



Universidade do Minho

Escola de Engenharia

MESTRADO EM ENGENHARIA INFORMÁTICA

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

UNIDADE CURRICULAR DE

PROJETO EM ENGENHARIA INFORMÁTICA

ANO LETIVO DE 2020/2021

Relatório Técnico

InFin

“Automating enterprises financial data management and analytics”

ID Documento	RT-20201230-PEI2020
Versão	1.0
Acesso	Restrito
Data de emissão	18 de dezembro de 2020
Autores	Grupo InFinions Bárbara Andreia Cardoso Ferreira, a80453, MIEI Bruno Manuel Chaves Martins, a80410, MIEI Catarina Araújo Machado, a81047, MIEI Eduardo Jorge Lima Pinto Barbosa, a83344, MIEI Filipe Pimenta Oliveira Monteiro, a80229, MIEI Jéssica Andreia Fernandes Lemos, a82061, MIEI João Manuel Pós de Mina Grenhas, pg33874, MEI João Pedro Machado Vilaça, a82339, MIEI Márcio Alexandre Mota Sousa, a82400, MIEI
Destinatário	PEI2020

Data de receção	
Responsável	
Avaliação	
Observações	

Relatório Técnico

InFin

“Automating enterprises financial data management and analytics”

Grupo InFinions

Resumo

O tema deste projeto nasce no âmbito da UC de PEI, ano letivo de 2020/2021, onde se lançou o desafio de levantar e implementar uma ideia suscetível de resultar num produto que apoiasse um modelo de negócio. A ideia subjacente a este projeto é uma **aplicação de apoio ao controlo financeiro de faturas, com foco nos fornecedores.**

Este documento constitui o relatório intermédio do projeto, no qual se inclui o levantamento e análise de requisitos, análise formal, gestão de projeto e desenho da solução.

Área de Aplicação: Desenho e implementação de aplicações web para a área financeira.

Palavras-Chave: Aplicações web. Aplicações financeiras. Engenharia de Requisitos. *Volere Requirements Specification Template*. Métodos formais. Gestão de projeto. Arquiteturas de software.

Índice

Resumo	2
Índice	2
Introdução	8
Contextualização, Motivação e Objetivos	8
Contextualização e Motivação	8
Objetivos	8
Estrutura deste Documento	8
Partes Interessadas	9
Convenções e Definições	9
Convenções	9
Glossário	9
administrador de conta	9
administrador de plataforma	9
categoria	10
conciliação bancária	10
conta de plataforma	10
data de vencimento	10
empresa/organização/contribuinte fiscal	10
etiqueta	10
grande empresa	10
média empresa	10
microempresa	11
orçamento	11
pequena empresa	11
utilizador	11
Siglas, acrónimos e abreviaturas	11
Factos e Assunções Relevantes	13
Factos	13
Digitalização	13
Faturas	13
Aplicações informáticas de faturação/ERP	13
SAFT	13
Portal da AT	13
Decreto-Lei n.º 372/2007, de 6 de novembro	14

Assunções Relevantes	14
Opções e Restrições	14
Entrega	14
Ambiente de Trabalho	14
Levantamento e Análise de Requisitos	15
Análise de Domínio	15
Âmbito do Trabalho	15
Modelo de Dados	15
Modelo de Domínio	15
Dicionário de Dados	16
Categoria	16
Conta Bancária	16
Documento	17
Empresa	17
Etiqueta	17
Fatura (Despesas)	17
Fatura (Receitas)	17
Movimento Bancário	17
Orçamento	18
Utilizador	18
Âmbito do Produto	18
Fronteira do Sistema	18
Casos de Uso do Produto	19
Lista de Entregáveis (Deliverables)	19
Aplicação web	19
Scraper de Faturas	19
Pacote de Análise (Analysis Kit)	19
Priorização	19
Requisitos Funcionais e de Dados	20
Requisitos Funcionais	20
#1 O sistema deve permitir o registo manual de faturas	20
#2 O sistema deve permitir a recolha automatizada das faturas no portal das finanças	20
#3 O sistema deve permitir a importação via CSV de faturas	21
#4 O sistema deve permitir a categorização das faturas	21
#5 O sistema deve permitir colocar etiquetas nas faturas	21
#6 O sistema deve permitir o registo de empresas	22

#7 As faturas, categorias e etiquetas pertencem ao perfil de uma empresa	22
#8 Uma empresa deve poder adicionar utilizadores	23
#9 Os utilizadores devem poder editar os dados de perfil da sua empresa	23
#10 O sistema deverá sugerir uma categoria quando uma fatura é inserida	23
#11 Cada categoria pode ter um orçamento associado	24
#14 O sistema deve disponibilizar a previsão da execução orçamental baseada na faturação atual	24
#15 O sistema deve gerar relatórios financeiros	24
#16 O sistema deverá emitir faturas	25
#17 O sistema deverá permitir a associação de faturas a fornecedores	25
#18 O sistema deve permitir a criação de faturas via scan	26
#21 O sistema deve gerar gráficos estatísticos	26
#22 O sistema deve permitir a definição de orçamentos	26
#23 O sistema deverá permitir registar receitas interativamente	27
#24 O sistema deverá permitir importar o SAF-T de faturação	27
#25 O sistema deverá permitir armazenar faturas no formato PDF	27
#26 O sistema deverá permitir importar faturas chegadas via email	28
#27 O sistema deverá permitir marcar faturas como pendentes ou pagas	28
#28 O sistema deverá permitir importar movimentos bancários via API	29
#29 O sistema deverá permitir importar movimentos bancários via CSV	29
#30 O sistema deverá permitir a associação de faturas a movimentos bancários	29
#31 O sistema deverá permitir associar movimentos bancários a operações financeiras da empresa	30
#32 O sistema deverá permitir fazer o pagamento de faturas	30
#33 O sistema deverá permitir simular a evolução financeira da empresa	31
#36 O sistema deve permitir predefinir categorias	31
#37 O sistema deve permitir predefinir etiquetas	31
Requisitos de Dados	32
#12 Cada fatura apenas pode estar associada a uma e uma só categoria	32
Requisitos Não Funcionais	32
Aparência	32
Usabilidade	32
#34 O sistema deverá ser fácil de usar por utilizadores com pouca cultura financeira	32
Desempenho	33

#39 O sistema deverá acautelar a performance e a disponibilidade ao efetuar cálculos complexos	33
Operacionais	33
#19 O sistema deverá disponibilizar uma aplicação web para interação com os seus utilizadores	34
#38 O sistema deverá separar a camada de acesso à BD numa API de dados	34
Manutenção, Portabilidade e Suporte	34
Segurança	34
#35 Toda a interação com o sistema requer autenticação prévia do utilizador	34
#40 Toda a informação associada a cada conta de plataforma deve estar encapsulada	35
Culturais e Políticos	35
Legais	35
#13 O sistema deverá suportar o padrão SAF-T contabilístico	35
#20 O sistema deverá respeitar o RGPD	36
Análise Formal	36
Metamodelo	37
Trace	37
Checks	38
Modelo do Processo	40
Gestão	40
Processo de desenvolvimento	40
Matriz de Rastreabilidade	41
Casos de Uso	41
Matriz de Rastreabilidade	41
Gestão de Recursos Humanos	42
Regras de base	42
OBS (Organizational Breakdown Structure)	42
Árvore da organização	42
Membros da equipa	43
Bárbara Andreia Cardoso Ferreira, a80453, MIEI	43
Bruno Manuel Chaves Martins, a80410, MIEI	43
Catarina Araújo Machado, a81047, MIEI	43
Eduardo Jorge Lima Pinto Barbosa, a83344, MIEI	43
Filipe Pimenta Oliveira Monteiro, a80229, MIEI	43
Jéssica Andreia Fernandes Lemos, a82061, MIEI	44

João Manuel Pós de Mina Grenhas, pg33874, MEI	44
João Pedro Machado Vilaça, a82339, MIEI	44
Márcio Alexandre Mota Sousa, a82400, MIEI	44
RAM (Responsibility Assignment Matrix)	44
Estratégia de Solução	45
Arquitetura Geral	45
Dados	45
Acesso do utilizador	46
Tecnologias	46
Diagramas de Blocos (Building Block View)	46
Diagrama de Componentes	46
Diagrama de Classes	48
Interface com o Utilizador	48
Lógica de Controlo	49
Instalação e Manutenção	49
Anexos	49
Modelo de Qualidade da Norma ISO 9126	50
Técnicas de Levantamento de Requisitos	51
O Cartão de Volere	51
Entrevistas	52
Entrevista a Laura Esteves (COO da Subvisual)	52
Entrevista a Francisco Maia (Keyruptive)	55
Entrevista a Adriano Martins (Explore.dev)	58
Observação	59
Personas	59
Introspeção	59
Detalhes da Análise Formal	60
Totalidade das assinaturas	60
Totalidade dos invariantes	61
Bibliografia	61

1. Introdução

1.1. Contextualização, Motivação e Objetivos

Nesta secção: Contexto em que nasce e o que motivou o projeto. O que se pretende alcançar com o projeto.

Contextualização e Motivação

No âmbito da participação de alguns membros do grupo em associações, constatou-se da dificuldade associada à gestão financeira de uma organização, por ser baseada em processos que se verifica estarem frequentemente associados à gestão manual de folhas de cálculo (cf. Excel), consumindo tempo e possibilitando o erro humano.

Isto associa-se, principalmente, a **faturas de fornecedores**, cujo pagamento é necessário controlar, em conjunto com a conta bancária. Frequentemente, ocorre dificuldade em encontrar as faturas materiais, para verificação.

Também há quem valorize o **pagamento de salários**, **controlo do pagamento de faturas emitidas**, o **cash-flow**, o cumprimento de **limites orçamentais** por departamento, além da **conciliação bancária** associada.

Analisando os serviços da AT, constatou-se ser possível obter automaticamente determinada informação sobre faturas de fornecedores, mensalmente carregadas pelos emissores no sistema da AT. Há facilidades tecnológicas que permitem, não só a digitalização, mas também o reconhecimento de caracteres.

Objetivos

Este projeto pretende chegar a um sistema que, automatizando processos, facilite significativamente a gestão financeira duma pequena organização, especialmente no que diz respeito às contas a pagar (referentes a faturas de fornecedores), sem esquecer que há quem valorize controlo financeiro adicional. Nomeadamente, pretende-se substituir o uso sistemático de folhas de cálculo para controlo financeiro.

1.2. Estrutura deste Documento

Este documento divide-se nos seguintes capítulos principais:

- **Levantamento e Análise de Requisitos**, onde se trata da análise de domínio e listagem pormenorizada dos requisitos funcionais e não funcionais, incluindo priorização, plasmados em cartões de Volere;
- **Modelo do Processo**, onde se discorre sobre o processo de desenvolvimento e sua gestão;
- **Matriz de Rastreabilidade**, um auxílio ao processo de desenvolvimento;
- **Gestão de Recursos Humanos**, com a descrição da equipa;
- **Estratégia de Solução**, descrevendo a arquitetura e opções de solução do problema; e
- Anexos.

1.3. Partes Interessadas

Tabela de partes interessadas no sistema:

Nome	Expectativas
Equipa de responsáveis da UC de PEI ; supervisores Profs. José Carlos Ramalho e Paulo Afonso	Um produto minimamente viável, comercializável. Um modelo de negócio viável associado.
Gestor financeiro de uma micro, pequena ou média empresa	Um sistema com informação financeira fiável. Poupar tempo e dinheiro com procedimentos administrativos e financeiros, nomeadamente precisa de controlar: contas a pagar/receber, níveis de despesa, e conciliação da conta bancária.
Autoridade Tributária e Aduaneira	Um sistema que não abuse dos serviços web por si disponibilizados e que efetue uma utilização idónea da informação.
Destinatários das faturas	Um sistema que não abuse dos seus direitos legais, em particular na utilização dos dados pessoais.

1.4. Convenções e Definições

Incluem-se nesta secção as convenções escolhidas e definição de termos.

1.4.1 Convenções

(Convenções estabelecidas no âmbito do projeto.)

1.4.2 Glossário

Inclui-se nesta secção a definição de alguns termos específicos do âmbito.

administrador de conta

Utilizador, duma conta de plataforma, com **poderes especiais** sobre essa conta -- nomeadamente poderá fazer CRUD dos utilizadores dessa conta (e apenas dessa conta). Terá de existir sempre pelo menos um administrador de conta.

administrador de plataforma

Utilizador com **poderes especiais no sistema**, nomeadamente CRUD de contas de plataforma.

categoria

Modo de classificar, enquadrar, uma fatura, uma receita ou uma despesa, tipicamente a partir de um conjunto pré-determinado em função do domínio de aplicação. Pode ser vista como o sector de uma empresa (é personalizável pela empresa).

conciliação bancária

Processo administrativo ou contabilístico, de comparação de movimentos financeiros de uma empresa contra valores, saldos e destinatários num extrato bancário (ou de conta-corrente). Em cada iteração, é suposto saber o que já foi conciliado e o que falta conciliar.

conta de plataforma

No âmbito do sistema, uma conta fica associada a uma empresa/organização, empresário em nome individual ou genericamente qualquer contribuinte fiscal. Sob essa conta será agregada toda a informação de sistema que lhe diga respeito. A conta de plataforma terá sempre um utilizador com nível de Administrador de Conta.

data de vencimento

Data limite de pagamento.

empresa/organização/contribuinte fiscal

Entidade **fiscal** e jurídica, identificada pelo NIF, com um domicílio fiscal (sede social), podendo ser uma pessoa coletiva ou em nome individual. No âmbito do sistema, identifica-se como uma **conta de plataforma**.

etiqueta

Conjunto versátil de palavras-chave para rápida classificação de faturas, de um modo mais personalizado que por via de uma categoria. Facilita a identificação/pesquisa e agrupamento.

grande empresa

(Instituto Nacional de Estatística, seguindo uma Recomendação da Comissão Europeia de 6 de maio de 2003.)

- Empresas com 250 ou mais pessoas ao serviço; ou
- empresas com volume de negócios superior a 50 milhões de euros e ativo líquido superior a 43 milhões de euros; senão
- PME, ou seja, pequenas e médias empresas.

média empresa

- Empresa que emprega menos de 250 pessoas, e com
- volume de negócios anual que não excede 50 milhões de euros ou balanço total anual não excede 43 milhões de euros, e que
- não está classificada como micro ou pequena empresa.

microempresa

Empresa que emprega menos de 10 pessoas e cujo volume de negócios anual ou balanço total anual não excede 2 milhões de euros.

orçamento

Limites de verba colocados pelo cliente para determinada categoria, de forma a receber avisos quando estes são ultrapassados.

pequena empresa

- Empresa que emprega menos de 50 pessoas, e com
- volume de negócios anual ou balanço total anual que não excede 10 milhões de euros, e que
- não está classificada como uma microempresa.

utilizador

Utilizador corrente do sistema, associado a uma única conta de plataforma, em nome da qual age.

