



Carlos Filipe Vilaça Rebelo

**Desmaterialização de processos de negócio  
com recurso a aplicações para dispositivos  
móveis**

**Universidade do Minho**  
Escola de Engenharia





**Universidade do Minho**

Departamento de Sistemas de Informação

Carlos Filipe Vilaça Rebelo

**Desmaterialização de processos de  
negócio com recurso a aplicações  
para dispositivos móveis**

Projeto de Dissertação de Mestrado

Desenvolvido sob a orientação do

**Prof. Doutor Miguel António Sousa Abrunhosa Brito**

Mestrado Integrado em Engenharia e Gestão de Sistemas de Informação.

Junho de 2020

---

## DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

### *Licença concedida aos utilizadores deste trabalho*



Atribuição  
CC BY

<https://creativecommons.org/licenses/by/4.0/>

## **AGRADECIMENTOS**

Terminando este projeto e o meu percurso académico, gostaria de deixar alguns agradecimentos a todos aqueles que me acompanharam e me auxiliaram durante estes anos todos.

Ao meu orientador, o Professor Doutor Miguel Brito, por todo o conhecimento e orientação que me forneceu ao longo de todo este projeto.

À minha namorada pelo apoio moral, emocional e académico que me deu ao longo de todo o meu percurso sem nunca desistir de mim.

Aos meus pais por sempre me apoiaram e me incentivaram a ser melhor.

E um especial agradecimento à PrimaveraBSS por me terem fornecido esta oportunidade e a todos aqueles que me acompanharam durante todo o meu projeto.

## **DECLARAÇÃO DE INTEGRIDADE**

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

## **Desmaterialização de processos de negócio com recurso a aplicações para dispositivos móveis**

### **Resumo**

De forma a estudar e perceber futuros desenvolvimentos a nível interno, a PrimaveraBSS recorre a um processo de desmaterialização de processos integrado com mobilidade empresarial, de forma a suportar uma visão de portabilidade e conseguir potencializar o uso do ERP V10.

De forma a dar resposta a este problema, será necessário o estudo e compreensão do modelo de negócio e processos de iteração e integração praticados pela mesma, de forma a identificar forma de potenciar uma maior eficácia no transporte e tratamento de informação.

Esta questão será trabalhada em ambiente de consultoria, tendo como principal objetivo o estudo de um cenário hipotético de interação de colaboradores da organização com clientes, seguindo o modelo de negócio atual praticado, simplificando o fluxo de trabalho e eliminando todos os processos redundantes, enquanto tenta emular o ERP atual em formato de aplicação móvel.

O ponto fulcral do problema será a integração dos sistemas internos com uma aplicação móvel que consiga dar resposta à mobilidade requerida num ambiente de negócio, disponibilizando toda a informação e serviços necessários que tornem mais eficiente todo o processo de negócio da organização.

A principal motivação para este projeto é a contínua evolução do ERP V10 da Primavera, continuando a sua inovação e conseguindo retirar todo o seu potencial como uma solução de gestão e contabilidade.

**Palavras Chave:** desmaterialização, mobilidade, otimização, processos, sistemas

## **Dematerialization of business processes using applications for mobile devices**

### **Abstract**

In order to study and understand future internal developments, PrimaveraBSS uses dematerialization of process integrated with enterprise mobility to support a vision of portability and to enhance the use of the ERP V10.

In order to respond to this problem, it will be necessary to study and understand the business model and the iteration and integration processes practiced by it, in order to identify ways to increase the efficiency of the transportation and processing of information.

This issue will be addressed in a consulting environment, with the main objective of studying a hypothetical scenario of interaction of employees of the organization with customers, following the current business model, simplifying the workflow and eliminating all redundant processes, while trying to emulate the current ERP in mobile application format.

The center of the problem will be the integration of internal systems with a mobile application that can meet the required mobility in a business environment, providing all the necessary information and services that make the entire business process of the organization more efficient.

The main motivation for this project is the continued evolution of Primavera's ERP V10, continuing its innovation and being able to tap its full potential as a management and accounting solution.

**Keywords:** dematerialization, mobility, optimization, processes, systems

---

## Índice

<i>Licença concedida aos utilizadores deste trabalho</i> .....	iii
<b>AGRADECIMENTOS</b> .....	<b>iv</b>
<b>Resumo</b> .....	<b>vi</b>
<b>Abstract</b> .....	<b>vii</b>
<b>Índice</b> .....	<b>viii</b>
<b>Índice de Figuras</b> .....	<b>x</b>
<b>Índice de Tabelas</b> .....	<b>xiii</b>
<b>Lista de Siglas e Acrónimos</b> .....	<b>xiv</b>
<b>1. Introdução</b> .....	<b>1</b>
1.1. Enquadramento.....	1
1.2. Problema e Motivação .....	1
1.3. Objetivos e resultados esperados .....	2
1.4. Estrutura do documento .....	3
<b>2. Metodologias de Desenvolvimento e Investigação</b> .....	<b>5</b>
2.1. Scrum Solo.....	5
2.2. Design Science Research .....	8
<b>3. Revisão de Literatura</b> .....	<b>10</b>
3.1. Desmaterialização de Processos .....	10
3.2. Fatores críticos de sucesso da desmaterialização de processos.....	11
3.3. Mobilidade empresarial .....	12
3.4. Desafios da Mobilidade .....	14
3.5. <i>Rich mobile applications</i> .....	14
3.6. Paginação em SQL .....	16
3.6.1. ROW_NUMBER/TOP.....	17
3.6.2. OFFSET / FETCH .....	18



---

<b>4. Infraestrutura de desenvolvimento .....</b>	<b>20</b>
4.1. ERP V10.....	20
4.2. Kalipso .....	22
<b>5. Plano de projeto .....</b>	<b>24</b>
<b>6. Trabalho realizado .....</b>	<b>27</b>
6.1. Análise e Desenho .....	27
6.2. Realização.....	30
6.3. Testes e Validação .....	32
6.4. Contratempos e limitações .....	33
<b>7. Aplicação móvel .....</b>	<b>35</b>
7.1. Estrutura de dados .....	36
7.2. Arquitetura Lógica e utilização.....	38
7.3. Paginação via listas personalizadas.....	47
<b>8. Processo de Scrum Solo .....</b>	<b>49</b>
<b>9. Teste de validação e aceitação do projeto .....</b>	<b>55</b>
9.1. Login .....	59
9.2. Listagem e filtragem de clientes/ documentos/ catálogo .....	60
9.3. Criação de clientes.....	61
9.4. Criação de documentos/Integração de documentos guardados.....	63
<b>10. Discussão sobre a realização do projeto .....</b>	<b>66</b>
<b>11. Conclusões .....</b>	<b>71</b>
<b>12. Referências.....</b>	<b>74</b>
<b>Anexo 1 – Documento de visão de processos .....</b>	<b>76</b>
<b>Anexo 2 – Manual de instalação .....</b>	<b>77</b>
<b>Anexo 3 – Manual de utilização .....</b>	<b>78</b>
<b>Anexo 4 – Documento de especificações técnicas .....</b>	<b>79</b>

## Índice de Figuras

Figura 1 - Scrum Process .....	7
Figura 2 - Ciclos presentes no Design Science Research .....	8
Figura 3 - Exemplo de funcionamento lógico de uma paginação. (Spaskovsky, 2014) .....	16
Figura 4 - Resultado obtido através da utilização de ROW_NUMBER() .....	18
Figura 5 - Código exemplo de utilização de ROW_NUMBER() .....	18
Figura 6 - Código exemplo de utilização de OFFSET/FETCH ORDER BY.....	19
Figura 7 - Funcionamento geral do sistema de informação do ERP V10 da PrimaveraBSS ....	21
Figura 8 - Processo de criação de uma lista personalizada via stored procedure.....	22
Figura 9 - Plano do projeto .....	24
Figura 10 - Ambiente ERP V10 .....	27
Figura 11 - Estrutura lógica inicial de navegação da aplicação.....	29
Figura 12 - Página exemplo do documento de visão, com mockup associado .....	29
Figura 13 - Ambiente de desenvolvimento com janela de código associada.....	30
Figura 14 - Pedido Http request.....	31
Figura 15 - Detalhes Http request.....	31
Figura 16 - Tabelas de dados presentes na aplicação .....	38
Figura 17 - Estrutura lógica final da aplicação V10.....	38
Figura 18 - Página de Login .....	39
Figura 19 - Menu Principal 2 .....	40
Figura 20 - Menu Principal 1 .....	40
Figura 21 - Menu Principal notificação .....	40
Figura 22 - Listagem de clientes com filtro .....	41
Figura 23 - Listagem de clientes.....	41
Figura 24 - Detalhe de cliente .....	41
Figura 25 - Criar cliente 1 .....	42
Figura 26 - Criar cliente 2 .....	42
Figura 27 - Detalhe de artigo .....	42
Figura 28 - Catálogo c/filtro .....	42
Figura 29 - Catálogo c/imagem.....	42

---

Figura 30 - Catálogo s/imagem .....	43
Figura 31 - Ações de listagem de documento.....	43
Figura 32 - Listagem de documentos com filtro .....	43
Figura 33 - Listagem de documentos.....	43
Figura 34 - Detalhe de linhas de documento.....	44
Figura 35 - Detalhe de documento (linhas) .....	44
Figura 36 - Detalhe de documento .....	44
Figura 37 - Criação de documento (escolha de artigo).....	45
Figura 38 - Criação de documento (linhas).....	45
Figura 39 - Criação de documento (cabeçalho).....	45
Figura 40 - Detalhe de documento guardado.....	45
Figura 41 - Listagem de documentos guardados.....	45
Figura 42 - Definições de Menu .....	46
Figura 43 - Definições de LogIn.....	46
Figura 44 - Input de querie na lista.....	47
Figura 45 – Endpoint gerado.....	47
Figura 46 - Stored procedure de paginação de artigos .....	48
Figura 47 – Chamada da stored procedure através das listas personalizadas.....	48
Figura 48 - Definição de requisitos segundo Scrum Solo .....	49
Figura 49 - Revisão de sprint (Deployment e Management).....	53
Figura 50 - Processo de LogIn com as credenciais pretendidas .....	59
Figura 51 - Dados de login validados .....	59
Figura 52 - Filtragem de documentos por tipo de doc .....	60
Figura 53 - Filtragem de documentos por data .....	60
Figura 54 - Filtragem de clientes.....	60
Figura 55 - Filtragem de catálogo via família e subfamília .....	61
Figura 56 - Filtragem de catálogo via código de produto, família e subfamília .....	61
Figura 57 - Criação de cliente 1.....	62
Figura 58 - Criação de cliente 2.....	62
Figura 59 - Cliente criado através da aplicação no ERP V10.....	62
Figura 60 - Listagem do cliente criado .....	63

---

Figura 61 - Seleção de artigo via catálogo .....	63
Figura 62 - Preenchimento do cabeçalho do documento .....	63
Figura 63 - Seleção dos dados do artigo após leitura do código de barras .....	64
Figura 64 - Seleção de artigo via código de barras .....	64
Figura 65 - Documento criado com sucesso no reportório do ERP V10.....	65

## Índice de Tabelas

Tabela 1 - Tarefas principais .....	25
Tabela 2 - Product Backlog com as sprints associadas a cada requisito .....	50
Tabela 3 - Testes de validação exploratórios realizados .....	55
Tabela 4 - Lista de pedidos criados .....	67

## Lista de Siglas e Acrónimos

- ERP – *Enterprise Resource Planning*;
- SI – Sistemas de Informação;
- TI – Tecnologias de Informação;
- API – *Application Programming Interface*;
- UI – *User Interface*;
- UX – *User Experience*;
- SP - *Stored Procedure*.

## 1. Introdução

Neste capítulo irá ser exposto o enquadramento do projeto, motivações, resultados esperados e a estrutura do documento.

### 1.1. Enquadramento

No ambiente atual do mercado é reconhecido, cada vez mais, a potencialidade e vantagem que as aplicações móveis trazem para um panorama geral dependente, de uma forma crescente, da portabilidade e desmaterialização dos seus processos de negócio.

Esta necessidade de agilizar processos e eliminar ambiguidades no transporte e tratamento de dados, apresenta-se como uma necessidade de que as organizações se vêm forçadas a acompanhar de forma a manterem-se relevantes no mercado.

A forma mais comum de dar resposta à desmaterialização de processos é recorrendo às integrações de infraestruturas de software já existentes, via Web API.

### 1.2. Problema e Motivação

Um processo de negócio agilizado e bem definido pode marcar a diferença entre uma organização ser competitiva ou cair na irrelevância no mercado atual. A presença de processos redundantes e analógicos pode atrasar todo o restante processo de negócio que deles depende.

Apesar da emergência e aumento constante de aplicações móveis e ferramentas de integração, é importante para uma organização ter o conhecimento de que funcionalidades irá tirar partido, em que processos chave terá de incidir e como irá ser tratada a informação. Assim, uma aplicação genérica não poderá tirar partido de toda a potencialidade que é pretendida, tendo em conta a existência da Web API do ERP da PrimaveraBSS, que se encontra preparada para a integração com uma aplicação móvel.

Este problema surgiu em ambiente de consultoria após o recente lançamento da Versão 10 (V10) do *software* principal de gestão e faturação da PrimaveraBSS, que visa inovar em vários campos no interior da organização. Um destes campos de inovação é o estudo de uma entrada oficial no campo das aplicações móveis, sendo também coincidente com a aquisição

da *framework* Kalipso para desenvolvimento de aplicações móveis. Esta situação vem combater a presença de meios analógicos de transporte de dados, resultante da ausência de plataformas que possam dar resposta à mobilidade empresarial.

O tema foi atribuído seguindo o requisito de conhecimento de trabalho com a *framework* segundo com trabalhos anteriormente realizados e conhecimentos de gestão de sistemas de informação.

### **1.3. Objetivos e resultados esperados**

No ambiente atual da organização os principais objetivos práticos para o desenvolvimento deste projeto são:

- Oferecer mobilidade e portabilidade de acesso ao ERP V10 da PrimaveraBSS, através de uma integração que utilize o mesmo como meio de processamento de dados, tendo uma aplicação móvel que funcione como uma interface móvel, potencializando o *software*;
  - Possibilitar a criação de documentos de venda, criação de clientes e consulta de catálogo através da aplicação móvel por meio *online* e *offline*;
  - Permitir a visualização de dados estatísticos através da aplicação;
  - Agilizar e eliminar processos redundantes de forma a manter o seu modelo de negócio competitivo;
- Potencializar o uso da ferramenta de desenvolvimento *mobile* presente na organização;
- Estudar e identificar lacunas e fraquezas presentes da Web API do ERP V10.

Relativamente à dissertação é esperado que:

- Seja entregue um protótipo da aplicação integrada com o ERP V10 que dê resposta aos requisitos apresentados pela organização;
- Seja criado um conjunto de documentos técnicos e teóricos que validem todo o projeto de forma a compreender o que foi feito;
  - Documento de visão de projeto (Anexo 1);
  - Manual de instalação (Anexo 2);
  - Manual de utilização (Anexo 3);



- Documento de especificações técnicas (Anexo 4).
- O protótipo apresente a qualidade esperada e que justifique uma análise mais profunda, que possa possibilitar uma implementação a nível oficial.

Na realização do presente documento de dissertação é esperado que:

- Seja apresentada a revisão de literatura que demonstre os conhecimentos relativos aos tópicos relacionados com o tema do projeto;
- Sejam explicadas as metodologias de desenvolvimento e investigação utilizados durante a realização do documento;
- Sejam exploradas as plataformas/tecnologias que irão possibilitar a realização deste projeto;
- Seja explorado o processo de desenvolvimento do protótipo e as suas diferentes fases e problemas;
- Seja construído um protótipo funcional que disponibilize uma integração funcional básica com o ERP V10;
- Seja apresentado o planeamento do trabalho, contendo as tarefas principais realizadas ao longo do projeto, de forma a perceber os diferentes estágios que o projeto apresentou.

#### **1.4. Estrutura do documento**

O presente documento irá ser estruturado em 7 capítulos, sendo que esta explicação está presente no capítulo introdutório, permitindo uma melhor compreensão da estrutura global do projeto.

No segundo capítulo, serão abordadas as metodologias utilizadas no atual projeto, ao nível da investigação científica e ao nível do desenvolvimento prático.

O terceiro capítulo contém toda a revisão de literatura acerca dos tópicos que se relacionam direta ou indiretamente com o tema do projeto, oferecendo uma camada de conhecimento que justifique e que crie uma base sólida para o restante documento.

O quarto capítulo será dedicado à exploração do *software* a utilizar na duração do projeto.

O quinto capítulo servirá para expor o planeamento do projeto com recurso a um diagrama de Gantt, onde irão ser exploradas as diferentes fases presentes no projeto e uma estimativa do tempo de duração total.

