



UNIVERSIDADE
LUSÓFONA

Sistema de gestão de produtos com ligação ao **InvoiceXpress**

Rodrigo de Castro Sampaio Cassanheira | 21703091

Orientador: Prof. Dr. Pedro Arroz Serra

Trabalho Final de Curso | LEI | 24/11/2019

www.ulusofona.pt

Direitos de cópia

Sistema de gestão de produtos com ligação ao InvoiceXpress Copyright de Rodrigo de Castro Sampaio Cassanheira, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Índice

Resumo	6
Abstract	7
1. Identificação do Problema	8
2. Viabilidade e Pertinência	10
3. Levantamento e análise de requisitos	11
3.1 Requisitos Funcionais	11
3.2 Requisitos Não Funcionais	18
4. Solução Desenvolvida.....	19
4.1 Abordagem.....	19
4.2 Solução.....	21
5. Benchmarking	22
6. Resultados	23
7. Método e planeamento	24
8. Conclusão.....	25
Manual de utilizador da aplicação	26
Bibliografia	28
Glossário	29

Índice de Figuras

Figura 1 - Número médio de SaaS por empresa	10
Figura 2 - Diagrama da arquitetura do sistema	20
Figura 3 - Login	26
Figura 4 – Menu Clientes.....	26
Figura 5 - Menu Artigos	27
Figura 6 - Menu Utilizadores	27

Índice de Tabelas

Tabela 1 - Projecto Colibri vs InvoiceXpress	22
Tabela 2 - Testes feitos à plataforma	23
Tabela 3 - Calendário.....	24

Resumo

Vivemos numa época em que o comércio tradicional está a evoluir para comércio eletrónico, gerando assim novas oportunidades de negócio. O presente trabalho consiste no desenvolvimento de uma aplicação web com a finalidade de fazer a gestão e controlo de produtos. A aplicação é apenas acedida pelos utilizadores autorizados, neste caso, os funcionários da empresa. Os principais objectivos da plataforma são fazer o controlo do stock a partir das entradas e saídas de produtos e estimar o custo dos produtos compostos.

A plataforma contém uma base de dados com clientes, fornecedores e produtos. Os clientes podem comprar produtos compostos ou serviços, e os fornecedores, por outro lado, vendem as suas matérias primas e / ou produtos. As matérias primas fornecidas têm origem mineral e irão resultar em peças de joalheria ou ourivesaria. Depois de as matérias primas serem adquiridas, um utilizador da plataforma, encarregar-se-á de preencher um formulário com todos os dados da fatura. Dentro destes dados, está incluída a lista de matérias primas e / ou produtos adquiridos que irão entrar em stock.

Sempre que é criado um produto composto de matérias primas, estas serão retiradas de stock proporcionalmente ao seu uso no produto em questão. Por fim, a plataforma ao vender um produto composto a um cliente, irá gerar uma fatura com todos os dados necessários.

Palavras-chave: Aplicação web. Cliente. Fatura. Fornecedor. Produtos.

Abstract

We live in a time where traditional commerce is evolving into e-commerce, generating new business opportunities. The present work consists in the development of a web application with the use of managing products. The application is only accessed by authorized users, in this case, company employees. The main objectives of the platform are to control the stock from the inflows and outflows of products and estimate the cost of composed products.

The platform contains a database with customers, suppliers, and products. Customers can buy composed products, and suppliers, on the other hand, sell their raw materials and / or products. The supplied raw materials have a mineral origin and are transformed into jewelry or goldsmithing pieces. Once these raw materials are purchased, a platform user is going to fill out a form with all invoice data. This data includes a list of raw materials and / or products that will enter in stock.

Whenever a composed product of raw materials is created, the raw materials are taken out of stock in proportion to their use in that product. Finally, the platform when selling a composed product will generate an invoice with the necessary data of the user.

Keywords: Client. Invoice. Products. Supplier. Web application.

1. Identificação do Problema

O negócio de empresas no sector da indústria transformadora baseia-se na compra de matérias primas e transformação dessas matérias em produtos finais prontos a serem vendidos ao cliente final. A compra das matérias primas é feita a diferentes cotações e, portanto, o preço de custo dos produtos acabados (compostos), depende de vários fatores:

- Preço de custo das matérias primas usadas
- Mão-de-obra
- Custos com maquinaria
- Eletricidade, consumíveis, etc.