1.4.3 Siglas, acrónimos e abreviaturas

API	<i>Application Programming Interface</i>
AT	Autoridade Tributária e Aduaneira (de Portugal)
BD	Base de Dados, o mesmo que DB
BUC	<i>Business Use Case</i>
Cf.	Confrontar, ver também
CRUD	<i>Create, Read, Update and Delete</i>
CSS	<i>Cascading Style Sheets</i>
DB	<i>Database</i> , o mesmo que BD
Endereço IP	Endereço de Protocolo da Internet, <i>Internet Protocol address (IP address)</i>
ERP	ERP é o acrónimo do termo inglês Enterprise Resource Planning (Planeamento de Recursos Empresariais). “O ERP é um software de gestão de processos de negócio que gere e integra as atividades de finanças, cadeia de fornecimento, operações, relatórios, fabrico e recursos humanos de uma empresa.” (in https://dynamics.microsoft.com/pt-pt/erp/what-is-erp/ , 01/11/2020)
Ex.	Exemplo
HTML	<i>HyperText Markup Language</i>

HTTP	<i>Hypertext Transfer Protocol</i> , Protocolo de Transferência de Hipertexto
MEI	Mestrado em Engenharia Informática, UMinho
MIEI, MiEI	Mestrado Integrado em Engenharia Informática, UMinho
MVC	<i>Model View Controller</i> , uma arquitetura de software
NIF	Número de Identificação Fiscal. Tem formato diferente conforme os países de domicílio (pode conter letras em certos países). Para países estrangeiros, é norma prefixar com um código definido para o país. Exemplo: ES B999(...) denota Espanha, NIF B999(...)
OCR	<i>Optical Character Recognition</i> , reconhecimento ótico de caracteres.
OOP	<i>Object Oriented Programming</i> , o mesmo que POO
PEI	Projeto em Engenharia Informática, uma UC de MEI e MiEI, UMinho.
PF	Programação Funcional
PLN	Processamento de Linguagem Natural
PME	Micro, pequena ou média empresa; Micro, pequenas e médias empresas.
POO	Programação Orientada a Objetos, o mesmo que OOP
PUC	<i>Product Use Case</i>
REST	<i>Representational State Transfer</i>
RGPD	Regulamento Geral de Proteção de Dados – cf. <i>General Data Protection Regulation – GDPR</i> .
SAF-T ou SAFT	SAF-T (PT): Standard Audit File for Tax purposes Portuguese edition) Ficheiro Normalizado de Exportação de Dados criado pela Portaria nº 321-A/2007, de 26/03. SAF-T (PT) “é um ficheiro normalizado (em formato XML) com o objetivo de permitir uma exportação fácil, e em qualquer altura, de um conjunto predefinido de registos contabilísticos, de faturação, de documentos de transporte e recibos emitidos , num formato legível e comum, independentemente do programa utilizado, sem afetar a estrutura interna da base de dados do programa ou a sua funcionalidade.” (in https://info.portaldasfinancas.gov.pt/pt/apoio_contribuinte/SAFT_PT/Paginas/news-saf-t-pt.aspx , 01/11/2020)
UC	Unidade Curricular
UI	<i>User Interface</i> , Interface com o utilizador
UM	Universidade do Minho

UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>

1.5. Factos e Assunções Relevantes

1.5.1 Factos

Digitalização

Existem tecnologias maduras disponíveis para digitalização e OCR.

Faturas

Existem diversos documentos financeiros de compra e venda de bens e serviços (fatura + recibo, fatura simplificada, fatura-recibo), e documentos de ajuste (nota de débito, nota de crédito). **Para simplificar, será comum referir-nos aos diversos documentos como faturas.** Entretanto, discrimine-se:

- **Fatura + recibo:** a fatura emite-se e é entregue ao destinatário, que assim toma conhecimento da dívida, além de prazo e meios de pagamento; quando o pagamento chega, é emitido o respetivo **recibo**, entregue ao mesmo destinatário.
- **Fatura simplificada e fatura-recibo:** fatura e recibo no mesmo documento; é emitida quando existe pagamento a pronto.
- **Nota de débito:** documento de ajuste semelhante à fatura em termos de liquidação e pagamento.
- **Nota de crédito:** documento de ajuste a favor do cliente, por exemplo por motivo de devolução; este documento pode ser usado para liquidar ou anular faturas.

Aplicações informáticas de faturação/ERP

Existem muitos fornecedores de aplicações de faturação no mercado, frequentemente referenciadas pelo acrónimo ERP. “A maior parte das empresas possuem um sistema de finanças e operações, mas a maior parte do software existente não vai mais além dos processos de negócio diários, não sendo capaz de auxiliar o crescimento futuro do negócio.” (in <https://dynamics.microsoft.com/pt-pt/erp/what-is-erp/>, 01/11/2020)

SAFT

As aplicações de faturação certificadas pela Autoridade Tributária e Aduaneira (AT) têm a possibilidade de gerar o SAFT mensal (bem como outro mais completo), depois carregado no portal da AT. Por esta via, a AT regista todo o movimento de faturas. O SAFT não exporta compras.

Portal da AT

Possibilita o carregamento de SAFT, e a consulta de faturas de compra. Possibilita a exportação dessa informação.

Decreto-Lei n.º 372/2007, de 6 de novembro

1 – A categoria das micro, pequenas e médias empresas (PME) é constituída por empresas que empregam menos de 250 pessoas e cujo volume de negócios anual não excede 50 milhões de euros ou cujo balanço total anual não excede 43 milhões de euros.

2 – Na categoria das PME, uma pequena empresa é definida como uma empresa que emprega menos de 50 pessoas e cujo volume de negócios anual ou balanço total anual não excede 10 milhões de euros.

3 – Na categoria das PME, uma micro empresa é definida como uma empresa que emprega menos de 10 pessoas e cujo volume de negócios anual ou balanço total anual não excede 2 milhões de euros.

1.5.2 Assunções Relevantes

A AT continuará a disponibilizar a sua plataforma de e-fatura e serviços associados.

1.6. Opções e Restrições

(Lembra-se os redatores desta secção que pode acontecer que certas restrições devam ser definidas como requisitos não funcionais...)

Entrega

O produto tem de estar pronto até 22 de janeiro de 2021, relatório inclusivé.

Deverá existir um endereço web para testar a aplicação, motivado por trabalho à distância dos diversos intervenientes (membros da equipa e professores), nomeadamente em referência a aulas, apresentações/demonstrações e avaliação à distância.

A solução e a documentação do projeto devem ser entregues via GitHub conforme acordado com o Professor JCR.

Documentos a entregar: Documento de Requisitos, Relatório Intermédio (draft da versão final), Relatório Final.

Ambiente de Trabalho

Cada elemento da equipa trabalhou no projeto, em grande medida, à distância, com os seus próprios meios ou nas instalações da UMinho.

Como *framework tool* para implementação da solução, escolheu-se a **Phoenix Framework**, baseada na linguagem Elixir, por facilidade e familiaridade com este ambiente.

2. Levantamento e Análise de Requisitos

Muitas secções deste capítulo estão em relação com o modelo Volere. Não há, no entanto, uma correspondência absoluta com o referido modelo.

2.1. Análise de Domínio

2.1.1 Âmbito do Trabalho

A situação atual do trabalho foi analisada contra a experiência de alguns elementos em gestão financeira de associações, e contra as entrevistas realizadas, transcritas em anexo.

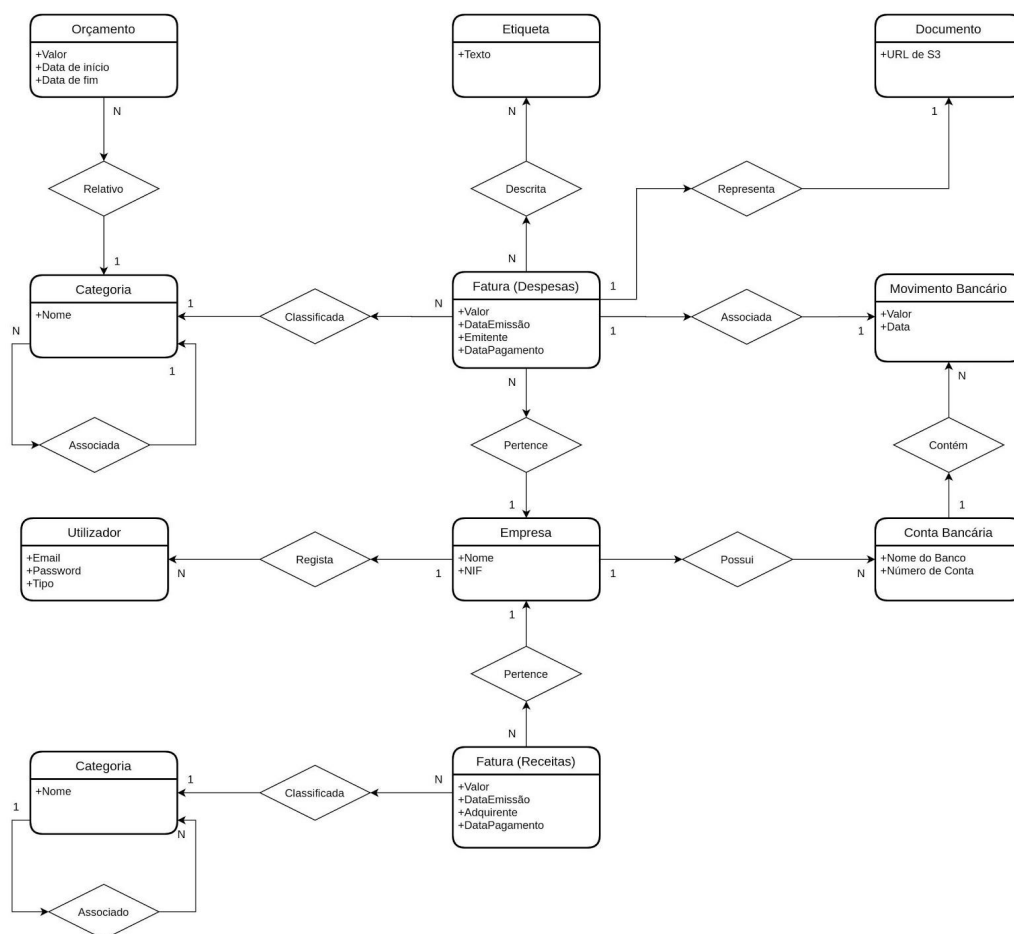
Verifica-se que as pessoas gastam algum tempo a gerir, manualmente, folhas de Excel para controlar orçamentos, conciliação bancária, recebimentos e pagamentos. Esta gestão pode ser colaborativa, pelo que a quantidade de tempo gasta nem sempre é rigorosamente medida.

Há soluções no mercado, quer para faturação, quer para gestão financeira, mas rapidamente se revelam complexas na sua aprendizagem e configuração.

Por vezes, um utilizador precisa de consultar uma fatura dum fornecedor, mas esta já foi entregue ao contabilista.

2.1.2 Modelo de Dados

Modelo de Domínio



(Nota: O modelo de domínio nem sempre é o modelo estritamente adotado pela solução, já que analisa o domínio em foco. No entanto, acabou por ser o modelo adotado neste projeto.)

Dicionário de Dados

Convenções -- tipos de dados abstratos:

- **C(n)**: Cadeia/string de caracteres de tamanho máximo n.
- **N(t, d)**: Numérico de tamanho total t, nº de casas decimais d; t inclui o ponto decimal e o sinal *menos*. Ex.: N(5,2) => de -9,99 a 99,99.
- **D**: Data de calendário.
- **L**: Lógico/booleano: verdadeiro ou falso.
- **M**: Memo (cadeia de caracteres de tamanho variável).

Categoria

Classificações económicas.

Atributos:

- ID: chave.
- Nome: descrição.

Conta Bancária

Contas abertas em instituições bancárias.

Atributos:

- ID: chave.
- Nome do Banco: banco de domiciliação da conta.
- Número de Conta: nº de conta fornecido pelo banco.

Documento

Ficheiros de imagem de documentos. Ex.: PDF, JPG, PNG, ...

Atributos:

- ID: chave.
- URL; M.

Empresa

Contas da plataforma.

Atributos:

- ID: chave.
- Nome: Nome da empresa
- NIF; C(15): N° de identificação fiscal PT.

Etiqueta

Atributos:

- ID: chave.
- Nome; M: descrição.

Fatura (Despesas)

Atributos:

- ID: chave.
- Emitente: NIF do emitente da fatura.
- Valor: valor total da fatura.
- DataEmissão, D: data de emissão da fatura.
- DataPagamento, D: data em que foi paga a fatura.

Fatura (Receitas)

Atributos:

- ID: chave.
- Adquirente: NIF do destinatário da fatura.
- Valor: valor total da fatura.
- DataEmissão, D: data de emissão da fatura.
- DataPagamento, D: data em que foi paga a fatura.

Movimento Bancário

Atributo multivalor de Conta Bancária.

Atributos:

- ID: chave.
- Valor; N: valor, positivo ou negativo, do movimento.
- Data; D: data do movimento.

- Data Valor; D: data valor do movimento.

Orçamento

Atributos:

- ID: chave.
- Valor: valor orçamentado.
- Data de início: início de vigência do orçamento.
- Data de fim: fim de vigência do orçamento.

Utilizador

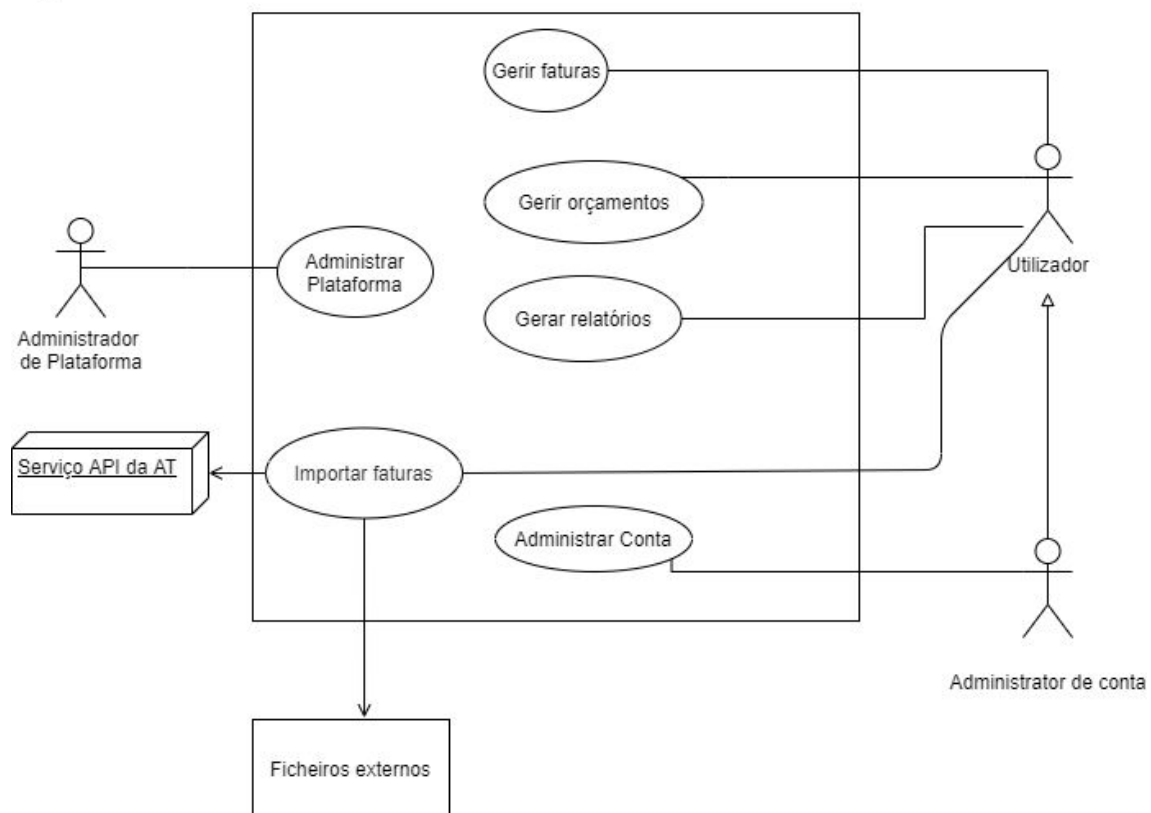
Atributos:

- ID: chave.
- Email: endereço de correio eletrónico.
- Password: palavra passe secreta.
- Tipo: tipo de utilizador na plataforma.

2.2. Âmbito do Produto

2.2.1 Fronteira do Sistema

Diagrama UML de casos de uso da fronteira do sistema



2.2.2 Casos de Uso do Produto

Atores:

- Administrador de plataforma
- Administrador de conta
- Utilizador

Casos de uso:

1. **Administrar Plataforma:** operação de administração com plenos poderes.
2. **Administrar Conta:** operação de administração (de utilizadores) e gestão (de preferências) da conta de plataforma.
3. **Importar faturas:** operação automática, a pedido, que faz ingestão de faturas, em lote ou não.
4. **Gerir faturas:** operações CRUD de faturas; gestão financeira.
5. **Gerir orçamentos:** operações CRUD de orçamentos.
6. Gerar relatórios: operações de geração de relatórios diversos baseados na informação da BD, incluindo análise automática e gráficos.

2.3. Lista de Entregáveis (*Deliverables*)

2.3.1 Aplicação web

Este módulo tem por objetivo servir de interface ao utilizador, tendo como responsabilidade gerir informação (sobretudo financeira) de faturas, tanto emitidas como recebidas. Privilegia a obtenção automática de dados, mas também permite a sua introdução manual.

2.3.2 Scraper de Faturas

Módulo de obtenção de informação do portal da AT, em formato estruturado.