Após a exposição teórica, o sexto capítulo irá abordar todo o trabalho prático executado ao longo do projeto, abordando o artefacto entregue e as abordagens utilizadas.

Por fim, o sétimo capítulo irá apresentar as conclusões finais acerca do trabalho exposto no presente documento.

## 2. Metodologias de Desenvolvimento e Investigação

### 2.1. Scrum Solo

O Scrum é uma *framework* que a Primavera *Bussines Software Solutions* utiliza para gerir o trabalho complexo de conhecimento que envolve toda a conceção e desenvolvimento de software. Esta metodologia é utilizada maioritariamente por pequenas equipas, em ambiente de desenvolvimento, distribui o trabalho em várias metas a serem completadas dentro de iterações, estas iterações são denominadas por *sprints*, em que não podem ultrapassar um mês e que normalmente duram entre duas a três semanas.

Um dos princípios chave do Scrum é o reconhecimento de que um cliente poderá ou irá mesmo mudar os requisitos do trabalho, ou seja irão sempre acontecer mudanças inesperadas. Tendo em mente esta visão de volatilidade dos requisitos, o Scrum encara este cenário, focando-se em maximizar o trabalho da equipa em curtos espaços de tempo de forma a responder a requisitos e problemas emergentes ao longo do desenvolvimento.

No caso deste projeto, é seguido o Scrum Solo (Pagotto, Fabrti, Lerario, & Gonçalves, 2016), uma versão de Scrum que visa desenvolvimento levado a cabo por apenas um individuo. O Scrum Solo teve a sua origem no Brasil visto que cerca de 60% das empresas brasileiras de software iniciavam as suas atividades com apenas um *developer*. (Pagotto, Fabrti, Lerario, & Gonçalves, 2016)

O fluxo do Scrum Solo e o Scrum convencional apresentam-se como bastante semelhantes, sendo distinguidos em algumas atividades, atores e artefactos.

#### Atividades:

- **Requirements:** Nesta primeira atividade é definido o âmbito do projeto partindo da interação do cliente com o *developer* e o orientador. Esta atividade dá origem ao *product backlog* e as especificações do protótipo.
- **Sprint:** Os sprints visam o desenvolvimento de um requisito selecionado previamente através do *Product Backlog*. Nesta atividade, o *developer*, terá de realizar a tarefa

selecionada, dando origem a funcionalidades do protótipo e um *sprint backlog*, dentro de um espaço temporal previamente definido.

- **Deployment:** No final de cada sprint é disponibilizado um protótipo ao cliente com as funcionalidades desenvolvidas na mesma.
- **Management:** Nesta última atividade é esperado uma gestão do estado do produto, recorrendo ao *produto backlog*, onde se irá analisar o estado da solução e quais as próximas tarefas. É esperado que nesta atividade seja gerado a estrutura analítica do projeto, um cronograma e uma tabela de custos.

#### Atores:

- **Cliente:** É o proprietário do produto, que irá interagir diretamente com o mesmo. Pode ser caracterizado como uma pessoa (ex. Contabilista, Secretário, Universidade) ou como um grupo de pessoas (ex. alguém que necessite de uma aplicação que indique o tempo);
- **Developer:** Ator responsável por desenvolver o produto;
- **Orientador:** Consultor que tem conhecimento de todo o processo e possui uma visão geral de toda a tecnologia utilizada para o desenvolvimento e do âmbito do processo.

#### Artefactos:

- **Âmbito:** Caracteriza o objetivo do projeto. Indica os aspetos inerentes ao mapeamento do problema do cliente e indica os principais pontos que podem ser aplicados na solução. Indica o perfil do cliente e as tarefas do *product backlog*;
- **Protótipo:** Imagem da interface do utilizador com a solução. É indicado o formato “.PNG” para visualização;
- **Product Backlog:** Lista de funcionalidades e requisitos a serem implementados na solução;
- **Repositório do processo:** Serviço de armazenamento cloud, que serve para guardar todos os artefactos gerados nas *sprints*;
- **Sprint Backlog:** Conjunto de funcionalidades armazenadas a serem implementadas durante uma dada *sprint*. Deverá ter presente o caso de uso do mesmo com todos os intervenientes e processos necessários;

- **Produto:** Parte ou totalidade do produto funcional que permita ao cliente acompanhar o progresso do mesmo.
- **Ata:** Registo da implementação de uma funcionalidade no final de uma *sprint*:
- **Plano de desenvolvimento:** Organiza os artefactos que foram utilizados na especificação de funcionalidades.
- **Estrutura analítica do projeto:** Organograma que tem como objetivo o âmbito do projeto, demonstrando o título, atividades e funcionalidades do mesmo.
- **Cronograma:** Diagrama de Gantt.
- **Plano de custo:** Mapeamento do custo durante a execução do projeto.

Na temos representado o fluxo de trabalho realizado segundo o *Scrum solo*.



Figura 1 - Scrum Process

Na figura 1 vemos o fluxo de trabalho segundo o *Scrum solo*, iniciando-se sempre com o *product backlog*, onde são analisadas todas as funcionalidades pressupostas para o desenvolvimento da solução que por sua vez são selecionadas um pequeno conjunto de funcionalidades a serem desenvolvidas num dado sprint, criando assim um *sprint backlog*. Após a seleção das funcionalidades, é esperado que estas sejam desenvolvidas no espaço de uma semana, criando uma *sprint* sem a necessidade de reuniões diárias, efetuando o desenvolvimento seguindo o plano de trabalho e a gestão previamente definidos. No final da sprint, a parte do projeto desenvolvido é entregue e avaliado pelos orientadores responsáveis que validam e analisam de forma crítica o trabalho desenvolvido. Este fluxo é repetido de forma sistemática até que a solução esteja completa.

## 2.2. Design Science Research

O Design Science Research (DSR) é o resultado baseado na metodologia da investigação de TI que oferece *guidelines* específicas para avaliação de projetos de investigação. O principal objetivo é desenvolver conhecimento que permita desenhar e desenvolver soluções que deem resposta aos problemas que estejam a resolver. (Hevner 2007)

São compreendidos três ciclos de *Design Science Research*, tal como a Figura 2 demonstra. (Hevner 2007)

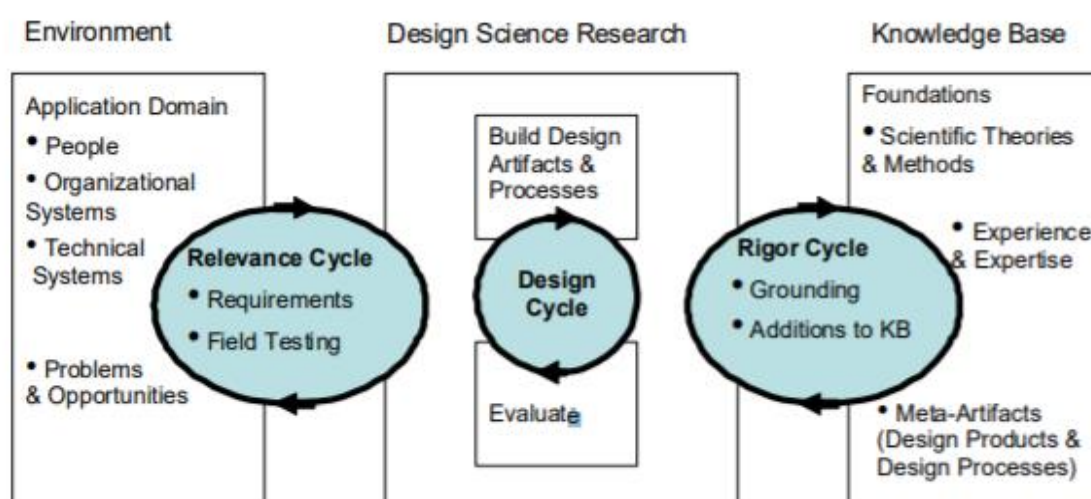


Figura 2 - Ciclos presentes no Design Science Research

- **Ciclo de relevância:** este ciclo inicia a investigação do desenho que providencia os requisitos que definem os critérios de aceitação para a avaliação final dos resultados de investigação. Este ciclo entende o envolvimento e interação de pessoas, sistemas organizacionais e sistemas técnicos com os seus problemas e oportunidades. Bons exemplos de DSR tem início pela representação de oportunidades e problemas no âmbito de aplicação.
- **Ciclo de rigor:** o ciclo de rigor fornece o conhecimento passada ao projeto de investigação, sendo que cabe aos envolventes pesquisar a base de conhecimento, de forma a garantir que os projetos produzidos funcionem como contribuições de pesquisa com base na aplicação de processos já conhecidos. Este ciclo baseasse em teorias científicas bem formuladas através de experiência.
- **Ciclo de *design*:** O ciclo central de *design* alterna entre as atividades nucleares de desenvolvimento e avaliação de artefactos de desenvolvimento e processos de investigação. É importante manter o balanço entre Design os esforços gastos em construção de validação do artefacto em avaliação.

O DSR reconhece ainda sete *orientações* («Design Science (Methodology)» 2019):

1. ***Design as an artifact:*** O DSR deve produzir um artefacto válido no formato de um modelo, método ou instância;
2. ***Problem relevance:*** O objetivo do DSR é desenvolver soluções que respondam a problemas importantes de negócio;
3. ***Design evaluation:*** A utilidade, qualidade e eficácia de um artefacto deverá ser demonstrado rigorosamente recorrendo a métodos de avaliação bem executados;
4. ***Research contributions:*** O DSR deve, efetivamente, fornecer contribuições verificáveis nas áreas do artefacto em desenvolvimento;
5. ***Research rigor:*** DSR depende de métodos rigorosos na construção e avaliação do artefacto em desenvolvimento;
6. ***Design as a search process:*** A pesquisa de um artefacto, requer a utilização de meios disponíveis de forma a alcançar os fins, satisfazendo o problema de ambiente;
7. ***Communication of research:*** DSR deve ter presente audiências orientadas para tecnologia e de gestão;

### 3. Revisão de Literatura

#### 3.1. Desmaterialização de Processos

A desmaterialização de processos tem como principais objetivos a simplificação e otimização de processo de negócios existentes que utilizem processos redundantes ou circuitos de informação com comunicação pouco eficiente (Ávila, Teixeira, e Almeida 2012).

Desmaterializar significa tornar um processo objetivo existente num mais simples, na maior parte das vezes digital, mantendo apenas a sua essência, descartando o seu “invólucro” (Ranoya 2012).

Com a entrada na era Digital, o advento da Internet e a evolução tecnológica, muitos processos foram sofrendo desmaterializações. Um exemplo disso é a forma de como vemos as notícias. O jornal, na sua essência, é ainda muito popular e muito requisitado, mas apenas em plataformas digitais, como meio físico é, hoje em dia, descartável, sendo um dos principais exemplos do descarte do seu “invólucro” enquanto mantém a ideia principal de uso, tal como a ideia da música, sendo que já não existe a necessidade de transportar CDs ou cassetes para as ouvir, sendo apenas necessário um mp3 ou telemóvel para as ouvir em formato digital. Outro exemplo relevante nos dias de hoje é a emulação de aparelhos antigos, por exemplo, sistemas operativos obsoletos ou programas antigos, que já não sejam suportados por *hardware* atual, podem continuar a ser utilizados com recurso a programas especializados de emulação que simulam o ambiente em que estes funcionavam. Tal como o Jornal, que apresentava informação em folhas de papel, os programas ou sistemas antigos utilizavam, de uma forma simplificada, estruturas de hardware e software específicos, que, com os programas referidos, podem ser simulados em *backend*, permitindo assim a utilização dos programas (Ranoya 2012).

A ideia de desmaterialização é muitas vezes compreendida como uma operação de simplificação que tem em vista a redução e muitas vezes a total eliminação de transporte de informação em meio físico, sendo isto mais relevante em ambiente de aprovação e armazenamento de documentos. O processo de desmaterialização de processos pode muitas vezes ser encarado como uma reengenharia de processos (Ávila, Teixeira, e Almeida 2012),



sendo que muitas vezes este processo é resultado de uma implementação de Sistemas de Informação ou de uma reestruturação de processos de negócio da organização.

Toda esta operação apresenta um conjunto de vantagens que pode tornar uma organização mais eficiente e eficaz, sendo algumas das principais vantagens as seguintes (Guedes, Santos, e Ferreira 2009):

- Automatização;
- Controlo de dados mais eficaz;
- Processo mais rápidos e eficientes;
- Eliminação de redundâncias;
- Uniformização de processos;
- Poupança de recursos e dinheiro.

### **3.2. Fatores críticos de sucesso da desmaterialização de processos**

Apesar da oferta extensa de tecnologias e facilidade de aceder a mesma, muitas empresas, mesmo de média dimensão, continuam a utilizar o formato físico para transporte e tratamento de informação, seja por má gestão de topo, resistência à mudança ou falta de financiamento. Por outro lado, temos o exemplo do setor público, que apesar de muitas vezes já ter a implementação de Sistema de informação e já existir uma desmaterialização de processos, é ainda comum a utilização de meios redundantes a todo o processo, independentemente da sua utilidade real.

Sendo que, como já referido, a otimização e simplificação são palavras chave e sinónimos de desmaterialização, quais as condições necessárias para que realmente sejam bem-sucedidos no seu processo? A resposta pode variar se olharmos para diferentes culturas, mercados, gerações ou infraestruturas. Tudo começa pela existência, ou ausência, de uma mentalidade aberta a mudança e inovação, pois sem esta, a própria ideia de desmaterialização poderá nunca ser ponderada e todo o processo será ignorado e descartado (Ávila, Teixeira, e Almeida 2012).

Dois fatores importantes que podem colidir e levar a uma falha da desmaterialização, é a gestão de topo e os colaboradores. Existem 2 cenários que podem levar a uma falha total ou parcial desta operação:

- **Resistência da gestão** (Ávila, Teixeira, e Almeida 2012): apesar de ser a gestão de topo que geralmente aceita a operação de desmaterializar os processos, não faz com que não seja uma possível causadora de problemas durante e após a operação. Poderá não existir uma direção estratégica coerente com os processos de negócio que estão a sofrer as mudanças, podendo levar a uma desmaterialização imperfeita ou ineficaz. Este problema pode também surgir devido a conflitos de financiamento, onde a gestão de topo não queira dispensar os fundos suficientes para que toda a operação seja feita no seu total potencial. Dito isto, mesmo que os colaboradores sejam capazes e se apercebam da necessidade desta operação, sem a total colaboração da gestão de topo, todo o processo pode falhar.
- **Resistência dos colaboradores** (Ávila, Teixeira, e Almeida 2012): Analisando o ponto anterior, o oposto pode acontecer. Num ambiente de desmaterialização de processos, a existência de colaboradores com os conhecimentos necessários, pode levar a uma incapacidade de compreender ou utilizar os novos processos, fazendo com que o processo em si se torne um pouco redundante.

Só a combinação da Gestão de topo de uma organização e os seus colaboradores, pode levar a uma operação de desmaterialização de processos bem-sucedida.

Para além dos fatores organizacionais acima referidos, podem existir fatores tecnológicos que impeçam esta operação. O principal fator tecnológico é a existência de infraestruturas capazes de suportar os novos processos, pois sem estas o processo poderá não ser possível (Ávila, Teixeira, e Almeida 2012).

### **3.3. Mobilidade empresarial**

Entende-se por mobilidade empresarial, o conceito de um colaborador ter a possibilidade de trabalhar e aceder aos dados da empresa, em ambiente externo às infraestruturas da mesma, continuando a conseguir utilizar os mesmo serviços («O que é mobilidade empresarial? + 10 benefícios específicos» 2019).

Não é raro no panorama empresarial atual, existirem empresas sem uma cede central ou mesmo sem qualquer escritório ou balcão físico, sendo que todos os colaboradores trabalham remotamente através de qualquer dispositivo. Uma empresa que não esteja disposta a

investir na área da mobilidade é quase certo que não se tornará competitiva, e ficará desatualizada perante aquelas que estejam dispostas a fazer esse mesmo investimento e, no pior dos casos, pode tornar-se obsoletas («Insight – A era da mobilidade empresarial moderna» 2017). Trabalhar fora das infraestruturas é cada vez mais obrigatório, seja apenas para uma consulta rápida de dados, confirmar de uma venda ou mesmo prestar serviço, tornando isto cada vez mais a norma daquilo que é trabalhar.

Um dos grandes culpados desta forma de encarar a questão do que é trabalhar para uma empresa é a computação *cloud*. A *cloud* permite aceder, carregar e alterar o nosso trabalho a partir de qualquer lugar, muitas vezes nem necessitando de uma aplicação especializada para o efeito, permitindo assim a um colaborador ter o seu escritório em qualquer lugar («Insight – A era da mobilidade empresarial moderna» 2017).

