2.5 Conclusões

Neste capítulo, mostra-se sobre quais recursos geológicos se fala neste projeto, a sua importância e o seu impacto na população. Tem, também, o objetivo de ajudar a compreender os problemas da solução atual, pois a solução atual está muito desatualizada. A falta de documentação sobre a versão da solução atual torna impossível aprofundar neste tópico. É dado um enquadramento acerca da ferramenta que está a ser utilizada, explicando a sua arquitetura e o funcionamento da tecnologia utilizada.

AVALIAÇÃO DE ALTERNATIVAS

Neste capítulo irei falar de várias ferramentas de gestão documental e enumerar algumas das suas funcionalidades. Infelizmente, todo este tipo de ferramentas só disponibilizam informação mais pormenorizada acerca das mesmas com a compra do produto, o que não me permite aprofundar o que cada uma das ferramentas tem para oferecer como gostaria. Irei também comparar com a ferramenta que foi escolhida para o desenvolvimento deste projeto e explicar o porquê da sua escolha.

3.1 Alternativas de Ferramentas de Gestão Documental

3.1.1 EdocLink

Tal como foi dito na introdução deste capítulo, a *Edoclink* é outra ferramenta de gestão documental. Algumas das funcionalidades disponibilizadas pelo *Edoclink* são as seguintes:

- Armazenamento em *Cloud*: armazena todos os documentos em *cloud*, o que significa que podemos acedê-los de qualquer lugar, basta estar conectado à internet.
- Organização de documentos: permite organizar documentos em pastas e subpastas, para que se possa encontrar facilmente o que se está a procurar.
- Partilha de documentos: pode-se partilhar documentos com outras pessoas através da *Edoclink*, permitindo que visualizem ou editem os mesmos.
- Controlo de versão: A *Edoclink* mantém um registo de todas as versões anteriores dos documentos, permitindo que se volte para uma versão anterior, se necessário.
- Segurança: utiliza medidas de segurança para garantir que os documentos estejam protegidos contra acesso não autorizado.
- Pesquisa de documentos: permite a pesquisa de documentos com base em palavras-chave, tornando mais fácil encontrar o que se precisa.

Com a *Edoclink*, é possível criar modelos e configurar fluxos de trabalho e propriedades usando uma interface gráfica de utilizador (GUI) baseada em *drag and drop*, o que significa

que se pode arrastar e soltar elementos para construir uma solução sem a necessidade de escrever código. No entanto, se se desejar criar uma solução personalizada e mais complexa, é possível usar a API do *Edoclink* para criar aplicações próprias e integrar com outros sistemas. A API do *Edoclink* está disponível em várias linguagens de programação, incluindo *Java*, *C#*, *Python* e *Ruby*. Ao usar a API do *Edoclink*, podem ser criadas aplicações personalizadas de gestão documental que se integram com os sistemas existentes. É possível criar formulários personalizados, automatizar fluxos de trabalho e definir as propriedades dos documentos com código. Pode ser útil se for precisa uma solução mais personalizada e flexível que não pode ser construída utilizando apenas a GUI do *Edoclink*. [3]

3.1.2 EdocLink vs Fabasoft

Tanto a *Edoclink* quanto a *Fabasoft* são soluções de gestão documental. Ajudam, por exemplo, as empresas a digitalizar os seus processos e melhorar a colaboração e a produtividade. Algumas diferenças entre a *Edoclink* e a *Fabasoft* incluem:

- Funcionalidades: Ambas as soluções oferecem recursos semelhantes, como gestão de documentos, fluxo de trabalho, colaboração e armazenamento em *cloud*. No entanto, a *Fabasoft* tem uma gama mais ampla de funcionalidades avançadas, como recursos de assinatura eletrónica, gestão de contratos e conformidade com regulamentos específicos do setor.
- Escala: A *Edoclink* é uma solução que pode ser escalada facilmente para atender às necessidades de empresas de pequeno e médio porte, enquanto a *Fabasoft* é frequentemente usada por grandes empresas e organizações governamentais.
- Preço: A *Edoclink* tem preços mais acessíveis e oferece planos que se adaptam a diferentes orçamentos e necessidades de empresas de pequeno e médio porte. Já a *Fabasoft* pode ser mais cara e é direcionada principalmente para empresas com orçamentos maiores.
- Interface: A *Edoclink* tem uma interface de utilizador mais moderna e amigável, enquanto que a interface da *Fabasoft* pode ser considerada mais complexa e técnica.
- Personalização da plataforma: O método de personalização da plataforma para a *Edoclink* é diferente do da *Fabasoft*. Enquanto a *Fabasoft* usa a linguagem de programação *App.Ducx* para personalizar a sua plataforma, a *Edoclink* suporta várias linguagens de programação para personalização da mesma, como *Java*, *C#*, *Python*, *Ruby* e *PHP*. Além disso, a *Edoclink* fornece uma API para interagir com a plataforma e personalizar as suas funcionalidades, enquanto que a *Fabasoft* oferece um conjunto de ferramentas e plugins para a integração do Eclipse e a personalização da plataforma. Portanto, embora ambas as plataformas permitam personalização e extensibilidade,

o método de personalização pode ser diferente, dependendo das ferramentas e linguagens de programação suportadas por cada plataforma.

Em última análise, a escolha entre a *Edoclink* e a *Fabasoft* dependerá das necessidades específicas da empresa e do orçamento disponível para investir numa solução de gestão documental.

3.1.3 SmartDocs

O *SmartDocs*, criado pela *Fujitsu*, é uma solução de gestão documental concebida para ajudar as organizações a gerir eficazmente os documentos e os fluxos de trabalho associados. Embora a funcionalidade específica possa variar consoante a versão e a configuração do software, eis algumas das características comuns do *SmartDocs*:

- Armazenamento e organização de documentos: O *SmartDocs* permite armazenar documentos digitalmente numa localização centralizada. Fornece funcionalidades para organizar e categorizar os documentos em pastas para facilitar a pesquisa e a recuperação posterior.
- Controlo de versões: O *SmartDocs* permite acompanhar e controlar as diferentes versões de um documento. Isto ajuda a garantir que as alterações são registadas e que as versões anteriores podem ser acedidas, se necessário.
- Colaboração e fluxos de trabalho: A solução facilita a colaboração entre utilizadores, permitindo-lhes partilhar documentos, comentar e fornecer feedback. Além disso, o *SmartDocs* suporta a configuração de fluxos de trabalho personalizados, permitindo que os documentos passem por aprovações e revisões antes de serem finalizados.
- Pesquisa avançada: O *SmartDocs* fornece capacidades de pesquisa avançada para ajudar os utilizadores a encontrar documentos com base em palavras-chave, metadados ou outros critérios específicos. Isto facilita a localização de informações relevantes de forma rápida e precisa.
- Integração com outros sistemas: O *SmartDocs* pode ser integrado noutros sistemas de gestão, tais como sistemas de gestão de conteúdos empresariais ou sistemas de gestão de relações com clientes. Isto permite que os documentos sejam acedidos e geridos sem problemas a partir destes sistemas.
- Segurança e controlo de acesso: O *SmartDocs* oferece funcionalidades de segurança avançadas para proteger documentos e dados sensíveis. Permite definir permissões de acesso com base nas funções e níveis dos utilizadores, garantindo que apenas as pessoas autorizadas podem ver, editar ou modificar documentos.

Estas são algumas das funcionalidades comuns encontradas no *SmartDocs*. É importante notar que funcionalidades específicas podem variar consoante a versão do software e as definições personalizadas, como referido anteriormente. [9]

3.1.4 SmartDocs vs Fabasoft

A *Fabasoft* e o *SmartDocs* são soluções de gestão documental diferentes, desenvolvidas por empresas diferentes. Segue-se uma comparação geral entre elas:

- Foco e âmbito: A *Fabasoft* é conhecida por oferecer uma vasta gama de soluções de gestão de documentos e processos empresariais, abrangendo tudo, desde a gestão de documentos empresariais à gestão de fluxos de trabalho complexos. Por outro lado, o *SmartDocs* centra-se mais especificamente na gestão de documentos e nos fluxos de trabalho associados.
- Funcionalidades: Ambas as soluções podem oferecer características como o armazenamento centralizado de documentos, o controlo de versões, os fluxos de trabalho personalizados, a colaboração e a pesquisa avançada. No entanto, a funcionalidade específica e a profundidade das características podem variar entre soluções. É importante ter em conta as necessidades específicas do projeto que vai ser desenvolvido e comparar a funcionalidade oferecida por cada solução para determinar qual delas deverá ser utilizada.
- Âmbito da aplicação: A *Fabasoft* é conhecida pela sua capacidade de abranger diferentes indústrias e oferecer soluções para várias áreas de negócio, incluindo sectores altamente regulamentados, como o sector público. Por outro lado, o *SmartDocs* tem um âmbito de aplicação mais alargado e pode ser utilizado em diferentes contextos e sectores de negócio, desde empresas corporativas a instituições governamentais.
- Integração e ecossistema: A *Fabasoft* oferece um ecossistema abrangente de produtos e soluções, incluindo ferramentas de colaboração, gestão de processos e gestão de conteúdos. O *SmartDocs* também pode oferecer integração com outros sistemas, como sistemas de gestão de conteúdos empresariais ou sistemas de gestão de relações com clientes, embora o âmbito da integração possa depender das configurações específicas da solução.

Estas são algumas das diferenças gerais entre a *Fabasoft* e o *SmartDocs* e, como podemos ver, são ferramentas que oferecem o mesmo tipo de funcionalidades, complementando muitos dos pontos essenciais que têm de ser tidos em conta para a escolha da ferramenta que se irá usar.

3.1.5 GFIDocs

A *GFIDocs* é uma ferramenta que serve para gerir documentos eletrónicos e fluxos de trabalho de negócios. Funcionalidades que devem ser referidas são:

- Gestão de documentos: O *GFIDocs* permite o armazenamento, organização e gestão de documentos eletrónicos de forma centralizada e segura, com controlo de versão e histórico de revisão. Também permite a criação de metadados personalizados e a

classificação dos documentos por tipo, categoria e outras informações que possam ser relevantes.

- Fluxo de trabalho: O *GFIDocs* permite a criação de fluxos de trabalho automatizados para lidar com processos de negócios, como aprovações, revisões e notificações. Podem ser definidas regras de negócios para automatizar certas ações, como enviar um documento para revisão ou aprovação para um utilizador específico ou grupo de utilizadores.
- Colaboração: O *GFIDocs* permite a colaboração entre utilizadores, permitindo a partilha de documentos e o trabalho em equipa em projetos e processos de negócios. Também permite a colaboração com utilizadores externos por meio de convites por e-mail para acesso a documentos compartilhados.
- Segurança: O *GFIDocs* fornece recursos de segurança avançados, incluindo controlo de acesso baseado na função do utilizador, auditoria de atividades do utilizador e criptografia de documentos.
- Integração: O *GFIDocs* pode ser integrado com outras ferramentas de negócio, como ERP (*Enterprise Resource Planning*), CRM (*Customer Relationship Management*) e sistemas de gestão de projetos, para oferecer uma solução completa de gestão de documentos e fluxo de trabalho.

Sumariamente, o *GFIDocs* é uma ferramenta completa de gestão de documentos eletrónicos e fluxo de trabalho, com recursos avançados de colaboração, segurança e integração. É uma ótima opção para empresas que desejam centralizar e automatizar processos de gestão de documentos e fluxos de trabalho. [5]

3.1.6 GFIDocs vs Fabasoft

Tanto o *GFIDocs* quanto o *Fabasoft* são sistemas de gestão documental e fluxo de trabalho, mas têm diferentes funcionalidades e abordagens de desenvolvimento. O *GFIDocs* é uma plataforma de gestão de documentos altamente personalizável e flexível, que permite a criação de soluções de gestão documental personalizadas e adaptadas às necessidades de cada organização. O *GFIDocs* é construído com base em tecnologia *Java* e pode ser desenvolvido em ambientes *Windows* ou *Linux*. O *Fabasoft*, por sua vez, é uma plataforma de gestão documental que se concentra na automação de processos de negócio. Oferece recursos avançados de gestão de fluxo de trabalho, como fluxos de trabalho de aprovação, rastreamento de *status* e relatórios de desempenho. O *Fabasoft* é construído com base na linguagem de programação *App.ducx* e é desenvolvido, pela maior parte dos utilizadores, com o Eclipse. Em termos de personalização da plataforma, o *Fabasoft* e o *GFIDocs* têm abordagens diferentes. Enquanto o *Fabasoft* oferece uma IDE (*Integrated Development Environment*) baseada em Eclipse para o desenvolvimento de soluções personalizadas, o *GFIDocs* utiliza uma abordagem de personalização baseada em modelos de solução.