2.3.3 Pacote de Análise (*Analysis Kit*)

Este módulo disponibilizará inteligência sobre os dados das faturas e outra informação guardada pelo sistema, fornecendo resultados como gráficos, relatórios e análise financeira.

2.4. Priorização

Por se considerar um padrão bem aceite e intuitivo, optou-se pela **priorização de requisitos** segundo a técnica **MoSCoW**:

- **Must** – Tem de ter (requisitos que **têm** de ser considerados);
- **Should** – Deveria ter (requisitos que **deveriam** ser considerados);
- **Could** – Poderia ter (requisitos **desejáveis**, mas não necessários);
- **Would/Won't** – Interessante ter (requisitos que **poderão, ou não**, ser considerados, no futuro).

2.5. Requisitos Funcionais e de Dados

(Nota: Confrontar informação de apoio no anexo Técnicas de Levantamento de Requisitos.)

Para apresentação dos requisitos optou-se pelo cartão Volere (descrito nos anexos). A numeração dos requisitos é globalmente **sequencial** (seja um requisito funcional ou não funcional).

O item *Description* no cartão é decalcado do título da secção respetiva. Esta pequena redundância destina-se a ter esse item no índice, para mais fácil consulta deste relatório.

2.5.1 Requisitos Funcionais

#1 O sistema deve permitir o registo manual de faturas

Volere requirement shell		
Requirement #: 1	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir o registo manual de faturas.		
Rationale: A obtenção automática pode não ser momentaneamente possível ou pretendida, ou por algum motivo o utilizador desejar introduzir faturas.		
Originator: Brainstorming		
Fit Criterion: Na área de faturas deverá existir uma opção para registo interativo da informação duma fatura, com pouca informação obrigatória.		
Customer Satisfaction: 5	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

#2 O sistema deve permitir a recolha automatizada das faturas no portal das finanças

Volere requirement shell		
Requirement #: 2	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir a recolha automatizada das faturas no portal das finanças.		
Rationale: A obtenção automática irá alimentar o sistema com a boa parte da informação pretendida.		
Originator: Brainstorming		
Fit Criterion: Depois de indicar um intervalo de datas, o sistema irá recorrer ao serviço da AT para obter as faturas (de fornecedores).		
Customer Satisfaction: 5	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts:
Supporting Materials:		
History: Criado a 10/2020.		

#3 O sistema deve permitir a importação via CSV de faturas

Volere requirement shell		
Requirement #: 3	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir a importação via CSV de faturas.		
Rationale: O portal da AT permite exportar informação de faturas no formato CSV. Poderá ser também um modo de permitir a ingestão de faturas, com origem noutros sistemas.		
Originator: Brainstorming		
Fit Criterion: Depois de selecionar a opção e indicar o ficheiro CSV, as faturas serão inseridas no sistema, associadas à conta de plataforma. Cf. categorização e etiquetagem de faturas.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts:
Supporting Materials:		
History: Criado a 10/2020.		

#4 O sistema deve permitir a categorização das faturas

Volere requirement shell		
Requirement #: 4	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir a categorização das faturas.		
Rationale: Permitirá associar uma categoria a cada fatura, que identifique, por exemplo, o sector a que respeita. Assim, as faturas poderão ser agrupadas para fácil organização e análise.		
Originator: Brainstorming, Entrevistas		
Fit Criterion: Na área de faturas deve ser possível atribuir uma categoria a uma ou mais faturas. No momento de inserção duma fatura também deverá ser possível esta associação. Ao consultar uma fatura, deve ser visível a categoria eventualmente associada. A categoria deve ser tida em conta na pesquisa e agregação de informação que tenha as categorias como foco.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

#5 O sistema deve permitir colocar etiquetas nas faturas

Volere requirement shell		
Requirement #: 5	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir colocar etiquetas nas faturas.		
Rationale: Com a aglomeração de faturas, no momento de pesquisar faturas,, a existência de etiquetas nestas irá facilitar a sua procura.		
Originator: Brainstorming		
Fit Criterion: Na área de faturas deve ser possível atribuir várias etiquetas a uma ou		

mais faturas. No momento de inserção duma fatura também deverá ser possível esta associação. Ao consultar uma fatura, devem ser visíveis as etiquetas eventualmente associadas. A etiqueta deve ser tida em conta na pesquisa e agregação de informação que tenha as etiquetas como foco.

Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Could	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

#6 O sistema deve permitir o registo de empresas

Volere requirement shell		
Requirement #: 6	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir o registo de empresas.		
Rationale: Só existindo no sistema uma conta de empresa esta poderá usufruir das funcionalidades da plataforma. A empresa identifica-se como a conta que irá agregar todo um conjunto de informação, financeira ou não, além de agregar os respetivos utilizadores. A informação da empresa só poderá ser gerida pelos seus utilizadores.		
Originator: Brainstorming		
Fit Criterion: Deverá existir uma opção para o administrador da plataforma inserir uma conta de empresa, à qual será de imediato atribuído um utilizador com nível de Administrador da conta, que também poderá inserir os respetivos utilizadores.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

#7 As faturas, categorias e etiquetas pertencem ao perfil de uma empresa

Volere requirement shell		
Requirement #: 7	Requirement Type:	Event/BUC/PUC #:
Description: As faturas, categorias e etiquetas pertencem ao perfil de uma empresa.		
Rationale: Para que os utilizadores de uma dada empresa tenham o acesso às faturas, categorias e etiquetas da sua empresa, estas são integradas no perfil da mesma.		
Originator: Brainstorming		
Fit Criterion: Na área reservada da empresa existirá uma opção para aceder às faturas, categorias e etiquetas dessa empresa.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020. Descontinuado a 07/11/2020, tendo originado dois novos requisitos, 36 e 37.		

#8 Uma empresa deve poder adicionar utilizadores

Volere requirement shell		
Requirement #: 8	Requirement Type: Funcional	Event/BUC/PUC #:
Description: Uma empresa deve poder adicionar utilizadores.		
Rationale: Os utilizadores duma empresa serão os utilizadores exclusivos da conta que ela constitui, servindo para regular o acesso e identificar operações por si realizadas.		
Originator: Brainstorming		
Fit Criterion: Na área reservada da empresa, existirá uma opção para CRUD de utilizadores dessa empresa. Só um utilizador com nível de Administrador da Conta deve ter acesso.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts:
Supporting Materials:		
History: Criado a 10/2020.		

#9 Os utilizadores devem poder editar os dados de perfil da sua empresa

Volere requirement shell		
Requirement #: 9	Requirement Type: Funcional	Event/BUC/PUC #:
Description: Os utilizadores devem poder editar os dados de perfil da sua empresa.		
Rationale: Os dados de configuração da conta de plataforma têm sempre alguma dose de dinamismo, pelo que a plataforma deve possibilitar a edição desses dados.		
Originator: Brainstorming		
Fit Criterion: Acedendo à zona reservada do perfil da empresa, será possível editar configurações, dependendo do grau de permissões do utilizador.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts:
Supporting Materials:		
History: Criado a 10/2020.		

#10 O sistema deverá sugerir uma categoria quando uma fatura é inserida

Volere requirement shell		
Requirement #: 10	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá sugerir uma categoria quando uma fatura é inserida.		
Rationale: Na existência de múltiplas categorias, poderá ser um processo demorado e tedioso a categorização de faturas. A sugestão facilitará o preenchimento.		
Originator: Brainstorming		
Fit Criterion: Quando é adicionada uma nova fatura ao sistema, este deve sugerir uma categoria para classificar a fatura em questão.		

Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Could	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

#11 Cada categoria pode ter um orçamento associado

Volere requirement shell		
Requirement #: 11	Requirement Type: Funcional	Event/BUC/PUC #:
Description: Cada categoria pode ter um orçamento associado.		
Rationale: As empresas gostam de ter controlo sobre as suas despesas, por isso a existência de limites ajuda a controlar essas despesas.		
Originator: Brainstorming		
Fit Criterion: Na sua conta, a empresa poderá formalizar limites para cada sector em que possui faturas, podendo alterar quando desejar.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020. Descontinuado a 08/11/2020 em favor do Req#22, mais concreto.		

#14 O sistema deve disponibilizar a previsão da execução orçamental baseada na faturação atual

Volere requirement shell		
Requirement #: 14	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve disponibilizar a previsão da execução orçamental baseada na faturação atual.		
Rationale: A previsão do futuro financeiro da empresa, para determinada janela de tempo, apesar de não ser uma certeza, ajudará na análise da evolução financeira da empresa, assim como poderá suscitar ideias para melhorar o desempenho.		
Originator: Brainstorming		
Fit Criterion: Na área desenvolvida para a previsão em referência, o utilizador, depois de indicar um limite temporal futuro, visualizará previsões de gastos e a evolução financeira da sua empresa.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

#15 O sistema deve gerar relatórios financeiros

Volere requirement shell		
Requirement #: 15	Requirement Type: Funcional	Event/BUC/PUC #:

Description: O sistema deve gerar relatórios financeiros.		
Rationale: Os relatórios financeiros irão facilitar a análise financeira por parte dos utilizadores do sistema.		
Originator: Brainstorming		
Fit Criterion: Os relatórios devem ser globais ou por categoria.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies: #21	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020. Desdobrado em 07/11/2020 no Req#21.		

#16 O sistema deverá emitir faturas

Volere requirement shell		
Requirement #: 16	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá emitir faturas.		
Rationale: Faz parte do estado na arte do âmbito. Este requisito transforma a aplicação numa faturação. Releve-se a possível necessidade de certificação pela AT e demais restrições legais.		
Originator: Brainstorming		
Fit Criterion: Perante uma fatura existente no sistema, existirá uma opção para imprimir em papel, ou gerar fatura eletrónica.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Won't	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 31/10/2020.		

#17 O sistema deverá permitir a associação de faturas a fornecedores

Volere requirement shell		
Requirement #: 17	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir a associação de faturas a fornecedores.		
Rationale: A associação será necessária em faturas de compra, sendo necessário indicar o emissor.		
Originator: Brainstorming		
Fit Criterion: A associação poderá fazer-se via NIF ou nome da empresa.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 31/10/2020.		

#18 O sistema deve permitir a criação de faturas via scan

Volere requirement shell		
Requirement #: 18	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir a criação de faturas via scan.		
Rationale: A ingestão de faturas via scan/OCR constituirá um meio alternativo de obter informação, a partir duma fatura em papel ou duma imagem/foto, de modo a evitar que o utilizador tenha de introduzir muitos dados manualmente.		
Originator: Brainstorming		
Fit Criterion: O utilizador, depois de introduzir a fatura no scanner, escolhe uma opção que o aciona e carrega a informação, via OCR, que será interpretada e unificada com a BD. Alternativamente, será aceite um ficheiro de imagem do documento, que permita OCR. O utilizador poderá ser chamado a identificar a informação.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Could	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 31/10/2020.		

#21 O sistema deve gerar gráficos estatísticos

Volere requirement shell		
Requirement #: 21	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve gerar gráficos estatísticos.		
Rationale: O sistema deverá ser capaz de gerar gráficos estatísticos para facilitar a análise financeira.		
Originator: Brainstorming		
Fit Criterion: Os gráficos podem ser visualizados separadamente ou incluídos em relatórios. Devem ser globais ou por categoria.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 07/11/2020 (desdobrado do Req#15).		

#22 O sistema deve permitir a definição de orçamentos

Volere requirement shell		
Requirement #: 22	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir a definição de orçamentos.		
Rationale: A definição de orçamentos permitirá balizar e controlar a despesa considerada aceitável num limite temporal à escolha. De modo a controlar gastos em orçamentos para determinadas atividades, sendo estas de períodos curtos ou longos, é fundamental o registo desta informação.		
Originator: Entrevista		
Fit Criterion: A qualquer momento, o utilizador poderá fazer CRUD de orçamentos de		

despesa, por departamentos da empresa. Os orçamentos irão definir limites de despesa em determinadas categorias (as “rubricas”). **Cada orçamento terá um período temporal associado.**

Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#23 O sistema deverá permitir registar receitas interativamente

Volere requirement shell		
Requirement #: 23	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir registar receitas interativamente.		
Rationale: Para controlar receitas da empresa e permitir o controlo de cash-flow, o sistema deve permitir o registo de receitas, de forma interativa manual ou automática.		
Originator: Entrevista		
Fit Criterion: A plataforma terá de proporcionar uma opção para registo das receitas, manual ou automático.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#24 O sistema deverá permitir importar o SAF-T de faturação

Volere requirement shell		
Requirement #: 24	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir importar o SAF-T de faturação emitida.		
Rationale: De modo a completar a informação no sistema para uma melhor utilização, este deverá poder receber um ficheiro SAF-T produzido pelos sistemas de faturação que cada empresa utiliza.		
Originator: Brainstorm		
Fit Criterion: A qualquer momento, o utilizador poderá importar o ficheiro SAF-T, sendo as novas faturas inseridas no sistema.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#25 O sistema deverá permitir armazenar faturas no formato PDF

Volere requirement shell		
Requirement #: 25	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir guardar faturas no formato PDF.		
Rationale: O armazenamento de digitalizações de faturas no sistema permitirá o		

acesso destas a qualquer momento, permitindo consultar a informação desta rapidamente.		
Originator: Entrevista		
Fit Criterion: Após guardar faturas, o utilizador poderá visualizá-las em qualquer altura.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#26 O sistema deverá permitir importar faturas chegadas via email

Volere requirement shell		
Requirement #: 26	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir importar faturas chegadas via email.		
Rationale: Para facilitar o registo de faturas reencaminhadas via email no sistema.		
Originator: Entrevista		
Fit Criterion: O utilizador, na sua conta de email externa ao sistema, poderá reencaminhar um email com uma fatura em anexo, destinado a um endereço de email do sistema, específico da empresa ou do utilizador. Depois, na zona de registo de faturas, deverá existir a opção de importar faturas chegadas por email.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Could	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#27 O sistema deverá permitir marcar faturas como pendentes ou pagas

Volere requirement shell		
Requirement #: 27	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir marcar as faturas como pendentes ou pagas.		
Rationale: Para uma gestão eficaz dos valores em dívida referentes a faturas de fornecedores, as faturas devem poder ser marcadas como pendentes ou pagas.		
Originator: Entrevista		
Fit Criterion: Na zona das faturas deverá existir uma opção que permita marcar faturas como pendentes ou pagas (neste caso, registando a data de pagamento).		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#28 O sistema deverá permitir importar movimentos bancários via API

Volere requirement shell		
Requirement #: 28	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir importar movimentos bancários via API.		
Rationale: O registo automático destes movimentos irá facilitar o controlo das contas bancárias da empresa.		
Originator: Entrevista		
Fit Criterion: Na área de gestão bancária, o sistema disponibilizará uma opção para carregar automaticamente os movimentos inexistentes, precisando apenas dos dados de acesso do cliente à plataforma bancária.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Could	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#29 O sistema deverá permitir importar movimentos bancários via CSV

Volere requirement shell		
Requirement #: 29	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir importar movimentos bancários via CSV.		
Rationale: Nas plataformas bancárias permite-se exportar os movimentos em formato CSV. O registo automático destes movimentos irá facilitar o controlo das contas bancárias da empresa.		
Originator: Entrevista		
Fit Criterion: Na área de gestão bancária, o sistema disponibilizará uma opção para carregar o ficheiro CSV, criando os movimentos inexistentes.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#30 O sistema deverá permitir a associação de faturas a movimentos bancários

Volere requirement shell		
Requirement #: 30	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir a associação de faturas a movimentos bancários.		
Rationale: Associando as faturas aos correspondentes movimentos bancários obtém-se uma organização que permitirá posterior análise dos dados sobre os		

movimentos bancários da empresa.		
Originator: Entrevista		
Fit Criterion: No registo das faturas deverá ser dada a opção de associar a fatura a um determinado movimento bancário.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#31 O sistema deverá permitir associar movimentos bancários a operações financeiras da empresa

Volere requirement shell		
Requirement #: 31	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir associar movimentos bancários a operações financeiras da empresa.		
Rationale: Ex.: pagamento de salários. Sendo possível a associação de movimentos bancários a operações financeiras da empresa, este registo permitirá a análise dos gastos da empresa. Também poderá ajudar a facilitar a conciliação bancária em relação a movimentos não relacionados com faturas.		
Originator: Entrevista		
Fit Criterion: No registo dos movimentos, será dada a possibilidade de associar a um dado documento de despesa.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#32 O sistema deverá permitir fazer o pagamento de faturas

Volere requirement shell		
Requirement #: 32	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir o pagamento de faturas.		
Rationale: Estando já as faturas registadas no sistema, seria cómodo poder executar pagamentos diretamente.		
Originator: Brainstorm		
Fit Criterion: A fatura passará de um estado “pendente” para “pago”, registando-se um movimento na conta bancária associada.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Won't	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#33 O sistema deverá permitir simular a evolução financeira da empresa

Volere requirement shell		
Requirement #: 33	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deverá permitir simular a evolução financeira da empresa.		
Rationale: Para permitir uma melhor gestão da empresa e dos seus recursos, o sistema deverá permitir fazer simulações da evolução (positiva ou negativa) da situação financeira da empresa, de acordo com a variação de determinados parâmetros.		
Originator: Entrevista		
Fit Criterion: Deverá existir uma zona de simulações, onde se apresentem campos para introduzir os valores ou variação de determinados parâmetros (a definir), e a partir daí fazer uma simulação do impacto dessas alterações na evolução financeira da empresa.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Should	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

#36 O sistema deve permitir predefinir categorias

Volere requirement shell		
Requirement #: 36	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir predefinir categorias.		
Rationale: A pré-definição de categorias associadas à conta da empresa permitirá fixar os registos pretendidos para uso posterior e proporcionar uma utilização expedita.		
Originator: Brainstorming		
Fit Criterion: Na área reservada da empresa existirá uma opção CRUD de categorias.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 08/11/2020 (JG). Originado pelo Req#7.		