Apesar da sua existência, ainda não é a norma a existir organizações sem escritórios físicos, sendo ainda a existência de uma sede e escritórios a forma convencional de atuar. Assim sendo, é também a norma, a existência de vários sistemas, base de dados e serviços espalhados pela mesma. O fator que torna, nos dias de hoje, a mobilidade empresarial uma mais valia, é a integração entre os sistemas e serviços da organização, permitindo uma intercomunicação eficiente e eficaz que completa aquilo que é uma solução imprescindível para uma organização. Esta integração permite a um colaborador ter acesso a todos os recursos da empresa à distância de um clique no telemóvel.

Esta realidade não está restrita aos colaboradores de uma organização, pode também ser uma ferramenta crucial de negócio para os seus clientes, sendo um exemplo disso, o mercado de retalho, que tem hoje uma forte componente digital, que não só permite a um cliente aceder a todo o catálogo de uma marca, mas também que este consiga encomendar os seus produtos a partir de qualquer lado do mundo em qualquer dispositivo.

É impossível pensar em mobilidade sem referir portabilidade, e este cenário é apenas possível graças à popularidade dos smartphones, computadores portáteis e tablets, pois é graças a estes que o conceito de mobilidade existe («Insight – A era da mobilidade empresarial moderna» 2017).

### **3.4. Desafios da Mobilidade**

Com o crescimento da mobilidade empresarial, foram surgindo alguns termos relativos a este cenário de negócio, sendo os mais relevantes o BYOD (*Bring Your Own Device*) e BYOA (*Bring Your Own App*).

BYOD reflete a organização que utiliza aplicações próprias especializadas para guardar e gerir dados, fazendo assim uma gestão interna. A BYOA vai aparecendo cada vez mais associada à anterior no sentido que reflete o próprio colaborador, que utiliza aplicações de uso pessoal, como aplicações de armazenamento de dados criando assim um ambiente pouco diferenciado entre o que é profissional e pessoal.

Estas realidades entram em conflito com novas normas de proteção de dados que proíbem esta indiferenciação e faz com que as organizações se vejam obrigadas a criar sistemas próprios fechados de forma a se manterem o seu negócio móvel.

Outro desafio recorrente é o aumento de utilizadores ou colaboradores de uma dada organização, que se pode ver obrigada a aumentar as suas infraestruturas tecnológicas de forma a acompanhar o seu crescimento e continuar a disponibilizar a mesma qualidade nos seus serviços. Para além de infraestruturas, é necessário que existam sistemas capazes de gerir e tratar os dados internos de forma a manter uma boa organização empresarial evitando erros institucionais que possam comprometer a sua integridade.

(«Insight – A era da mobilidade empresarial moderna» 2017)

### **3.5. Rich mobile applications**

A introdução da internet na década de 90, melhorou de forma dramática a experiência dos utilizadores perante os computadores, o que estimulou um maior crescimento de utilização de aplicações utilitárias através dos mesmos. Este uso exponencial deu origem ao conceito de *Rich Internet Applications*(RIA), um termo que indica a experiência realçada por aplicações ou programas que contribuem com um grande número de funcionalidades para a experiência do utilizador dando origem ao termo *Rich User eXperience* (RUX), sendo que ambas se complementam. (Abolfazli et al., 2014)

Com o emergir dos *smartphones*, os computadores rapidamente se tornaram obsoletos perante a tecnologia móvel, sendo que esta tendência deu origem a um novo termo, as *Rich Mobile Applications* (RMA). As RMA herdam a maioria das suas características das RIA, sendo distinguidas especialmente pela sua interface flexível, elevada interatividade, resposta rápida, portabilidade de código, comunicação assíncrona, entre outras. (Abolfazli et al., 2014)

Por sua vez as aplicações móveis estão a sofrer uma mudança onde cada vez mais o foco recai no *software* em vez de *hardware*. Idealmente uma aplicação móvel satisfaz 3 requisitos principais: fornecer funcionalidades ricas similares a aplicações de computador apesar de limitações inerentes de *hardware*; estar disponível em várias plataformas; e por último, sendo o aspeto mais importante do projeto, conseguir apresentar fiabilidade em várias condições de rede, consoante a sua degradação ou até mesmo a sua ausência. (March et al., 2011)

Apesar do crescente aumento de conectividade no ambiente empresarial de aplicações móveis, a utilização de aplicações em modo *offline* é ainda prevalente, sendo que utilizadores veem a flexibilidade de uso como uma componente principal na hora de escolher uma aplicação para uma dada tarefa. (Shah, 2018)

Estudos mostram que um utilizador favorece o uso de uma aplicação em ambiente com degradação de rede, uso da aplicação de forma instantânea sem qualquer condição *online* que impeça o seu funcionamento normal e a possibilidade de salvar dados de forma *offline* no dispositivo a qualquer altura. (Shah, 2018)

Para o desenvolvimento de uma aplicação *offline* a utilização de uma base de dados local, tirando partido do armazenamento do dispositivo, sendo que para este efeito o uso de SQLite é um dos mais populares, sendo que complementar este sistema com uma funcionalidade de sincronização é um aspeto que se deve ter em consideração, de forma a manter os dados presentes atualizados. (Kozlov, 2017)

A compreensão destes termos ajuda a compreender e aprofundar o conhecimento de fatores inerentes ao desenvolvimento de uma aplicação que possam auxiliar a complementar a utilidades, flexibilidade e acima de tudo tirar o máximo de partido da componente mobilidade do projeto.

### 3.6. Paginação em SQL

As atuais aplicações *web* e móveis estão a redefinir a comunicação entre cliente e servidor em relação à troca de dados existente, no sentido de que o aumento de informação disponível e a escalabilidade das aplicações, obrigam à existência de novas formas de contornar a questão da gestão de informação. (Gábor, 2007)

O aumento de informação leva a um aumento direto do tráfego da mesma, assim, uma das questões mais prevalentes é a de qual será a melhor forma de aceder a esta informação sem sobrecarregar a aplicação e dar origem a tempos de espera extensos. (Gábor, 2007)

O principal contratempo presente neste cenário é as limitações de recursos de *hardware* presente muitas vezes no lado de cliente (ex. RAM), que limita o processamento de pedidos provenientes do lado do servidor. A abordagem, que é defendida no método de paginação, é a segmentação de dados presentes nas tabelas das BD criando assim um intervalo menor de recolha de dados, baixando exponencialmente a utilização dos recursos do lado do cliente, tornando os processos presentes nas aplicações mais agilizados. (Spaskovsky, 2014)

Na imagem apresentada é possível observar um exemplo de funcionamento lógico de uma paginação.

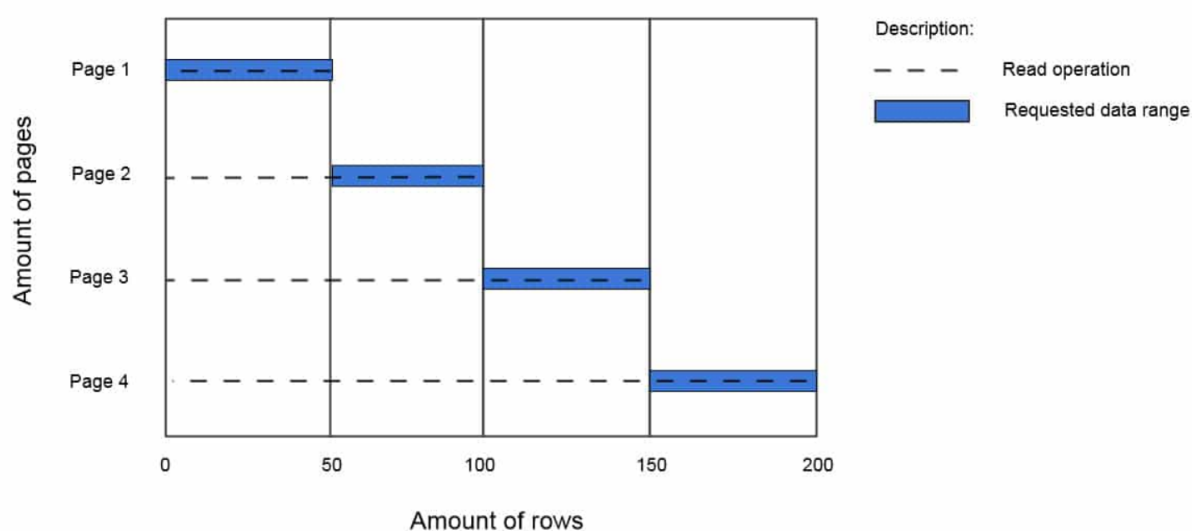


Figura 3 - Exemplo de funcionamento lógico de uma paginação. (Spaskovsky, 2014)

Durante a paginação, existem alguns tipos de auxiliares que podem ser utilizados para otimizar o processo, tal como a existência de valores exatos de procura, ordenação de resultados ou número máximo de resultados. Estes auxiliares funcionam como componentes que preveem um auxílio direcionado ao lado do cliente, otimizando assim a apresentação da informação relevante para o mesmo. (Spaskovsky, 2014)

SQL permite que a paginação seja efetuada na aplicação, no servidor ou mesmo no próprio SQL. A paginação no próprio servidor SQL apresenta-se como a mais viável devido a que este efetua a paginação e divide a informação pretendida e só depois efetua o envio dessa mesma informação, oferecendo um tráfego de informação menor que as outras opções, visto que estas necessitam de recolher todos os dados antes do tratamento. (D i z, 2020)

#### 3.6.1. ROW\_NUMBER/TOP

Em versões mais antigas do SQL Server a função ROW\_NUMBER() era a mais utilizada para efetuar uma paginação. Esta enumerava em primeiro lugar o total de linhas dos dados existentes na BD e só depois é que era possível efetuar a recolha do número de linhas pretendido pelo utilizador. (D i z, 2020)

Este é um método bastante taxativo ao servidor SQL pois cada vez que é pretendido um novo conjunto de linhas, é necessário efetuar todo o processo de contar linhas para obter o resultado pretendido, não sendo um processo eficiente. (D i z, 2020)

As seguintes imagens explicitam um exemplo de utilização deste método: (D i z, 2020)

```

-- código #2
declare @TamPagina tinyint, @NumPagina smallint;
set @TamPagina= 15; -- quantidade de linhas, por página
set @NumPagina= 3; -- número da página solicitada

declare @DataInicial datetime, @DataFinal datetime;
set @DataInicial= convert (datetime, '1/10/1996', 103);
set @DataFinal= convert (datetime, '31/12/1996', 103);

--
USE NorthWind;

with Consulta as (
SELECT OrderDate, OrderID, ShipCountry, ShippedDate,
      seq= row_number() over (order by OrderDate, OrderID)
  from Orders as O
  where O.OrderDate between @DataInicial and @DataFinal
)
SELECT top (@TamPagina)
      convert (char(10), OrderDate, 103) as OrderDate,
      OrderID, ShipCountry,
      convert (char(10), ShippedDate, 103) as ShippedDate
  from Consulta
  where seq > ((@NumPagina -1) * @TamPagina)
  order by seq;

```

Figura 5 - Código exemplo de utilização de ROW\_NUMBER()

	OrderDate	OrderID	ShipCountry	ShippedDate
1	07/11/1996	10348	Germany	15/11/1996
2	08/11/1996	10349	USA	15/11/1996
3	11/11/1996	10350	France	03/12/1996
4	11/11/1996	10351	Austria	20/11/1996
5	12/11/1996	10352	Portugal	18/11/1996
6	13/11/1996	10353	Austria	25/11/1996
7	14/11/1996	10354	Mexico	20/11/1996
8	15/11/1996	10355	UK	20/11/1996
9	18/11/1996	10356	Germany	27/11/1996
10	19/11/1996	10357	Venezuela	02/12/1996
11	20/11/1996	10358	France	27/11/1996
12	21/11/1996	10359	UK	26/11/1996
13	22/11/1996	10360	France	02/12/1996
14	22/11/1996	10361	Germany	03/12/1996
15	25/11/1996	10362	France	28/11/1996

Figura 4 - Resultado obtido através da utilização de ROW\_NUMBER()

Este processo lógico inicia-se pela contagem das linhas que se encontram na BD que logo após seleciona apenas aquelas que se encontram no intervalo entre “@DataInicial” e “@DataFinal”. É efetuado um *Sort* que recebe as linhas e que as ordena como pedido.

Este processo é complementado com uma cláusula *Where* que serve como filtro para as linhas e por fim é utilizado um *Top* que seleciona as linhas iniciais.

### 3.6.2. OFFSET / FETCH

Outro método também utilizado para criar uma paginação é o par *OFFSET/FETCH* que se tornou disponível em versões mais recentes do SQL Server e que é utilizado em conjunto com a cláusula *ORDER BY*. (D i z, 2020)

Este método é mais eficiente que o *ROW\_NUMBER()* devido ao facto de não necessitar de efetuar uma contagem inicial ao número total de linhas presentes na BD, logo não sobrecarrega os recursos do servidor SQL. (D i z, 2020)

As imagens seguintes apresentam um exemplo de utilização deste método: (D i z, 2020)



```
-- código #3
declare @TamPagina tinyint, @NumPagina smallint;
set @TamPagina= 15; -- quantidade de linhas, por página
set @NumPagina= 3; -- número da página solicitada

declare @DataInicial datetime, @DataFinal datetime;
set @DataInicial= convert (datetime, '1/10/1996', 103);
set @DataFinal= convert (datetime, '31/12/1996', 103);

USE NorthWind;

--
SELECT convert (char(10), OrderDate, 103) as OrderDate,
        OrderID, ShipCountry,
        convert (char(10), ShippedDate, 103) as ShippedDate
from Orders
where OrderDate between @DataInicial and @DataFinal
order by OrderDate, OrderID
offset ((@NumPagina -1) * @TamPagina) rows
fetch first @TamPagina rows only;
```

Figura 6 - Código exemplo de utilização de OFFSET/FETCH ORDER BY

Analisando este processo é fácil de concluir que apresenta um código bastante mais simples que o método *ROW\_NUMBER()* e com o mínimo de linhas possível, tornando a sua execução mais eficiente.

Este processo é iniciado através de um filtro *Where* que define, no caso apresentado, um intervalo de datas. Após isto é definido a ordenação das linhas retornadas e é efetuado um *offset* de 30 ((3 - 1) \* 15,) logo irá retornar de forma imediata as primeiras 15 linhas do Offset.

## 4. Infraestrutura de desenvolvimento

Neste capítulo serão abordadas as ferramentas e plataformas que foram utilizadas ao longo do processo de realização e desenvolvimento de projeto. A escolha destas tecnologias baseou-se na componente de inovação e melhoramento dos processos existentes na organização.

### 4.1. ERP V10

Atualmente na sua décima versão, o ERP primavera apresenta-se como uma solução orientada a médias e grandes empresas sendo integrada com soluções cloud e on premises. Este ERP tem como objetivo, permitir aos seus clientes rentabilizar os investimentos efetuados em tecnologia.

Este ERP assenta numa plataforma transversal, sendo caracterizado pela sua robustez, fiabilidade, integridade e segurança. Este é constituído por diferentes módulos integrados, que dão resposta às diferentes áreas de negócio funcionais, tais como:

- Logística;
- Financeira;
- Contactos e Oportunidades;
- Projetos e Serviços;
- Recursos Humanos;
- Indústria;
- Construção Civil;
- Equipamentos e Ativos.

Integração e extensibilidade é o princípio em que assenta o ERP Primavera. A possibilidade de personalizar a solução base, adicionando ou alterando funcionalidades, de forma a responder aos diferentes requisitos das diferentes áreas de negócios dos clientes. Esta natureza permite à PrimaveraBSS incorporar, de forma adicional, informação e processos complementares que possam realizar tarefas específicas que antes seria impossível na forma base do ERP.

Um exemplo de como o ERP pode ser complementado é de com aplicações genéricas, tais como:

- Envio de e-mails;
- Criação de formulários;
- Novos menus;
- Diferentes perfis de utilizados.

Constituindo a generalidade da sua composição, temos as rotinas, que são conjuntos de componentes. Estas rotinas, são a característica principal que oferece ao ERP a possibilidade de se alterar consoante as necessidades requeridas por diferentes clientes.

As rotinas são acedidas através de uma interface gráfica que segmenta toda a informação em módulos de gestão, permitindo uma fácil adaptação e uso intuitivo.

No projeto atual, o ERP terá o papel de gestão e validação de dados, sendo a comunicação tratada pela Web API da primavera que dará, à aplicação móvel, acesso aos dados presentes na base de dados da primavera.