Isso significa que, no *GFIDocs*, a personalização é feita por meio da configuração de modelos de solução que definem os tipos de documentos, campos de metadados, fluxos de trabalho e permissões de acesso. Embora essa abordagem possa ser menos flexível do que a personalização através da programação, pode ser mais acessível de se usar para utilizadores que não têm tanta experiência em programação.

3.1.7 Fabasoft: Porquê?

Para além de tudo o que foi descrito na comparação entre as ferramentas que escolhi para falar acerca, era crucial encontrar a ferramenta mais eficiente para a gestão documental. Neste sentido, a Fabasoft oferece uma solução completa e fiável, com várias funcionalidades que simplificam o processo de gestão de documentos. Desde o controlo de versões, à definição de perfis de utilizador e gestão de fluxos de trabalho, a Fabasoft torna todo o processo de gestão documental mais simples e eficiente. Além disso, esta ferramenta é altamente personalizável, permitindo adaptar-se às necessidades específicas de cada organização. Com a Fabasoft, é possível gerir documentos de forma centralizada, acedê-los de forma segura a partir de qualquer dispositivo e colaborar em tempo real com colegas ou parceiros. Tudo isto faz da Fabasoft uma opção a considerar para quem procura uma ferramenta de gestão documental fiável e completa. Finalmente, por se tratar de uma ferramenta antiga conhecida pela **Unipartner** e já haver vários projetos desenvolvidos utilizando esta mesma ferramenta, não fazia sentido mudar. Ainda faz sentido frisar que a versão anterior foi desenvolvida utilizando uma versão mais antiga da Fabasoft, logo esta solução seria muito mais fácil de desenvolver utilizando a Fabasoft como ferramenta de gestão documental.

ARQUITETURA E ORGANIZAÇÃO DA SOLUÇÃO

Neste capítulo iremos ter uma visão geral da arquitetura do projeto, quais os seus grandes componentes e por quais processo de desenvolvimento tive passar.

4.1 Arquitetura da Solução

A solução proposta para este projeto foi implementar diferentes componentes de software para otimizar a solução e torná-la mais confiável do que a solução anterior.

Neste projeto, teremos dois módulos/componentes principais:

- O primeiro agrega dois outros módulos diferentes. Um deles será a *Estatística dos Recursos Geológicos*, em que os recursos geológicos são *Minas*, *Pedreiras* e *Águas*, dada a sua semelhança em termos do modelo conceptual. Temos os seguintes processos estatísticos incluídos neste módulo:
 - *Inquérito Único*, um para as *Minas* e outro para as *Pedreiras*;
 - *Estatística* para os recursos geológicos: *Águas*, *Minas* e *Pedreiras*;
 - *Relatório técnico das pedreiras*;
 - *Relatório de acidentes nas pedreiras*.

Uma classe foi criada para cada um destes processos estatísticos e um formulário associado a ele. Todas as propriedades associadas a cada um desses formulários/-classes também foram criadas, com base na solução atual que temos. De notar que todas essas propriedades foram alocadas no componente de software que faz mais sentido, na medida em que possa ser reutilizado e gastar menos recursos. Para isso, foi necessário pesquisar todas as propriedades de todos os formulários e classes e entender quais propriedades eram utilizadas em dois ou mais formulários.

Falando sobre o Power BI, há um componente de Business Intelligence que faz parte da solução proposta. Esta proposta está alinhada com a estratégia definida pelo cliente, ou seja, será feita de forma a não invalidar a sua escalabilidade a partir de outros projetos que o cliente tenha em andamento.

Para este módulo, existem alguns *mockups* de como devem ser criados os formulários estatísticos:

The mockup shows a 'Criar' (Create) form with a dark blue header bar containing the title 'Criar' and a close button. Below the header, there is a search bar with the placeholder text 'O que deseja criar?' and a search button. The search bar contains the text 'Estatística'. Below the search bar, there are two sections: 'Favoritos (0)' and 'Classes de objecto (3)'. The 'Classes de objecto (3)' section lists three items: 'Boletim de estatística de mina', 'Estatística de pedreira', and 'Estatística de águas'. At the bottom right of the form, there is a 'Cancelar' (Cancel) button.

Figura 4.1: *Mockup* que demonstra como escolher o processo a criar

Só temos de escrever o nome do processo que queremos criar e, como mostrado na figura 4.1, alguns resultados com essa pesquisa são listados. Depois disso, temos que escolher uma das opções listadas e um formulário será aberto, como consta na figura 4.2

Agora, o cliente só precisa preencher o formulário e quando estiver concluído, clicar em "Seguinte" para concluir o processo. Esta operação de criação de um documento/processo ou qualquer outro tipo de objeto faz parte da ferramenta de gestão documental, ou seja, uma vez que tudo esteja devidamente instalado, existem algumas operações que não têm necessidade de serem desenvolvidas.

Como a própria legenda indica, na figura acima podemos ver um diagrama temporal que nos diz quais são os passos necessários para a criação de um objeto com a ferramenta Fabasoft. É um processo muito simples e de fácil execução por parte de qualquer utilizador.

Figura 4.2: Mockup que demonstra o processo escolhido para criar

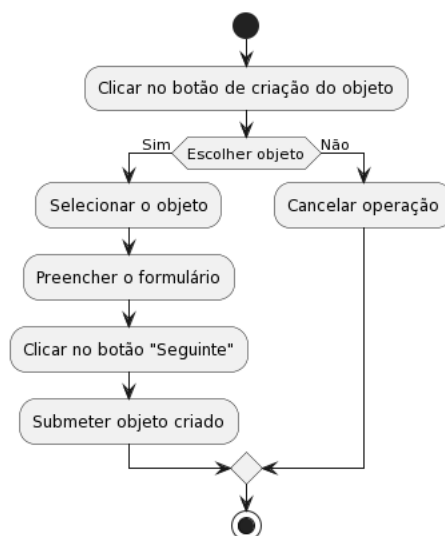


Figura 4.3: Criação de um objeto em Fabasoft

O outro módulo é o das *Instalações Físicas* de cada um dos recursos geológicos mencionados anteriormente: *Águas*, *Minas* e *Pedreiras*. Este módulo é como a identificação de cada um desses recursos geológicos, onde podemos obter alguns dados, como sua localização, nome, etc. As *Águas* têm três tipos de instalações físicas distintas:

- *Água mineral*;
- *Água de nascente*;
- *Água mineroindustrial*.

As Minas têm a sua própria instalação física, assim como as *Pedreiras*.

O objetivo era manter o modelo conceptual atual, onde a **instalação física** é a peça chave que une os processos de negócios. Além disso, como parte do objetivo, a evolução desses processos de negócios para a versão mais recente do software é um assunto muito importante para garantir que tudo esteja a funcionar de acordo com as expectativas. O processo de criação dos formulários de instalação física é idêntico ao que foi mostrado nos *mockups* acima. O cliente só precisa escrever qual instalação física quer criar, escolher e preencher o formulário.

Tudo o que foi escrito acima faz parte do módulo principal: *Estatísticas de recursos geológicos*, ou seja, todas as informações que foram construídas em torno da solução principal possibilitam obter os dados que permitem fazer estatísticas sobre os três recursos geológicos mencionados. A atualização deste módulo principal consiste em replicar os processos/instalações físicas mencionados anteriormente na versão mais atual do Fabasoft. Todos os formulários, fluxos ou automações/ações serão as mesmas, exceto por mudanças inerentes à evolução da aplicação Fabasoft.

- O outro módulo/componente principal é o componente de *Business Intelligence*. Este consiste no desenvolvimento de uma estrutura base e um modelo analítico do componente de Estatísticas dos Recursos Geológicos. Será necessária a exportação de dados do componente de Estatísticas dos Recursos Geológicos para preenchimento dos relatórios.

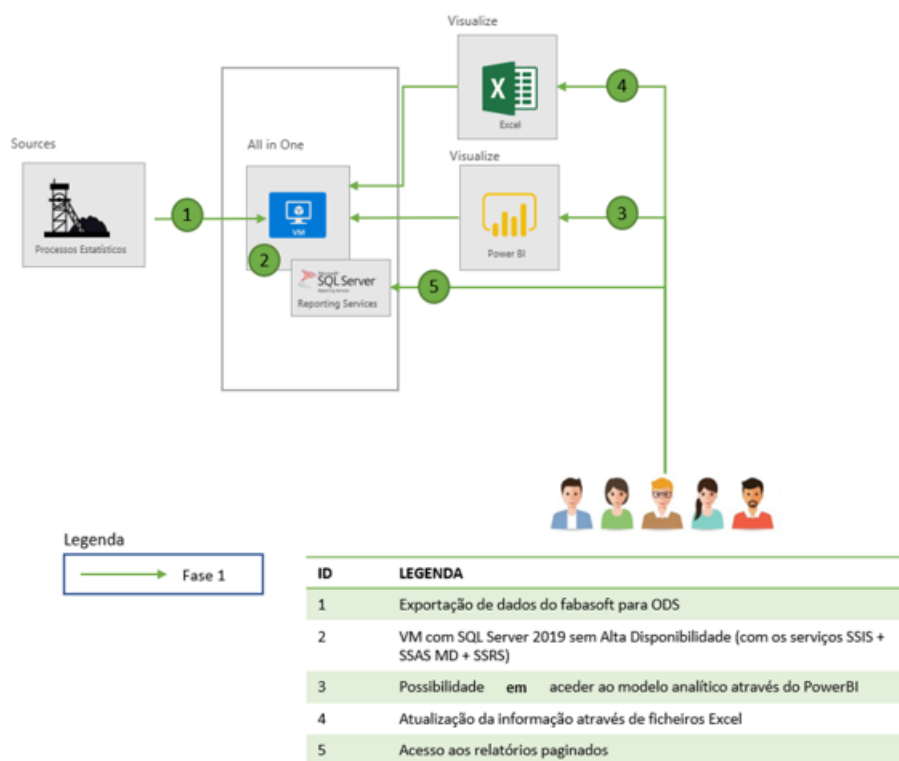


Figura 4.4: Componente de BI

Com a implementação deste componente, é possível definir novas estratégias para o uso de algoritmos de inteligência artificial, isto numa perspetiva futurista. Todos os dados obtidos dos formulários são utilizados para construir conjuntos de dados, conforme mencionado anteriormente. Para isso, é necessário existir um protocolo completo para que os dados possam ser tratados, isto é, analisados, escolhidos e salvos na base de dados.

Interessa referir que não fui eu a pessoa encarregue pela implementação deste componente. No entanto, estive envolvido com a equipa de BI durante o desenvolvimento do projeto, passando desde já a explicar em que parte.

Como foi dito anteriormente, é necessário obter os dados que estão nos formulários dos boletins estatísticos, instalações físicas, e.t.c. Para isso foi necessário fazer uma migração de dados do ambiente desatualizado para o novo ambiente que foi implementado. Nos próximos diagramas temporais - dividi a fase de exportação e a fase de importação em dois diagramas diferentes para não ser um diagrama demasiado complexo - estão as etapas pelas quais foram necessárias passar para migrar todos os dados que constam no âmbito deste projeto e, assim, poder exportar esses mesmos dados do novo ambiente para a Base de Dados pedida pela equipa de BI para poderem começar a fazer todo o trabalho de *Business Intelligence* requerido no âmbito deste projeto.