É indispensável a estas empresas um software que lhes permita fazer uma gestão de produtos e de stock e que seja capaz de calcular com precisão o preço de custo de cada produto. Uma vez criados os produtos com todas as informações necessárias é então necessário que este software seja capaz de registar as vendas aos clientes finais dando assim saída de stock.

Fazer um software de faturação é um processo complexo, moroso, e que requer certificação por parte da autoridade tributária. Além disso, são necessárias atualizações constantes, já que há uma dependência muito forte da legislação em vigor. No entanto existem várias empresas que fornecem APIs para faturação na Cloud. Estas soluções SaaS (software as a service), garantem todas as funcionalidades esperadas de um software de faturação com atualizações constantes e backups automáticos.

Este trabalho foi dividido em duas partes:

1. Componente produção: criação de produtos compostos e introdução de matérias primas;
2. Componente de faturação: criação de clientes e lançamento de faturas utilizando o InvoiceXpress.

Este TFC focou-se apenas na parte 2. A parte 1 estava planeada ser desenvolvida por outro aluno, mas acabou por não se realizar. Irá ser desenvolvida no futuro.

Após o desenvolvimento da plataforma houve algumas funcionalidades que não ficaram como inicialmente esperado. Nomeadamente na criação de faturas, faturas-recibo e notas de crédito.

O utilizador da plataforma apenas consegue inserir um produto em vez de vários por cada operação referida.

2. Viabilidade e Pertinência

As soluções cloud (SaaS) para a gestão comercial das empresas têm vindo a mudar a forma como estas trabalham. Atualmente usar um software cloud para fazer a gestão das faturas de uma empresa é muito mais conveniente, uma vez que se evita o processo de certificação [1]. Um software certificado de faturação necessita de ter uma manutenção constante devido às alterações na lei. Caso o software não esteja a gerar documentos de acordo com a lei, pode resultar em coimas pesadas para a empresa. Por outro lado, a maioria dos programas de faturação que existem não permitem a criação de produtos novos a partir de produtos em stock, ou seja, a criação e gestão de produtos compostos.

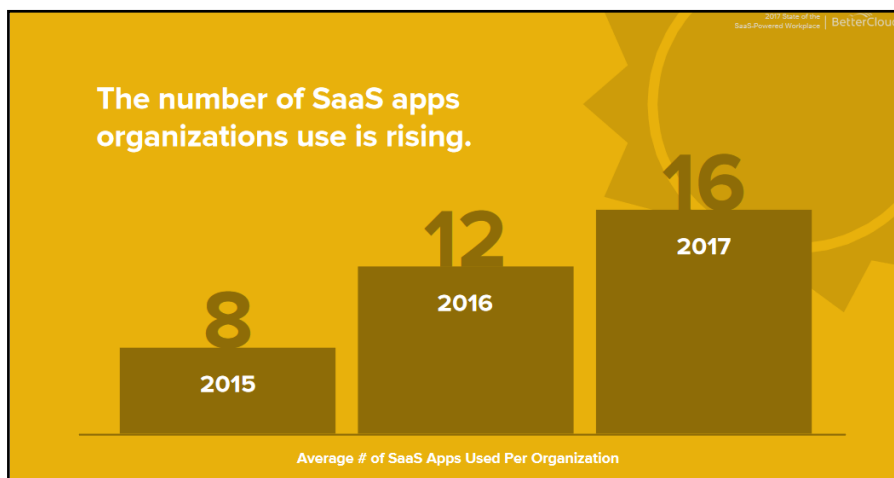


Figura 1 - Número médio de SaaS por empresa

Como se pode ver na Figura 1 as empresas nos dias de hoje usam até 16 SaaS nas suas soluções de negócio [3].

O presente trabalho tem como objetivo substituir a plataforma concorrente no mercado. A plataforma concorrente necessita de mais manutenção o que representam mais gastos por parte da empresa. O software de gestão comercial usado também não é o mais indicado por trazer complexidade desnecessária à solução devido a ser um software standalone.

A plataforma desenvolvida não tem o problema mencionado devido a usar um software de faturação que foi desenvolvido em cloud.

3. Levantamento e análise de requisitos

Após o levantamento de requisitos foi feita uma análise criteriosa em que se identificou os seguintes requisitos:

3.1 Requisitos Funcionais

Requisito Funcional 1 (RF01) – Cumprimento Integral

Autenticação do utilizador da plataforma.