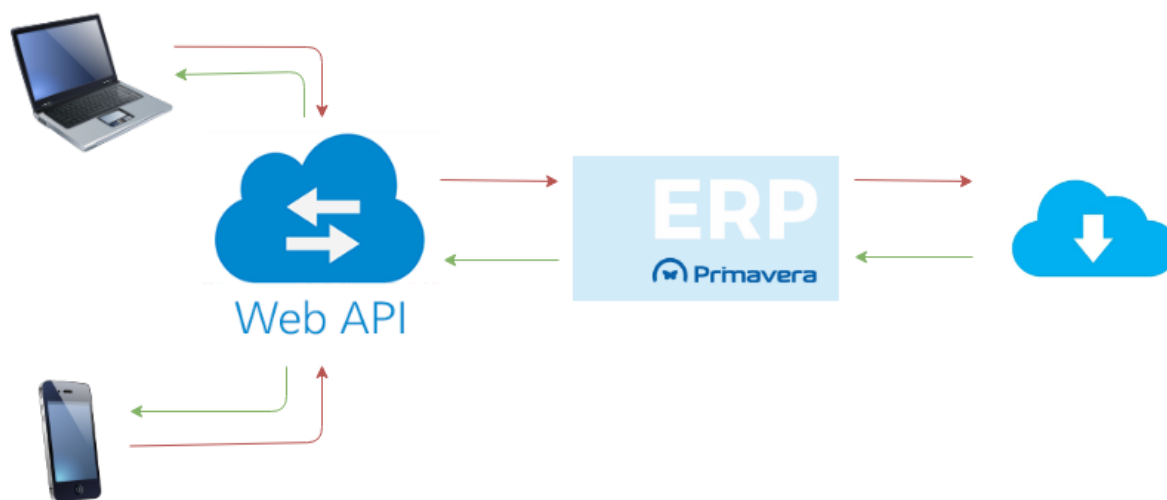


Figura 7 - Funcionamento geral do sistema de informação do ERP V10 da PrimaveraBSS

Devido ao estado incompleto de pedidos na Web API utilizada, será também utilizado um dos recursos únicos ao ERP V10, as listas personalizadas. As listas personalizadas do ERP são utilizadas para a construção de *endpoints* que efetuem pedidos diretos ao ERP.

As listas personalizadas funcionam através de *queries* SQL que efetuem pedidos personalizados segundo as necessidades objetivas do programador de uma forma direta à BD do ERP. Estes pedidos podem ser feitos de duas formas. A primeira forma é a inserção da

*querie* diretamente na lista do ERP, sendo que esta servirá apenas para pedidos simples e diretos sem recurso a parâmetros dinâmicos. Caso seja necessário o transporte dados através de *queries* o utilizador pode optar pela criação de um *stored procedure(SP)* com as devidas variáveis e efetuar a chamada da mesma na lista do ERP. Neste caso a lista permite que sejam programados parâmetros que coincidam com as variáveis presentes na SP criando assim um pedido dinâmico e flexível.

Após a criação de uma lista, é criada uma nova instância na tabela de *queries* da BD do ERP com o pedido e parâmetros associados. Esta instância é identificada por um *guid* único, sendo este utilizado posteriormente como o *id* do novo *endpoint* que servirá para a aplicação efetuar o pedido à Web API. A seguinte imagem demonstra o processo de utilização de uma lista personalizada.

```
SELECT * FROM CabecDoc
WHERE (TipoDoc LIKE '%' + @doc + '%'
OR Entidade LIKE '%' + @entidade + '%')
AND (CabecDoc.Data >= CAST(@datai AS datetime) AND CabecDoc.Data <= CAST(@dataf AS datetime) )

ORDER BY TipoDoc ASC OFFSET @Linhas1 ROWS
FETCH NEXT @Linhas2 ROWS ONLY
```

Guid	Descricao	SQL
8F133DBB-630B-11EA-AAC9-C01885A6374A	Listas_documento_sp_data	Offset_documentos @doc = '@@documento@', @entid...

{{apiUrl}}Plataforma/Listas/CarregaLista/adhoc/?listid=8F133DBB-630B-11EA-AAC9-C01885A6374A&listParameters=FA,SOFRIO,20190315,202003,15,0

Figura 8 - Processo de criação de uma lista personalizada via stored procedure

## 4.2. Kalipso

A ferramenta Kalipso é uma das principais plataformas de desenvolvimento *mobile* presente na organização. No seguimento de agilização de processos, o software Kalipso apresenta-se como uma ferramenta de desenvolvimento *mobile* que se destaca pela abstração de código, que de certa forma, é por si só um exemplo de desmaterialização de processos, eliminando o conceito de total programação, oferecendo uma base agilizada de construção de aplicações

moveis através de sistema de *drag and drop*. Este sistema permite o desenvolvimento de uma aplicação funcional num intervalo de tempo substancialmente mais curto que numa *framework* de desenvolvimento normal.

Apesar de algumas limitações gráficas que possam impedir a construção de interfaces mais complexas, a ferramenta compensa com a utilização de recursos do próprio dispositivo, tal como a utilização da capacidade gráfica do sistema operativo nativo ou a utilização do armazenamento interno do mesmo.

A UI é construída através do sistema *drag and drop* mencionado, eliminando a necessidade de programar uma interface complexa, e consequentemente, perder tempo de desenvolvimento. O *backend* é manuseado através de blocos, sendo que cada bloco pode comunicar entre si se pretendido. Estes blocos são divididos através de ações (ex. click, open form, close form, etc) que permite ao programador organizar e dividir o seu trabalho de uma forma organizada e eficiente.

Por último é importante salientar a potencialidade desta ferramenta em permitir gerar o mesmo projeto numa aplicação que seja compatível com Android, iOS ou Windows, permitindo um ambiente *cross-platform* nativo.

## 5. Plano de projeto

De forma a manter uma organização geral do projeto, a nível temporal, e assegurar que todos os prazos são cumpridos, foi desenhado um plano de projeto que refletisse todo o processo esperado no desenvolvimento do projeto prático.

Este plano (apresentado na figura 9) foi construído e elaborado com recurso à ferramenta “Microsoft Excel” onde são apresentadas todas as atividades planeadas e entregáveis programados a entregar, seguindo o planeamento inicial do projeto.

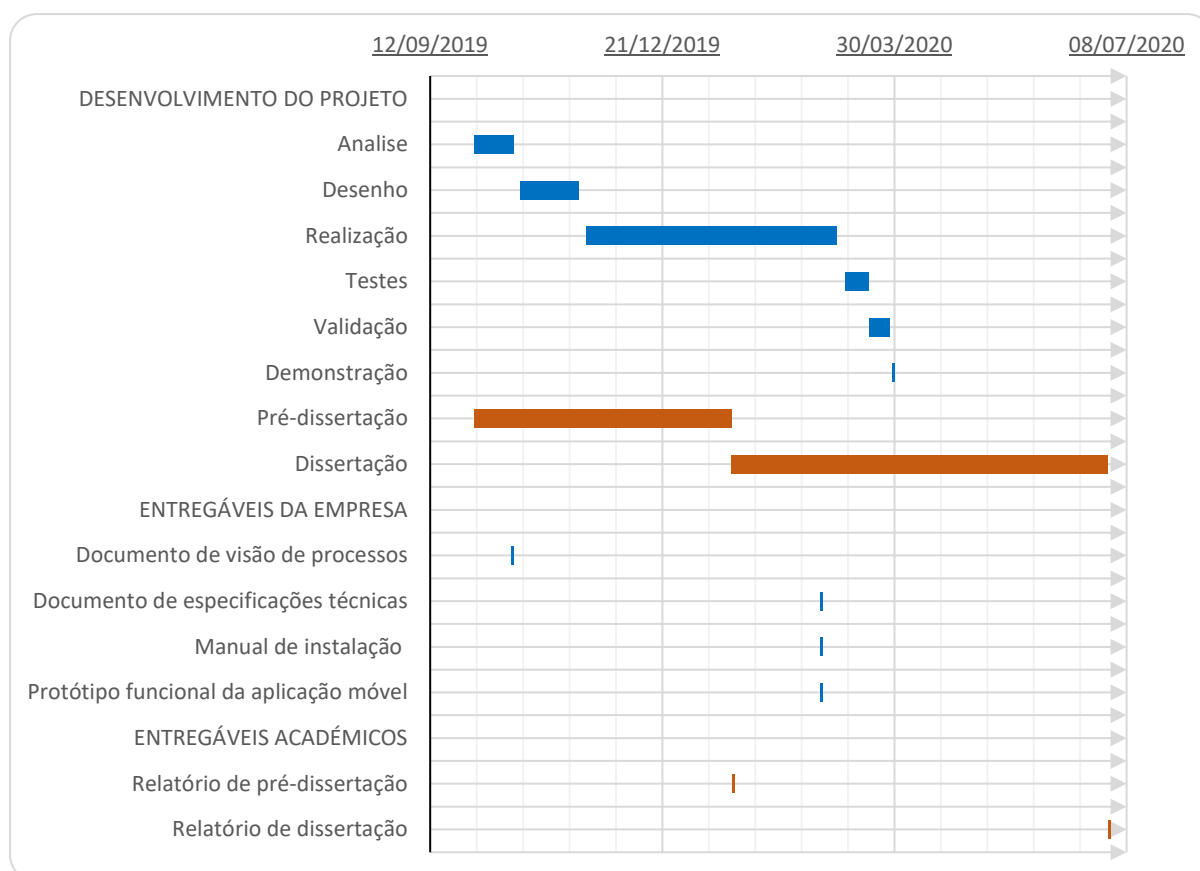


Figura 9 - Plano do projeto

No plano apresentado, na figura 9, são representados dois conjuntos de entregáveis: da empresa e académicos.

Os entregáveis da empresa são os seguintes:

- **Documento de visão e processos:** documento de visão do protótipo que segue a metodologia de gestão MIP que explica e demonstra a estrutura e funcionamento dos processos;

- **Documentação de especificações técnicas:** este documento tem como função explicar o processo de instalação e preparação para o uso da aplicação;
- **Manual de instalação:** Este manual servirá para o apoio de configuração de utilização da aplicação em qualquer máquina de desenvolvimento.
- **Protótipo funcional da aplicação móvel:** protótipo funcional que incorpora todos os requisitos solicitados e acordado no início e ao longo do desenvolvimento do projeto.

Os entregáveis académicos são constituídos pelo relatório de pré-dissertação e pelo documento de dissertação, sendo ambos realizados ao longo de todo o projeto.

Na Tabela 1 são descritas, de forma resumida, as principais tarefas requisitadas pela organização e pela instituição académica de forma a que o projeto alcance os resultados esperados.

*Tabela 1 - Tarefas principais*

<b>Atividades</b>	<b>Descrição</b>
Análise	Parte introdutória do projeto que visa o estudo das ferramentas e processos que serão abordados.
Desenho	Processo de conceção de <i>mockups</i> de forma a construir uma estrutura inicial do protótipo a desenvolver, seguindo os processos estudados anteriormente.
Realização	Período de desenvolvimento do protótipo.
Testes	Testes finais do protótipo de forma a consolidar os requisitos implementados.
Validação	Período de validação prática do protótipo por parte do orientador da empresa.
Demonstração	Reunião de apresentação do protótipo às partes responsáveis e interessadas da empresa de forma a validar e concluir o processo de consultoria realizado.
Pré-dissertação	Relatório realizado na duração do primeiro semestre
Dissertação	Relatório realizado na duração do segundo semestre

As atividades compreendem duas partes, a parte acadêmica que engloba as diferentes componentes teóricas da Pré-dissertação e Dissertação, que servem como prova e trabalho final de tudo que foi realizado ao longo do projeto, e a parte Organizacional, que engloba todo o processo de desenvolvimento do trabalho a nível prático, desde a análise até a demonstração, sendo que este processo será auxiliado pelas metodologias ágeis de desenvolvimento e investigação.

As atividades de análise e de desenho, apesar de terem períodos designados de realização, são atividades que estão sujeitas a alterações periódicas, dependendo do acréscimo de novos requisitos. Apesar de serem também atividades independentes, apenas representam o rematar de todo o processo de realização, devido a todos os requisitos terem de ser aprovados e testados ao longo de todo o processo de desenvolvimento da aplicação.



## 6. Trabalho realizado

### 6.1. Análise e Desenho

Na primeira fase do projeto, foi efetuado um estudo de utilização do ERP V10 da Primavera, de forma a compreender, de forma direta, o funcionamento lógico do software. Este exercício teve em vista a interpretação da dinâmica de funcionamento e tratamento de dados presente no ERP.

Para efetuar este estudo, foram instalados um conjunto de componentes teste na máquina fornecida pela empresa que permitisse um ambiente favorável à análise a efetuar. Esta instalação compreendeu uma base de dados de teste com dados previamente inseridos, V10 com acesso de administrador, Web API que permitisse a comunicação do mesmo e um projeto Visual Studio de extensibilidade ao ERP que permitisse a inserção de componentes extra se necessário.

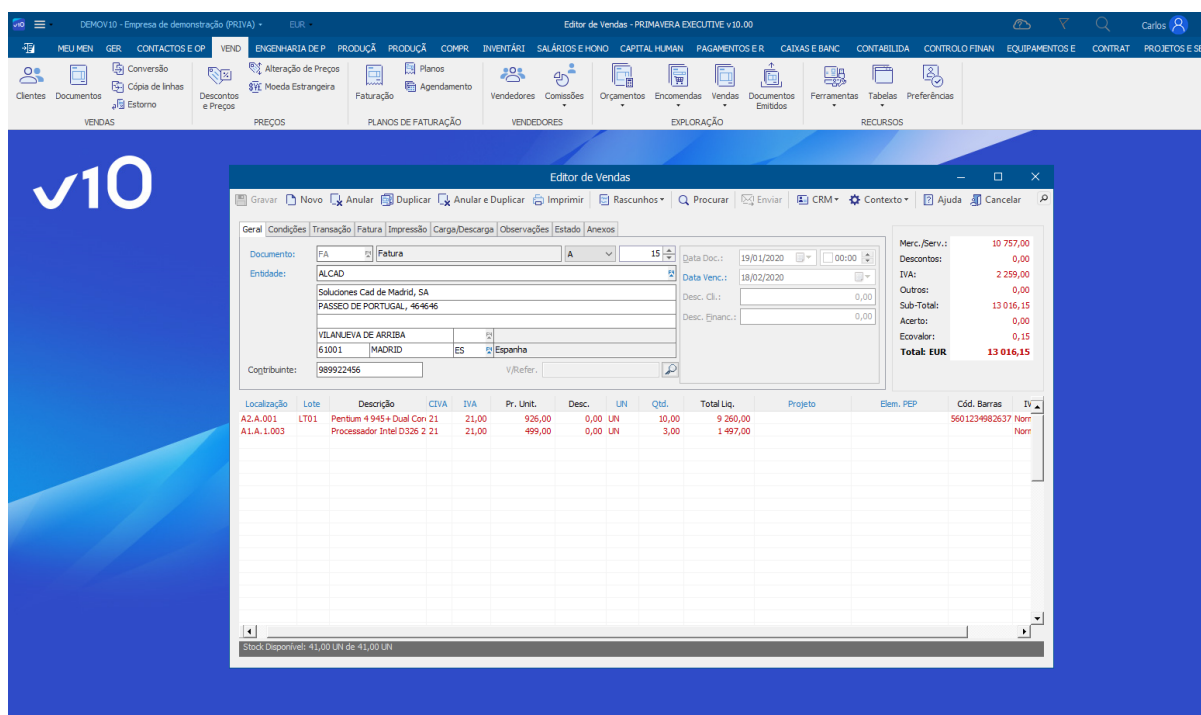


Figura 10 - Ambiente ERP V10

Uma das componentes principais do ERP é o seu manuseamento de dados, sendo que foi imperativo a compreensão do seu processo lógico de forma que, numa fase posterior, fosse possível emular este processo em ambiente *mobile*. Durante o estudo foi detetado o problema de importação de dados, sendo compreendido a importância da quantidade

elevada de informação que o ERP gere, tentando perceber quais os mais relevantes a serem tratados via aplicação.

Este processo serviu, também, para isolar os processos mais cruciais e relevantes a serem auxiliados pela desmaterialização de processos a ser realizado, visto que, sem este conhecimento, não seria possível apresentar uma solução com o máximo de rigor.

Após o estudo do ERP, foram estudadas aplicações modelos já existentes no mercado que dessem resposta a problemas similares ao estudo. Foram analisadas métricas relevantes como processos lógicos de processamento de dados, estruturas de apresentação de dados e número de funcionalidades uteis oferecidas. A partir destas aplicações, foram levantados alguns requisitos extra e funcionalidades opcionais que pudessem vir a ser úteis no desenrolar do projeto.

Esta análise seguiu as guidelines *Problem relevance* e *Design as a search process* que o DSR compreende, estruturando toda a investigação através das ferramentas intervenientes no projeto de forma a que fosse levantado todo o conhecimento que permitisse a construção de uma solução relevante para o problema em estudo.