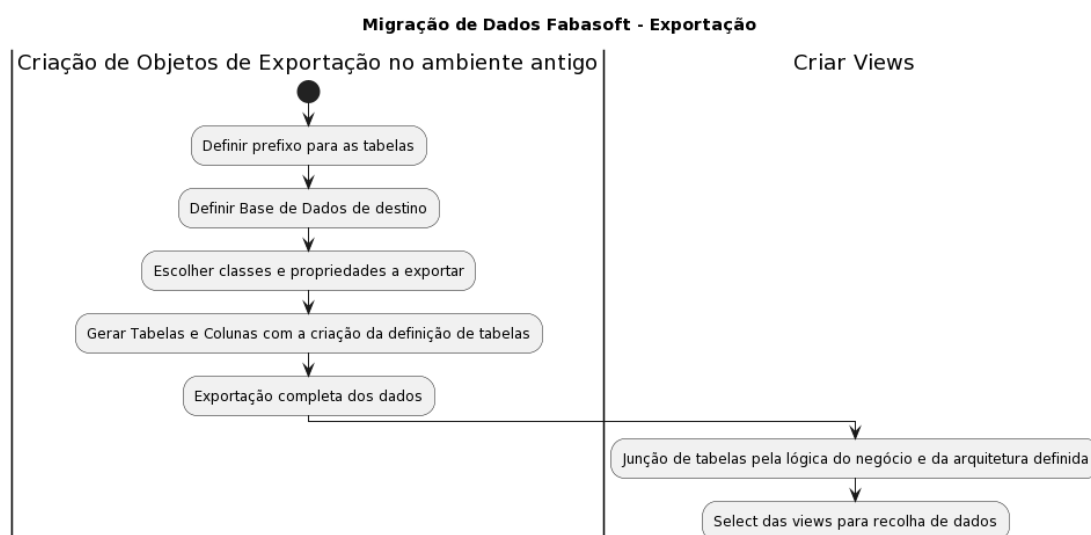


Figura 4.5: Migração de Dados - Exportação

A figura 4.5 tem várias etapas dentro de cada atividade necessária realizar quando se fala da Migração de Dados. Primeiro, foi necessário aceder ao ambiente antigo que contém todos os dados. Tiveram de ser criados objetos de exportação; Neste caso foram criados objetos de exportação por cada um dos tipos de processos estatísticos existentes e para cada um dos tipos de instalação física existentes. Em cada um deles, através do formulário

do objeto de exportação, foi necessário definir o prefixo das tabelas que foram exportadas, definir a Base de Dados de destino e escolher as classes e respectivas propriedades que queria exportar. Depois de concluído esse processo, é preciso selecionar a ação de criação da definição da exportação de tabelas. Isso criará todas as tabelas com as respectivas colunas que irão ser exportadas. São criadas tabelas por cada classe, objeto do tipo composto ou objetos múltiplos, isto é, por exemplo, um conjunto de *strings*. Uma vez criadas todas as tabelas e colunas, é necessário correr a exportação completa para exportar todos os dados referentes às tabelas e colunas.

Quando acabada a exportação dos dados, foi necessário criar *views* para poder juntar as tabelas que diziam respeito ao mesmo processo. Por exemplo, como referi anteriormente, era criada uma tabela para objetos do tipo composto, o que fazia com que esses objetos não constassem na tabela correspondente à classe na qual foram definidos.

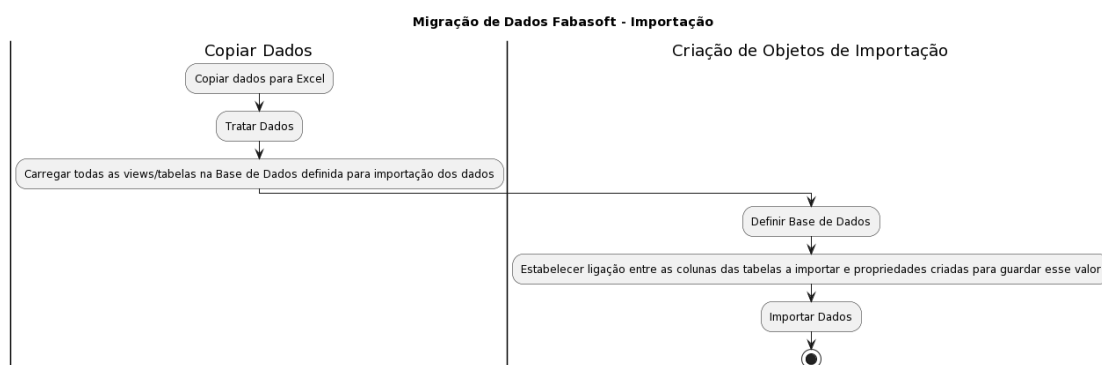


Figura 4.6: Migração de Dados - Importação

Após criação da *view*, teve de se executar um *select* dessa mesma *view* e copiar toda a tabela retornada para um ficheiro excel onde os dados pudessem ser tratados. Devido a alguns erros de conversão dos dados que estavam a ser copiados do excel para a Base de Dados que ia servir para guardá-los e, posteriormente, importá-los para o novo ambiente desenvolvido, teve de haver um tratamento dos mesmos. Uma vez feito o tratamento de dados, a *view* é carregada na Base de Dados.

Já no ambiente atualizado e desenvolvido, tiveram de ser criados objetos de importação para cada uma das *views* criadas. No formulário de cada um destes objetos era definida a Base de Dados de onde os objetos foram importados, especificando a tabela a que se referiam. Depois disso, foi preciso estabelecer a ligação entre as colunas da tabela que se queria importar e as propriedades que tinham sido desenvolvidas nesta nova versão. Foi nessas propriedades que os dados passaram a estar guardados. Concluído este processo, é só correr a importação de dados e esperar que conclua. Uma vez concluída, os dados constavam no novo ambiente desenvolvido.

4.2 Conclusões

Com este capítulo, é possível entender melhor a divisão deste projeto, o que é requerido para a sua conclusão e que desenvolvimentos tiveram de acontecer para o término do mesmo. Houve uma abordagem quanto à arquitetura da solução, embora não pormenorizada, uma vez que isso consta no capítulo seguinte. Foi explicado ao pormenor como é feita a migração de dados, de forma a entender todos os passos necessários, dentro de cada etapa, para que os dados possam ser migrados da versão anterior para a que foi desenvolvida.

DESCRIÇÃO DA IMPLEMENTAÇÃO

O capítulo seguinte foca-se na implementação de todo o *backoffice* proposto para o projeto que esta dissertação descreve. Ao ter todos os requerimentos pedidos pelo cliente bem estabelecidos, a arquitetura dos diferentes componentes e classes a construir bem definida, este capítulo começa por descrever a arquitetura definida para a parte de *backoffice*, dizendo quais componentes e classes foram implementadas em cada um desses componentes. É também explicado o porquê da existência de determinadas classes. Depois disso, será explicada a implementação de classes e propriedades, sendo fundamentado com certos exemplos. Será também explicado que automatismos foram necessário ser desenvolvidos. De notar que certos desenvolvimentos, feitos utilizando a Fabasoft eGov-Suite, requerem a utilização de diferentes DSL, portanto, cada desenvolvimento terá a indicação de qual DSL foi necessário ser utilizado.

5.1 Abordagem da Solução

5.1.1 Instalações Físicas

Como mencionado anteriormente, um dos objetivos foi manter o modelo conceptual o mais próximo possível da versão anterior, o que influenciou a escolha e organização dos componentes. Conforme descrito ao longo deste documento, foram criados componentes em torno de duas grandes áreas do projeto: Instalações Físicas e Estatísticas relacionadas com essas instalações.

Adicionalmente, a solução foi projetada para ser flexível e escalável, permitindo a sua aplicação em projetos futuros ou em propostas feitas pelo cliente. Com esse objetivo em mente, foi criado o componente **Instalação Física - Base**, que contém uma classe abstrata e é a classe hierarquicamente superior do projeto. Este componente foi criado apenas para estruturar a solução e ter em vista projetos futuros.

Um dos componentes que serve como base para as Instalações Físicas dos recursos geológicos é o componente **Instalação Física - Base - Recursos Geológicos**. Neste componente, foi criada uma classe base que engloba as propriedades comuns a cada um dos tipos de Instalações Físicas. Por exemplo, se há propriedades presentes em dois tipos de

formulários diferentes, tanto na identificação da *Mina*, tanto na identificação da *Pedreira* - como identificação da Instalação Física - são instanciadas nesta classe. Além disso, em alguns formulários existem objetos que são ponteiros para classes criadas no âmbito desta solução. Dessa forma, as classes e ponteiros que são utilizados em mais do que um tipo de Instalação Física devem ser criados neste componente, também. Esta lógica foi seguida ao longo da implementação, sendo as classes criadas num componente de acordo com a sua utilização.

Continuando com a mesma abordagem escolhida inicialmente, foram criados mais dois componentes: um para as *Minas* e *Pedreiras* e outro para as *Águas*. Não foram criados três componentes devido às similaridades que existem entre as *Minas* e as *Pedreiras*, tendo sido juntadas as duas classes (*ClassIFMina* e *ClassIFPedreira*) no mesmo componente. O nome deste componente é **Instalação Física - Pedreiras e Minas**, enquanto que o referente às *Águas* é **Instalação Física - Águas**. No componente das *Águas* existe uma classe para cada tipo de água existente: *ClassIFAguaMineral*, *ClassIFAguaNascente* e *ClassIFAguaMineroindustrial*.

O exemplo abaixo serve para perceber a hierarquia seguida neste componente das Instalações Físicas.

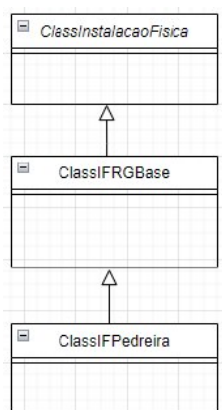


Figura 5.1: Exemplo da hierarquia seguida para cada tipo de Instalação Física

5.1.2 Estatísticas dos Recursos Geológicos

Foi criado um componente para as estatísticas diretamente relacionadas com cada uma das Instalações Físicas, com o objetivo de servir de base para possíveis projetos futuros relacionados com recursos geológicos. Este componente chama-se **Processos - RG - Base** e contém apenas uma classe abstrata (*ClassProcRGBase*) que servirá de base para todos os processos.

Também foi criado um componente que terá uma classe base para todos os processos estatísticos referentes às Instalações Físicas, a *ClassProcRGEstatBase*. Este componente -

Processos - RG - Estatísticas - Base - foi criado porque existem muitas propriedades que são usadas nos diferentes formulários das estatísticas dos diferentes tipos de Instalações Físicas. Para uma melhor gestão de recursos, tudo o que é comumente usado por estes formulários foi instanciado nesta classe.

Por fim, tal como nas Instalações Físicas, foram criados componentes para os processos referentes às *Águas* - **Processos - RG - Estatísticas - Águas** - e para as *Pedreiras e Minas* - **Processos - RG - Estatísticas - Pedreiras e Minas**. Este último seguiu a lógica explicada anteriormente de criar apenas um componente para as *Pedreiras e Minas* devido às similaridades existentes entre os formulários de ambos os recursos. No componente referente às *Águas*, foi criada a *ClassEstatisticaAguas*. Já no componente criado para as *Pedreiras e Minas* foram criadas seis classes, uma por cada processo estatístico existente no âmbito do projeto: *ClassInqueritoUnicoPedreiras*, *ClassInqueritoUnicoMinas*, *ClassEstatisticaPedreira*, *ClassEstatisticaMina*, *ClassAcidentePedreira* e *ClassRelatorioTecnicoPedreira*. A hierarquia seguida no componente das estatísticas segue a mesma lógica que no componente das Instalações Físicas.

5.1.3 Localizações, Regiões e Portal de Serviços

Devido a certas particularidades requeridas pelo cliente, houve a necessidade de criar mais três componentes: das **Regiões**, o das **Localizações** e para o **Portal de Serviços**. No componente das **Regiões** existe a *ClassRegiao* que contém várias propriedades de sistema e outras que foram necessárias para ir em conformidade do que foi pedido pelo cliente. Também é importante mencionar que existem classes baseadas na *ClassRegiao*, isto é, que têm a *ClassRegiao* definida como classe base, uma vez que foi necessária uma distinção entre Freguesias, Concelhos e Distritos. Todas são regiões, mas requerem uma diferente nomeação consoante a sua dimensão. Esta abordagem foi necessária para depois haver um componente das **Localizações**, que contém a *ClassLocalizacao*. Nesta classe existem várias propriedades que foram criadas no componente das **Regiões**, o que a torna hierarquicamente superior a esta componente das **Localizações**. A *ClassLocalizacao* é necessária uma vez que cada Instalação física necessita de ter identificada a sua localização, claro está, para dados estatísticos.