Precondição:

O sistema disponibilizará ao utilizador um ecrã para fazer o seu login na plataforma.

O utilizador tem de estar registado.

Critérios de aceitação:

O sistema permite a autenticação do utilizador em caso de sucesso.

A autenticação deve ser feita através da introdução do nome de utilizador e da palavra-passe.

Criação de um produto.

É constituído pelos seguintes dados:

1. Descrição
2. Natureza
3. Categoria
4. Taxa de IVA
5. Razão de Isenção
6. País
7. Unidades
8. Peso da unidade
9. PVP

Precondição:

O utilizador tem de estar autenticado.

Critério de aceitação:

O produto é criado com os dados fornecidos.

Criação de um cliente.

É constituído pelos seguintes dados:

1. Prefixo
2. Nome
3. Data de nascimento
4. Número de telemóvel
5. Medidas (RF04)
6. Género
7. Estado civil
8. Cônjuge
9. NIF
10. Número do cartão de cidadão
11. Número do passaporte
12. Número de telefone
13. Número alternativo de telefone
14. Email
15. Site
16. Empresa
17. Fax
18. Morada
19. Mailing list
20. Código postal
21. Cidade
22. País
23. NIB
24. Descrição
25. Desconto
26. Data da última visita

Precondição:

O utilizador tem de estar autenticado (RF01).

Critério de aceitação:

O cliente é criado com os dados fornecidos.

Requisito Funcional 4 (RF04) – Cumprimento Integral

Criação da medida do cliente ou cônjuge.

É constituída pelos seguintes dados:

1. Mindinho esquerdo
2. Mindinho direito
3. Anelar esquerdo
4. Anelar direito
5. Dedo do meio esquerdo
6. Dedo do meio direito
7. Indicador esquerdo
8. Indicador direito
9. Polegar esquerdo
10. Polegar direito
11. Pulso esquerdo
12. Pulso direito
13. Pescoço

Precondição:

Existir um cliente ou cônjuge associado à medida.

Critérios de aceitação:

A medida é criada com os dados fornecidos.

É associado ao cliente ou cônjuge respetivo.

Criação do fornecedor.

É constituído pelos seguintes dados:

1. Nome
2. Acrónimo
3. Email
4. NIF
5. Número de telemóvel
6. Número de telefone
7. Número alternativo de telefone
8. Fax
9. Site
10. Mailing list
11. Morada
12. Código postal
13. Cidade
14. Contacto

Precondição:

O utilizador tem de estar autenticado (RF01).

Critério de aceitação:

O fornecedor é criado com os dados fornecidos.

Requisito Funcional 6 (RF06) – Cumprimento Integral

Criação do utilizador da plataforma.

É constituído pelos seguintes dados:

1. Nome
2. Palavra-passe
3. Nível

Precondição:

N/A

Critérios de aceitação:

O utilizador é criado com os dados fornecidos.

A palavra-passe é guardada na base de dados cifrada.

Requisito Funcional 7 (RF07) – Cumprimento Integral

O sistema deve oferecer informação a sistemas externos via uma interface.

A informação pode ter as seguintes formas:

- Cliente
- Produto

Precondição:

A informação deve existir no sistema (RF03 / RF04).

Critérios de aceitação:

A informação enviada é a desejada.

Requisito Funcional 8 (RF08) – Cumprimento Integral

O sistema deve oferecer a sua informação (RF07) a sistemas externos no formato XML através de uma interface que está disponível a clientes REST.

Precondição:

A informação deve existir no sistema (RF03 / RF04).

Critérios de aceitação:

A informação enviada é a desejada.

Requisito Funcional 9 (RF09) – Cumprimento Parcial

O sistema deve permitir ao utilizador introduzir os dados para a criação de uma fatura a um cliente.

Precondição:

O utilizador tem de estar autenticado (RF01).

Existir o cliente.

Critérios de aceitação:

O stock dos produtos é ajustado.

A fatura é constituída pelos dados inseridos.

Requisito Funcional 10 (RF10) – Não Implementado

O sistema deve permitir ao utilizador introduzir os dados de uma fatura de um fornecedor.

Precondição:

O utilizador tem de estar autenticado (RF01).

Existir o fornecedor.

Critérios de aceitação:

O stock dos produtos é ajustado.

Requisito Funcional 11 (RF11) – Cumprimento Parcial

O sistema deve permitir ao utilizador criar uma nota de crédito a um cliente.