Após os requisitos base fundamentais serem decididos e organizados, deu-se início ao desenho da estrutura da aplicação. Este desenho, apesar de ser construído a partir do zero, teve como base estrutural aplicações desenvolvidas anteriormente pela Primavera, de forma a seguir uma linguagem comum entre os diversos produtos. Tendo isto em conta, foram construídos um conjunto de *mockups* iniciais de forma a criar um ponto de partida no desenvolvimento. Durante a construção dos *mockups* foi realizado um documento de visão para manter um acompanhamento do percurso do processo de desenho. Este documento visou também a formalização da aplicação no ambiente de consultoria da Primavera,

sofrendo alterações à medida que o projeto se desenrola. A finalização da primeira iteração marca o início do desenvolvimento da aplicação.

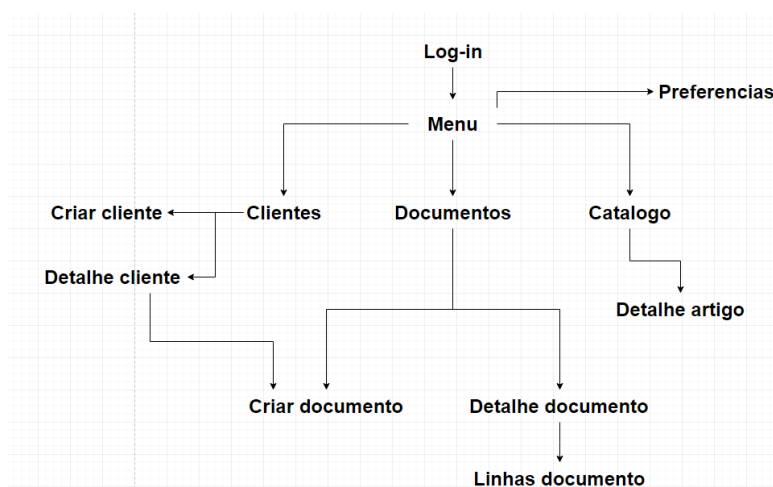


Figura 12 - Estrutura lógica inicial de navegação da aplicação

#### 2.1.3.10. Detalhe da Fatura

Nesta página, navegando através da seleção de uma fatura na página de Vendas, o utilizador tem acesso aos detalhes de uma fatura. Esta página permite também o download e/ou partilha da fatura através de e-mail.

Esta página apresenta os seguintes dados:

- Documento;
- Entidade;
- País;
- Morada;
- Código Postal;
- Nº. Contribuinte;
- Observações;

Esta página apresenta a seleção de artigos na forma de uma tabela que apresenta todos os artigos que constituem a fatura. Esta tabela apresenta as seguintes colunas:

- Produto;
- Quantidade;
- Valor;

Produto	Desc	Qtd	Valor(€)
A001	Computador	3	1000
A002	Ecrã HD	6	500

Data do documento: 06/10/2019  
Data de vencimento: 20/10/2019

Figura 11 - Página exemplo do documento de visão, com mockup associado

## 6.2. Realização

Como mencionado previamente, este projeto tem também como uma das motivações a potencialização da *framework* de desenvolvimento *mobile* presente na Primavera, Kalipso. Graças à capacidade gráfica do Kalipso, tanto a realização das *mockups* como o desenvolvimento adjacente foram realizados no mesmo ambiente, sendo as *mockups* o ponto de partida para todo o desenvolvimento.

Nesta fase, foi também considerada a parte de desmaterialização de processos relativo ao desenvolvimento, ou seja, apesar de não ser a única *framework* de desenvolvimento presente na empresa, esta foi escolhida por ser uma forma de agilizar tanto a parte *frontend* como o *backend*, tirando partido da abstração de código presente na mesma, permitindo um desenvolvimento mais rápido e acessível quando comparado a uma *framework* de desenvolvimento mais convencional (ex. Ionic).

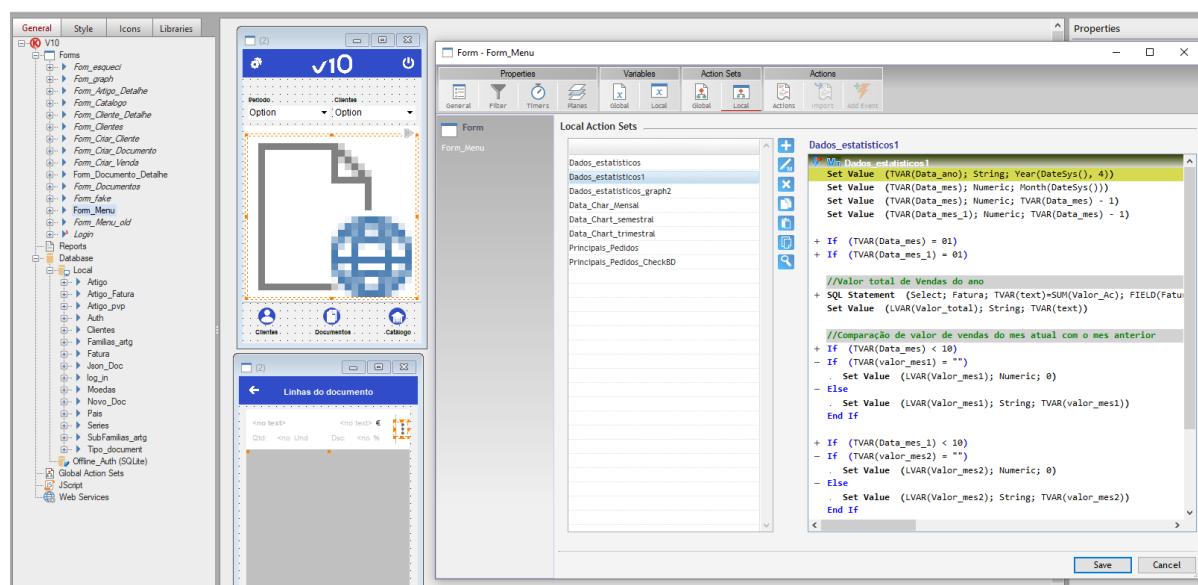


Figura 13 - Ambiente de desenvolvimento com janela de código associada

Pode-se considerar 4 estágios diferentes no desenvolvimento da aplicação. O primeiro estágio refere-se à construção de todo o modelo *mockup* completo com navegação funcional de forma a que se consiga ter uma referência visual que auxilie um raciocínio lógico da aplicação. A partir desta base, é possível uma avaliação preliminar da gestão de dados a ser realizada, conseguindo assim desenhar esquemas que consigam dar resposta às necessidades adjacentes.

Uma segunda fase é a construção da comunicação da aplicação Kalipso com a Web API V10. Um dos recursos fornecidos adicionalmente, de forma posterior, ao início do projeto foi uma *Collection Postman*, que continha estruturas de comunicação e exemplos de pedidos à Web API. Esta *Collection* ofereceu um ponto de partida de obter a comunicação entre os dois componentes.

Os pedidos Http foram distribuídos em 3 grupos, o Login, essencial para o funcionamento de todos os restantes pedidos, listagens e criação. Uma característica presente no ERP é a possibilidade de criação de listas personalizadas, ou seja, regra geral um *endpoint* está associado a um *excerto* presente no seu código que devolve um conjunto de dados pré-programados. As listas personalizáveis do ERP permitem ao utilizador criar uma *query* SQL que devolve os dados pretendidos da Base de Dados, criando um Id que por sua vez, cria um *endpoint* que devolve esses mesmos dados. Foi tirado partido deste sistema nas listagens da aplicação criando listas que devolvam conjuntos de dados personalizados, evitando assim um excesso de tráfego de dados na aplicação.



Figura 14 - Pedido Http request

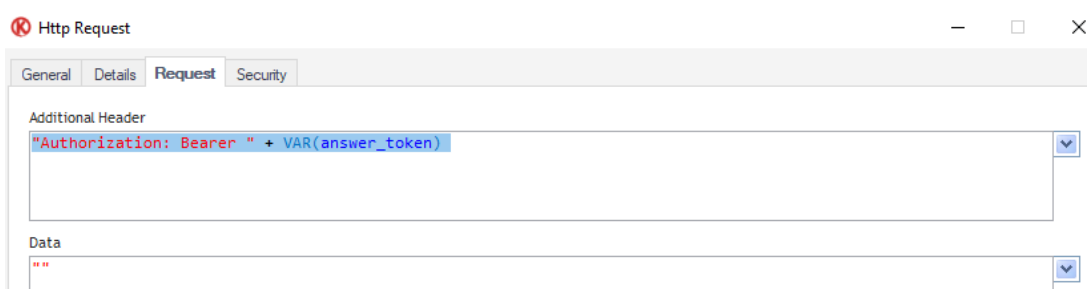


Figura 15 - Detalhes Http request

Na terceira fase, foram implementadas as funcionalidades complementares e extra requisitadas. Uma das principais funcionalidades extra foi a capacidade de a aplicação funcionar em modo *offline*. Este requisito foi implementado recorrendo à utilização do armazenamento dos dispositivos móveis, onde a aplicação guarda as informações de

autenticação relativas ao utilizador de forma a tirar partido da portabilidade mesmo fora de alcance de rede. Também, de forma a usufruir de portabilidade, temos a função de leitura de código de barras, uma utilidade, que apesar de simples, remete mais uma vez para a portabilidade da aplicação. Em geral, estes tipos de requisitos dão resposta à mobilidade requerida no projeto, acrescentando utilidade, indo de encontro à *guideline* de *Research contributions* presente na DSR, que por sua vez torna a aplicação mais versátil e flexível.

A quarta e última etapa resume-se ao controlo de qualidade segundo os parâmetros requeridos pela Primavera BSS, seja a nível de qualidade como a nível de apresentação. Depois de construída a aplicação na parte funcional e lógica, foi necessário efetuar um controlo de UE (*User Experience*). As reuniões tiveram lugar de forma sistemática para que a aplicação fosse refinada de uma perspetiva de utilizador, com isto, pretende-se complementar a boa funcionalidade e gestão de dados presente no *backend* seja complementada por uma apresentação de *frontend* construída de forma lógica, intuitiva e apelativa, para que seja a melhor representação possível de qualidade produzida pela Primavera.

### **6.3. Testes e Validação**

Durante a fase de realização são efetuados testes constantes, de forma a monitorizar o comportamento das diferentes funcionalidades presentes na aplicação. É efetuada uma abordagem de testes manuais Ad hoc cada vez que se adiciona uma nova funcionalidade, ou seja, é construído e efetuado um conjunto de testes especializados de forma a refinar o seu funcionamento, até que não se apresente qualquer tipo de erro ou falha, sendo que, de forma lateral, são testadas todas as funcionalidades que possam comunicar ou relacionar-se com a última que foi desenvolvida, mantendo sempre um nível de controlo geral da estrutura lógica. Este tipo de análise é adequado ao ambiente de consultoria, onde é imperativo o trabalho de acompanhamento gradual durante toda a realização da aplicação.

Os testes realizados, são testes dinâmicos, ou seja, são executados os componentes do produto de forma a realizar uma análise geral dos requisitos gerais de negócio sugeridos pelos clientes, e como explicado anteriormente, quando um requisito é construído ou sofre alterações, são efetuados os testes correspondentes, tanto ao requisito como a toda a

aplicação, principalmente a componentes que comuniquem ou se relacionem diretamente com a funcionalidade desenvolvida, sendo estes, testes de sistema.

Dito isto, um dos estágios finais do desenvolvimento do projeto é a fase de teste, sendo que, ao contrário dos testes regulares realizados, diferencia-se por ser realizado em ambiente real, com quantidades de dados reais e manuseado por elementos da empresa externos ao projeto, de forma a compreender e avaliar um comportamento real.

Os testes finais são chamados de testes de aceitação, que, mais uma vez, são definidos seguindo os parâmetros de qualidade da Primavera e avaliados segundo os critérios e requisitos de negócio estipulados pelo cliente, sendo aprovado posteriormente pelos responsáveis e orientadores do projeto, de forma a avaliar todas as funcionalidades requisitadas.

#### **6.4. Contratempos e limitações**

Durante o desenvolvimento existiram alguns fatores que limitaram e/ou atrasaram o avanço do projeto. De entre todas as limitações e contratempos salientam-se os seguintes:

**Falta de conhecimento no uso do ERP V10:** Na primeira fase de análise, a falta de compreensão do funcionamento do ERP levou a algum atraso. De forma a contornar esta falta de conhecimento, foram efetuados pequenos testes de criação e consulta de documentos e clientes, que sendo estes 2 dos módulos cruciais à realização da aplicação, tentou-se compreender quais os campos obrigatórios para a criação dos mesmos, permitindo assim compreender o raciocínio lógico da gestão de dados. Estes testes foram feitos com a orientação da equipa de consultoria e desenvolvimento;

**Processos da API de ambiente *web* em desenvolvimento:** Na fase da realização, na ligação da aplicação ao ERP, surgiram problemas com alguns *requisitos* em que, após reuniões com responsáveis do desenvolvimento do ERP, se descobriu que estes continham funcionalidades que se encontravam em desenvolvimento para suportar funcionamento em ambiente *web*. Neste contratempo, devido à sua natureza, a única solução para os ultrapassar eram as reuniões com os responsáveis de desenvolvimento que prontamente atenderam às necessidades postas pela equipa da consultoria;

**Limitações gráficas na framework Kalipso:** Na fase final de aprimoramento, onde foram realizadas reuniões com os responsáveis da UX, foram propostos vários requisitos gráficos que a framework não suportava, salientando a apresentação de estatísticas em formato *dashboard*. De forma a contornar este problema, foi utilizada a API da *Google Graphs*, uma API que, partindo do pressuposto que exista uma fonte de dados, consegue fornecer gráficos de variadas formas, conseguindo assim construir uma *dashboard* que cumpra os requisitos propostos;

**Configuração do ambiente de teste:** Por fim, para a fase de testes, foi utilizada a própria máquina de trabalho como um servidor *localhost* acedendo à base de dados de teste instalada e preenchida no início do desenvolvimento do projeto. Esta configuração serviu também como um meio exploratório do ERP, tirando partido das infraestruturas internas da organização e de um conhecimento de redes, de forma a conseguir manusear as mesmas e otimizar o ambiente de desenvolvimento.



## 7. Aplicação móvel

Neste capítulo será explorada a aplicação desenvolvida ao longo desta dissertação destacando os processos abordados pela aplicação, as abordagens efetuadas, as funcionalidades presentes na aplicação e a gestão de dados praticada.

O estudo preliminar dos processos começou pela análise de um documento de venda. Foram analisados um conjunto de cenários diferenciados que colocavam em estudo o modelo de negócio presente no ERP de faturação V10 e em que medida uma aplicação móvel poderia tirar partido das necessidades presentes.

Em estudo foram distinguidos 3 cenários diferentes:

1. Cenário interno: este analisou a interação direta de um utilizador com as infraestruturas do ERP V10. Neste cenário o utilizador encontra-se ligado diretamente a uma máquina/dispositivo com a acesso local às funcionalidades necessárias tal como a visualização de catálogo, clientes e criação de documentos.
2. Cenário externo: neste cenário o utilizador encontra-se fora da organização que contem as infraestruturas do ERP e encontra-se em contato direto com o cliente. Nesta situação o utilizado irá anotar as informações de um cliente, informações do documento requisitado e os produtos pretendidos pelo mesmo de forma analógica. Após esta interação o utilizador irá inserir os dados diretamente no ERP e concluir assim o processo de criação de documento;
3. Cenário espontâneo: Este cenário compreende um misto dos dois anteriores e prevê uma situação espontânea de se deparar com a necessidade de criar um documento de venda. Neste cenário o input de informação será variável sendo que poderá ser proveniente de forma direta de um cliente ou através de um colaborador. Este cenário apresenta uma interação imprevisível em que o utilizador se pode deparar com lacunas de informação que poderá não permitir um preenchimento completo de um documento. Esta situação poderá decorrer no interior ou no exterior da organização.

Após analisar os diferentes cenários, foram encontradas as seguintes necessidades:

- Aceder a informações de clientes;
- Aceder ao catálogo de artigos e serviços;
- Criação\Edição de documentos;
- Gestão de informação de documento.

Após a recolha destas necessidades que os diferentes cenários apresentaram, foi efetuado um cruzamento das mesmas com os pré-requisitos expostos pela Primavera na fase inicial.

Este trabalho pretende então, conseguir eliminar todas as necessidades e lacunas expostas nos cenários e modelo de negócio presente.

## **7.1. Estrutura de dados**

A aplicação foi desenvolvida utilizando a ferramenta de desenvolvimento Kalipso. Esta ferramenta especializa-se em tirar partido dos recursos presentes nos dispositivos, conseguindo funcionar melhor através da utilização do armazenamento local dos mesmos.

Os dados do ERP estão presentes num servidor de SQL que estão associados a diferentes utilizadores. Foi definido que a gestão de dados será feita no servidor, sendo que o dispositivo apenas será utilizado como armazenamento dos mesmos.