O componente do **Portal de Serviços** foi necessário ser criado porque tem de haver alguém encarregue por cada Inquérito Único criado. Portanto, tiveram de ser criadas a *ClassPortalServicos* e a *ClassAmbienteOrganizacao*. Ambas foram criadas para identificação. A *ClassPortalServicos* para identificação do utilizador do Portal de Serviços, a *ClassAmbienteOrganizacao* para identificar a Organização. Cada utilizador tem de estar associado a uma organização, e as organizações têm vários utilizadores que são nomeados como gestores dessas organizações.

5.1.4 Importância da Hierarquia

O processo de *deployment* de certos componentes é mais complexo que outros, isto é, temos de ter em atenção o que é carregado primeiro. Por exemplo, se carregarmos primeiro o componente das **Localizações** e só depois o das **Regiões**, tal não vai ser possível, uma vez que estamos a tentar carregar propriedades que ainda não estão criadas. Portanto, antes de ser feito qualquer carregamento/*deployment* de qualquer componente, tem de se verificar as dependências entre eles e qual é a ordem pela qual os componentes devem ser carregados.

Importa ainda salientar que, devido à utilização de diferentes DSL, a implementação da solução foi personalizada para cada uma das áreas do projeto, tendo em conta as suas especificidades e necessidades.

5.2 Particularidades da Ferramenta

Todas as ferramentas têm as suas particularidades, e a *Fabasoft egov-Suite* não foge à regra. De seguida, estão exemplos que demonstram como são desenvolvidas certas particularidades que fazem parte de qualquer linguagem orientada a objetos, fazendo distinção por qual DSL deve ser utilizada para certos desenvolvimentos.

5.2.1 Desenvolvido na DSL *Object Model*

Todos os exemplos que se seguem são desenvolvidos numa DSL específica da *Fabasoft eGov-Suite*, que é a *Object Model*. Pela denominação, podemos perceber que é aqui que modelamos todo o projeto, definindo classes, propriedades, e.t.c.

5.2.1.1 Classes e propriedades associadas

Todas as classes mencionadas anteriormente foram criadas na DSL *object model*, que é onde é definido todo o modelo e arquitetura da solução. Para haver hierarquias entre classes, quando cada classe é criada, tem de haver uma classe base associada. A classe que acabou de ser criada, herda todas as propriedades existentes na classe definida como sua classe base, explicando assim o porquê de ter tido a abordagem explicada na secção anterior em relação à criação de determinadas componentes.

Exemplo da criação de uma classe:

```
1 class<CompoundObjectClass> ClassIFAguaMineral: IFRGBASE@103.4800:  
  ClassIFRGBase {  
2     float AttrFloatAreaConcessaoAguas;  
3  }
```

Listagem 5.1: Criação da classe ClassIFAguaMineral

Para que uma propriedade possa ser utilizada posteriormente numa classe, basta declarar o tipo e criar uma referência para nomeação da mesma, como é o caso do que acontece na listagem 5.1 com a propriedade do tipo *float* criada. Caso seja uma propriedade já existente noutro componente, basta escrever a sua referência nessa mesma classe e importar o componente onde essa classe foi criada.

Na prática, para as estatísticas relacionadas diretamente com uma Instalação Física, as propriedades utilizadas para identificar a qual instalação física um determinado formulário correspondia, tiveram de ser propriedades da componente correspondente, uma vez que os formulários se referiam diretamente a essa Instalação Física. Qualquer tipo de propriedade/objeto pode ser criado, uma vez que esse mesmo tipo exista.

5.2.1.2 Propriedades do tipo composto

Existem propriedades simples e compostas, isto é, as compostas podem conter mais do que uma propriedade associada.

São criadas da seguinte forma:

```
1 struct TypeAggrPeriodoFuncionamento {  
2     datetime AttrDtInicio;  
3     datetime AttrDtFim;  
4     string AttrStrObservacoesPeriodoFunc;  
5 }
```

Listagem 5.2: Criação de uma struct

No caso da listagem 5.2, é criada uma *struct* composta com 3 propriedades associadas. Funciona como se de uma classe se tratasse, isto é, os objetos são criados declarando o tipo e, neste caso, o tipo é a propriedade do tipo composto criada.

```
1 class<CompoundObjectClass> ClassTeste: IFRGBASE@103.4800:  
  ClassTesteBase {  
2     TypeAggrPeriodoFuncionamento AttrAggrPeriodoFuncionamento;  
3     TypeAggrPeriodoFuncionamento[] AttrAggrPeriodoFuncionamento1;  
4  }
```

Listagem 5.3: Criação de uma propriedade do tipo composto

Este tipo de propriedades são uma realidade muito utilizada pois, ao inserir esta propriedade num formulário, esta funcionará quase como um quadro, em que um registo

poderá ter um valor por cada propriedade associada. No exemplo coloquei duas propriedades em que, para além de terem referências diferentes, uma pode conter vários valores (**AttrAggrPeriodoFuncionamento1**) e a outra não (**AttrAggrPeriodoFuncionamento**). Isto serve para poder ter vários registos associados a um só agregado.

5.2.1.3 Propriedades do tipo enumerado

Outro tipo de propriedades que faz sentido indicar pois é muito utilizado no âmbito do *Fabasoft eGov-Suite*, são as propriedades do tipo enumerado, que são propriedades cujo valor é escolhido de entre um conjunto de opções/listagem possível.

Podemos ver a criação de um enumerado na listagem a seguir:

```

1  enum TypeEnumEstadoCaptacao {
2      suspensao = 0,
3      reserva = 1,
4      producao = 2,
5      inativa = 3,
6      cimentado = 4,
7      selado = 5
8  }
```

Listagem 5.4: Criação de uma propriedade do tipo enumerado

Outra grande funcionalidade oferecida pelo *Fabasoft eGov-Suite* são os *backlinks*. Têm de ser definidos, tal como as propriedades faladas anteriormente, na DSL *Object Model*. Esta funcionalidade serve para estabelecer uma ligação entre propriedades, ou seja, e pegando num exemplo concreto deste projeto, uma propriedade ponteiro para a classe dos Inquéritos Únicos das **Pedreiras** deve estar associada à Instalação Física da Pedreira a qual esse Inquérito diz respeito. Essa associação é feita no formulário do próprio Inquérito, pois todos têm uma propriedade que é um ponteiro para um determinado objeto da classe da Instalação Física correspondente. Em contrapartida, irá haver na Instalação Física uma listagem de todos os Inquéritos Únicos da mesma. Sempre que a propriedade ponteiro é preenchida no formulário do Inquérito Único, esse mesmo Inquérito é listado na propriedade *backlink* existente na Instalação Física.

Para isso acontecer, é necessário implementar o seguinte:

```

1  class <CompoundObjectClass> ClassInqueritoUnicoPedreiras:
2      PROCRGESTATBASE@103.4800:ClassProcRGestatBase {
3          abstract = false;
4          common = true;
5          ClassIFPedreira AttrPtrIFPedreira {
6              link = AttrBackPtrIUPedreiras;
7              allow {
8                  ClassIFPedreira create;
9              }
10 }
```

```

11     extend class ClassIFPedreira {
12         backlink<ClassInqueritoUnicoPedreiras>[] AttrBackPtrIUPedreiras;
13     }

```

Listagem 5.5: Associação de propriedades

Podemos verificar que há a extensão de uma classe pertencente a outro componente que está hierarquicamente acima do componente das **Estatísticas**. Desde que as regras de hierarquia sejam respeitadas, esta funcionalidade pode ser utilizada para adicionar propriedades a uma determinada classe criada noutra componente. Visto que o componente das Instalações Físicas está hierarquicamente acima do componente das Estatísticas, não era seguida a hierarquia definida ao criar uma propriedade ponteiro dos Inquéritos das Pedreiras no componente das Instalações Físicas. A inclusão das propriedades *backlink* nos formulários vai ser abordada mais à frente, na secção 5.2.2.2.

Assim, com estes exemplos, já é de mais fácil percepção que propriedades foram desenvolvidas ao longo deste projeto e como é organizado o mesmo.

5.2.2 Desenvolvido na DSL *User Interface*

Os formulários da ferramenta já são desenvolvidos noutra DSL diferente, que é a *User Interface*. Tudo o que é implementado utilizando esta DSL serve para definir como e o quê que é visto pelo utilizador. Estão, de seguida, exemplos do que é necessário para poder criar um formulário, associá-lo a uma ou várias classes e como formatá-lo.

5.2.2.1 Criação de páginas/separadores

Todos os formulários existentes no ambiente final disponibilizado contêm, no mínimo, um separador. No mínimo porque os formulários podem ter mais do que um separador. É nesta DSL que definimos que separadores constam num determinado formulário e qual a sua ordem, mais à frente está um exemplo disso.

Para criar uma página/separador, é preciso fazer o seguinte:

```

1     formpage PageIUMinas {
2         audience = enduser;
3         dataset {
4             objname;
5             AttrIntReferenteAno;
6             AttrDtDataPedido;
7             AttrPtrIFMina;
8         }
9         layout {
10            row {
11                objname {
12                    colspan = 8;
13                    labelposition = left;
14                }

```

```
15     }
16     row {
17         AttrIntReferenteAno {
18             colspan = 4;
19             labelposition = left;
20             mustbedef = expression {
21                 true;
22             }
23         }
24         AttrDtDataPedido {
25             colspan = 4;
26             labelposition = left;
27             changeable = expression {
28                 false;
29             }
30         }
31     }
32     row {
33         AttrPtrIFMina {
34             colspan = 8;
35             labelposition = left;
36             mustbedef = expression {
37                 true;
38             }
39         }
40     }
```

Listagem 5.6: Criação de página de um formulário

À página é dada uma referência, que pela listagem 5.6 podemos perceber que é **PageIUMinas**. São definidas as propriedades que fazem parte dessa mesma página, inserindo-as no *dataset*, e depois é também definida a ordem e a forma como essas mesmas propriedades são apresentadas quando o utilizador interage com esta página em específico, definido no *layout* da página. Cada linha pode conter várias propriedades e é definido o seu tamanho com a definição da *colspan* que cada propriedade tem. Também há várias condições que a ferramenta disponibiliza:

- ***mustbedef*** - esta função define que a propriedade tem de estar definida aquando a submissão do formulário se a condição implementada for verdadeira;
- ***visible*** - esta função define que a propriedade seja visível se se verificar que a condição definida é verdadeira;
- ***changeable*** - esta função define se uma propriedade é alterável aquando o preenchimento do formulário se a condição definida se verificar.

Faz sentido abordar este tipo de funcionalidades da ferramenta uma vez que isto faz parte de muitos dos pedidos feitos pelo cliente. Quando pede que, por exemplo, certas propriedades tenham de estar preenchidas aquando a submissão do formulário, ou que só um determinado utilizador tem a permissão de alterar uma certa propriedade.

5.2.2.2 Criação/extensão de formulários

Depois de concluída a estruturação da página, tem de se definir em que formulário ou formulários queremos que a mesma conste.

Para isso, foi necessário adicionar a referência da página que está no exemplo a um formulário:

```
1 form FormIUMinas {  
2   PageIUMinas;  
3 }
```

Listagem 5.7: Adição de uma página a um determinado formulário

Uma vez concluído este processo, é necessário associar este formulário a pelo menos uma classe existente. Não é possível ter formulários sem estar definido a que classe o mesmo se refere.