#37 O sistema deve permitir predefinir etiquetas

Volere requirement shell		
Requirement #: 37	Requirement Type: Funcional	Event/BUC/PUC #:
Description: O sistema deve permitir predefinir etiquetas.		
Rationale: A pré-definição de etiquetas associadas à conta da empresa permitirá fixar uma base de escolha de etiquetas pretendidas para uso posterior e proporcionar uma utilização expedita.		
Originator: Brainstorming		

Fit Criterion: Na área reservada da empresa existirá uma opção CRUD de etiquetas.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 08/11/2020 (JG). Originado pelo Req#7.		

2.5.2 Requisitos de Dados

#12 Cada fatura apenas pode estar associada a uma e uma só categoria

Volere requirement shell		
Requirement #: 12	Requirement Type: Dados	Event/BUC/PUC #:
Description: Cada fatura apenas pode estar associada a uma e uma só categoria		
Rationale: As faturas são correspondentes a um centro de custos e, como tal, apenas podem estar associadas a uma categoria.		
Originator: Brainstorming		
Fit Criterion: Todas as faturas do sistema deverão ter apenas uma categoria associada.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

2.6. Requisitos Não Funcionais

2.6.1 Aparência

(O essencial da aparência do produto.)

2.6.2 Usabilidade

(Facilidade de utilização e outras considerações de usabilidade.)

#34 O sistema deverá ser fácil de usar por utilizadores com pouca cultura financeira

Volere requirement shell		
Requirement #: 34	Requirement Type: Não Funcional	Event/BUC/PUC #:
Description: O sistema deverá ser fácil de usar por utilizadores com pouca cultura financeira.		
Rationale: Sendo objetivo da plataforma que seja utilizada por pessoas fora da área		

da contabilidade, o sistema terá de ser fácil de manipular.		
Originator: Brainstorm		
Fit Criterion: Um utilizador deverá, ao fim de um dia de exploração, ser capaz de navegar facilmente na plataforma.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 7/11/2020.		

2.6.3 Desempenho

(Quão rápida, quão extensa, quão precisa uma funcionalidade tem de ser.)

#39 O sistema deverá acautelar a performance e a disponibilidade ao efetuar cálculos complexos

Volere requirement shell		
Requirement #: 39	Requirement Type: Não Funcional	Event/BUC/PUC #:
Description: O sistema deverá acautelar a performance e a disponibilidade ao efetuar cálculos complexos.		
Rationale: Certos cálculos, relacionados com relatórios e análises, poderão ser exigentes e não deverão impedir uma utilização eficiente e fluida da aplicação. Poderão ter de se tomar medidas na aplicação para não ocupar demasiado o servidor, porque outros utilizadores poderão experimentar uma redução da fluidez.		
Originator: Brainstorm		
Fit Criterion: Encontrada uma medida de tempo “psicológico” (p.ex., 5 segundos), ou desde o início dos cálculos, deve ser disponibilizado um progresso da tarefa, com possibilidade de cancelar. O resultado poderá ser mostrado numa janela nova, de modo a que o utilizador possa continuar a interagir com a aplicação.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Could	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/11/2020.		

2.6.4 Operacionais

(Ambiente de operação do produto, físico ou não, e considerações sobre esse ambiente.)

#19 O sistema deverá disponibilizar uma aplicação web para interação com os seus utilizadores

Volere requirement shell		
Requirement #: 19	Requirement Type: Operacional	Event/BUC/PUC #:
Description: O sistema deverá disponibilizar uma aplicação web para interação com os seus utilizadores		
Rationale: O sistema deve estar disponível para utilização em qualquer lugar com ligação à internet, por ser um padrão de utilização atual.		
Originator: Brainstorming; padrão atual.		
Fit Criterion: A aplicação será acessível por navegador web.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 06/11/2020.		

#38 O sistema deverá separar a camada de acesso à BD numa API de dados

Volere requirement shell		
Requirement #: 38	Requirement Type: Operacional	Event/BUC/PUC #:
Description: O sistema deverá separar a camada de acesso à BD numa API de dados		
Rationale: Padrão arquitetural. Separar radicalmente a camada entre UI e BD irá conferir mais independência a modificações nessas camadas. Adicionalmente, a referida API poderia ser aproveitada por outras aplicações, sob autenticação.		
Originator: Brainstorming; Padrão arquitetural.		
Fit Criterion: Acesso à BD feito exclusivamente por via da referida API.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Could	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 09/11/2020.		

2.6.5 Manutenção, Portabilidade e Suporte

(As alterações esperadas e o tempo permitido para as efetuar.)

2.6.6 Segurança

(A segurança e confidencialidade do produto.)

#35 Toda a interação com o sistema requer autenticação prévia do utilizador

Volere requirement shell		
Requirement #: 35	Requirement Type: Segurança	Event/BUC/PUC #:

Description: Toda a interação com o sistema requer autenticação prévia do utilizador.		
Rationale: Requisito essencial para regular o acesso à aplicação, aos dados elegíveis, e para possível log da atividade de cada utilizador.		
Originator: Padrão de segurança.		
Fit Criterion: Para aceder à aplicação, será disponibilizado um meio de acesso para autenticação do utilizador, e sem obter sucesso nesta não conseguirá aceder.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 08/11/2020.		

#40 Toda a informação associada a cada conta de plataforma deve estar encapsulada

Volere requirement shell		
Requirement #: 40	Requirement Type: Segurança	Event/BUC/PUC #:
Description: Toda a informação associada a cada conta de plataforma deve estar encapsulada.		
Rationale: Requisito essencial de segurança. Todo o utilizador espera que o acesso a informação de gestão seja restringida à empresa, a detentora legal, representada pelos seus utilizadores. Confiança no sistema. Sigilo de informação fiscal e de gestão.		
Originator: Padrão de segurança.		
Fit Criterion: Cada utilizador da plataforma só conseguirá aceder aos registos de BD associados à sua conta de plataforma. O próprio administrador da plataforma só poderá aceder à gestão de contas de plataforma e não à informação de cada empresa.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies: Req#35	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 02/12/2020.		

2.6.7 Culturais e Políticos

(Requisitos especiais que surgem por causa das pessoas envolvidas no desenvolvimento e operação do produto.)

2.6.8 Legais

(Leis e normas que se aplicam ao produto.)

#13 O sistema deverá suportar o padrão SAF-T contabilístico

Volere requirement shell

Requirement #: 13	Requirement Type: Legal	Event/BUC/PUC #:
Description: O sistema deverá suportar o padrão SAF-T contabilístico.		
Rationale: O SAF-T é um padrão aceite e requerido pela AT.		
Originator: Brainstorming		
Fit Criterion: Os ficheiros SAF-T produzidos pelas ferramentas de faturação serão utilizados no sistema pelo utilizador.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 10/2020.		

#20 O sistema deverá respeitar o RGPD

Volere requirement shell		
Requirement #: 13	Requirement Type: Legal	Event/BUC/PUC #:
Description: O sistema deverá respeitar o RGPD.		
Rationale: O RGPD é um regulamento legal em vigor.		
Originator: Brainstorming		
Fit Criterion: A utilização da plataforma não deverá apresentar nenhuma irregularidade relativamente ao RGPD.		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority: Must	Dependencies:	Conflicts: Nenhum
Supporting Materials:		
History: Criado a 07/11/2020 (JG).		

2.7. Análise Formal

Esta secção propõe-se apresentar as conclusões da verificação formal de requisitos. Para uma explicação mais detalhada, reservou-se o anexo Detalhes da Análise Formal.

A análise formal dos requisitos, e do sistema, foi feita utilizando a ferramenta Electrum. O Electrum é uma extensão do Alloy que acrescenta operadores de lógica temporal à lógica relacional do Alloy. Estes operadores temporais permitem a especificação de ações, simplificando a análise de sistemas dinâmicos. Por sua vez, o Alloy é uma linguagem formal de especificação utilizada para raciocinar sobre soluções de software.

Em Alloy, um modelo consiste num conjunto de relações e restrições sobre as mesmas. Estas relações e respetivas restrições especificam o sistema a ser modelado de forma abstrata. Os modelos especificados em Alloy podem ser validados e verificados, automaticamente, utilizando o Alloy Analyzer. Esta capacidade, de verificar predicados sobre o modelo, permite conferir que estados indesejados não são possíveis de serem alcançados.

O modelo apresentado, parcialmente, neste relatório, tem como foco os recursos do sistema: utilizadores, empresas, faturas, etc. O objetivo foi validar que cada utilizador apenas pode editar e visualizar os recursos da sua empresa, não havendo fugas de informação privada. O modelo pode ser consultado na íntegra no repositório do grupo, sendo neste relatório apresentados os excertos mais relevantes.

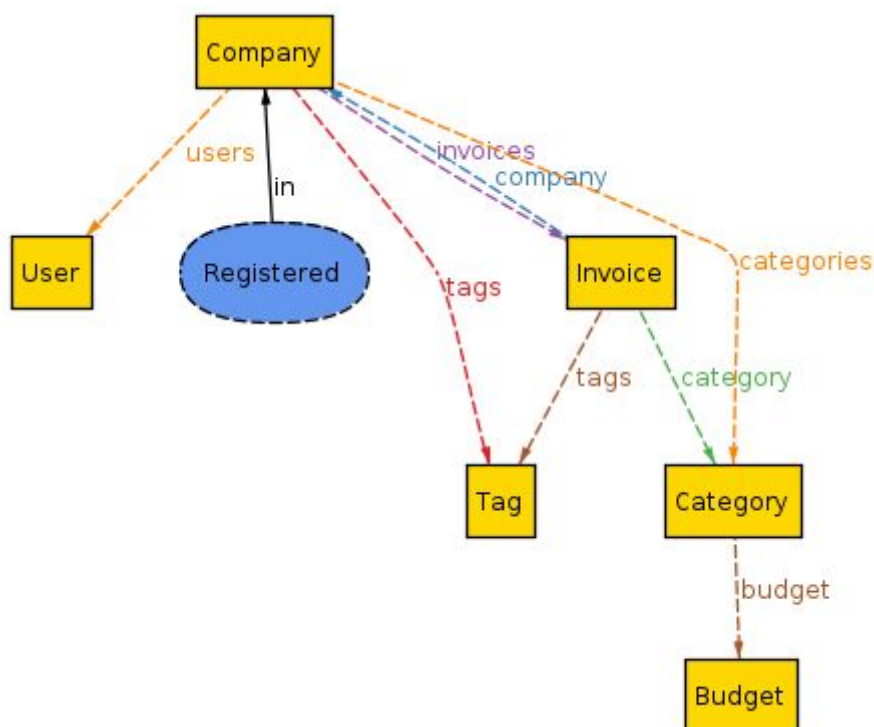
De forma a tornar esta secção mais sucinta, as assinaturas do modelo encontram-se nos anexos.

Finalmente, o modelo encontra-se em constante evolução, sendo esta a versão mais atual à data de entrega deste documento.

2.7.1 Metamodelo

O metamodelo representa os recursos do nosso sistema, bem como as relações entre eles. O metamodelo apresenta relativa simplicidade, sendo que a parte mais complexa é: Uma empresa, assim que é registada no sistema, passa o subconjunto “Registered”.

Analisando o metamodelo é possível reparar que existe informação redundante, sendo que é preciso garantir que as ações do nosso sistema levam a estados coerentes. Por exemplo, uma empresa não pode ter uma fatura, cuja empresa da fatura não seja a empresa original.



2.7.2 Trace

As ações no nosso sistema são traduzidas em predicados no modelo. Estes predicados que representam ações são agrupados num facto do modelo. Desta forma, o modelo apenas evolui consoante as ações do sistema. Se estas ações/predicados quebram invariantes desejáveis ou levam a estados inconsistentes, o sistema tem falhas que devem ser corrigidas (assumindo que não existem falhas na modelação).

As ações modeladas foram as seguintes:

- Registrar uma empresa e um utilizador
- Adicionar um utilizador a uma empresa registada
- Processar uma fatura
- Criação de tags por parte de uma empresa
- Criação de categorias por parte de uma empresa
- Alteração da categoria de uma fatura
- Adição de tags a uma fatura
- Remoção de tags de uma fatura

```
fact traces {  
  no Registered  
  always (  
    some u : User | some c : Company | some i : Invoice | some t : Tag | some cat : Category |  
      registerUser[u, c] or addUserToCompany[u, c] or  
      addInvoiceToCompany[i, c, cat, t] or addTagsToCompany[t, c] or addCategoryToCompany[cat, c] or  
      changeInvoiceCategory[i, cat] or addInvoiceTags[i, t] or removeInvoiceTags[i, t] or  
      stutter  
    )  
}
```

2.7.3 Checks

Tendo o sistema modelado e as ações especificadas, é então possível analisar se estas ações respeitam os nossos invariantes. Esta análise é feita escrevendo um predicado com a “keyword” “check”.

A título de exemplo, exemplificamos três invariantes que devem ser respeitados:

- Uma fatura pertence a uma empresa apenas
- As categorias são únicas a cada empresa
- As tags são únicas a cada empresa

Invariantes mais complexos encontram-se nos anexos.


```

check InvoicesBelongToOnlyOneCompany {
    always (
        all disj c1, c2 : Registered | no c1.invoices & c2.invoices
    )
}

check CategoriesBelongToOnlyOneCompany {
    always (
        all disj c1, c2 : Registered | no c1.categories & c2.categories
    )
}

check TagsBelongToOnlyOneCompany {
    always (
        all disj c1, c2 : Registered | no c1.tags & c2.tags
    )
}

```

3. Modelo do Processo

Esta secção debruça-se sobre o processo de software usado.

3.1. Gestão

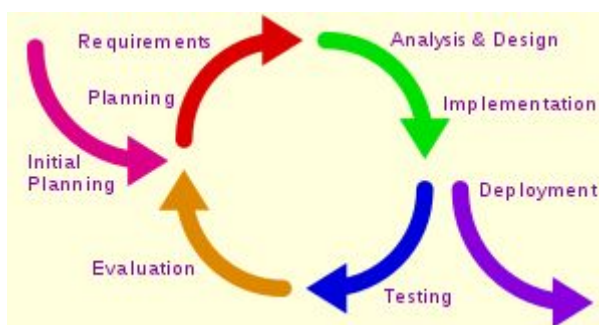
O processo é gerido pela equipa de coordenadores em estreita colaboração democrática com a equipa. Devido à emergência da covid-19, quase toda a interação tem sido online, nomeadamente com os professores.

Há reuniões regulares da equipa completa, sendo normalmente semanais, onde se debatem as dificuldades, requisitos e opções a tomar. Há reuniões online de trabalho por subgrupo, seja da análise e desenvolvimento seja do modelo de negócio. Tirando isto, toda a gestão do tempo é individual, em conjunto com a UC de Dissertação, e eventuais UC atrasadas.