Numa fase inicial foram decididas que tabelas de dados seriam essenciais para um funcionamento correto da aplicação. Foram acordadas as seguintes tabelas:

- Documentos;
- Artigos;
- Clientes;
- Login.

Estas tabelas funcionarão como principal pilar de manuseamento de informação da aplicação pois as principais páginas e formulários irão tirar partido da informação presente nas mesmas.

A tabela Documentos irá servir para guardar toda a informação relativa aos documentos de venda já criados e presentes no ERP V10, servindo para efetuar listagens e cálculos

estatísticos. Após um estudo aprofundado foi notado que a informação relativa a um documento de venda esta dividido entre 2 tabelas no servidor SQL:

- Cabeçalho: Contém informação relativa ao tipo e formato de documento e a informação relativa à entidade compradora de produtos ou serviços;
- Linhas: Lista todos os artigos e serviços relativos ao dado documento;

As tabelas de clientes e artigos servirão para a listagem de clientes e a visualização do catálogo. Para além disto serão também pilares importantes que a criação de um documento irá depender.

A tabela Login irá guardar toda a informação relativa ao utilizador, tais como preferências, definições da aplicação e credenciais do mesmo. Estas informações irão ser sempre apresentadas como registo único na tabela de forma a conter apenas um utilizador por login.

Em segundo lugar foi analisada a funcionalidade de criação de um documento de venda, onde se verificou que seriam necessárias tabelas que guardassem os vários tipos de documentos possíveis de criação no ERP, bem como as suas diferentes séries contabilísticas associadas. No catálogo foi também necessário a criação de duas tabelas de família e subfamília de artigos de forma a ser possível a implementação de um filtro que permita a procura de artigos através destas duas categorias.

Para a criação de cliente foi também necessário a listagem dos diferentes países que a Primavera abrange, bem como as moedas respetivas dos mesmos.

Juntando algumas tabelas de apoio secundárias, o resultado final das tabelas de dados é apresentado na seguinte imagem.

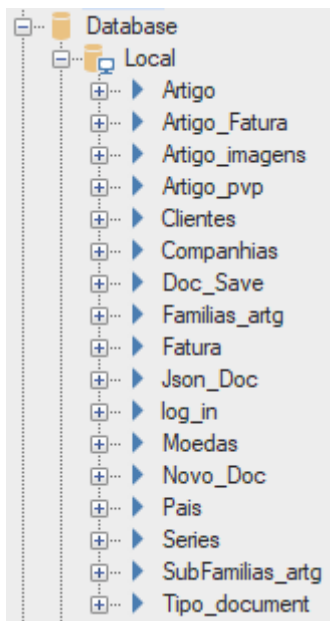


Figura 16 - Tabelas de dados presentes na aplicação

## 7.2. Arquitetura Lógica e utilização

Após serem definidos as tabelas de dados foi compreendida a estrutura inicial desenhada na fase inicial do projeto. A partir deste desenho foi aprimorada a navegação lógica da aplicação.

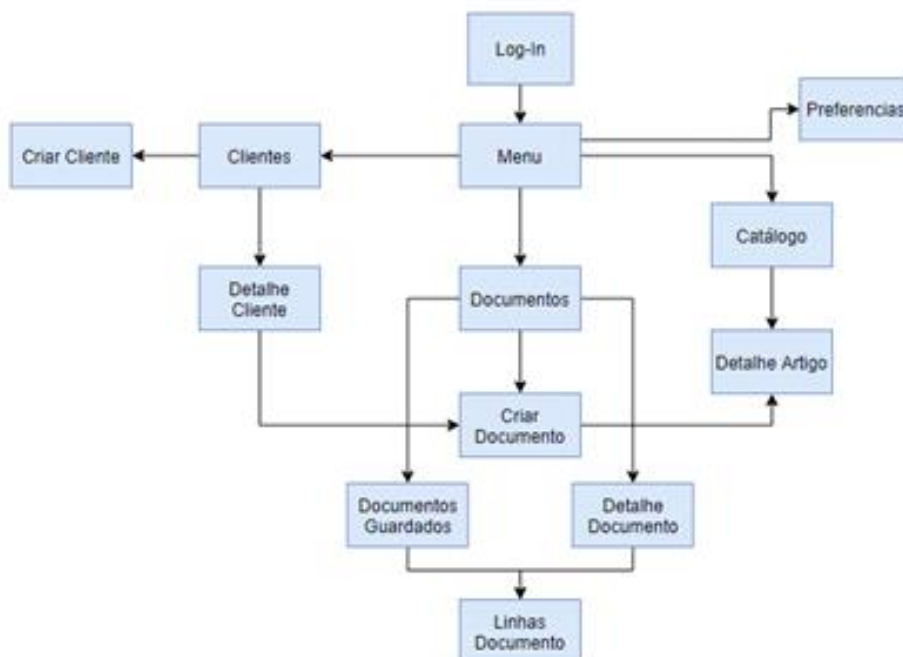


Figura 17 - Estrutura lógica final da aplicação V10

A arquitetura final acordada é composta por uma página de login que dará acesso ao menu principal. Através do menu principal será possível aceder à listagem de documentos, documentos guardados (caso existam), listagem de clientes, catálogo e definições.

A página de listagem de documentos dará também acesso ao detalhe de cada documento, à página de criação de documento e documentos guardados. A criação de documentos terá também acesso ao catálogo de forma a ser possível listar os artigos.

A listagem de clientes, à semelhança da listagem de documentos, será possível aceder ao detalhe de cada cliente e à página de criação de clientes. A página de detalhe de cliente permitirá ao utilizador navegar para a criação de documento de forma a criar uma venda para o cliente selecionado.

O catálogo permitirá aceder ao detalhe de todos os artigos.

A aplicação irá apresentar uma página inicial de login que irá ser disponibilizada no primeiro login que o utilizador irá realizar, sendo que após realizar o mesmo com sucesso, sempre que iniciar a aplicação irá entrar automaticamente, oferecendo assim maior rapidez de uso ao utilizador.

O menu de login é composto por Utilizador, *Password*, *Url*, *Port* e companhia. Estes campos oferecem flexibilidade de acesso ao utilizador, permitindo a este inserir a companhia que pretender, partindo do princípio que está associado à mesma.

O *Url* e *Port* são definidos pela organização.



Figura 18 - Página de Login

Após o login irá ser disponibilizado o menu principal da aplicação. O menu disponibiliza o centro principal de navegação da aplicação, estando dividido em duas páginas que disponibilizam ao utilizador informações estatísticas sobre o panorama de vendas.

As estatísticas são disponibilizadas em gráficos de barras que podem ser manipulados com recurso a combobox que permite ao utilizador selecionar a periodicidade e companhia que pretende que seja demonstrada. É através do menu que o utilizador pode navegar para a listagem de clientes, documentos e catálogo.



Figura 20 - Menu Principal 1

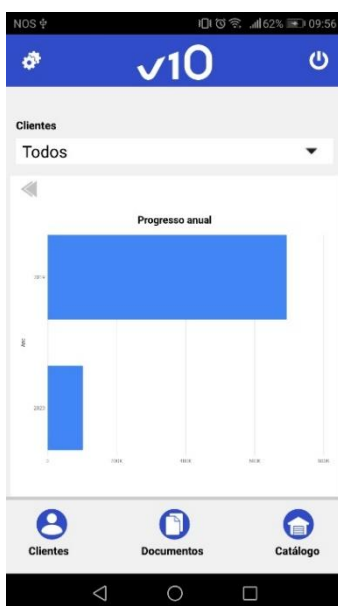


Figura 19 - Menu Principal 2



Figura 21 - Menu Principal notificação

No cenário de existirem documentos guardados na aplicação, será demonstrada uma barra de aviso amarela de forma a chamar o utilizador à atenção para que estes não sejam esquecidos.

Através do ícone de Clientes presente na página de Menu é possível aceder à listagem de clientes da companhia associada ao *LogIn*. Nesta Listagem é possível filtrar os clientes por nome através do ícone “search” presente no canto superior direito.

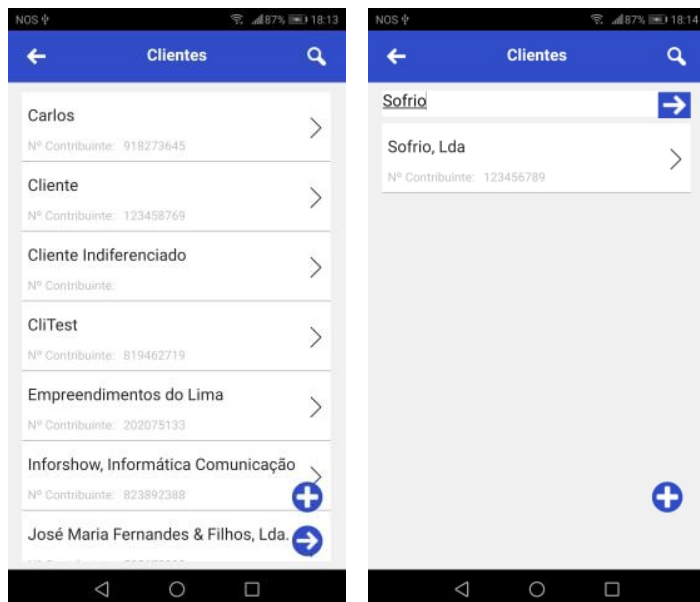


Figura 23 - Listagem de clientes

Figura 22 - Listagem de clientes com filtro

Selecionando um cliente a partir da lista de clientes é possível verificar as informações detalhadas do cliente selecionado. Através desta página é possível navegar para a página de criação de documento selecionando o ícone no canto superior direito.

Selecionando a Morada é possível aceder à mesma no [googlemaps](#) de forma direta. Através do contacto é possível efetuar uma chamada direta e/ou selecionando o email, é possível aceder diretamente ao email.



Figura 24 - Detalhe de cliente

Através do ícone “+” presente na lista de clientes é possível aceder à página de criação de cliente. Após o preenchimento dos dados estes são enviados diretamente para o ERP V10.

Figura 25 - Criar cliente 1

Figura 26 - Criar cliente 2

Através do ícone de Catálogo presente na página de Menu é possível aceder à listagem de artigos da companhia associada ao *LogIn*. Nesta Listagem é possível filtrar os artigos por código de artigo, descrição, família e subfamília através do ícone “search” presente no canto superior direito.



Figura 29 - Catálogo c/imagem



Figura 28 - Catálogo c/filtro



Figura 27 - Detalhe de artigo



A lista de artigos é apresentada em várias páginas, variando de forma dinâmica dependendo da quantidade de artigos presentes na base de dados. É possível também alternar entre uma lista simples e lista com imagens.



Figura 30 - Catálogo s/imagem

Através do ícone de Documentos presente na página de Menu é possível aceder à listagem de documentos da companhia associada ao *LogIn*. Nesta Listagem é possível filtrar os documentos de forma cronológica, tipo de documento e entidade através do ícone “search” presente no canto superior direito.

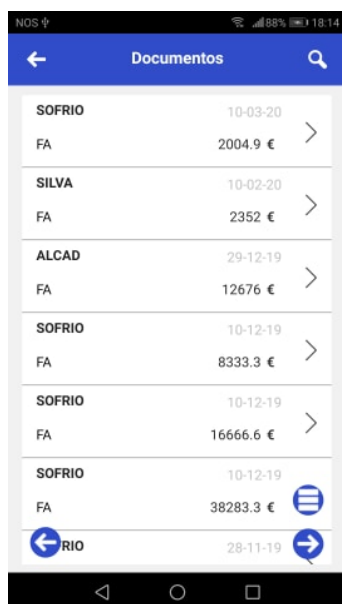


Figura 33 - Listagem de documentos

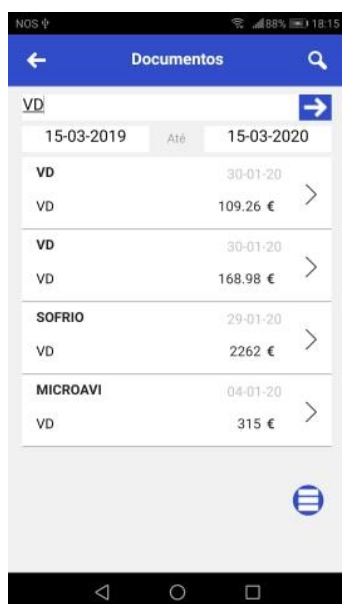


Figura 32 - Listagem de documentos com filtro



Figura 31 - Ações de listagem de documento

Ao selecionar um documento apresenta na lista de documentos, é possível aceder aos seus detalhes. É possível também partilhar e efetuar o download do documento selecionado.

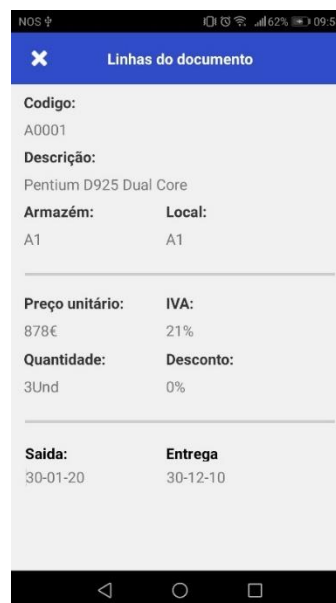
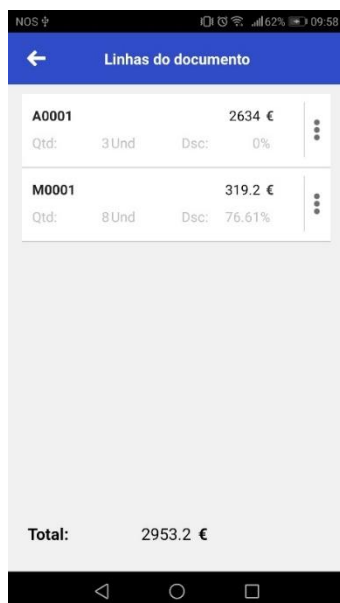


Figura 36 - Detalhe de documento

Figura 35 - Detalhe de documento (linhas)

Figura 34 - Detalhe de linhas de documento

Através da página de listagem de documentos, é possível aceder á página de criação de documentos. Durante a criação do documento o utilizador pode escolher entre guardar ou enviar o documento. Para guardar, o utilizador tem de selecionar o ícone presente na página de cabeçalho do documento no canto superior direito.

A criação de documento é distribuída em 3 fases:

- Cabeçalho;
- Linhas;
- Seleção de artigo.

Nas páginas de cabeçalho e linhas o utilizador pode escolher inserir as informações de forma manual ou recorrendo às informações presentes na base de dados.

Se o utilizador escolher selecionar os artigos recorrendo às informações presentes no dispositivo pode escolher através do catálogo ou através de uma leitura de código de barras recorrendo ao ícone presente na página de linhas no canto superior direito.

Acedendo ao artigo através de página de criação de documento, o detalhe de artigo é apresentado apenas com as informações relevantes e escolha de quantidade e desconto que

o utilizador escolha. Nesta página, são apresentados os 3 descontos mais utilizados para o produto selecionado.

Caso o utilizador esteja a aceder à aplicação de forma *offline*, apenas terá a opção de guardar o documento.

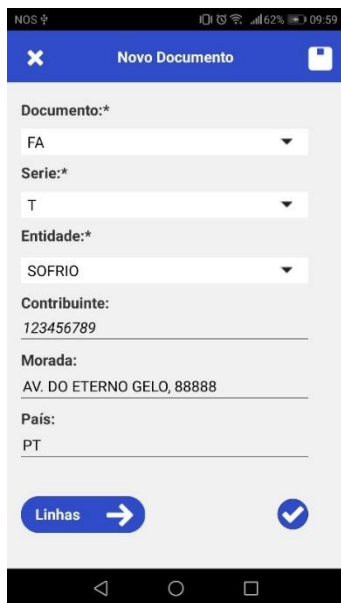


Figura 39 - Criação de documento (cabecalho)

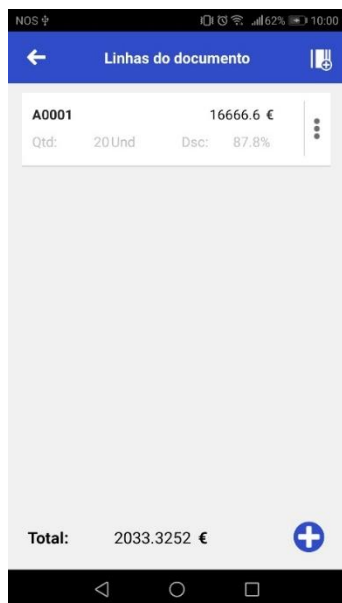


Figura 38 - Criação de documento (linhas)

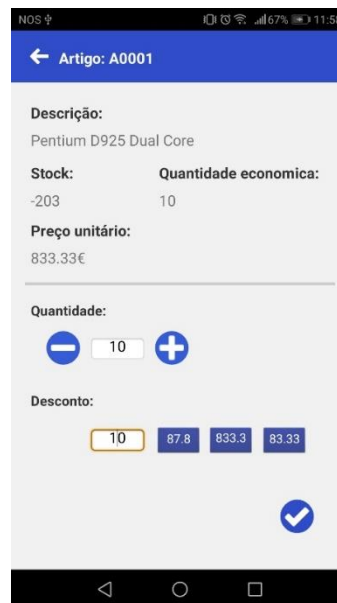


Figura 37 - Criação de documento (escolha de artigo)

Através da página de listagem de documentos é possível aceder à página de documentos guardados. Nesta página, o utilizador pode integrar os documentos que escolheu guardar na aplicação com o ERP V10.