Para isso, é necessário o seguinte código:

```
1 forms for ClassInqueritoUnicoMinas {  
2   default, ObjectConstructor, EditObjectAttributes,  
3     ReadObjectAttributes, SearchObjects {  
4     FormIUMinas;  
5   }  
6 }
```

Listagem 5.8: Associação do formulário à classe

Uma vez concluído estes desenvolvimentos descritos, basta carregar o componente a que este DSL está inserido e este formulário estará disponível. Depois de carregado, este formulário em específico servirá para a criação de objetos da classe **ClassInqueritoUnicoMinas**. Só têm de ser preenchidos os campos em conformidade com a vontade do utilizador, seguindo os requerimentos que cada campo/propriedade contenha, e submeter no final. Uma vez submetido, o objeto fica criado e pode ser acedido e editado de acordo com as permissões do utilizador que está autenticado.

Pegando no exemplo e servindo de introdução para o que está de seguida, vou voltar a falar dos *backlinks*, que está explicado na listagem 5.5.

Os formulários, tal como as propriedades e classes, dão para ser estendidos e fazer alterações, tanto por questões que surgem devido à hierarquia mas também por questões que possam surgir com novos desenvolvimentos.

No desenvolvimento do projeto, houve a necessidade de estender os formulários referentes às Instalações físicas, uma vez que havia o requerimento de conter um separador,

por cada formulário da Instalação física, que contivesse a lista de Inquéritos associados à mesma. Como foi explicado anteriormente, basta ter uma página que contenha a propriedade *backlink*, que neste caso é múltipla, e assim aparecem todos os Inquéritos que dizem respeito à Instalação física em questão.

Foi necessário criar a página e estender o formulário em questão adicionando a página que foi criada:

```

1  formpage PageIUPed {
2      audience = enduser;
3      dataset {
4          AttrBackPtrIUPedreiras;
5      }
6      layout {
7          row {
8              ESTATPEDREIRASMINAS@103.4800:AttrBackPtrIUPedreiras {
9                  colspan = 8;
10                 labelposition = left;
11             }
12         }
13     }
14 }
15 extend form FormIFPedreiras {
16     PageIUPed;
17 }

```

Listagem 5.9: Adição de um separador estendendo um formulário

Toda a interface com que o utilizador irá interagir foi desenvolvida seguindo o que foi descrito e respeitando toda a arquitetura e hierarquia da solução. Desta forma, é mais fácil entender como foi desenvolvida a parte que diz respeito ao *front-end* do projeto.

5.2.3 Desenvolvido na DSL *Use Case*

É nesta DSL, *Use Case*, onde todos os automatismos são desenvolvidos. Todos os automatismos que faziam parte do âmbito do projeto estão associados a quadros. Esses quadros, num contexto mais tecnológico e adequado à ferramenta, são agregados/propriedades do tipo composto em que também são propriedades que podem conter mais do que um registo, ou seja, são propriedades múltiplas. Esta introdução serve para dar a perceber no que se baseiam os automatismos, pois com os valores de vários registos num quadro, podemos ter totais e, nada melhor que esses totais se calcularem automaticamente após a sua inserção ou após conclusão do preenchimento do formulário. Por exemplo, e falando de casos concretos, existe a contagem de trabalhadores efetivos, trabalhadores do sexo feminino e trabalhadores do sexo masculino no que diz respeito ao pessoal encarregue do engarrafamento de uma determinada água. Como explicado acima, após inserção dos

números em cada um dos campos, é necessário ter valores concretos para que depois possam ser feitas estatísticas precisas e verdadeiras.

Para isso acontecer, tem de haver o seguinte desenvolvimento:

```

1 usecase TotaisPessoalEng () {
2     mlname = {}
3     variant ESTATAGUAS@103.4800:ClassEstatisticaAguas {
4         expression {
5             integer homEfTemp = 0;
6             integer mulhEfTemp = 0;
7             integer homTTemp = 0;
8             integer mulhTTemp = 0;
9             for (integer pos = 0; pos < count(cooobj.
10                 AttrAggrPessoalEngarraamento); pos++) {
11                 homEfTemp += cooobj.AttrAggrPessoalEngarraamento[pos].
12                     AttrIntNumTrabEfetivosHomens;
13                 mulhEfTemp += cooobj.AttrAggrPessoalEngarraamento[pos].
14                     AttrIntNumTrabEfetivosMulheres;
15                 homTTemp += cooobj.AttrAggrPessoalEngarraamento[pos].
16                     AttrIntNumTrabTempHomens;
17                 mulhTTemp += cooobj.AttrAggrPessoalEngarraamento[pos].
18                     AttrIntNumTrabTempMulheres;
19             }
20             cooobj.AttrAggrTotaisPessoalEng.AttrIntNumTrabEfetivosHomens =
21                 homEfTemp;
22             cooobj.AttrAggrTotaisPessoalEng.AttrIntNumTrabEfetivosMulheres
23                 = mulhEfTemp;
24             cooobj.AttrAggrTotaisPessoalEng.AttrIntNumTrabTempHomens =
25                 homTTemp;
26             cooobj.AttrAggrTotaisPessoalEng.AttrIntNumTrabTempMulheres =
27                 mulhTTemp;
28         }
29     }
30     variant Object {
31         empty;
32     }
33 }

```

Listagem 5.10: Desenvolvimento de um automatismo

Pela listagem 5.10 podemos ver o desenvolvimento por detrás do preechimento dos campos que dizem respeito aos totais, neste caso referente ao pessoal por detrás do engarraamento. Primeiramente tem de ser dada uma referência/nome ao automatismo desenvolvido. Mais à frente vamos ver como é feita a associação deste mesmo automatismo à propriedade em questão e esta será a referência dada como identificação. Toda a *expression* tem de ser associada a uma classe, pois é no formulário dessa classe que será executado o automatismo. De seguida, são criadas variáveis temporárias de forma a que estas possam

ser alteradas várias vezes durante a execução do automatismo. Dentro do ciclo, que é executado tantas vezes quanto o número de registos existentes no quadro, está a ser somado a cada variável temporária o valor do registo associado à posição em que o mesmo se encontra. Depois de concluído o ciclo, cada propriedade que guarda o valor do total referente, é igualada à variável temporária que guardou a soma.

Algumas particularidades da ferramenta valem a pena ser mencionadas. como por exemplo, para conseguir chegar ao valor da propriedade tem de ser seguido todo um caminho. Para, por exemplo chegar à propriedade que diz respeito ao número de trabalhadores efetivos do sexo masculino, tem de se aceder ao objeto em questão, que neste caso é o **cooobj**, que diz respeito ao objeto em que este automatismo está a ser executado. Por exemplo, este automatismo é executado no preenchimento de um formulário correspondente às Estatísticas de uma determinada **Água**, logo o objeto é essa mesma Estatística que está a ser preenchida. Depois, ainda tem de se aceder à propriedade do tipo composto que contém a propriedade à qual queremos chegar. Só depois é escrita a referência da propriedade em questão, ficando com o seguinte aspeto:

1 `cooobj.AttrAggrTotaisPessoalEng.AttrIntNumTrabEfetivosHomens`

Listagem 5.11: Caminho para chegar a uma determinada propriedade

Finalmente, após concluir a implementação deste método, teve de se associar o mesmo a uma propriedade, para que quando o formulário fosse submetido, o automatismo fosse executado.

Esta associação tem de ser feita na definição da propriedade, portanto teve de ser feita na DSL *Object Model*:

```
1 TypeAggrPessoal[] AttrAggrPessoalEngarrafamento {  
2     attractset = TotaisPessoalEng;  
3 }
```

Listagem 5.12: Associação do automatismo à propriedade

Desta forma é perceptível a abordagem que foi tomada e como funciona o desenvolvimento de automatismos com a ferramenta *Fabasoft eGov-Suite*.

5.3 Gestão de acessos dos utilizadores criados

A componente de *BackOffice* está dividida em duas áreas, como foi descrito anteriormente: área das **Minas e Pedreiras** e área das **Águas**. Como tal, também teve de haver uma gestão de quais utilizadores podem aceder ao ambiente desenvolvido e a que tipo de classes e objetos têm permissões de acesso, isto tudo tendo em conta a área a que estão atribuídos dentro da organização do cliente. Por isso, na criação dos utilizadores, todos os acessos são definidos tendo em conta o perfil atribuído a cada. Se for atribuído o perfil da área das **Águas**, o utilizador quando se autentica só pode pesquisar e criar objetos que digam respeito ao componente criado para essa área. Se for atribuído o perfil da área das **Minas e Pedreiras** o utilizador quando se autentica só pode pesquisar e criar objetos que digam respeito ao componente criado para essa área. No capítulo seguinte estará uma demonstração da validação desta gestão, servindo para confirmação do bom funcionamento das atribuições de perfil por utilizador.

5.4 Conclusões

Tendo em conta o que foi explicado em todo o capítulo, fica sucinto o que foi desenvolvido no decorrer da fase de implementação de toda a componente de *BackOffice*. É perceptível que teve de haver uma adaptação a uma nova ferramenta para poder ser feito o desenvolvimento de forma otimizada e para que tudo funcionasse de encontro com os requerimentos do cliente. O projeto requereu a construção de uma arquitetura complexa com vários princípios de hierarquia a serem necessários para o bom funcionamento do produto final.

AVALIAÇÃO DA APLICAÇÃO REALIZADA

O projeto no qual esta dissertação é baseada e escrita é composta por vários componentes, como foi descrito anteriormente. Claro está, existe sempre uma perspectiva em relação ao que se espera de uma solução proposta de maneira a controlar o rumo da mesma e também para saber o que esperar no que toca ao resultado final. Neste capítulo está descrito o que era esperado, aquando a realização da proposta, de cada um dos componentes. Após feita essa descrição, será feito um ponto de situação do componente em si e da minha influência e encargo sobre o mesmo, isto porque como já foi referido, este é um projeto que tem o envolvimento de várias equipas.

6.1 Portal de Serviços

Um dos grandes objetivos do projeto passava pelo desenvolvimento de um **Portal de Serviços**, desenvolvido em tecnologia atual. Uma vez implementado, seria testado e se tivesse o comportamento esperado, estava de acordo com o proposto. O comportamento esperado é o de, através de uma interface de utilizador simples, onde serão requeridos os dados de autenticação (nome de utilizador e *password*), ser efetuado o *login* pelo proprietário/explorador do recurso geológico em questão. Uma vez autenticado, procede ao preenchimento do inquérito ou boletim estatístico associado ao recurso geológico de acordo com os números referentes ao ano que passou. Quando concluísse o preenchimento do formulário, seria só submeter para, posteriormente, ser validado por alguém por parte do cliente.

Também teriam de ser feitos alguns testes de carga em que eram analisados alguns fatores como a capacidade de resposta e a escalabilidade, vendo se o número de utilizadores a utilizar o portal simultaneamente for elevado, que impacto terá. Isto é, teria de ser garantido que uma percentagem de todos os utilizadores existentes, definida consoante as necessidades do cliente, consegue aceder ao portal concorrencialmente. Teria de se estudar qual é a média de inquéritos preenchidos por dia para poder ser feita a simulação e verificar se tudo ocorreria como esperado, preenchendo e submetendo os formulários sem qualquer problema.

Se o acesso fosse efetuado com sucesso e o formulário chegasse à fase de validação sem problemas, podíamos concluir que esta componente estava bem implementada e concluída.

Infelizmente e por se tratar de um projeto empresarial, em que as adversidades são reais, não foi possível concluir a implementação desta componente de forma a mostrar resultados positivos do mesmo.

6.2 *BackOffice*

O componente de *BackOffice* foi desenvolvido por mim, como foi documentado no decorrer da dissertação. O objetivo deste grande componente era o de desenvolvimento de todo um novo módulo para a **Estatística dos Recursos Geológicos**. Este mesmo módulo contém as instalações físicas de cada área e os processos estatísticos respetivos. Sabendo que o modelo conceptual é para ser mantido tendo em conta a versão anterior, teve de se fazer uma implementação com os mesmos princípios, apesar de haver diferenças que resultaram de erros anteriores.