Precondição:

O utilizador tem de estar autenticado (RF01).

Existir o cliente.

Critérios de aceitação:

O cliente fica com a nota de crédito associada.

Requisito Funcional 12 (RF12) – Não Implementado

O sistema deve permitir ao utilizador criar um recibo a um cliente.

Precondição:

O utilizador tem de estar autenticado (RF01).

Existir o cliente.

Critérios de aceitação:

O recibo é constituído pelos dados inseridos.

3.2 Requisitos Não Funcionais

Requisito Não Funcional 1 (RNF01) – Cumprimento Integral

O sistema deve usar a base de dados relacional MySQL 8.0.18

Precondição:

N/A

Critérios de aceitação:

O sistema consulta os dados da base de dados relacional.

4. Solução Desenvolvida

4.1 Abordagem

As tecnologias definidas para o desenvolvimento da plataforma são o Django, MySQL e por fim o uso da API do InvoiceXpress.

O Django é uma framework web em Python que inclui ferramentas para lidar com tarefas comuns do desenvolvimento web, como por exemplo, a autenticação e uma interface admin. Esta framework web segue o padrão de arquitetura model-view-template. O model é uma camada de acesso a dados e é responsável por os manter consistentes. A view é a camada responsável por receber os dados, assim como os métodos HTTP, para que os dados sejam passados para o model. Por fim, a template consiste em partes do HTML desejado assim como alguma sintaxe especial do Django denominada por “Django template language”.

O model do Django vem com uma base de dados de origem a SQLite, no entanto como é pouco escalável será substituída pelo MySQL. O MySQL é um SGBD que é bastante usado para data warehousing. Quando usado com Python irá interagir em tempo real para exibir rapidamente informações categorizadas e estruturadas a um usuário do sistema. Um dos recursos mais poderosos do Django é o ORM (object relational model), que permite que se interaja com a base de dados independentemente da tecnologia usada para esta. Facilitando assim a manipulação e a consulta de dados.

O InvoiceXpress é um programa de faturação online que disponibiliza a sua API para a criação de faturas. Esta API será integrada na plataforma para gerir a faturação.

Na Figura 2 está representado um pequeno diagrama da arquitetura do sistema.

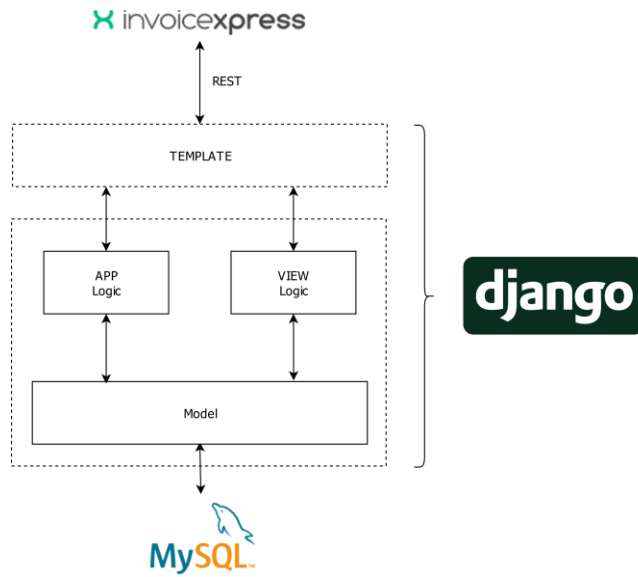


Figura 2 - Diagrama da arquitetura do sistema

4.2 Solução

Durante o desenvolvimento da plataforma foram encontrados diversos obstáculos que implicaram a instalação de vários pacotes em Python através do pip. Tais como 'betterforms', 'django_tables2', 'django_filters', 'xmldict'.

O betterforms foi usado com o objetivo de ter vários formulários de models diferentes e os poder relacionar todos na mesma view e template do Django.

O django_tables2 permite com a ajuda do django_filters obter uma tabela de um model para conseguir fazer uma pesquisa filtrada de um objeto que exista na base de dados. Esta tabela é bastante útil porque só assim o utilizador sabe a primary key de um cliente para conseguir enviar uma fatura para a api do Invoicexpress.

Por fim foi usado o xmldict que transforma um ficheiro XML em dicionário, assim é mais fácil fazer a manipulação dos dados.