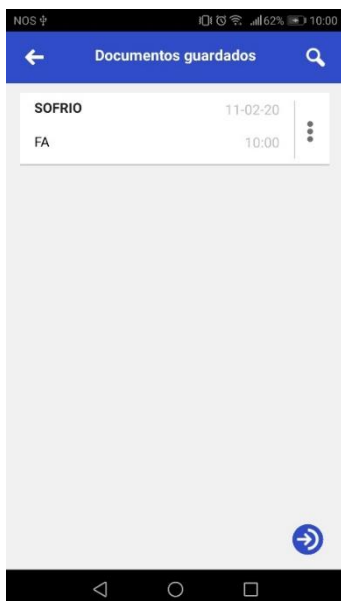


Figura 41 - Listagem de documentos guardados



Figura 40 - Detalhe de documento guardado

Na página de preferências de menu, é possível escolher as preferências gerais da aplicação e importar dados do ERP de forma manual.



Figura 42 - Definições de Menu

Na página de definições de Login é possível escolher as preferências da aplicação relacionados com dados de utilizador. Quando um destes dados é atualizado, a aplicação efetua um Login automático de forma a atualizar os dados da aplicação.

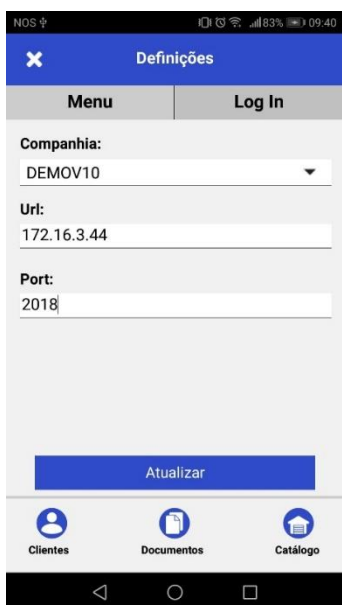
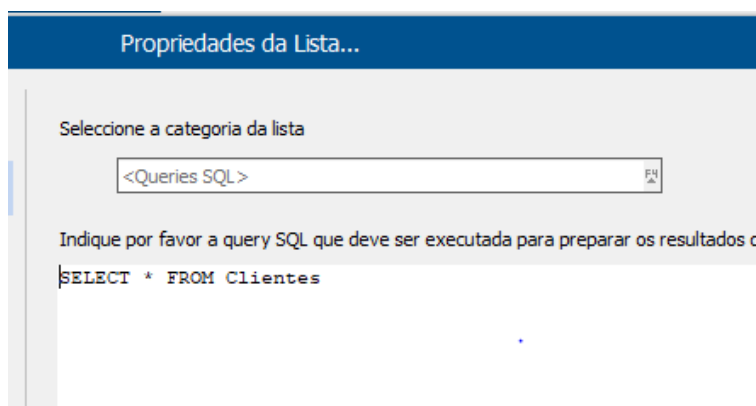


Figura 43 - Definições de Login

### 7.3. Paginação via listas personalizadas

Como discutido no capítulo de explicação do ERP V10, foi utilizado o sistema de listas personalizadas como meio de completar os pedidos existentes da Web API. Estas listas serviram, numa primeira fase, para efetuar chamadas simples a tabelas específicas na BD do ERP. A imagem seguinte demonstra a utilização de uma *querie* simples.



Propriedades da Lista...

Seleccione a categoria da lista

<Queries SQL>

Indique por favor a query SQL que deve ser executada para preparar os resultados c

```
SELECT * FROM Clientes
```

Figura 44 - Input de querie na lista

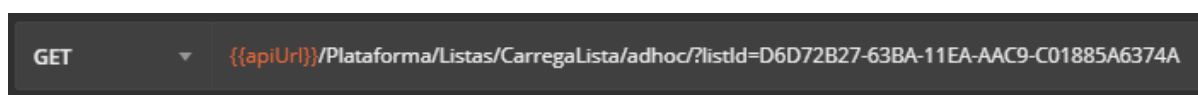


Figura 45 – Endpoint gerado

Na fase final do desenvolvimento, foi utilizado este sistema de listas para a utilização de paginação nas listagens de clientes, documentos e artigos. O sistema de paginação foi utilizado como meio de substituição a importar todos os clientes, artigos ou documentos presentes no ERP para a aplicação no dispositivo. Devido à aplicação consumir os dados de listagem diretamente do armazenamento local do dispositivo, o que implica uma importação prévia de todos os dados através da Web API, e a possível existência de milhares de registos na BD, foi necessário a implementação deste sistema como meio de processamento dos dados pretendidos de listagem.

Para efetuar este método, foram construídas um conjunto de SP's que respondessem a todas as listagens necessárias (Clientes, documentos e artigos). De forma a potencializar o uso de paginação através de SP, foi incorporado nestas o sistema de filtragem da aplicação, criando assim *queries* mais complexas com um conjunto de variáveis que correspondem aos diferentes campos da BD.

Cada SP contém duas variáveis que correspondem ao número de linhas que definem a paginação. A primeira variável é a utilizada de forma dinâmica através dos pedidos da Web API, enquanto que a segunda é estática, contendo um número fixo definido na lista do ERP. As restantes variáveis variam consoante a listagem pretendida, sendo geralmente orientadas a informação de identificação de um cliente, artigo ou documento.

A imagem seguinte demonstra uma SP que origina um pedido através de uma lista personalizada.

```

]ALTER PROCEDURE [dbo].[Offset_artigos1]

    @artigo varchar(255) = NULL,
    @familia varchar(255) = NULL,
    @subfamilia varchar(255) = NULL,
    @Linhas1 integer,
    @Linhas2 integer

AS
]BEGIN

    SET NOCOUNT ON;

]
    SELECT * FROM Artigo
        WHERE Artigo LIKE '%' + REPLACE(@artigo,'per','%') + '%'
        AND Familia LIKE '%' + REPLACE(@familia,'per','%') + '%'
        AND SubFamilia LIKE '%' + REPLACE(@subfamilia,'per','%') + '%'

        ORDER BY artigo ASC OFFSET @Linhas1 ROWS
        FETCH NEXT @Linhas2 ROWS ONLY
]

```

Figura 46 - Stored procedure de paginação de artigos

Propriedades da Lista...

Seleccione a categoria da lista

<Queries SQL>

Indique por favor a query SQL que deve ser executada para preparar os resultados da

```

Offset_artigos1 @artigo = '@@artigo@@", @familia =
'@@familia@@", @subfamilia = '@@subfamilia@@", @Linhas1 =
@@linhas1@@", @Linhas2 = 25|

```

Figura 47 – Chamada da stored procedure através das listas personalizadas

A utilização deste sistema de paginação permitiu a uma implementação de listagem mais flexível, mas acima de tudo, diminuiu o tempo de espera de pedidos de importação de dados para o dispositivo. O tempo de espera foi reduzido para mais de metade, tornando assim a experiência de utilização da aplicação muito mais fluida e eventualmente mais rápida.

## 8. Processo de Scrum Solo

A metodologia Scrum Solo foi utilizada apenas durante o desenvolvimento da solução, não tendo sido incluída no processo de investigação e construção do presente documento. Esta foi implementada seguindo as boas práticas do Scrum de forma a promover um bom desenvolvimento e processamento do projeto.

Este projeto passou por uma fase inicial de definição de requisitos que teve como objetivo a definição do âmbito da solução e o *product backlog*. De forma a uma melhor recolha dos requisitos, foram feitas várias reuniões com o orientador do projeto ou um representante da empresa, mantendo assim os requisitos em ordem com o âmbito. Estas reuniões foram feitas de uma forma frequente na fase inicial do projeto, passando a reuniões espontâneas com o surgimento de novos requisitos em fases avançadas do projeto.

Na seguinte imagem vemos a interação dos vários atores durante a definição dos requisitos.

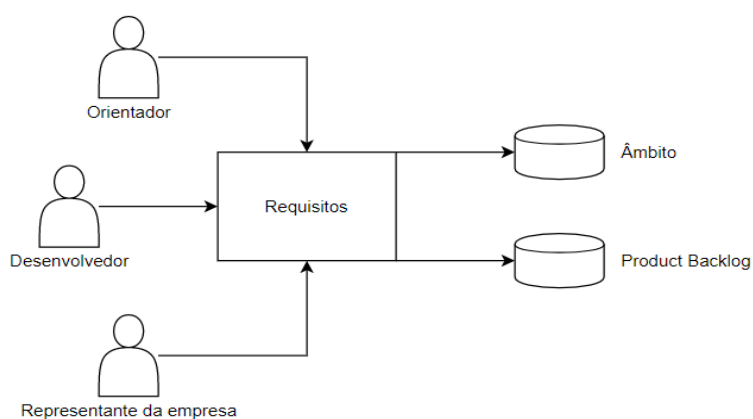


Figura 48 - Definição de requisitos segundo Scrum Solo

Através deste processo de recolha de requisitos foi possível a aquisição dos requisitos que formaram o *product backlog*, que por sua vez, após a conclusão destas, foi possível obter uma solução que se adequasse ao perfil do cliente.

Numa seguinte fase foram acordadas as sprints a serem seguidas. Foi acordado que estas teriam durações variada conforme os módulos abordados. Estes módulos representaram numa forma inicial, as diferentes componentes constituintes da aplicação (ex. Menu, Página de clientes, catálogo, etc) sendo que passaram a ser divididas de forma personalizada consoante o acréscimo de novas funcionalidades ao longo do projeto.

As sprints tiveram início a 18 de outubro de 2019, estando já o processo de design e visão do projeto completo. O tempo de trabalho no projeto engloba 3 dias por semana sendo que a seguinte imagem representa o *product backlog* e as devidas sprints realizadas.

Na tabela 2 vemos a listagem do *Product Backlog*, descrevendo todas as tarefas realizadas com a sua sprint respectiva.

Tabela 2 - *Product Backlog* com as sprints associadas a cada requisito

Módulo	Descrição	Sprint		Retrabalhado
		Início	Conclusão	
Login	Criação de interface de Login	18/11/2019	20/11/2019	Sim
Login	Integração de Login com o ERP	18/11/2019	20/11/2019	Sim
Menu	Criação de interface do Menu principal	25/11/2019	25/11/2019	Sim
Menu	Implementação de sistema de processamento de dados estatísticos	06/01/2020	08/01/2020	Sim
Menu	Conceção e implementação de gráficos estatísticos na <i>dashboard</i> do menu	06/01/2020	06/01/2020	Sim
Menu	Criação de interface de definições da aplicação	13/01/2020	15/01/2020	Não
Menu	Implementação de opções de filtragem dos dados estatísticos	13/01/2020	15/01/2020	Não
Menu	Sistema de integração de dados global com o ERP	13/01/2020	15/01/2020	Sim
Menu	Implementação de opção de visualização de catálogo	18/02/2020	19/02/2020	Não
Menu	Implementação de alteração de informações de Login	18/02/2020	19/02/2020	Sim



Cientes	Criação de interface da página de cliente	26/11/2019	3/12/2019	Sim
Cientes	Listagem de clientes através do ERP	26/11/2019	3/12/2019	Não
Cientes	Criação de interface de detalhe de cliente	26/11/2019	3/12/2019	Sim
Cientes	Implementação de filtro de pesquisa de clientes	24/12/2019	24/12/2019	Sim
Cientes	Criação de interface de menu de criação de cliente	20/01/2020	21/01/2020	Sim
Cientes	Integração de criação de cliente com o ERP	20/01/2020	21/01/2020	Não
Cientes	Ligação de detalhe de cliente com o menu de criação de documento	29/01/2020	04/02/2020	Não
Cientes	Integração da morada de cliente com <i>googlemaps</i>	29/01/2020	04/02/2020	Não
Cientes	Integração do contacto e email de cliente com sistema nativo do dispositivo	29/01/2020	04/02/2020	Não
Cientes	Implementação de sistema de paginação de listagem de clientes	24/02/2020	04/03/2020	Não
Documentos	Criação de interface da página de documentos	04/12/2019	11/12/2019	Sim
Documentos	Listagem de documentos através do ERP	04/12/2019	11/12/2019	Não
Documentos	Criação de interface de detalhe de documento	04/12/2019	11/12/2019	Sim
Documentos	Implementação de filtro de pesquisa de documentos	24/12/2019	24/12/2019	Sim

Documentos	Criação de interface do menu de criação de documento	22/01/2020	28/01/2020	Sim
Documentos	Integração de criação de documento com o ERP	22/01/2020	28/01/2020	Não
Documentos	Integração de sistema de download de documentos via pdf	05/02/2020	10/02/2020	Não
Documentos	Integração de sistema de partilha de documentos via aplicações sociais.	05/02/2020	10/02/2020	Não
Documentos	Sistema de processamento de dados personalizados de criação de documento	05/02/2020	10/02/2020	Sim
Documentos	Implementação de sistema de paginação de listagem de documentos	24/02/2020	04/03/2020	Não
Documentos Guardados	Criação de interface da página de documentos guardados	11/02/2020	17/02/2020	Não
Documentos Guardados	Listagem de documentos guardados	11/02/2020	17/02/2020	Não
Documentos Guardados	Criação de interface de detalhe de documentos guardados	11/02/2020	17/02/2020	Não
Documentos Guardados	Implementação de filtro de pesquisa de documentos guardados	11/02/2020	17/02/2020	Não
Documentos Guardados	Sistema personalizado de integração de documentos guardados no dispositivo com o ERP	11/02/2020	17/02/2020	Sim
Catálogo	Criação de interface da página de catálogo	16/12/2019	23/12/2019	Sim

Catálogo	Listagem de artigos através do ERP	16/12/2019	23/12/2019	Não
Catálogo	Criação de interface de detalhe de artigo	16/12/2019	23/12/2019	Sim
Catálogo	Implementação de filtro de pesquisa de artigos	24/12/2019	24/12/2019	Sim
Catálogo	Implementação de leitura de código de barras com catálogo	22/01/2020	22/01/2020	Não
Catálogo	Integração de imagens de artigos no catálogo	18/02/2020	19/02/2020	Não
Catálogo	Implementação de sistema de paginação de listagem de artigos	24/02/2020	04/03/2020	Não

As tarefas listadas no *product backlog* foram distribuídas por módulos que correspondessem à funcionalidade global presente na aplicação, ou seja, cada tarefa pertence a uma sub tarefa global a realizar. Na tabela, é também descrito o intervalo de tempo necessário (sprint). É também referido se a mesma foi novamente abordada após a finalização da respetiva sprint, indicando que algum aspeto teve que sofrer algum trabalho extra não previsto.

Após cada sprint, eram discutidos, através de reuniões com o orientador ou um representante da empresa, os requisitos completados de forma a validar e a discutir o progresso da aplicação, mantendo assim todos os intervenientes em contacto com a evolução do projeto.

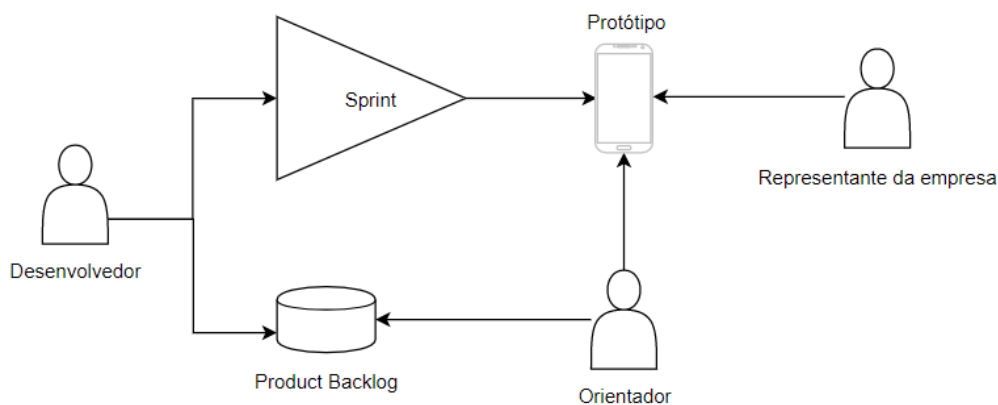


Figura 49 - Revisão de sprint (Deployment e Management)

Esta validação é feita através da entrega de um protótipo com os novos requisitos implementados e processada da forma que a figura 49 demonstra.