Para se testar todo este componente, existem vários testes que tiveram de ser feitos. Mas, antes disso, irei dizer o que era esperado para concluir que tudo estava dentro da conformidade do que foi proposto:

- Primeiro, ao serem carregados os componentes, teria de se seguir uma ordem para garantir que a hierarquia estabelecida era respeitada. Se não houvesse erros com o carregamento dos componentes, estaria tudo de acordo com o que tinha sido implementado e com a hierarquia definida.
- Depois de carregados os componentes, teria de ser feita a atribuição dos perfis a cada utilizador e confirmar que os acessos estavam em conformidade com os perfis associados.
- Teriam de ser feitos testes de funcionalidade. Os formulários teriam de estar em conformidade com a implementação, isto é, os campos pela ordem definida, com as dimensões definidas e no formulário associado. Verificação se todos os campos constam no formulário com a *label* que foi definida. Por exemplo, se os campos/propriedades do tipo *string* estivessem bem criados, qualquer valor podia ser preenchido. Teria de ser feita a verificação se os campos que tinham alguma condição, essa mesma condição estaria a funcionar segundo a implementação da mesma. Teria também de se verificar os acessos a determinadas funcionalidades, páginas de certos formulários, preenchimento de alguns formulários, e.t.c., dependendo do utilizador que estivesse autenticado. Teria de se ver se as propriedades múltiplas têm a funcionalidade de ter mais do que um registo associado. Se as propriedades múltiplas forem *unique*, só poderá existir um registo com um determinado valor. Se houver outro registo com esse valor que tenha sido adicionado anteriormente, nada é adicionado.

- Quanto à migração de dados, teriam de ser feitos testes de qualidade dos dados, isto comparativamente com a versão anterior. Teriam de ser feitos testes de comparação entre casos específicos para ser feita a verificação da qualidade dos dados e se os dados estivessem preenchidos na propriedade associada corretamente, poderia-se concluir que a migração tinha sido efetuada com sucesso. Também seria necessário fazer testes quantitativos dos dados, isto é, por exemplo, se existissem 10000 objetos de inquéritos únicos referentes às pedreiras, após a migração dos dados, a nova versão também teria de conter esse mesmo número de objetos.

Assim, no decorrer desta secção, irão estar algumas imagens que comprovam o bom funcionamento dos pontos referidos anteriormente em relação ao que deveria ser testado. Cada imagem terá uma pequena explicação para que se entenda o seu significado e o porquê de nos indicar o sucesso de implementação.

Componentes carregados e desenvolvidos para BO			
File Área de transferência Vista Tools Versions			
			
	Nome	Classe de objecto	Endereço
☐	 Instalação física - Base - Recursos Geológicos	 Componente do Software	COO.103.4800.1.238404
☐	 Instalação física - Base	 Componente do Software	COO.103.4800.1.238203
☐	 Regiões	 Componente do Software	COO.103.4800.1.243233
☐	 Localizações	 Componente do Software	COO.103.4800.1.243109
☐	 Processos - Estatística - Águas	 Componente do Software	COO.103.4800.1.240510
☐	 Processos - Estatística - Pedreiras e Minas	 Componente do Software	COO.103.4800.1.240314
☐	 Processos - RG - Estatística - Base	 Componente do Software	COO.103.4800.1.240108
☐	 Processos - RG - Base	 Componente do Software	COO.103.4800.1.239007
☐	 Instalação física - Águas	 Componente do Software	COO.103.4800.1.238808
☐	 Instalação física - Pedreiras e Minas	 Componente do Software	COO.103.4800.1.238606
☐	 Portal de serviços	 Componente do Software	COO.103.4800.1.244727
Adicionar registo Procurar e adicionar			

Figura 6.1: Todos os componentes desenvolvidos no âmbito do módulo de *BackOffice*

Na figura 6.1 está a prova que todos os componentes foram carregados com qualidade e pela ordem correta. Isso indica que foi seguida e respeitada a hierarquia definida e não houveram quaisquer erros de compilação. Resta verificar se toda a implementação está de acordo com o que foi definido.

Na figura 6.2 são mostradas quais classes um utilizador que, com o perfil das águas,

Perfil - Estatística de Águas (Perfil do utilizador):Ler		
Objecto	Área de transferência	Vista Ferramentas Definições Versões Arquivo
Classes de objecto para criar		
 ☐	Nome	Objecto ACL
☐ 	Pessoa	ACL predefinido para objectos gerais
☐ 	Organização	ACL predefinido para objectos gerais
☐ 	Pasta	ACL predefinido para objectos gerais
☐ 	Recurso Geotérmico	ACL predefinido para objectos gerais
☐ 	Tipo de captação	ACL predefinido para objectos gerais
☐ 	Caraterização Química Termalismo	ACL predefinido para objectos gerais
☐ 	Captação	ACL predefinido para objectos gerais
☐ 	Produtos brancos	ACL predefinido para objectos gerais
☐ 	Marca comercial	ACL predefinido para objectos gerais
☐ 	Tipo de Águas	ACL predefinido para objectos gerais
☐ 	Aproveitamento	ACL predefinido para objectos gerais
☐ 	Utilização	ACL predefinido para objectos gerais
☐ 	Caraterização química (engarraçamento)	ACL predefinido para objectos gerais
☐ 	Água mineroindustrial	ACL predefinido para objectos gerais
☐ 	Água de nascente	ACL predefinido para objectos gerais
☐ 	Água mineral natural	ACL predefinido para objectos gerais
☐ 	Estatística de águas	ACL predefinido para objectos gerais

Figura 6.2: Classes com permissão de criação de objetos para utilizadores com o perfil das Águas

consegue criar objetos. Como já foi referido, a ferramenta já tem funcionalidades implementadas aquando a sua instalação e a criação de objetos é uma delas. Logo, para testar se apenas estas classes são permitidas para a criação de objetos de um utilizador com o perfil das Águas atribuído, temos de fazer *login* com um desses utilizadores e verificar que classes são disponibilizadas aquando a opção de criação de um objeto.

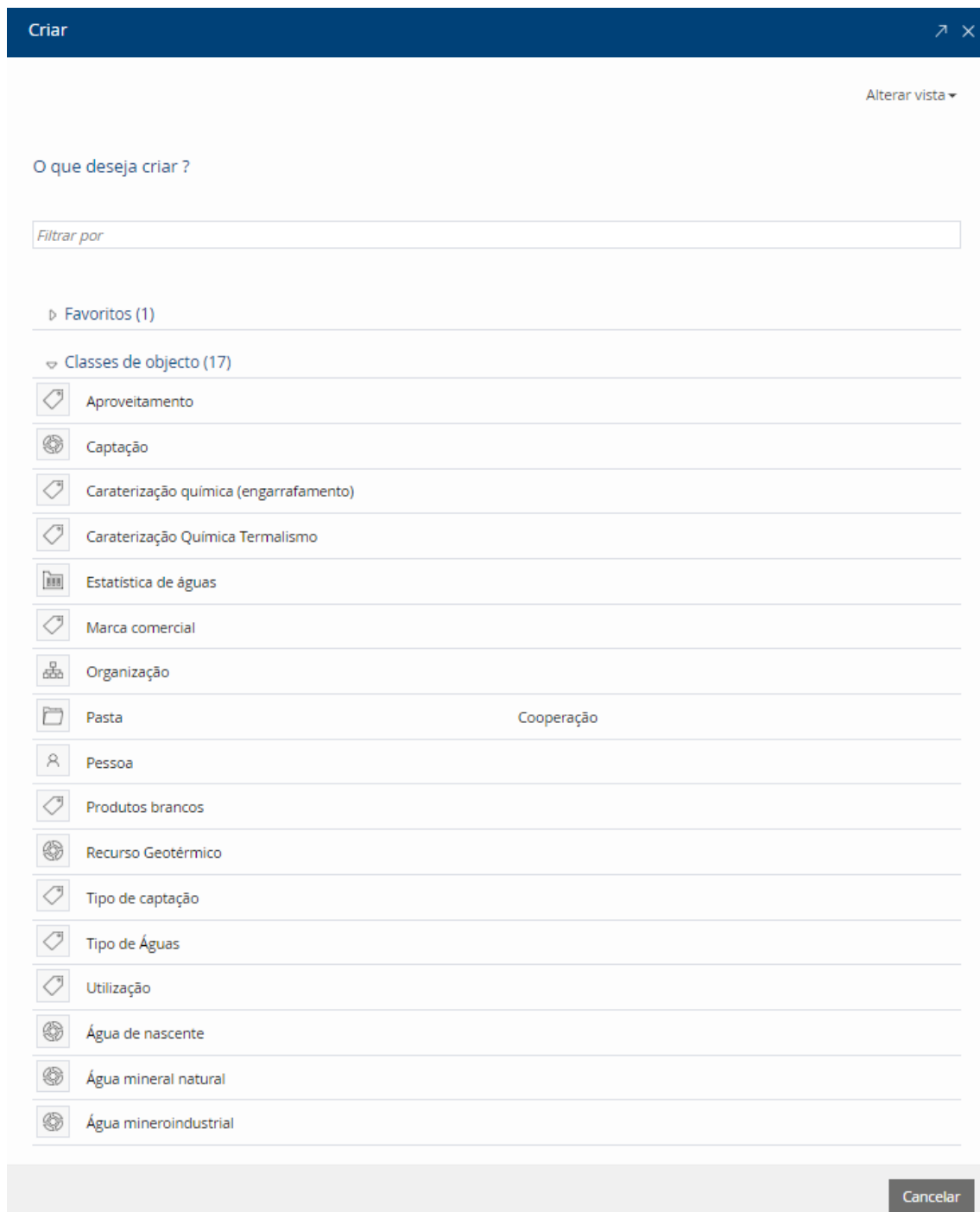


Figura 6.3: Classes disponibilizadas para a criação de objetos para utilizadores com o perfil das Águas

E como se pode verificar na figura 6.3, as classes disponibilizadas quando o utilizador quer criar um objeto são apenas as que foram selecionadas e associadas ao perfil em questão, neste caso o perfil das Águas. Assim, concluímos que as permissões estão de acordo com o que foi definido.

O que acabou de ser explicado acontece com a criação de objetos, mas o processo

para a procura de objetos é idêntico. Se um utilizador quer procurar um objeto de uma determinada classe, tem de ter permissão para isso. Essas permissões são definidas de forma semelhante com a de criação de objetos, como está na seguinte figura:

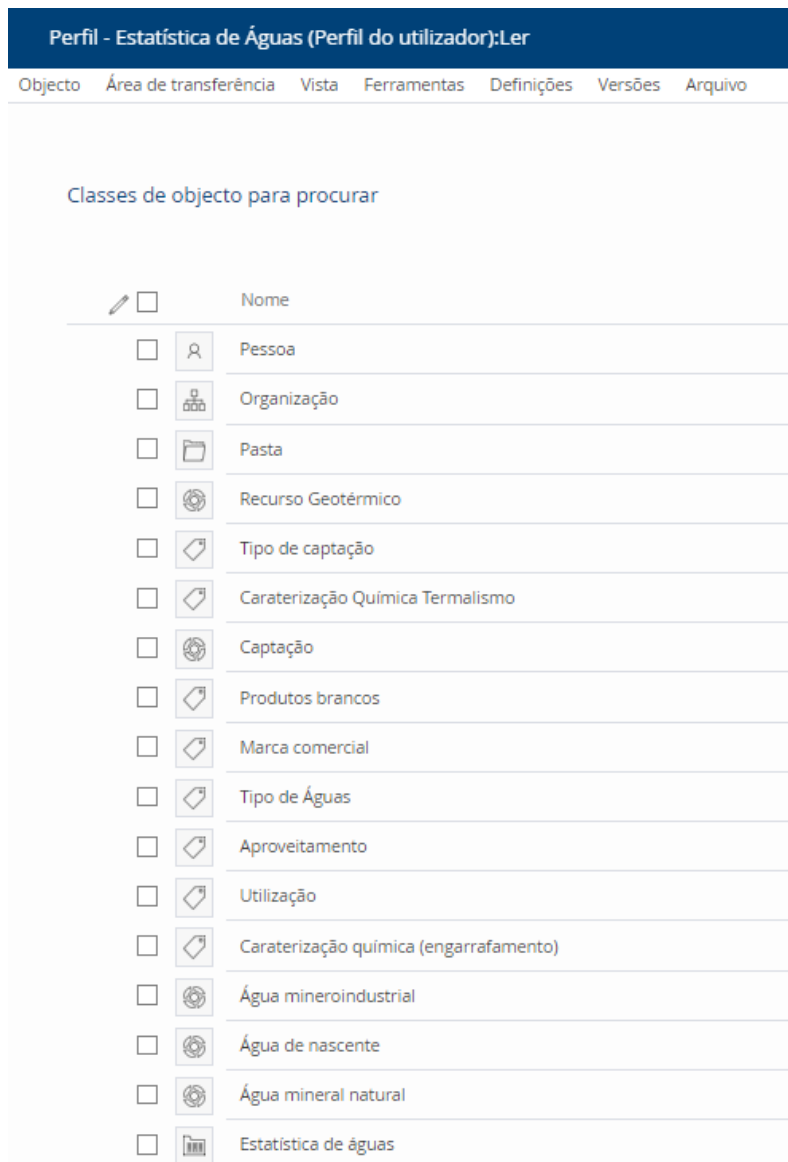


Figura 6.4: Classes com permissão de procura de objetos para utilizadores com o perfil das *Águas*

E para se verificar se as permissões estão de acordo com o definido, é muito idêntico ao que foi explicado anteriormente em relação à operação de criação de objetos. Foi através da autenticação de um utilizador com o perfil das águas que, ao executar a operação de procura de um objeto, comparou-se que classes eram disponibilizadas.