Foram escolhidas as seguintes plataformas para gestão e desenvolvimento do projeto:

- **GitHub**: repositório de código e documentos finais como atas e relatórios a entregar;
- **Google Drive, Google Docs, Notion**: edição colaborativa de conteúdos;
- **Slack**: comunicação chat intra-equipa;
- **Jitsi**: comunicação síncrona da equipa e dos subgrupos;
- **WhatsApp**: comunicação entre o professor orientador JCR e a equipa.
- **Phoenix Framework**: Linguagem Elixir e ambiente de codificação. Complementado com **Bulma** (Framework para CSS).

3.2. Processo de desenvolvimento



Modelo: Parece-nos adequado reconhecer o modelo **iterativo** como padrão do nosso processo de desenvolvimento. Efetivamente, iteramos ciclos de **elaboração de requisitos**, com posterior **escolha de requisitos**, **depois analisados, implementados e testados**. Apesar das versões sistematicamente guardadas no github, possivelmente apenas a última versão poderá ser classificada como *deployment ready*.

Revisão pelo cliente: Reuniões semanais de acompanhamento do projeto com o orientador tecnológico da equipa, Professor José Carlos Ramalho, e quinzenais com o orientador do modelo de negócio, Professor Paulo Afonso, que, adicionalmente, poderemos imaginar num papel de mediadores entre a equipa e um desejável cliente final.

4. Matriz de Rastreabilidade

Segundo Tiago Palhoto ([Pal16], pág. 38), a matriz de rastreabilidade efetua um **mapeamento de funcionalidades com requisitos**, permitindo assim controlar a ligação entre estes. A sua vantagem está em, perante a alteração a um requisito funcional, identificar rapidamente os casos de uso afetados.

Para elaborar uma matriz de rastreabilidade, é previamente necessário estabelecer a lista de casos de uso, já que esta matriz apresenta o relacionamento dos requisitos a montante (*stakeholders*) e a jusante (casos de uso, casos de teste, classes, estado, etc.).

4.1. Casos de Uso

Identificaram-se os seguintes casos de uso *top*, do produto, obtidos da secção 2.2.2.

Casos de uso:

1. **Administrar Plataforma:** operação de administração com plenos poderes.
2. **Administrar Conta:** operação de administração e gestão da conta de plataforma.
3. **Importar faturas:** operação automática, a pedido, que faz ingestão de faturas, em lote ou não.
4. **Gerir faturas:** operações CRUD de faturas; gestão financeira.
5. **Gerir orçamentos:** operações CRUD de orçamentos.
6. **Gerar relatórios:** operações de geração de relatórios diversos baseados na informação da BD, incluindo análise automática e gráficos.

4.2. Matriz de Rastreabilidade

Com os casos de uso anteriores e os requisitos funcionais produziu-se a seguinte matriz de rastreabilidade:

RF \ UC	1	2	3	4	5	6	8	9	10	12	14	15	16	17	18	21	22	23
1						x												
2							x	x										
3		x	x												x			
4	x			x	x				x	x			x	x				x

5																	x	
6											x	x					x	

RF \ UC	24	25	26	27	28	29	30	31	32	33	36	37						
1																		
2																		
3	x	x	x		x	x												
4		x	x	x			x	x	x		x	x						
5										x								
6										x								

5. Gestão de Recursos Humanos

5.1. Regras de base

Responsabilidade e empenho. Seguir os requisitos e orientações acordadas. Dedicar pelo menos metade da semana ao projeto, consequência das unidades de crédito da UC PEI. Respeito pelos diferentes membros da equipa.

Sempre que surgirem dúvidas difíceis de resolver sem uma decisão estratégica, solicitar apoio aos coordenadores ou à equipa.

5.2. OBS (*Organizational Breakdown Structure*)

Árvore da organização

A árvore da organização do projeto pode descrever-se pela seguinte estrutura:

- **Cliente:** Equipa docente de PEI.
- **Docentes orientadores:** Professores José Carlos Ramalho e Paulo Afonso.
- **Mentores:**
 - Finiam - José Gomes
 - Subvisual - Laura Esteves
 - UTrust - Liliana Brito
 - Explore.dev - Adriano Martins
 - Keyruptive - Francisco Maia
- **Equipa:**
 - Equipa de requisitos e análise: toda a equipa participa;
 - Relatório: coordenação, redação e revisão de João Grenhas.
 - Equipa de métodos formais: Eduardo Barbosa, Márcio Sousa.
 - Equipas de programação (desenho e implementação):

- BD: Bruno Martins, João Vilaça, Eduardo Barbosa.
- Frontend: Jéssica Lemos, Catarina Machado, Bárbara Cardoso.
- Backend: Bruno Martins, Eduardo Barbosa, João Vilaça, Márcio Sousa.
- AI: Filipe Monteiro.

Membros da equipa

Esta secção pretende identificar as competências mais fortes, de cada membro da equipa, composta por nove membros, não numa perspetiva de curriculum vitae ou de potencialidade completamente descrita, que será necessariamente mais vasta, mas antes naquilo em que cada membro se sente pronto a contribuir na sua máxima eficácia e qualidade.

Tem por objetivo ajudar os coordenadores do projeto a alocar tarefas. Elenca-se os nomes por ordem alfabética, o nº de aluno e o curso.

Bárbara Andreia Cardoso Ferreira, a80453, MIEI

Áreas científicas: Métodos Formais, Verificação Formal, Engenharia do Conhecimento, Análise de Dados.

Frameworks e SGBD: MySQL, Neo4J, MongoDB, Vue.

Linguagens de programação: Haskell, Java, C, WhyML.

Bruno Manuel Chaves Martins, a80410, MIEI

Áreas científicas: Desenvolvimento de Software, Arquiteturas de Software, POO, PF, Testes de Qualidade, Machine Learning.

Frameworks e SGBD: Phoenix, NodeJS, ExpressJS, .NET, MongoDB, PostgreSQL, MySQL, Neo4J, React, Flutter.

Linguagens de programação: Elixir, Go, Java, JavaScript, C#, Prolog, Dart, C, C++, Python.

Catarina Araújo Machado, a81047, MIEI

Áreas científicas: Desenvolvimento de Software, Engenharia de Aplicações, Sistemas Inteligentes.

Frameworks e SGBD: Ruby on Rails, React, Vue.js, PostgreSQL, Neo4J.

Linguagens de programação: Ruby, Java, JavaScript, Python, Haskell.

Eduardo Jorge Lima Pinto Barbosa, a83344, MIEI

Áreas científicas: Teoria das Linguagens de Programação, Métodos formais, Desenvolvimento de Software, Teoria das Categorias

Frameworks e SGBD: Phoenix, PostgREST, Servant, PostgreSQL, MySQL

Linguagens de programação: Agda, Haskell, Elixir, Go

Filipe Pimenta Oliveira Monteiro, a80229, MIEI

Áreas científicas: Ciência de Dados, Sistemas Inteligentes, Machine Learning, Desenvolvimento de Software.

Frameworks e SGBD: Tensorflow, Keras, Scikit-learn, Flask, NodeJS, React.

Linguagens de programação: Python, R, JavaScript, Java, C, C++.

Jéssica Andreia Fernandes Lemos, a82061, MIEI

Áreas científicas: Engenharia de Aplicações, Sistemas Inteligentes, Machine Learning.

Frameworks e SGBD: Phoenix, NodeJS, PostgreSQL, React, Vue.js, Django, Flask.

Linguagens de programação: Java, JavaScript, Python, Elixir.

João Manuel Pós de Mina Grenhas, pg33874, MEI

Áreas científicas: Engenharia de Software. Engenharia de Requisitos. Relatório de projeto. Teste ágil de software. Desenvolvimento de software. Arquiteturas microserviços. POO. PF. PLN. Ontologias.

Frameworks e SGBD: Alaska Xbase++ via Módulo C *framework*. Node.JS. Express.JS. Protégé. SGBD: DbfDbt, MongoDB, GraphDB.

Linguagens de programação: Xbase++, JavaScript, Python.

João Pedro Machado Vilaça, a82339, MIEI

Áreas científicas: Desenvolvimento de Software, DevOps, Sistemas Distribuídos, Engenharia do Conhecimento, Análise de Dados.

Frameworks e SGBD: Phoenix, NodeJS, PostgreSQL, SQLServer.

Linguagens de programação: Elixir, Terraform, Ansible, JavaScript, TypeScript.

Márcio Alexandre Mota Sousa, a82400, MIEI

Áreas científicas: Métodos Formais, Verificação Formal, Criptografia, Segurança, Desenvolvimento de Software.

Frameworks e SGBD: Flask, MySQL, Neo4J.

Linguagens de programação: C, Haskell, Java, Python, Prolog, WhyML, R.

5.3. RAM (*Responsibility Assignment Matrix*)

Segundo António Miguel ([Mig13], pág. 251), esta matriz é usada para ilustrar as ligações entre o trabalho a realizar e as partes interessadas do projeto. O formato tabular permite ver todas as atividades associadas a uma pessoa, ou, inversamente, todas as pessoas associadas a uma atividade. Em projetos de grande dimensão, as matrizes de responsabilidade podem ser desenvolvidas a vários níveis. Exemplo: Uma MR de alto nível pode definir qual a **equipa** responsável por cada componente da EDT, ao passo que MR de nível mais baixo podem ser usadas **em cada equipa** para designar papéis, responsabilidades e níveis de autoridade para atividades específicas.

Exemplo de uma RAM para uma equipa de desenvolvimento de software:

Legenda: A = Aprova; C = Consultado; E = Exigida a revisão; I = Informado; R = Responsável.

Atividade	Parte interessada					
	Sponsor	Cliente	Gestor de projeto	Analista	Programador	Quality Assurance
Plano do projeto	A		R	C	I	E
Requisitos	C	C; A	C	R		E
Desenho	I		A	C	R	E
Desenvolvimento	I		A	C	R	E
Integração	I		A	R	C	E
Testes	A	C	C	C	C	E
Rollout	I	A	C	C	C	

6. Estratégia de Solução

6.1. Arquitetura Geral

A arquitetura projetou-se como solução web, desenhada numa aplicação principal, com persistência de dados via um SGBDR, e recorrendo a API internas e externas (AT e bancos).

As opções refletiram uma abordagem cliente-servidor, *state of the art* em termos tecnológicos e de interface com o utilizador.

Dados

A persistência dos dados será conseguida com recurso a um SGBDR, **PostgreSQL**. Alguns registos de *log* poderão ser sobre ficheiro de texto *ascii*.

Como modelo, adotou-se o modelo de dados da análise de domínio.

Acesso do utilizador

Todo o utilizador acederá à aplicação via **navegador de internet**. Terá de se autenticar para aceder, e terá acesso às funcionalidades dentro das balizas definidas para cada tipo de utilizador.

Cf. requisitos 35 e 40 (não funcionais, de segurança).

Tecnologias

Foram escolhidas as seguintes tecnologias.

Como *framework tool* para implementação da solução, escolheu-se a **Phoenix Framework**: Linguagem **Elixir** e ambiente de codificação. Facilidades de alojamento.

Complementos: **Bulma** (Framework para CSS).

Para construir o componente de estatísticas e Machine Learning foi utilizada a linguagem Python, com recurso à *framework* Flask para criar um ambiente de servidor. Para executar pedidos a esse servidor é usada a linguagem de *query GraphQL*. Ainda neste serviço, é utilizada a biblioteca Pandas para facilitar o manuseamento de dados e o *Prophet*, uma biblioteca de previsão de dados.

O componente de *scrapping* foi desenvolvido em typescript, com recurso a nodeJS para a criação do servidor.

6.2. Diagramas de Blocos (*Building Block View*)

6.2.1 Diagrama de Componentes

Na imagem seguinte podemos verificar que a aplicação segue uma arquitetura de tipo quase monolítico, embora apoiada em dois serviços desenhados para retirar stress computacional da mesma, e numa base de dados.

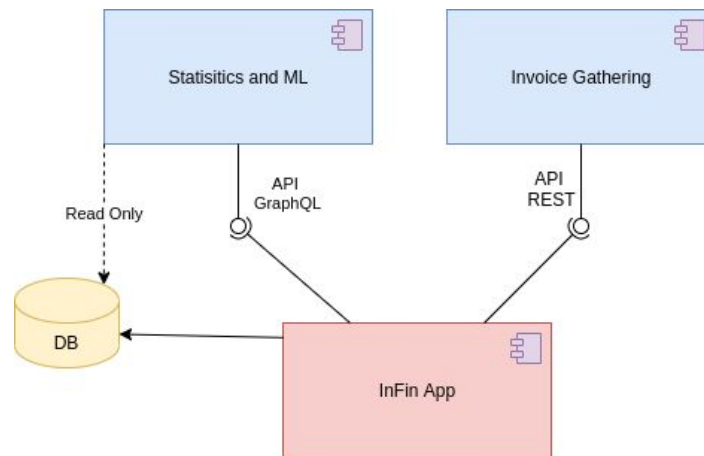
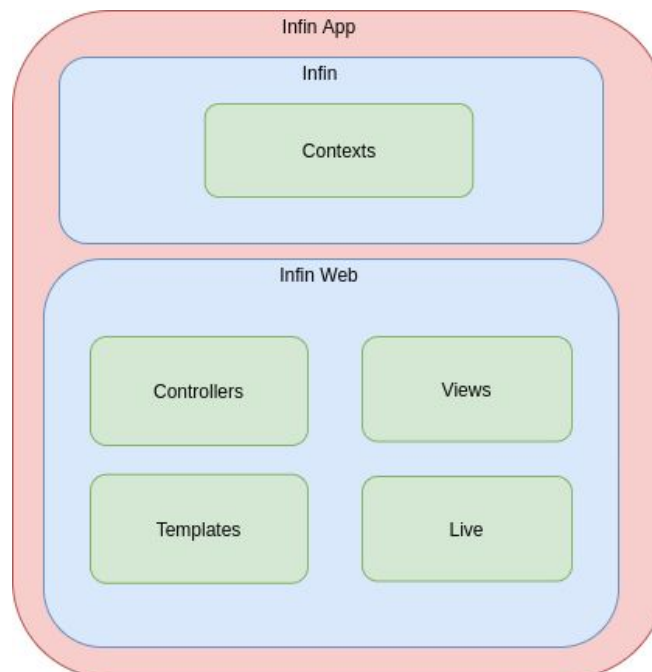


Fig. - Esquema do diagrama de componentes

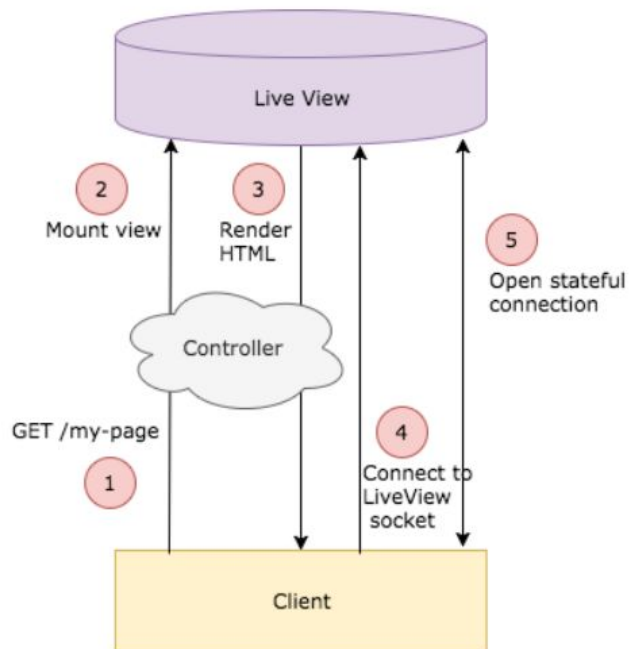
O componente **InFin App** é responsável por toda a lógica de negócio da aplicação, utilizando uma Base de Dados relacional para manter toda a persistência de dados necessária, por exemplo, contas de utilizadores, faturas das empresas, entre outros. Como este componente segue uma arquitetura (quase) monolítica, encapsula toda a lógica de *frontend* pela qual um cliente utiliza o sistema, recorrendo a *server-side rendering*. No entanto, apesar do referido monolitismo, poderia ser assumido que existem limitações no que toca à interação do utilizador com a página, sendo necessário um *render* completo novamente da página. Phoenix, a *framework* utilizada, resolve este problema mantendo uma conexão aberta *stateful*, por onde são enviados eventos para o *backend*, que por sua vez responde com dados relevantes para a atualização da página, não sendo necessário o envio de uma página completamente nova.