5. Benchmarking

A plataforma concorrente no mercado, neste momento, está a usar o software Projecto Colibri para fazer a sua gestão comercial.

O Projecto Colibri é um software que funciona através de uma instalação standalone. Para se tornar funcional necessita de estar exposta num webservice desenvolvido por uma pessoa. Não oferece estabilidade, porque, devido às possíveis mudanças na legislação em vigor o webservice necessita de ser atualizado, logo, obriga o pagamento de horas ao programador que o fez para o atualizar.

Uma solução cloud como o InvoiceXpress é muito mais viável. Não é necessário estar a instalar software nem em adquirir servidores. E está sempre atualizado de acordo com a legislação em vigor. Uma grande vantagem de uma solução cloud é que na sua acessibilidade, ao contrário de software standalone, pode ser acedida em qualquer computador. No caso da escalabilidade um software cloud a empresa pode escolher a qualquer momento melhorar o seu plano, num software standalone é necessário comprar mais servidores se houver falta de armazenamento [2].

Software	Necessário fazer instalação	Acessível remotamente	Atualizado com a legislação	Escalável ao momento	Necessário fazer manutenção
Projecto Colibri	Sim	Não	Não	Não	Sim
InvoiceXpress	Não	Sim	Sim	Sim	Não

Tabela 1 - Projecto Colibri vs InvoiceXpress

6. Resultados

NRº	Tipo	Descrição	Resultado Esperado	Resultado Obtido
1	Funcional	Testar o RF01 com valores incorretos.	A autenticação não é efetuada.	A autenticação é negada e ocorre um erro que informa o utilizador.
2	Funcional	Testar o RF02 preenchendo os valores do formulário obrigatórios.	O produto é criado.	O produto é criado com os valores fornecidos no InvoiceXpress e na base de dados.
3	Funcional	Testar o RF03 preenchendo os valores do formulário obrigatórios	O cliente é criado com uma medida, e um cônjuge associado e esse mesmo cônjuge com uma medida associada. As medidas são únicas e distintas.	Resultado esperado. O cliente também é criado no InvoiceXpress.
4	Funcional	Testar o RF06 com valores que já estão a ser usados.	O utilizador não é criado.	Resultado esperado. Ocorre um erro que informa o utilizador.
5	Funcional	Testar o RF07 com clientes inexistentes.	Um erro é ocorrido.	Ocorre o erro 404 da API, visto que o cliente não existe no InvoiceXpress.
6	Funcional	Testar o RF07 com produtos inexistentes.	Um erro é ocorrido.	Ocorre o erro 404 da API, visto que o produto não existe no InvoiceXpress.

Tabela 2 - Testes feitos à plataforma

7. Método e planeamento

Atividade \ Mês	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maió	Junho
Levantamento de Requisitos								
Análise de Requisitos								
Implementação								
Testes								
Lançamento								

Tabela 3 - Calendário

Na primeira semana de Novembro foram retirados os requisitos de alto-nível.

Até Dezembro está prevista a análise de todos os requisitos para o desenvolvimento do trabalho.

De Dezembro até meados de Abril será feita a implementação do trabalho.

De Abril até Junho serão feitos testes na plataforma.

Em Abril foi feito o primeiro lançamento da plataforma através do Heroku.

E por fim em Junho será feito o lançamento da plataforma.

Em Junho foi feita uma atualização na plataforma devido à chave da API estar expirada.

8. Conclusão

Após a realização deste TFC adquiri competências que me poderão a vir a ser úteis no futuro. O desenvolvimento por mim feito no TFC não esgotou o tema podendo assim estar em aberto para novas alterações e aprofundamentos.

Manual de utilizador da aplicação

1. Entrar no site <https://django-deisi95.herokuapp.com/>
2. Digitar o nome de utilizador e palavra-passe (deisi95 e tfc, respetivamente)



The screenshot shows a login form titled "Login". It contains two input fields: "Username*" and "Password*", both with empty text boxes. Below the fields is a green button labeled "Login".