Na figura 6.5 podemos verificar que as classes disponibilizadas para o utilizador autenticado são as mesmas definidas para um utilizador com o perfil da área das *Águas*.

Dessa forma, comprova que a atribuição das classes por perfil foi bem executada.

Assim, ficam verificadas as permissões de procura e criação de objetos para cada perfil.

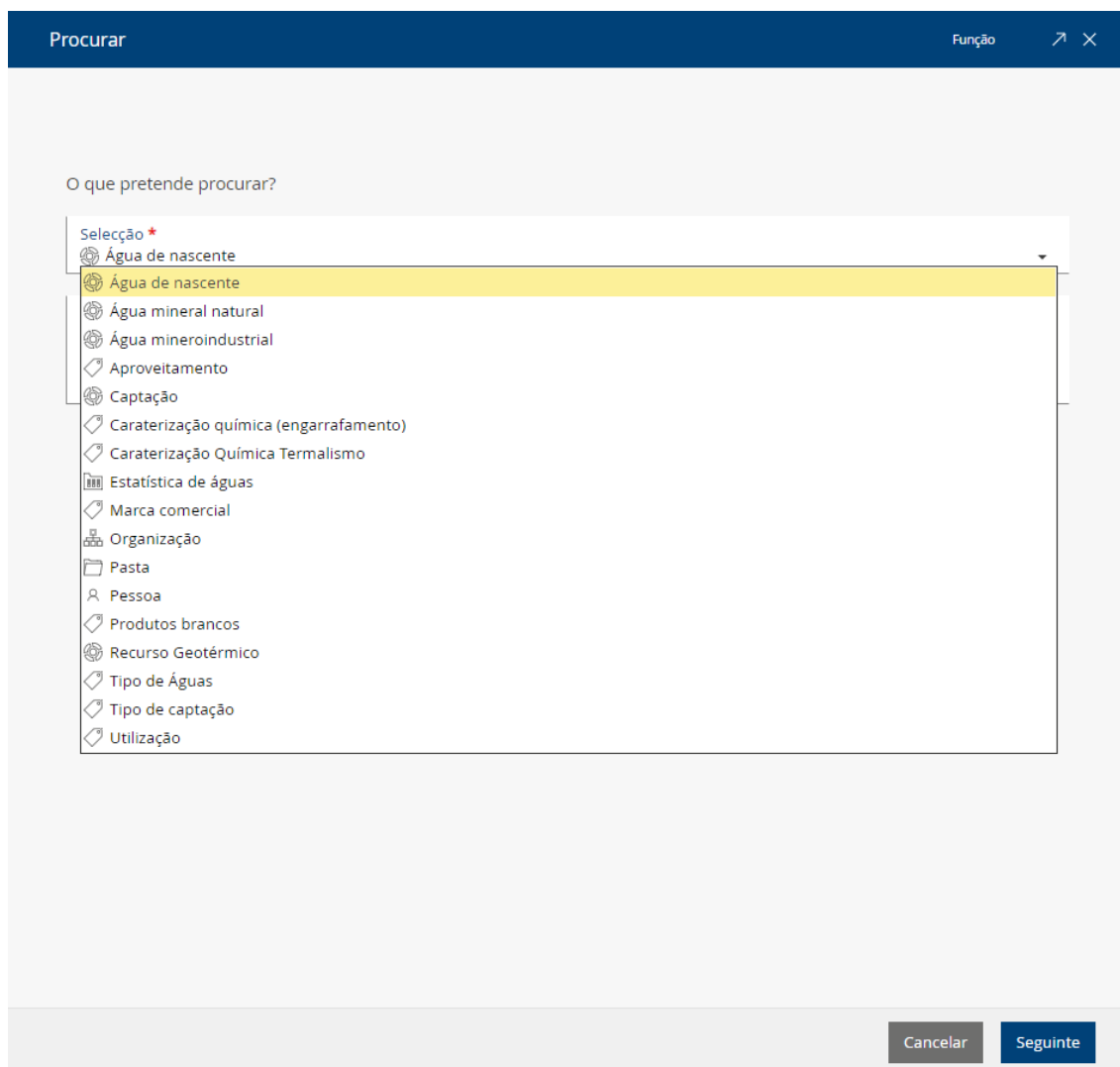


Figura 6.5: Classes disponibilizadas para a procura de objetos para utilizadores com o perfil das *Águas*

Depois de verificado o bom funcionamento em relação à gestão de acessos dos utilizadores, tiveram de ser feitos testes de funcionalidade. A seguinte imagem (6.6) vai conter um dos formulários desenvolvidos e, que se estiver de acordo com a implementação feita, podemos concluir que o preenchimento será feito da forma requerida pelo cliente.

Na figura podem ser verificadas inúmeras funcionalidades associadas a várias propriedades.

Campos como a *data de validação*, *data de correção* e *data de anulação* não podem ser alteradas, pois foi implementada essa condição. Já a *data de submissão* tem o tipo requerido, que só permite o preenchimento do campo com uma data.

A propriedade *referente ao ano* despoleta um erro por invalidade do valor inserido no campo, isto é, a propriedade é do tipo *integer*, logo não aceita outros valores que não sejam numéricos.

Estatística de águas

Foram encontrados problemas (1). Verifique os campos destacados do formulário.

Estatística das águas

Número

Referente ao ano contém um valor inválido.
Referente ao ano !
ddddd

Ano de submissão
2023

Estado do pedido *
Submetido

Data de submissão
DD/MM/YYYY hh:mm:ss

Água
DGEG RG Água de Nascente N.º NAS097000 - FONT SALEM

Data de validação

Data de correção

Data de anulação

Classificação
01101/Estatística

Responsável
Utilizador
Grupo
Posição

Documento de confirmação
Mostrar detalhes (0)
Esta lista não contém nenhuma entrada.

Observações

Observações/Correções

Cancelar Aplicar Seguinte

Figura 6.6: Formulário correspondente à Estatística das Águas

A condição associada à propriedade *estado do pedido* está bem desenvolvida, indicando a necessidade do campo estar definido antes da submissão do formulário. Essa verificação é feita com a existência do *asterísco (*)* assinalado no campo correspondente. Também se comprova que essa mesma propriedade é um enumerado em que a listagem de valores possíveis é visível através do clique na pequena seta (figura 6.7).

Figura 6.7: Exemplo da seleção de valor de uma propriedade do tipo enumerado

Quanto à formatação do formulário, está de acordo com a ordem e dimensão desenvolvida e pode-se constatar que todos os campos têm as *labels* associadas corretamente. Também se verifica a existência de todos os separadores, do lado esquerdo do formulário, e pela ordem desejada.

Figura 6.8: Exemplo de um agregado com múltiplos valores

Para finalizar a demonstração de todos os tipos de teste de funcionalidade que foram feitos, na figura 6.8 está a demonstração de mais um tipo de propriedade, uma propriedade do tipo composto e múltipla, em que constam todas as propriedades definidas e podemos ter mais do que um registo pertencente ao agregado. Mais uma vez com o resultado esperado.

Depois de uma análise ao detalhe de todas as propriedades, separadores e ao próprio formulário, podemos chegar à conclusão que tudo foi implementado corretamente e o resultado é o esperado.

Para finalizar os testes necessários ligados à componente de *BackOffice*, têm de ser feitos testes em relação à migração dos dados. Os testes são feitos por amostragem, ou seja, são escolhidos vários conjuntos de dados que necessitem de verificação para se conseguir avaliar a qualidade do que foi migrado da versão anterior para a nova.

Acedendo à versão anterior, primeiro verificamos se o número de dados migrados correspondem ao que existe na versão anterior.

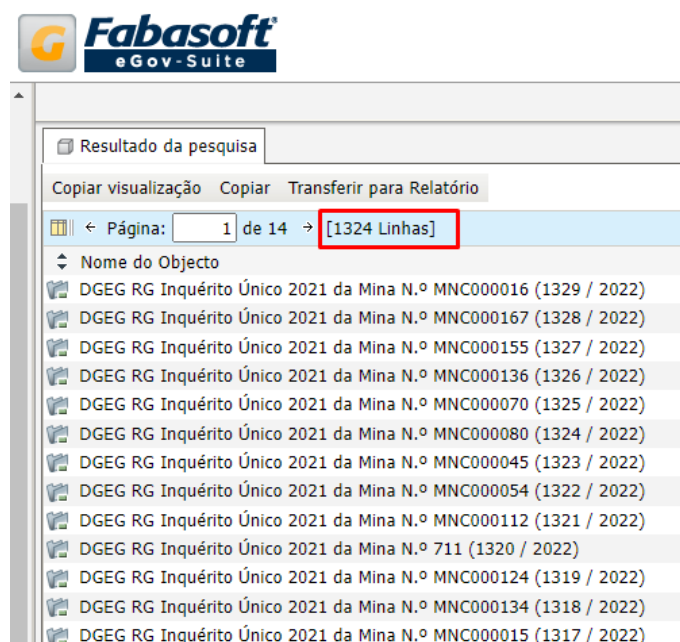


Figura 6.9: Número de Inquéritos Únicos de Minas existentes na versão anterior

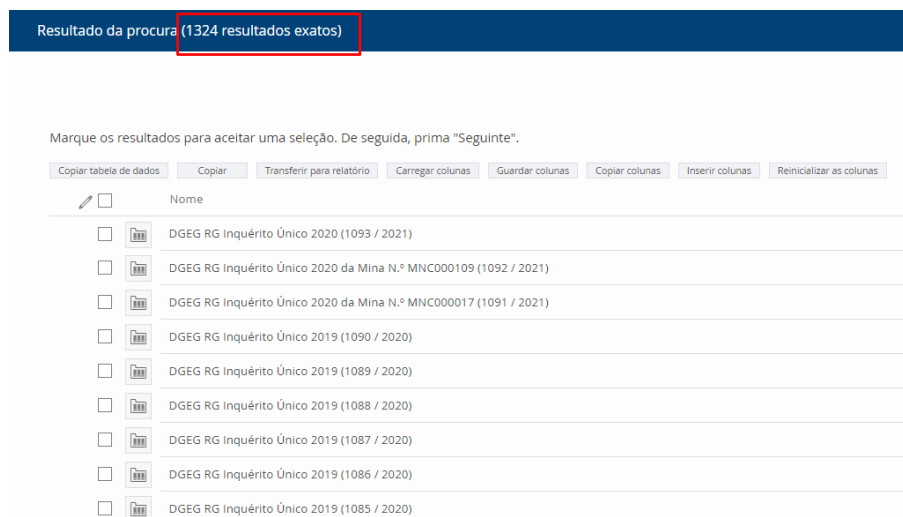


Figura 6.10: Número de Inquéritos Únicos de Minas existentes na nova versão após migração de dados

Como se pode comprovar pelas figuras 6.9 e 6.10, quando se faz uma pesquisa por inquéritos únicos referentes às Minas, podemos verificar que o número é o mesmo entre as duas versões. Com isso, sabemos que em relação ao volume dos dados migrados, está tudo de acordo com o esperado.