Como podemos verificar na imagem anterior, o componente **InFin App** segue uma arquitetura interna MVC, onde os Modelos estão especificados dentro dos *Contexts* e

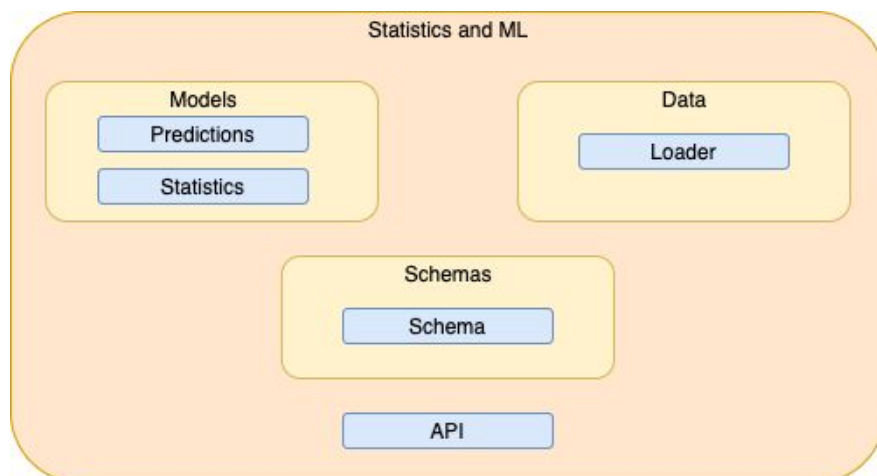
as suas funções de camada de negócio definem-se em ficheiros dentro do módulo **infin**. No módulo **Infin Web** estão definidos os controladores e as *views* e *templates*. Os controladores são o ponto de comunicação da parte do cliente com o *backend*.

O módulo **Live**, como referido anteriormente, é um dos módulos mais importantes do sistema, permitindo manter uma conexão aberta entre o *backend* e *frontend*, por onde eventos são registados e geram uma reação, não sendo necessária uma nova renderização completa. No diagrama seguinte, podemos verificar como o processo ocorre.



O componente **Invoice Gathering**, através de *scrapping* executa o *login* no *site* das finanças, mais tarde recolhe todas as faturas emitidas em nome da empresa utilizadora da plataforma através de um pedido GET, e disponibiliza uma interface REST para que estes pedidos sejam executados.

O componente **Statistics and ML**, através do acesso *read-only* à Base de Dados, recolhe todos os dados relevantes para oferecer aos clientes estatísticas, e ao mesmo tempo construir *datasets* para a previsão de métricas relevantes.




Futuramente, prevê-se desejável implementar o requisito #38, “O sistema deverá separar a camada de acesso à BD numa **API de dados**”, API que disponibilizaria uma interface a utilizar imediatamente por *Statistics and ML*, e desejavelmente por *InFin App*. Isto aumentará a complexidade em termos de **segurança**, nomeadamente na necessidade de autenticação e regulação do acesso a dados sensíveis ou privados, provavelmente a resolver via JWT. Poderá ocorrer um decréscimo de **eficiência**, devido a mais trânsito de dados e a protocolos https. Facilitará muito a **manutenção** futura, da BD e do sistema de software, porque apenas a API vai conhecer o esquema de BD e efetuar a sua gestão e manutenção (nomeadamente criação de novas tabelas, índices e atributos, reformulação e migração de dados, mudança de SGBD), necessitando, porém, de disponibilizar uma **interface externa** competente, que satisfaça as necessidades do sistema.

6.2.2 Diagrama de Classes

6.3. Interface com o Utilizador

Na página principal, o utilizador consegue obter todas as informações sobre o serviço disponível e efetuar o registo ou *login*.


[Invoices](#)
[Bank Accounts](#)
[Incomes](#)

[Register](#)
[Log In](#)


The only software you need to run your business financials

Manage money, not spreadsheets

Let's get you connected! We've designed a simple, efficient service for companies to automate their financial processes.


[Start Now!](#)

What's So Great About Infin?




Lovely product, very good, very nice.

José Gomes
CEO at Finiam



Amazing service, I need this in my company.

Laura Esteves
COO at Subvisual



Incredible work, can't live with this no more.

Francisco Maia
CEO at Keyruptive

Standard	Pro	Enterprises
\$ 4.99 /month	\$ 49.99 /month	Custom Plan
Up to 100 invoices/month	Up to 1000 invoices/month	Custom user and invoice limit
Up to 10 standard users	Up to 50 standard users	Adjusted price
Unlimited auditing users	Unlimited auditing users	
Register	Register	Contact Sales

Contact the Infinions




Let's talk about the future of financials

We're here to answer your questions and discuss the automation of your company financial processes.

📍 Departamento de Informática - Universidade do Minho, Braga

✉ contact@infin.io

[Submit](#)

Infin. Manage money, not spreadsheets.

[Features](#)
[Enterprise](#)
[Support](#)
[ICO](#)

© 2020 Infinions. All right reserved.

Para aceder ao sistema, o utilizador deve registar os seus dados bem como os da empresa.

BULMA Invoices Bank Accounts Incomes Register Log in

Register

1 2 3
User Company Finish

Infin

1

Previous Next

Infin. Manage money, not spreadsheets. [Features](#) [Enterprise](#) [Support](#) [ICO](#)

[Twitter](#) [Facebook](#) [Instagram](#) © 2020 Infinions. All right reserved.

Ao aceder à página da empresa, o utilizador pode editar os dados de perfil, bem como gerir as categorias da mesma.

BULMA Invoices Bank Accounts Incomes infin@gmail.com Infin Log out

Infin

Settings

Name

Infin

Tax Id

1

Update

Categories


Education

New Category

Infin. Manage money, not spreadsheets. [Features](#) [Enterprise](#) [Support](#) [ICO](#)

[Twitter](#) [Facebook](#) [Instagram](#) © 2020 Infinions. All right reserved.

Poderá registar novas categorias, editar ou remover, consistentemente.

InvoicesBank AccountsIncomes

infin@gmail.com

infin

Log out

Category

Name

Education

Save

Back

Delete


Infin. Manage money, not spreadsheets.

[Features](#) [Enterprise](#) [Support](#) [ICO](#)



© 2020 Infinions. All right reserved.

O utilizador pode ainda editar os dados da sua conta no sistema.

InvoicesBank AccountsIncomes

infin@gmail.com


infin

Log out


Settings

Change email

Email

 infin@gmail.com


Current password




Change email

Change password


New password



Confirm new password



Current password



Change password

Infin. Manage money, not spreadsheets.

[Features](#) [Enterprise](#) [Support](#) [ICO](#)

O sistema permite consultar todas as *invoices* no sistema, bem como criar uma nova ou editar. É, ainda, permitido importá-las do portal da AT, e-fatura.

Invoices

Document	Emission Date	Total Value (EUR)	Emitter Tax Id	Emitter Name	Category	
1	2020-12-16	2.50	2	FNAC	Education	>
2	2020-12-18	27.80	2	FNAC	Education	>

< >

Page 1 of 1

[New Invoice](#) [Import Invoices](#)

Infin. Manage money, not spreadsheets.

[Features](#) [Enterprise](#) [Support](#) [ICO](#)

© 2020 Infinions. All right reserved.

Invoice

Id Document

Emission Date

Total Value (EUR)

Emitter Tax Id

Emitter Name

Category

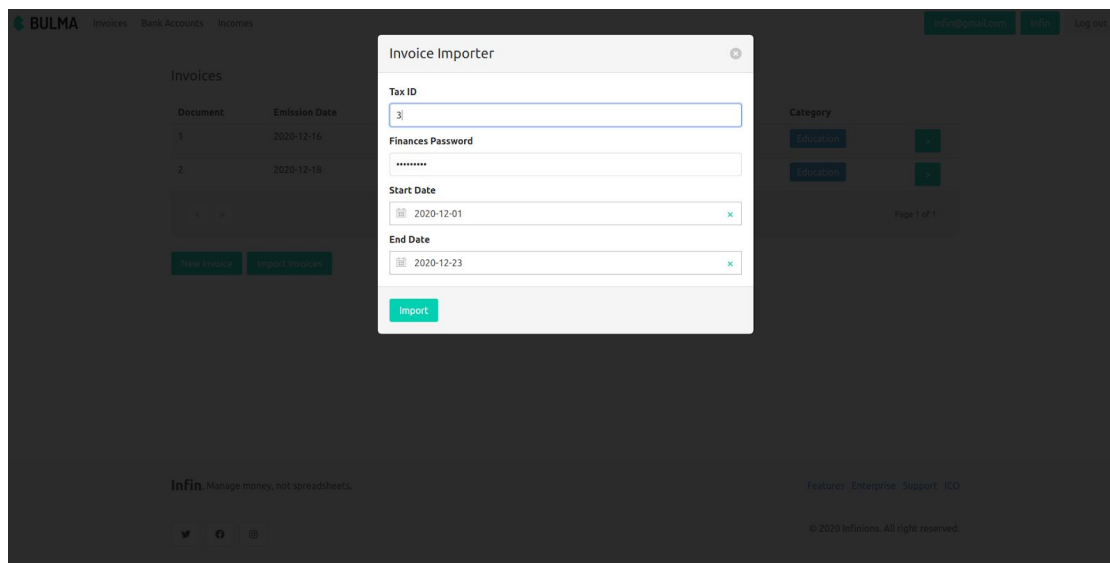
[Save](#)[Back](#) [Delete](#)

Infin. Manage money, not spreadsheets.

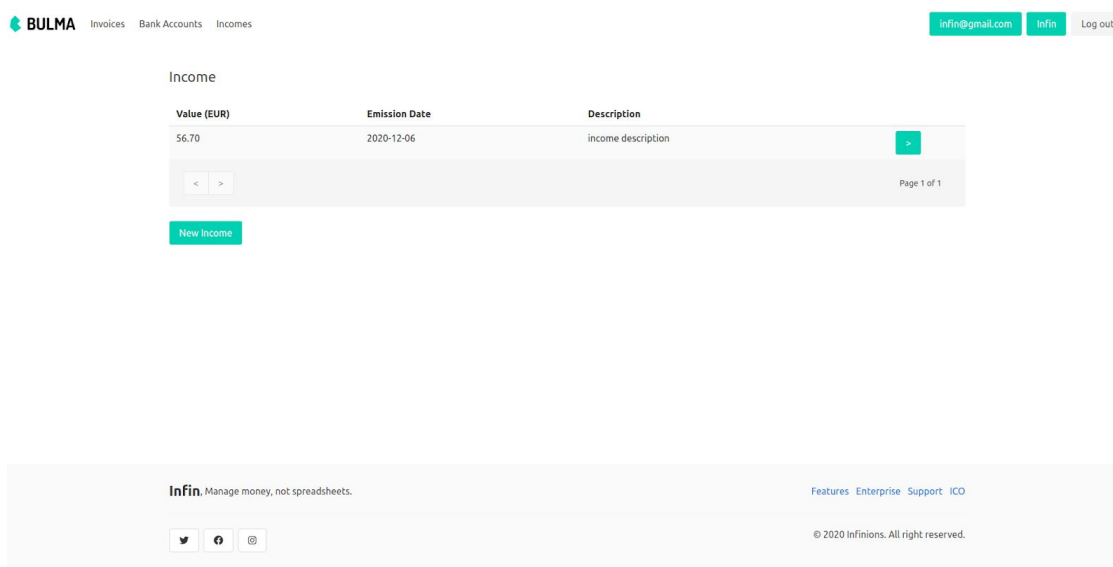
[Features](#) [Enterprise](#) [Support](#) [ICO](#)


© 2020 Infinions. All right reserved.

Para fazer a importação das *invoices*, o utilizador deve aceder à sua conta do portal, através das respetivas credenciais.



O utilizador pode ainda consultar e registar as suas receitas.



InvoicesBank AccountsIncomes

infin@gmail.cominfinLog out

New Income

Value

35.00

Date




2020-12-07

Description

Save


Infin. Manage money, not spreadsheets.

FeaturesEnterpriseSupportICO













© 2020 Infinions. All right reserved.

O utilizador pode associar contas bancárias, fazendo o login com as credenciais adequadas.

InvoicesBank AccountsIncomes

infin@gmail.cominfinLog out

Banks




	BBPI	BANCO BPI, SA
	BST	BANCO SANTANDER TOTTA, SA
	ABPT	BANCO ACTIOVBANK, SA
	BCPPT	BANCO COMERCIAL PORTUGUES, SA
	CGDPT	CAIXA GERAL DE DEPOSITOS, SA
	CEMG	CAIXA ECONOMICA MONTEPIO GERAL
	GCA	CAIXA CENTRAL - CAIXA CENTRAL DE CREDITO AGRICOLA MUTUO, CRL
	CEMAH	CAIXA ECONOMICA DA MISERICORDIA DE ANGRA DO HEROISMO
	BIG	BANCO DE INVESTIMENTO GLOBAL, SA
	BPG	BANCO PORTUGUES DE GESTAO, SA

<>

Page 1 of 3

Infin. Manage money, not spreadsheets.

FeaturesEnterpriseSupportICO



© 2020 Infinions. All right reserved.

Consent for BST

PT50001800035437859102009


Jessica Lemos

Consent

Back



Desta forma, pode verificar e associar todas as transações efetuadas num determinado período de tempo.

 **EMPRESA DE TESTE**

Bank: BANCO BPI, SA (BBPI)

IBAN: PT5000100005578399000180

Authorized: 30.00 EUR

Expected: 30.00 EUR

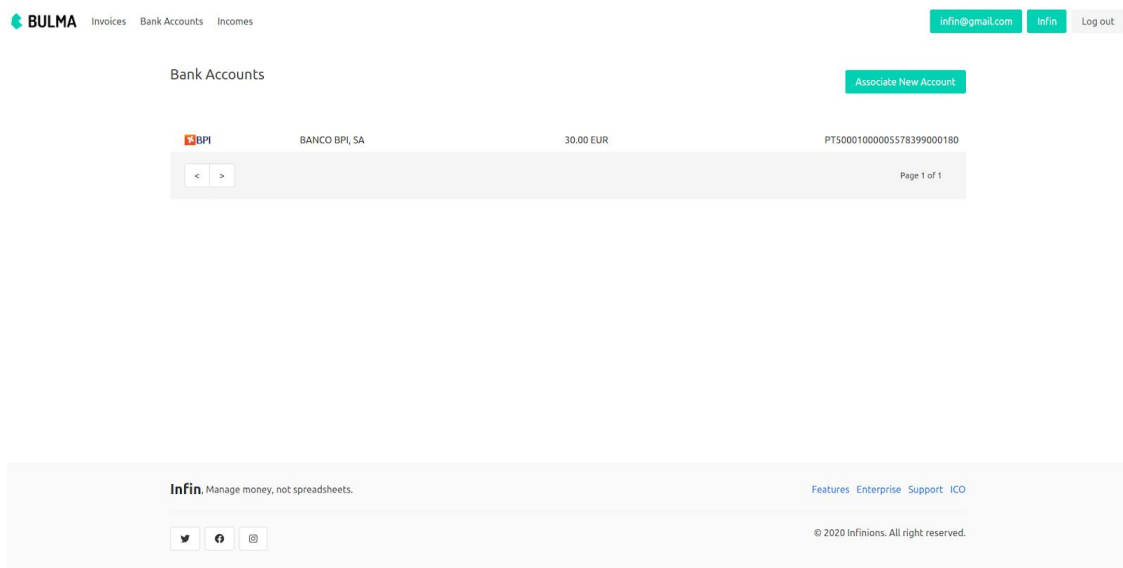
Transactions

Start Date

mm/dd/yyyy

Fetch Transactions

2018-10-25	TRF 0031574 DE ALFREDO	1.00 EUR
2018-10-25	TRF 0031573 DE MARIA	1.00 EUR
2018-10-25	TRF 0031572 DE ANA	1.00 EUR
2018-10-25	TRF 0031571 DE HELENA	1.00 EUR
2018-10-25	TRF 0031570 DE PAULO	1.00 EUR
2018-10-25	TRF 0031569 DE BRUNO	1.00 EUR
2018-10-25	TRF 0031568 DE PEDRO	1.00 EUR
2018-10-25	TRF 0031567 DE RICARDO	1.00 EUR
2018-10-25	TRF 0031566 DE CRISTINA	1.00 EUR
2018-10-25	TRF 0031565 DE TERESA	1.00 EUR

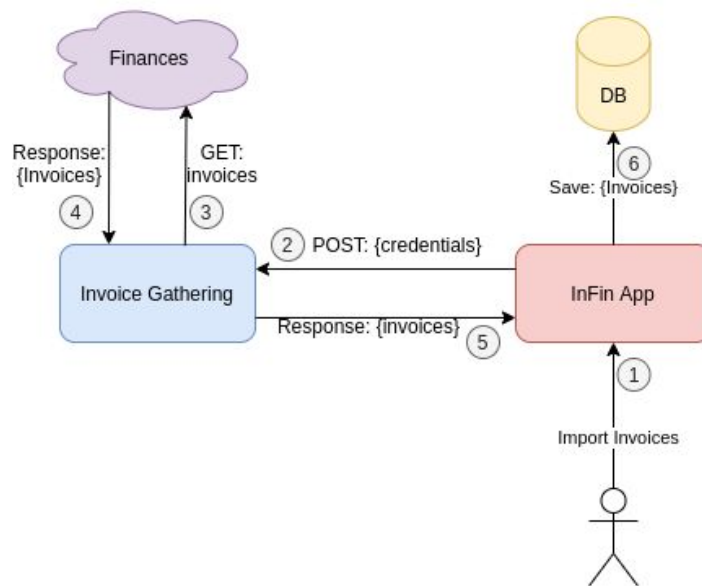


6.4. Lógica de Controlo

Um algoritmo pilar do nosso sistema é a importação de faturas, do portal da AT, para o sistema. Este processo necessita dos serviços **InFin App** e **Invoice Gathering**. Quando o utilizador executa este processo, insere as suas credenciais de acesso ao portal das finanças, e o intervalo de datas onde as faturas a importar se encontram. Após esta ação, a **InFin App** executa um pedido *POST* ao serviço **Invoice Gathering**; o pedido é *POST* pois é necessário que as credenciais sejam encapsuladas no body para obter uma camada extra de proteção.