Figura 3 - Login

3. Criar novo cliente – carregar em **Clientes > Novo Cliente**

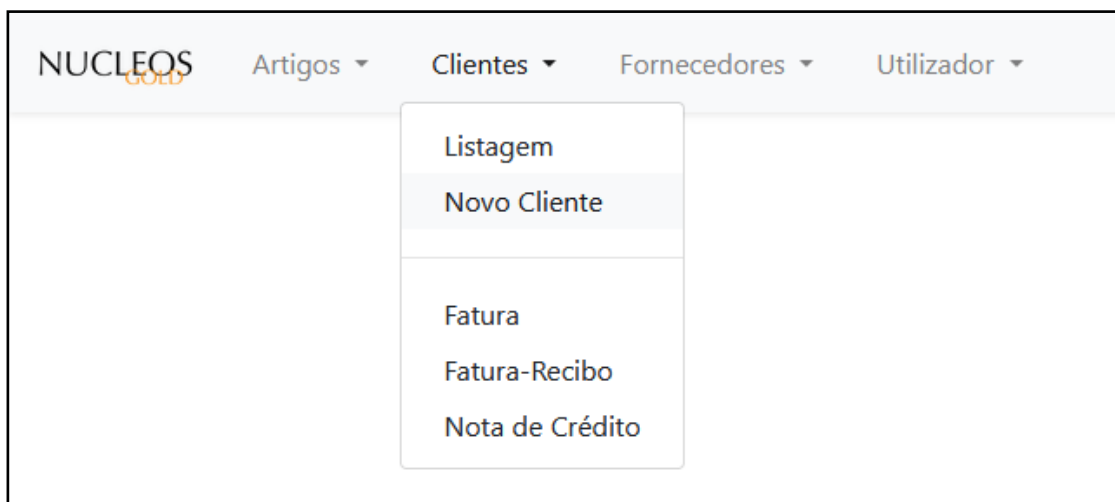


Figura 4 – Menu Clientes

4. Criar fatura – Carregar em **Clientes > Fatura**
5. Criar produto – Carregar em **Artigos > Novo produto**

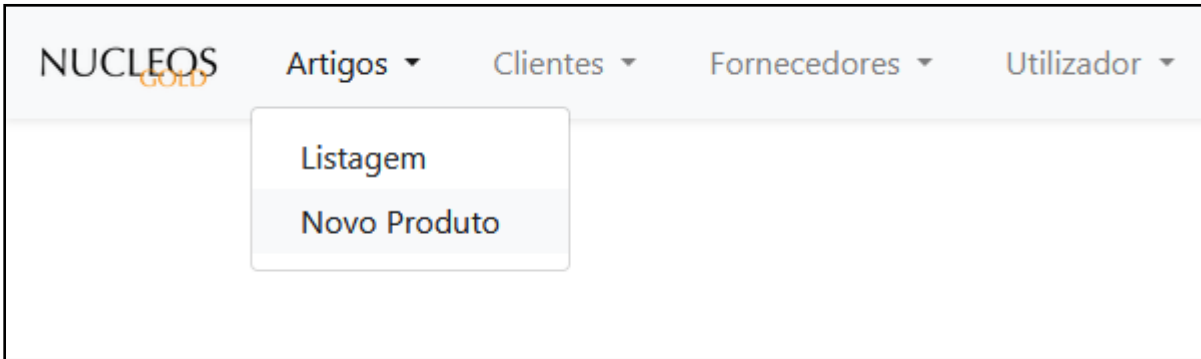


Figura 5 - Menu Artigos

6. Criar utilizador – Carregar em **Utilizador > Novo utilizador**



Figura 6 - Menu Utilizadores

Após a escolha da operação na plataforma devem ser preenchidos todos os campos do formulário marcados com asterisco (*), pois são os obrigatórios.

Bibliografia

- [1] M. Creed, “clinked,” [Online]. Available: <https://blog.clinked.com/standalone-vs.-cloud-software-for-accountants>. [Acedido em 26 1 2020].
- [2] E. Team, “Elorus,” [Online]. Available: <https://www.elorus.com/blog/10-good-reasons-your-company-should-use-cloud-invoicing/>. [Acedido em 26 1 2020].
- [3] D. Politis. [Online]. Available: <https://www.bettercloud.com/monitor/state-of-the-saas-powered-workplace-report/>. [Acedido em 29 01 2020].

Glossário

API – permite que um software interaja com sistemas externos.

REST – arquitetura que permite interagir com recursos através de métodos HTTP.

SaaS – forma de distribuição e comercialização de software.

Standalone – programa autossuficiente que precisa de ser instalado.

Stock – quantidade atual de um produto.

XML – formato usado para descrever documentos ou dados.

Pip – gerenciador e instalador de pacotes