Encargos de Exploração			
[10 Linhas]			
	Encargo de exploração	Valor (€)	
1	Aluguer de máquinas e equipamentos	3.550.791,98	EUR
2	Encargos com ambiente	313.590,14	EUR
3	Encargos com Segurança e Saúde	852.549,24	EUR
4	Investigação e desenvolvimento	0,00	EUR
5	Manutenção e reparação de máquinas e equipamentos	27.646.229,69	EUR
6	Rendas e matagens	316.332,83	EUR
7	Seguros	1.650.742,39	EUR
8	Subcontratações (trabalhos especializados)	36.472.201,63	EUR
9	Telecomunicações	187.956,56	EUR
10	Outros	81.404.619,00	EUR
Total - Encargos de Exploração			
Valor (€)		152.395.013,46	EUR
Alterações e observações			

Figura 6.11: Dados referentes a um quadro de um Inquérito Único das Minas específico na versão anterior

Encargos de exploração

Mostrar detalhes (10)

	Encargo de exploração		Valor (€)
1	 Aluguer de máquinas e equipamentos	EUR	3 550 791,98
2	 Encargos com ambiente	EUR	313 590,14
3	 Encargos com Segurança e Saúde	EUR	852 549,24
4	 Investigação e desenvolvimento	EUR	0,00
5	 Manutenção e reparação de máquinas e equipamentos	EUR	27 646 229,69
6	 Rendas e matagens	EUR	316 332,83
7	 Seguros	EUR	1 650 742,39
8	 Subcontratações (trabalhos especializados)	EUR	36 472 201,63
9	 Telecomunicações	EUR	187 956,56
10	 Outros	EUR	81 404 619,00

Total - Encargos de exploração

Valor (€)

152 395 013,46

Alterações e observações

Cancelar

Editar

Seguinte

Cancelar Editar Seguinte

Figura 6.12: Dados referentes a um quadro de um Inquérito Único das Minas específico na nova versão

Quanto à qualidade dos dados, é verificada com a comparação de valores entre o mesmo objeto da versão anterior e na nova versão, depois de migrado. Como se pode comprovar pelas figuras 6.11 e 6.12, todos os dados estão iguais, o que comprova que os dados foram bem migrados.

Estas imagens servem apenas de exemplificação dos testes que tiveram de ser efetuados para a garantia do bom funcionamento da nova versão de *BackOffice* implementada. Claro

que teve de ser feito todo este processo a muitos mais objetos e em diferentes casos. Mas desta forma consegue entender-se o que é necessário para saber se o resultado se iguala ao que é esperado pelo cliente. Assim, podemos finalizar esta secção tendo uma noção de tudo o que foi feito após o carregamento final.

6.3 Componente Analítica

Para esta componente, apesar de não me competir diretamente a mim a testagem da mesma, posso afirmar que se os resultados forem positivos, transmitem também resultados positivos do trabalho executado na componente de *BackOffice*, isto porque todos os conjuntos de dados usados para testes finais são os dados que foram migrados da versão anterior para esta nova versão. Podemos até mesmo dizer que esta componente não seria possível de desenvolver sem esta ponte entre a componente de *BackOffice* e a *Componente Analítica*. Portanto, se os *dashboards* tiverem os resultados esperados e em conformidade com os *dashboards* da versão anterior, isso quer dizer que a migração de dados também foi um sucesso.

Assim, já podemos perceber que tipos de teste teriam de ser efetuados para comprovar o bom funcionamento desta componente. Um deles, como referido anteriormente, seria o de comparação de *dashboards* entre as duas versões referentes ao mesmo conjunto de dados, pois são feitos testes por amostragem, e verificar a sua similaridade. Esta componente vai ser disponibilizada para vários utilizadores, o que implica que têm de ser feitos testes de carga para perceber a capacidade que o sistema terá para não prejudicar o funcionamento desta componente dependente dos utilizadores que estejam conectados.

Infelizmente, e por, mais uma vez, este projeto ser realizado em ambiente empresarial, não se pôde concluir este componente e proceder aos testes de qualidade do mesmo. A equipa de BI ainda tem trabalho a desenvolver o que torna impossível a amostragem de testes relativos a esta componente e para a conclusão deste projeto.

6.4 Conclusões

O principal objetivo da avaliação do que foi desenvolvido e do projeto de uma visão geral, foi o de perceber se os resultados finais iam de encontro ao que foi requerido inicialmente e também para expor alguns problemas e inconsistências que pudessem, de alguma forma, impactar a experiência que queríamos fornecer ao utilizador.

Foi possível perceber, em cada uma das componentes, o processo de desenvolvimento de cada e o que era esperado como resultado final. Deu, de certo modo, para experienciar algumas das funcionalidade fornecidas pela própria ferramenta, *Fabasoft eGov-Suite*, como de algumas funcionalidade implementadas por mim. Volto a referir que por se tratar de um projeto com várias componentes, com a intervenção de várias equipas e em ambiente empresarial, sempre em constante contacto com o cliente, não foi possível finalizar por completo todo o projeto porposto até à data final de entrega desta dissertação. No entanto,

foram demonstrados resultados satisfatórios de todo o trabalho com a qual tive envolvimento neste mesmo projeto e que serviu para acrescentar valor ao mesmo, seguindo o rumo esperado em relação às tarefas que me propus a desenvolver no início de todo este processo.

CONCLUSÕES E TRABALHO FUTURO

O capítulo seguinte é uma visão geral das conclusões que podem ser tiradas desta dissertação, onde os objetivos são revistos e é feita uma comparação com o que realmente foi desenvolvido com sucesso. Finalmente, uma secção será dedicada para falar sobre o trabalho futuro, onde serão abordados temas que façam sentido ser falados com o cliente com a perspetiva de melhoria e acompanhamento da solução entregue.

7.1 Conclusão

O principal objetivo deste projeto foi o de atualizar/desenvolver toda a ferramenta de gestão documental e atualizar o modelo analítico que irá ser utilizado para as estatísticas dos recursos geológicos. Como já foi dado a entender anteriormente e, por se tratar de um projeto desenvolvido em contexto empresarial, houve problemas que me ultrapassaram e tornaram impossível a conclusão de algumas partes que foram faladas como objetivos a concluir no início do documento. Apesar disso, posso concluir que o componente de *BackOffice* foi concluído com sucesso e vai de encontro com todos os pontos requeridos. Faz sentido destacar este componente uma vez que foi aquele que sempre foi da minha inteira responsabilidade desenvolver e garantir uma solução viável, de acordo com o que foi acordado com o cliente. A ferramenta *Fabasoft eGov-Suite* foi atualizada para a versão mais recente, sempre tendo em conta os princípios de hierarquia definidos e utilizando o mesmo modelo conceptual da versão anterior. Ter esta visão em relação aos desenvolvimentos deste componente é de grande importância uma vez que todos os dados foram migrados da versão anterior para a nova, o que implica que a veracidade dos dados tinha de ser mantida. Os dados foram migrados para a nova versão e, apesar da grande quantidade de dados existente na versão anterior, tudo aconteceu como definido. Isso seria impossível sem seguir as regras que foram estabelecidas e sem ter uma arquitetura muito bem definida a partir do momento que se começaram a desenhar soluções para ir de encontro ao pedido final do cliente. Pessoalmente, este mesmo processo de migração foi o mais desafiante. Perceber a arquitetura desenhada na versão anterior e como tinham de ser construídas as exportações de cada uma das classes que continham objetos para ser

migrados foi um processo demorado e que requeria extrema sensibilidade. De seguida, tratar dos dados que muitas vezes continham alguns caracteres que podiam desformatar os *datasets* exportados, foi complexo e demorado. No meio de tanta informação, teve de ser feito o seu tratamento e solucionar os problemas que iam aparecendo, mantendo a veracidade do que foi exportado. Por fim, teve de ser feita uma associação das propriedades existentes na versão anterior para a versão nova, o que também requereu bastante atenção. Foi um processo de extrema importância para chegar a uma solução próxima da proposta, uma vez que outras componentes dependiam do sucesso desta migração de dados. Resumidamente, a atualização da ferramenta foi concluída, de acordo com o que foi definido. O portal de serviços não se conseguiu concluir, estando em fase de desenvolvimento, tal como a componente de BI.

7.2 Trabalho Futuro

Primeiramente, terá de ser concluído o que nos propusemos inicialmente e ficou em falta. Os componentes do portal de serviços e de BI têm de chegar à solução proposta, independentemente da minha influência nas mesmas. O portal de serviços tem de ser desenvolvido para que se possa continuar a fazer uma recolha dos dados por cada instalação associada a um recurso geológico e, posteriormente, ser usado para a estatística desse mesmo recurso. A equipa de BI tem de concluir o que foi proposto para que o acesso à estatística seja facilitado e rigoroso face à realidade. Com a atualização desse componente e fornecimento das estatísticas atualizadas, conjugado com todas as funcionalidades oferecidas com esta atualização, fica concluído tudo o que foi proposto ao cliente aquando a apresentação desta proposta.

Obviamente e, como já foi mencionado, o projeto foi construído de forma a estar preparado para ser extendido a novos componentes relacionados com a área ou com novos pedidos que façam sentido ser desenvolvidos. Isto tendo em conta todas as funcionalidades que a *Fabasoft eGov-Suite* pode oferecer. Vários processos podem ser criados e ter um fluxo de atividades associado. Se num contexto futuro quiserem adicionar mais dados estatísticos, criar novas classes e formulários ou então extender os mesmos com a adição de propriedades, o ambiente desenvolvido está completamente preparado para endereçar qualquer tipo de pedido que seja acordado com o cliente. Um dos objetivos deste projeto passa por projetar e aumentar o número de serviços digitais disponibilizados ao cliente, tornando a desmaterialização de processos uma realidade transversal a tantas áreas quanto possíveis.

E, claro está, o acompanhamento regular ao cliente terá de ser uma realidade. Não pode acontecer o que aconteceu com a versão anterior, senão todo o trabalho desenvolvido teria sido em vão. Tem de ser feita uma verificação periódica a atualizações da ferramenta e ao aparecimento de novas versões. Endereçar problemas que possam surgir, relacionados com incompatibilidade entre aplicações, problemas relacionados com a implementação dos componentes ou com o carregamento dos mesmos no ambiente de produção, problemas

de acessos, dificuldade de utilização da ferramenta ou qualquer outra dificuldade que surja. Este acompanhamento é imprescindível para evitar a obsolescência tecnológica.

Por fim, qualquer melhoria a qualquer desenvolvimento anterior que seja visível e para a qual haja uma solução viável de evolução do que já existe, será também um trabalho a ter em conta no futuro.

BIBLIOGRAFIA

- [1] *A Importância Económica e Social da Indústria Extrativa*. URL: https://www.dct.fct.unl.pt/sites/www.dct.fct.unl.pt/files/exploracao_de_minerios_em_portugal_0.pdf (ver p. 9).
- [2] *An Introduction to Fabasoft app.ducx*. URL: <https://manualzz.com/doc/6982125/--fabasoft-app.ducx> (ver p. 15).
- [3] *EdocLink*. URL: <https://linkconsulting.com/what-we-do/products/edoclink-2/> (ver p. 18).
- [4] Fabasoft. *Software Product Information*. Fabasoft, 2022 (ver p. 14).
- [5] *GFIDocs*. URL: <https://simpleflow.pt/solucao/funcionalidades/> (ver p. 21).
- [6] *Pedreiras*. URL: <https://www.asae.gov.pt/newsletter2/asaenews-n-98-junho-2016/pedreiras.aspx> (ver p. 8).
- [7] *Pedreiras: a riqueza mineral de Portugal*. URL: <https://www.pai.pt/artigos/126-pedreiras-a-riqueza-mineral-de-portugal> (ver p. 8).
- [8] *Recurso Geológico*. URL: <https://www.apiam.pt/conteudo/Legisla%C3%A7%C3%A3o/-/49> (ver p. 9).
- [9] *SmartDocs*. URL: <https://www.fujitsu.com/pt/solutions/smartdocs/> (ver p. 19).
- [10] *The use case oriented development environment for your business applications*. URL: <https://www.fabasoft.com/en/products/fabasoft-appducx> (ver p. 14).