Este serviço autentica-se no portal das finanças através *scrapping* e, posteriormente, faz a recolha das faturas no portal das finanças através de uma *interface* disponibilizada. Depois de receber a resposta, envia, no formato *JSON*, a resposta, que contém os dados detalhados das faturas do contribuinte, no intervalo de tempo especificado. Esses dados são processados e inseridos na plataforma para que o cliente consiga depois executar as ações que entender.

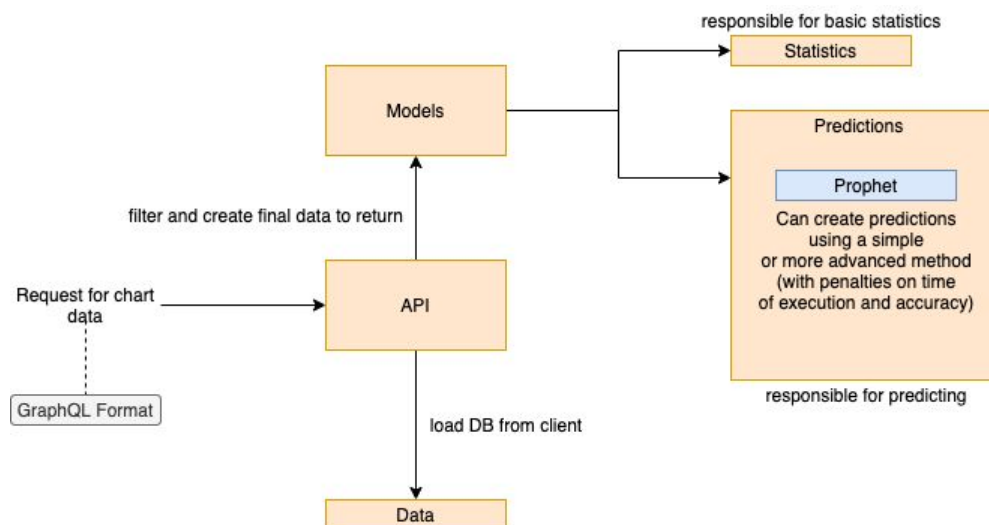
Na imagem seguinte podemos ver um diagrama que especifica este *flow* de comunicação.



No que toca à análise dos dados e exploração destes, um dos pontos importantes é a biblioteca **Prophet**, produzida pelo *Facebook*, responsável pela previsão das séries temporais dos clientes. Este recebe um conjunto de dados filtrado para o cliente **X** e gera um modelo capaz de justificar os dados tendo em conta padrões. Com este modelo criado é então inferido sobre um período de tempo futuro onde iremos receber uma previsão. Uma nota importante é que é extremamente difícil conseguir boas previsões com poucos dados por parte do cliente, sendo que o uso de dados de outros é errado devido à variância que cada empresa terá no seu contexto. Dito isto, o modelo faz uma análise rápida mais básica, ficando depois um serviço responsável por descobrir melhores modelos. Para a melhor previsão possível seria necessário a interação de um cientista de dados competente para ajustar as diferentes componentes do modelo em relação aos dados em questão.

No futuro espera-se, já com a existência de grandes quantidades de dados por parte dos clientes, que se possa explorar outros métodos de previsão mais eficientes e precisos.

No diagrama seguinte podemos ver o flow deste algoritmo.

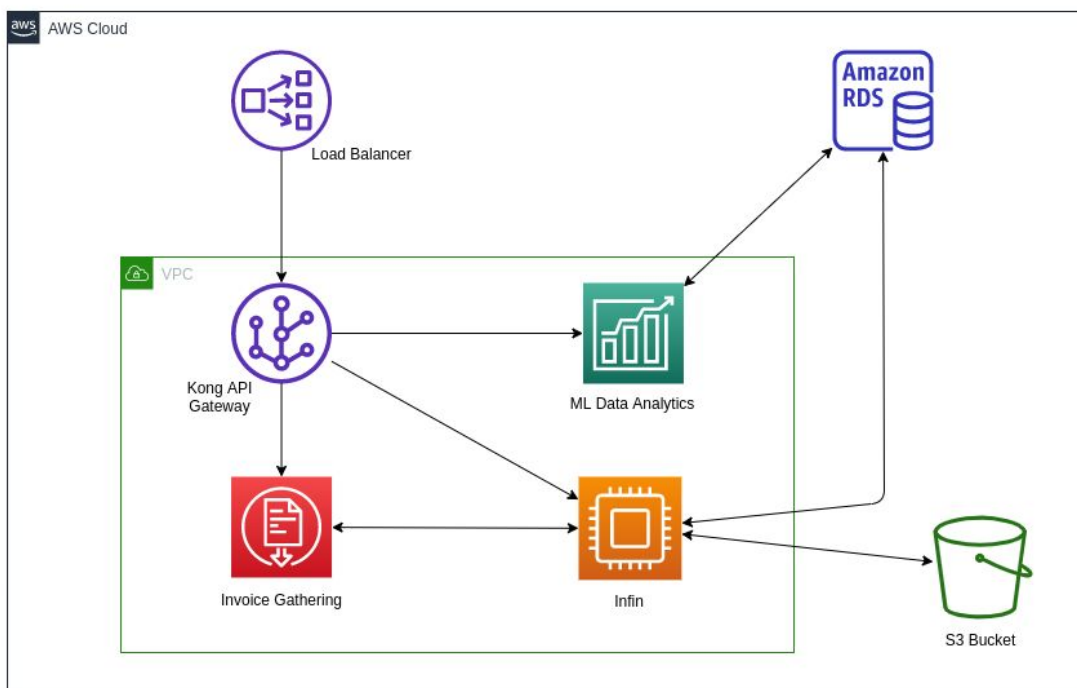


7. Instalação e Manutenção

A instalação deste sistema será feita no *cloud provider* AWS e fará uso de diversas ferramentas lá disponibilizadas. Em primeiro, e como ponto de entrada dos pedidos do utilizador, será usado um Load Balancer da AWS capaz de receber tráfego do exterior e direcioná-lo para o API Gateway da aplicação, dentro do VPC onde estarão os vários serviços. O Kong API Gateway será então responsável por analisar o URL do pedido e, de acordo com o mesmo, escolher qual o serviço ao qual este deverá ser entregue. Depois disso, no VPC estão ainda contidos os 3 serviços que compõem então a parte aplicacional da Infin.

Para além disso, recorreremos ainda a mais 2 componentes da AWS. Em primeiro lugar, o Amazon RDS, que é o sistema de base de dados relacionais gerido pela AWS, onde teremos uma instância de PostgreSQL, que servirá de suporte ao serviço Infin e ao serviço ML Data Analytics. A utilização de RDS oferece várias vantagens, entre as quais, a alta disponibilidade do mesmo e replicação de dados.

Por fim, será também utilizado um S3 Bucket, que tal como o RDS oferece alta disponibilidade e a replicação de dados, mas que está otimizado para armazenamento de ficheiros, sendo neste que serão guardadas as digitalizações e ficheiros das faturas dos utilizadores da plataforma.



Anexos

Anexo I. Modelo de Qualidade da Norma ISO 9126

A norma 9126 foca-se na qualidade do produto de software, propondo **Atributos de Qualidade**, distribuídos em seis características principais e respectivas subdivisões, conforme se pode ver na figura abaixo.

In https://pt.wikipedia.org/wiki/ISO/IEC_9126 .



Fig. - Atributos de qualidade

Anexo II. Técnicas de Levantamento de Requisitos

O Cartão de Volere

Nota: Por uma questão de habituação a equipas com membros de diferentes línguas, optou-se pelo modelo com etiquetas em Inglês, ainda que neste relatório o cartão seja preenchido em Português.

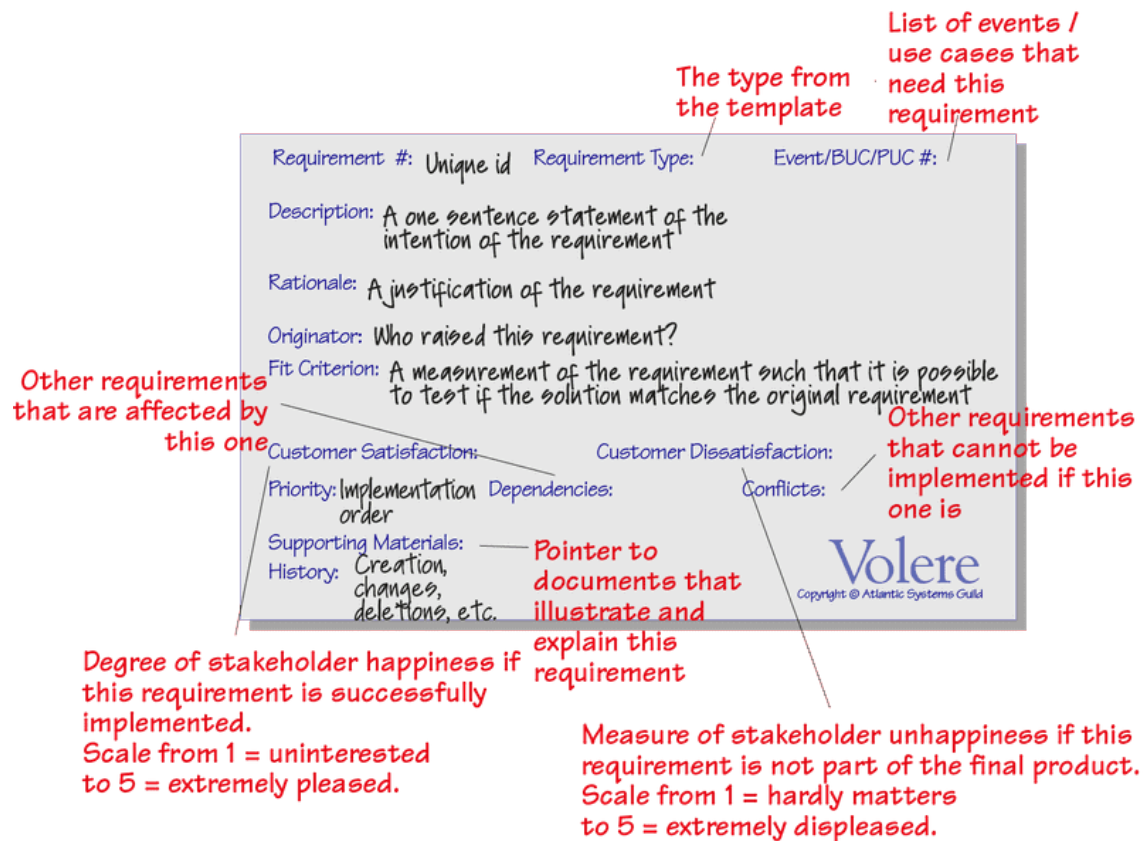


Fig. - Cartão de Volere comentado

No contexto deste relatório, elaborou-se a seguinte apresentação:

Volere requirement shell		
Requirement #:	Requirement Type:	Event/BUC/PUC #:
Description:		
Rationale:		
Originator:		
Fit Criterion:		
Customer Satisfaction:	Customer Dissatisfaction:	
Priority:	Dependencies:	Conflicts:
Supporting Materials:		
History: Criado a <data>. [Implementado a <data>.]		

Requirement Type: Tipo de requerimento. Colocar em relação com as secções da lista de requisitos deste documento. Por exemplo, abreviando:

- Funcional
- Dados
- Aparência
- Desempenho
- Operacional
- Manutenção
- Segurança
- Cultural
- Legal

Valores para **Originator**:

- Estado da arte (padrão de facto que se impõe)
- Brainstorming (resultante da discussão em grupo)
- Introspeção (quando nasce da nossa imaginação individual)
- Entrevista com S...
- Sr./Sra. N...
- Padrão

History: Parece útil ir afixando uma etiqueta “Implementado.”, quando o requisito tiver fechado o seu ciclo de desenvolvimento.

Entrevistas

Coleciona-se nesta secção as entrevistas efetuadas para o levantamento de requisitos.

Entrevista a Laura Esteves (COO da Subvisual)

Entrevistador: João Vilaça.

04/11/2020. 37 minutos. Não é a entrevista literal, são excertos, **sintetizados e adaptados**, da entrevista:

JV: Gerir as faturas: ... como fazem? Dá muito ou pouco trabalho?

LE: Trabalho há cerca de sete anos nesta área e já testei vários softwares, mas nenhum agradou. Revelaram-se **ou demasiado extensos e complexos ou pouco configuráveis**, não sendo possível ajustar às necessidades da empresa. Resultado: todos os anos crio um **Excel para a gestão financeira**, com pequenas alterações de ano para ano. O processo base é muito simples: digitalizo tudo o que é papel; no final do mês vou buscar todos os **extratos bancários**, introduzo-os todos num Excel. Cada movimento tem um ID, que serve para associar aos documentos. Fica tudo arquivado no Google Drive, em particular a folha Excel, para edição colaborativa.

JV: Uma coisa que não nos ocorreu foi essa **associação aos movimentos bancários**. Achas importante essa questão?

LE: Acho que é **essencial**. É uma parte necessária à contabilidade, que se chama **Conciliação Bancária**. É isto que a contabilidade faz -- para cada movimento tem de haver um documento. Para reduzir o nº de pedidos da contabilidade, dá-se-lhes toda a informação de que precisam.

JV: Imagina um novo software disponível na internet. Sentirias **conforto** que tenha acesso à tua conta bancária? De que modo, diretamente ou indiretamente, de forma manual?

LE: **O ideal seria, de forma automática, irem caindo lá os movimentos**. O sigilo bancário não seria um entrave. O interesse é evitar trabalho, especialmente trabalho aborrecido.

JV: Por semana, **quanto tempo** gastas em tarefas deste género, a introduzir dados no Excel?

LE: Antigamente realizava essas tarefas semanalmente, atualmente é no fim do mês, o que também pode ser um problema porque só nessa altura se ganha a visão real dos movimentos. Atualmente gasto **oito horas mensais**. (No fim da entrevista: talvez mais, depende dos problemas que surgem...)

JV: Acontece introduzir **erros** nesse processamento manual?

LE: É claro, algumas vezes. Tenta-se configurar o Excel para evitar erros. Uma questão é a diferença de formatos entre extratos de banco para banco, seja nos numéricos seja nas colunas usadas. Assim, uso uma folha com páginas que preparei para colar o extrato, e costumo elaborar algumas validações, mas **mesmo assim ocorre erro humano**.

JV: Neste momento, estamos a pensar: importar as faturas via portal das finanças, porque todas lá vão parar, embora tenha a desvantagem da altura em que os dados são carregados na AT...

LE: Até ao dia 15...

JV: ... Exato. Com a fatura digital, os dados poderão chegar mais cedo. Outra via que estamos a pensar é por digitalização de faturas, ou mesmo por inserção manual. Este é o foco principal, com um sistema para tratar as faturas. Categorias associadas a faturas, permitindo adicionalmente uma visão geral, em particular via gráficos e relatórios. O que achas destas funcionalidades ou outras e dados de que precisas?

LE: A análise que eu faço é sempre de **cash-flow e não contabilística** (que funciona por centros de custo). Além da categoria, será necessária a subcategoria. Exemplo: ter a área de marketing e lá dentro poder ter vários projetos. Ou seja, poder personalizar as categorias. Até porque os responsáveis das diferentes áreas depois querem saber como está a ser executado o seu orçamento, se estão ou não a cumprir, como podem equilibrar as coisas. Uma parte muito importante é a da conciliação bancária, porque ajuda-me a resolver problemas posteriores com a contabilidade. Primeiro organizo a

despesa orientada pela conciliação bancária: para cada movimento, em face do documento (fatura) respetivo, indico a categoria/subcategoria, adiciono uma descrição e altero o ID automático do movimento para o ID da fatura. A pensar na contabilidade. Outra necessidade é conhecer as receitas, o que foi faturado e do que foi faturado o que ainda não foi pago (recebido).

JV: Também pensámos em etiquetas (livres) em vez de subcategorias. A mesma etiqueta poderia ser associada a qualquer categoria. Isto poderia ser útil.

LE: No meu caso não, o que me interessa é saber o que gastei com dado projeto. Por exemplo, **a grande categoria é “Clientes”** e a **subcategoria é “Custos com projetos”**, subdividida por projeto. Servirá para criar relatórios de cada projeto e saber como estão a evoluir os nossos custos e os nossos lucros. Não tenho subcategorias repetidas: são um centro de custo que me interessa. Não me interessa saber que gastei X em alimentação, mas sim o que gastei num projeto.

JV: Em relação ao *income* (receitas), certamente usam software para emitir faturas. Como inseres no Excel?

LE: A ferramenta que nós usamos é o Invoice Express e é bom; quando quero fazer *tracking* de *income* é lá que eu vou, mas nem toda a equipa têm acesso (tem um custo de licenciamento), e o que faço é passar alguns valores para o Excel, atualizando tabelas de *invoicing* e recebimento, estes através dos movimentos (bancários). O objetivo é as pessoas saberem quanto se está a faturar e previsão de receita -- política de transparência. O que faço é copiar e colar. **Por cliente** coloco quando faturei, depois tenho uma tabela abaixo, **por categoria** de *income*, que tenho de atualizar -- pelo tipo de cliente sei que tipo de *income* é. Organizado assim: a categoria é o *income* (o cliente?), a subcategoria é o tipo de *income*, e uma coluna para o projeto. Acho ainda importante ter o *budget*, para saber quanto e como (acima ou abaixo do orçamento) se andou a gastar mensalmente em cada uma das subcategorias. Pode levar a ajustes de orçamento (para o mês seguinte). Todo este controlo é útil para saber quantos meses de sobrevivência temos sem depender dos clientes.

JV: Em termos de gráficos, há alguma coisa que gostasses de ter?

LE: O que eu uso é um gráfico de 3 ou 4 linhas com *income*, *outcome* e valor faturado. Ajuda a saber se tivemos um mês bom ou mau.

JV: E como geres as despesas com colaboradores?

LE: Uma das categorias é a equipa e como subcategorias os salários, os pagamentos ao Estado (IRS, Segurança Social, ...).

JV: Faria sentido um gráfico de distribuição por centro de custo?

LE: Para mim não, porque essa é uma perspetiva contabilística. O que me é útil no dia a dia é a análise de cash-flow por categoria e subcategoria.

JV: Faz sentido ter uma previsão de receitas e gastos, a seis doze meses, dois anos?

LE: Faço isso através do budget, que me permite conhecer a despesa média mensal e objetivos de despesa e income. Fundamental saber, para planejar, são também custos que só caem em dado mês, como anuidades de serviços.

JV: Ocorre-te mais alguma necessidade?

LE: Tenho uma folha com gastos com salários e me permite simular o aumento de despesa decorrente de aumentar salários.

JV: Achas que faz sentido todos os colaboradores terem acesso?

LE: Prefiro ferramentas em que possa adicionar todos os utilizadores que quiser, **uns que só possam consultar e outros “ativos” que possam alterar, com registo de quem fez certas alterações**. A folha excel que eu disponibilizo deixa meter comentários mas não deixa editar.

JV: Seria aceitável disponibilizar um nº máximo de utilizadores ativos e um valor livre de utilizadores de consulta?

LE: Sim.

JV: Faria sentido uma espécie de “audit log” de alterações, com indicação por utilizador.

LE: Sim.

JV: Quanto estarias disposta a pagar por uma solução destas? Quanto aceitavas pagar por utilizador ativo?

LE: Até 100 euros mensais para usar a ferramenta seria aceitável. Já agora, vi uma ferramenta que achei muito interessante, que poderiam analisar: Expensify (<https://www.expensify.com/>). Permite tirar uma foto a um documento e importar.

Fim da entrevista.

Entrevista a Francisco Maia (Keyruptive)

Entrevistador: João Vilaça.

04/11/2020. 44 minutos. Não é a entrevista literal, são excertos, **sintetizados e adaptados**, da entrevista:

JV: Como fazem a gestão das faturas que vão recolhendo (compras)?

FM: Temos três situações: a empresa que nos faz a contabilidade (EditValue), as faturas que emitimos e as faturas que emitem para nós. Para o contabilista, há um dia de entrega de documentos. Para a gestão das faturas emitidas, usamos o Moloni, o resto é feito manualmente. Tento manter uma cópia em papel. Usamos uma folha de excel para os **orçamentos**. Por vezes há falhas.

JV: Uma situação problemática é que por vezes entrega-se a fatura ao contabilista e depois é preciso consultar e ela já não está connosco.

FM: Exatamente.

JV: Como funciona então a tal folha de cálculo, que dados lá colocam e gerem?

FM: Metemos lá os **encargos com pessoal**, a **conta-corrente**, e fazemos uma **projeção** do que vamos faturar. Como somos uma empresa pequena, o que nos interessa saber são nossos custos fixos correntes para saber se é preciso fazer ajustes. O que não é fixo é muito pontual, como a compra de um computador.

JV: Não gerem as faturas de despesas do dia-a-dia?

FM: É muito pouco. O que nos faria mais jeito seria saber de faturas do passado e evolução ao longo do tempo. Há uma ferramenta da EditValue que tem alguma coisa sobre isso, mas ainda não chegámos ao ponto de utilizar isso muito bem.

JV: Com que frequência precisas de editar essa folha de excel.

FM: Ultimamente bastante, quase semanalmente, para fazer um ponto da situação.

JV: Quanto tempo gastas com essas tarefas?

FM: Uma hora e meia por semana.

JV: Supõe então uma ferramenta que atualizasse as faturas (de compra) pelo portal das finanças e permitisse introduzir no sistema faturas de modo automatizado, de modo a ter todas as faturas da empresa. Facilitaria também a conciliação bancária com automatismos. Um tal sistema seria proveitoso? Gastam tempo a gerir as faturas?

FM: Em relação à conciliação bancária, seria com um sistema que, dado um movimento bancário, assinalasse que falta o respetivo justificativo. Acontece que eu faço conciliação bancária mas depois a EditValue (contabilista) também faz. Acaba por ser trabalho duplicado. Se a conciliação bancária ficasse logo disponível para a EditValue seria ótimo. Como faço: imprimo o extrato de movimentos e vou conferindo o extrato, e depois a EditValue faz a mesma coisa. Às vezes falha algum, verifica-se que falta a fatura e tem de se conseguir uma segunda via.

JV: Pensamos fazer parcerias com empresas como a Primavera e contabilistas. Será possível produzir um relatório que eles possam usar.

FM: Quanto à questão do OCR, pode ser difícil por todas as faturas terem formato diferente. Mas será bom que reconheça o mais possível da informação, sem obrigar a preencher muitos campos.

JV: A ideia seria identificar o mínimo necessário, sem obrigar a preencher muita informação. Outra questão: Usando um software como o que estamos a projetar, sentir-te-ias confortável em ter lá a informação bancária e associá-lo ao acesso à conta bancária?

FM: 100% confortável não, mas o automatismo compensaria. Como será uma solução cloud, terá de ser justificada a **segurança da informação**, para não se extraviar.

JV: Quanto aos **centros de custo**, pensamos em associar **categorias e subcategorias**.

FM: Os centros de custos são uma necessidade mais da contabilidade que nossa. Contudo, será bom **associar faturas a projetos** para fazer somatórios e saber quanto custou o projeto. Uma coisa boa será ter uma **previsão e respetiva execução de gastos**.

JV: O projeto seria a categoria.

FM: Sim. Um centro de custos fica associado a um projeto. Irá haver um *bugdet* e execução. Tudo isto só será útil se tiver associada uma componente de recursos humanos.

JV: Onde fazem o processamento de salários?

FM: É informação que a EditValue gera, produzindo um recibo de vencimento que será o documento de custos. Em termos de lógica, não precisam de processar o vencimento, mas devem poder referenciar o recibo de maneira especial. Assim, teremos movimentos bancários de despesa que não são faturas, mas antes vencimentos e impostos.

JV: Para gerar gráficos que elucidem sobre a situação financeira, estamos a pensar pedir valores manuais.

FM: Será bom ter uma visão global. O mais importante é pode ser as faturas (receitas e despesas) e ter num gráfico esses totais, salários e impostos. Com quatro cores consegue-se ter uma visão rápida do que está a acontecer. Não será necessário muito detalhe.

JV: Lembras-te de mais alguma funcionalidade que gostasses de ter?

FM: Seria bom um sistema que tentasse automatizar coisas como os pagamentos de salários...

JV: Estamos focados em substituir o excel e facilitar esse tipo de gestão. E substituir o papel para arquivar digitalmente.

FM: Posso dar um exemplo com uma app de gestão de voos. Cheguei a ter uma. Compravas o teu voo, fazias a reserva e só tinhas de fazer *forward* do email para um email específico que a aplicação gerava e te fornecia. A app determinava quem estava a enviar o email (o meu), processava a reserva de voo e preenchia na minha app os dados do voo. Funcionava espetacularmente bem. Podiam fazer algo adaptado a faturas, fazendo *forward* para uma zona de faturas pendentes. Há impressoras que permitem fazer scan e enviar por email.

JV: Como gostarias que fosse a subscrição duma aplicação deste género?

FM: O titular devia ser a empresa, poder escolher que quer aceder e como pagar, mensalmente ou anualmente.

JV: Pensamos em utilizadores ativos e passivos.

FM: Cuidado com o preçário. Um utilizador passivo pode ter pouco interesse, porque depois não pode corrigir nada. Será mais interessante um plano ilimitado, a baixo custo.

JV: Será útil um *log* de operações por utilizador?

FM: Numa empresa maior, penso que será útil saber quem cometeu dado erro.

JV: Supões um plano até 10 utilizadores ativos e ilimitado nos passivos. Quanto estarias disposto a pagar?

FM: Dez euros por mês, tendo em conta que continua a ser necessário pagar um sistema de faturação (emissão) e ao contabilista.

JV: Achas que faz sentido o preço depender do nº de faturas?

FM: Penso que sim, mas tornava-se mais difícil prever os custos (tinha de se contar as faturas).

JV: Alguma coisa a acrescentar?

FM: Seria bom, mais tarde, esse sistema poder **importar dados de outras aplicações**.

JV: Interessado em experimentar mais tarde um protótipo?

FM: Com certeza!

Fim da entrevista.

Entrevista a Adriano Martins (Explore.dev)

Entrevistador: João Vilaça.

04/11/2020. 45 minutos. O resumo possível da entrevista, sintetizado e adaptado (a entrevista foi difícil de sintetizar devido à quantidade de diálogo informal):

JV: Estamos a considerar obter faturas do portal das finanças, e também disponibilizar o scan de faturas e obter a informação principal. Também será possível introduzir faturas manualmente, evitando o excesso de campos. Prevê-se ainda implementar a conciliação bancária. As faturas serão associadas a categorias.

A: Usamos o **toconline**. Contudo, é um sistema que está sempre a mudar, e apesar de fazer muitas coisas, é mais **direcionado a contabilistas e muito complexo**. Tem uma parte para gestão financeira. Uma dificuldade que tenho é processar faturas que chegam. Gerir as faturas recebidas, algumas por email, será importante.

JV: Questões fulcrais: a conciliação bancária e gestão de faturas.

A: **Poderia ser vantajoso fazer a ponte com o toconline** (sistema este que continuo a considerar muito complicado, apesar de já lidar com ele há um ano). A contabilidade,

ou a preparação dela, acaba por levar muito tempo. Ter uma plataforma, a preço acessível, para empresas a começar, seria muito bom.

...

A: toconline integra com o Millennium BCP (banco).

...

A: Uma gestão de tarefas poderia ser útil.

...

Fim da entrevista.

Observação

Não foi realizada observação de atores a desempenhar processos, no entanto alguns membros da equipa já realizaram tarefas de controlo financeiro em associações.

Personas

Dada a rápida evolução do projeto, ainda não foram elaboradas personas. Remete-se para as entrevistas como proposta de descoberta de personas mais concretas.

Introspeção

Toda a introspeção foi validada sob brainstorming pela equipa.

Anexo III. Detalhes da Análise Formal

Neste anexo podemos encontrar os detalhes da Análise Formal (cf. secção homónima).

Totalidade das assinaturas

```
sig Company {  
    var users: set User,  
    var invoices: set Invoice,  
    var categories: set Category,  
    var tags: set Tag  
}  
var sig Registered in Company {}  
  
sig User {}  
sig Tag {}  
sig Budget {}  
  
sig Category {  
    var budget: lone Budget  
}  
  
sig Invoice {  
    var company: one Company,  
    var category: one Category,  
    var tags: set Tag  
}
```


Totalidade dos invariantes

```
check UsersBelongToOnlyOneCompany {
  always (
    all disj c1, c2 : Registered | no c1.users & c2.users
  )
}

check CompanyHasAtLeastOneUser {
  always (
    all c : Registered | some c.users
  )
}

check InvoicesBelongToOnlyOneCompany {
  always (
    all disj c1, c2 : Registered | no c1.invoices & c2.invoices
  )
}

check CategoriesBelongToOnlyOneCompany {
  always (
    all disj c1, c2 : Registered | no c1.categories & c2.categories
  )
}

check TagsBelongToOnlyOneCompany {
  always (
    all disj c1, c2 : Registered | no c1.tags & c2.tags
  )
}

check BudgetsBelongToOnlyOneCompany {
  always (
    all disj c1, c2 : Registered | all cat1 : c1.categories | all cat2 : c2.categories |
    no cat1.budget & cat2.budget
  )
}

check BudgetsBelongToOnlyOneCategory {
  always (
    all c : Registered | all disj cat1, cat2 : c.categories | no cat1.budget & cat2.budget
  )
}
```

```

check InvoiceCompanyIsConsistent {
  always (
    all c : Registered | all i : c.invoices | one i.company and i.company = c
  )
}

check InvoiceCategoryBelongsToCompany {
  always (
    all c : Registered | all i : c.invoices | one i.category and i.category in c.categories
  )
}

check InvoiceTagsBelongToCompany {
  always (
    all c : Registered | all i : c.invoices | i.tags in c.tags
  )
}

```

Bibliografia

- [Est19] António J. Esteves (2019), Universidade do Minho. ***“Escrita de Requisitos de acordo com o Modelo de Volere”***, apresentação de diapositivos.
- [Fer15] João M. Fernandes (2015), Universidade do Minho. ***Curso “Engenharia de Requisitos”***, apresentação de diapositivos.
- [FM15] João M. Fernandes, Ricardo J. Machado (2015). ***Requirements in Engineering Projects***.
- [Mig13] António Miguel, 2013. ***Gestão Moderna de Projetos***. 7ª ed. Lisboa: FCA.
- [Pal16] Tiago Palhoto, 2016. ***Desenvolvimento Ágil de Software***, Guia Prático. 1ª ed. Lisboa: FCA.