De forma a poderem ser corrigidos erros ou *bugs*, antes do início de cada nova *sprint* era realizado uma verificação global da aplicação. Após a finalização dos módulos iniciais definidos no início do projeto, as subseqüentes sprints foram definidas segundo requisitos adicionais que completem ou adicionem valor à solução final. Estes requisitos adicionais seguiram o mesmo processo presente na imagem anterior (Figura 38).

O seguimento desta metodologia em conjunto com o documento de visão realizado previamente, permitiu a construção de um manual de especificações técnicas detalhado que permite a um futuro desenvolvedor dar continuação ao protótipo.

Apesar do detalhe que esta metodologia permite abordar o processo de desenvolvimento, esta não contém medidas de como lidar ou prevenir contratemplos externos que possam impedir o cumprimento da sprint no intervalo de tempo estabelecido. Os contratemplos identificados durante a realização do projeto foram essencialmente associados ao estado de desenvolvimento da Web API em relação às funcionalidades do próprio ERP que requereram um esforço adicional não previsto no plano de trabalho inicial e a um uso de recursos extra. Também a utilização de um servidor de teste *localhost*, dificultou a tarefa de fornecer versões de validação no final de cada *sprint*, tendo recorrido a vários protótipos construídos com as funcionalidades desenvolvidas para a avaliação da mesma.

A existência de requisitos extra serviu também como uma forma de combater estes possíveis contratemplos, através da utilização de recursos extra foi possível explorar novas formas de complementar o projeto. Desta forma, foi possível maximizar o tempo de trabalho tirando partido do processo de *sprints* e de validação, promovendo uma expansão do *product backlog* sem nunca perder de vista o âmbito da solução.

## 9. Teste de validação e aceitação do projeto

Após a realização do projeto e construção do protótipo, foram efetuados um conjunto de casos de teste que visaram a validação das principais funcionalidades da aplicação. De forma ao protótipo ser testado na sua integridade foram listados um conjunto de cenários, tal como a tabela 3 demonstra.

Tabela 3 - Testes de validação exploratórios realizados

Teste	Cenário	Passos	Resultado esperado
<b>Login</b>	Verificar o funcionamento do processo de login.	<ol style="list-style-type: none"> <li>1. Instalar o protótipo</li> <li>2. Abrir a aplicação</li> <li>3. Inserir credenciais</li> <li>4. Efetuar login</li> </ol>	<ol style="list-style-type: none"> <li>1. O login ser efetuado com sucesso</li> <li>2. A aplicação navegar posteriormente para o menu</li> <li>3. Dados de login inseridos serem expostos na página de definições</li> </ol>
<b>Listagem e filtragem de clientes</b>	Verificar o funcionamento de listagem e filtragem da lista de clientes presente na página de clientes.	<ol style="list-style-type: none"> <li>1. Efetuar login na aplicação;</li> <li>2. Navegar para página de cliente através da página de menu;</li> <li>3. Verificar se a lista se encontra ativa;</li> <li>4. Inserir nome de um cliente existente na listagem.</li> </ol>	<ol style="list-style-type: none"> <li>1. Os clientes presentes no erp devem ser apresentados na listagem da página de clientes;</li> <li>2. Após a inserção de um nome de cliente na filtragem ser apresentado o mesmo na listagem.</li> </ol>

<b>Listagem e filtragem de documentos</b>	Verificar o funcionamento de listagem e filtragem da lista de documentos presente na página de clientes.	<ol style="list-style-type: none"> <li>1. Efetuar login na aplicação;</li> <li>2. Navegar para página de cliente através da página de menu;</li> <li>3. Verificar se a lista se encontra ativa;</li> <li>4. Inserir o código de documento válido para filtragem;</li> <li>5. Inserir intervalo de datas para filtragem.</li> </ol>	<ol style="list-style-type: none"> <li>1. Os documentos presentes no ERP, associados ao presente utilizador, devem ser apresentados na listagem da página de documentos;</li> <li>2. Após a inserção do código de documento na filtragem devem ser apresentados todos os documentos do mesmo;</li> <li>3. Após a inserção do intervalo de tempo na filtragem devem ser apresentados todos os documentos presentes no mesmo.</li> </ol>
<b>Listagem e filtragem de artigos</b>	Verificar o funcionamento de listagem e filtragem do catálogo presente na	<ol style="list-style-type: none"> <li>1. Efetuar login na aplicação;</li> <li>2. Navegar para página de catálogo através da página de menu;</li> <li>3. Verificar se a lista se encontra ativa;</li> </ol>	<ol style="list-style-type: none"> <li>1. Os artigos presentes no ERP associados ao presente utilizador, devem ser apresentados na listagem da página de catálogo;</li> </ol>

	página de catálogo.	<ol style="list-style-type: none"> <li>4. Inserir código de artigo de um cliente existente na listagem;</li> <li>5. Selecionar na <i>combobox</i> uma família e subfamília subsequentemente.</li> </ol>	<ol style="list-style-type: none"> <li>2. Após a inserção de código de artigo na filtragem ser apresentado o artigo na listagem;</li> <li>3. Após a inserção da família e subfamília devem ser apresentados os artigos associados aos mesmos.</li> </ol>
<b>Criação de cliente</b>	Verificar se o processo de criação de clientes consegue comunicar com o ERP.	<ol style="list-style-type: none"> <li>1. Efetuar login na aplicação;</li> <li>2. Navegar para página de clientes através da página de menu;</li> <li>3. Navegar para a página de criação de cliente;</li> <li>4. Inserir os dados</li> <li>5. Enviar o pedido de criação</li> </ol>	<ol style="list-style-type: none"> <li>1. Ser apresentado a página de criação de cliente;</li> <li>2. Ser enviado com sucesso um novo cliente para o ERP;</li> <li>3. Conseguir listar o mesmo cliente na página de cliente.</li> </ol>
<b>Criação de documento</b>	Verificar se o processo de criação de documentos consegue comunicar com o ERP e se os	<ol style="list-style-type: none"> <li>1. Efetuar o login na aplicação;</li> <li>2. Navegar para a página de documentos;</li> <li>3. Abrir a opção de criar documento;</li> <li>4. Preencher o cabeçalho;</li> </ol>	<ol style="list-style-type: none"> <li>1. Ser preenchido o cabeçalho com sucesso;</li> <li>2. Serem apresentadas as informações de artigo</li> </ol>

	processos inerentes à criação funcionam com sucesso.	<ol style="list-style-type: none"> <li>5. Navegar para a seleção das linhas do documento;</li> <li>6. Efetuar a seleção de um artigo via catálogo;</li> <li>7. Efetuar a seleção de um artigo via leitor de código de barras;</li> <li>8. Enviar pedido de criação</li> </ol>	<p>correspondente ao cliente selecionado;</p> <ol style="list-style-type: none"> <li>3. Aplicação encontrar o artigo através de leitor de código de barras;</li> <li>4. Ser registado com sucesso o documento no ERP V10.</li> </ol>
<b>Integração de documentos offline</b>	Verificar se o processo de integração de um documento guardado na aplicação consegue comunicar com o ERP V10.	<ol style="list-style-type: none"> <li>1. Efetuar o login na aplicação;</li> <li>2. Seguir o mesmo processo de criação de documento;</li> <li>3. Gravar o documento guardado no dispositivo;</li> <li>4. Navegar para o menu principal;</li> <li>5. Verificar a existência de uma notificação na zona superior da app;</li> <li>6. Navegar para a página de documentos guardados através da notificação presente;</li> <li>7. Efetuar a integração do documento guardado pretendido.</li> </ol>	<ol style="list-style-type: none"> <li>1. Ser guardado um documento guardado com sucesso;</li> <li>2. O processo de integração ser efetuado com sucesso;</li> <li>3. Ser listado o documento integrado no ERP V10.</li> </ol>



A Tabela 3 apresenta listados todos os cenários de testes realizados, sendo estes correspondentes às funcionalidades chave da aplicação. Acompanhando os cenários temos a coluna de “Teste” que representa uma breve descrição do que se está a tentar validar no processo efetuado. A seguir temos as colunas de paços e resultados esperados, sendo o primeiro o processo efetuado necessário à obtenção dos dados pretendidos. Estes passos representam o funcionamento genérico que o cliente procura. Na obtenção dos dados temos os pontos necessários a serem atingidos para que o processo seja validado e encarado como estando completo e funcional.

## 9.1. Login

De forma a efetuar o login, o utilizador deverá inserir a suas credenciais de utilizador e password da mesma forma que acede à aplicação *desktop* do ERP. Seguidamente, é requerido que insira a companhia à qual se pretende aceder, a port e nome/ip do servidor da mesma. Este nível de detalhe permite ao utilizador não só aceder a várias companhias a que esteja associado, mas também a várias infraestruturas nas quais se possa estar a trabalhar em. Este funcionamento vai de encontro ao método de negócio já praticado pela Primavera, oferecendo assim o mesmo nível de funcionalidade que a versão *desktop*.

Na imagem seguinte, vemos a ação de login seguida da sua validação na página de definições da mesma.



Figura 50 - Processo de LogIn com as credenciais pretendidas

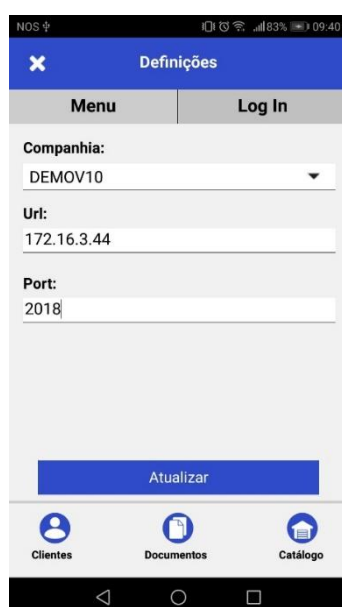


Figura 51 - Dados de login validados

Tal como foi explicado no capítulo 7.2, a alteração de qualquer um destes dados irá despoletar um novo login que resultará numa atualização global de todos os dados da aplicação, promovendo assim uma resposta rápida às necessidades de fluidez requerida.

## 9.2. Listagem e filtragem de clientes/ documentos/ catálogo

Para a validação de listagem apenas se requereu a verificação e navegação das páginas enquanto que para a filtragem foi necessário a inserção de dados de pesquisa correspondente à página em que o utilizador se encontra. Para isto todos os componentes do sistema de filtro foram utilizados tal como as figuras 52, 53, 54, 55 e 56 demonstram.

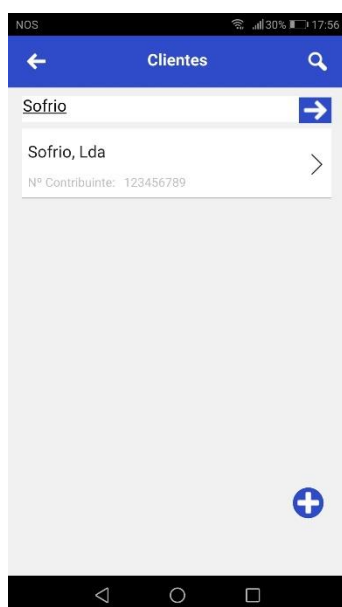


Figura 54 - Filtragem de clientes



Figura 53 - Filtragem de documentos por data

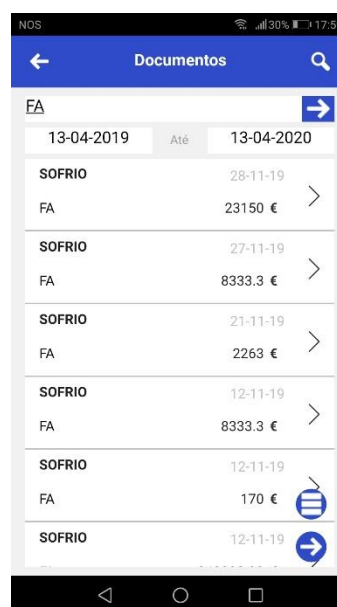


Figura 52 - Filtragem de documentos por tipo de doc

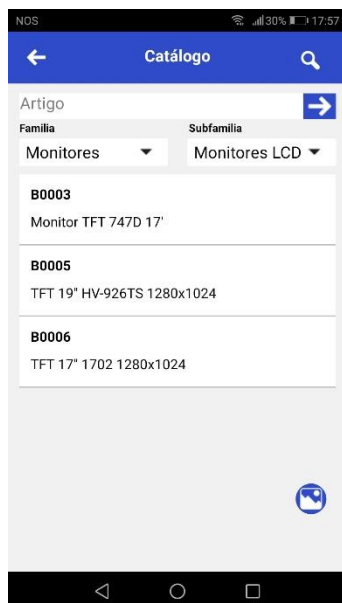


Figura 56 - Filtragem de catálogo via código de produto, família e subfamília

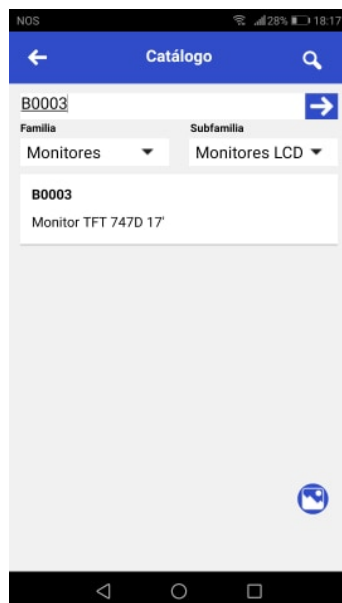


Figura 55 - Filtragem de catálogo via família e subfamília

O sistema de filtragem funciona, tal como explicado em capítulos anteriores, em conjunto com o sistema de paginação, sendo que cada vez que é efetuada uma filtragem, é requisitada uma nova paginação, que apesar de aumentar o tempo de processamento, aumenta a eficácia de processamento de informação na aplicação.

### 9.3. Criação de clientes

O processo de criação de um cliente apresenta-se de forma linear, com um conjunto de campos referidos como obrigatórios divididos em 2 páginas. De salientar que o campo de número de contribuinte e contacto telefónico apenas aceitam input numéricos com tamanho não inferior ou superior a 9 dígitos. As seguintes imagens demonstram o processo de criação de um cliente e a sua validação de criação do mesmo no ERP.

Novo Cliente

Cliente: \*  
PrimaveraBSS

Nome: \*  
APBSS

Contribuinte: \*  
123456789

Morada:  
Edifício Primavera-lamaças

Localidade:  
Braga

Código-Postal:  
4719-006

1/2

Seguinte

Figura 57 - Criação de cliente 1

Novo Cliente

País: \*  
Portugal

Distrito:  
Braga

Telefone: \*  
919204462

Fax:

Endereço Web:  
https://pt.primaverabss.com/pt/

Moeda: \*  
Euros

2/2

Voltar Guardar

Figura 58 - Criação de cliente 2

Após a criação do cliente é possível efetuar a sua listagem na aplicação, através da página de listagem de clientes, tal como é demonstrado na seguinte imagem.

Cliente

Gravar Novo Anular Listas Imprimir CRM Contexto Ajuda Cancelar

Cliente: APBSS PrimaveraBSS Anulado

Retenções Anexos Contabilidade Encargos com Títulos Cobrança Trans. Eletrónicas Encargos Bancários Campos do Utilizador

Moradas Bancos Dados Fiscais Dados Comerciais Outros Dados Crédito Observações

Morada: Edifício Primavera-lamaças

Localidade: Braga

Código Postal: 4719-006

Telefone: 919204462 Telef. 2: Fax:

Endereço Web: https://pt.primaverabss.com/pt/

Distrito: Tipo Terceiro:

Outras Moradas

Código	Morada	Morada (2)

Figura 59 - Cliente criado através da aplicação no ERP V10

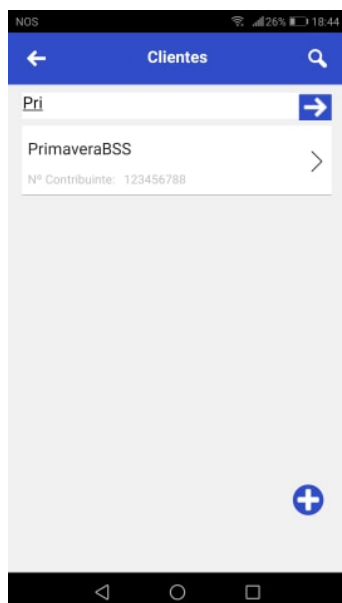


Figura 60 - Listagem do cliente criado

## 9.4. Criação de documentos/Integração de documentos guardados

A criação de um documento é considerada a precursora da criação e desenvolvimento do atual projeto, sendo que para a sua validação foram escolhidas várias métricas. Em primeiro lugar, temos a criação do cabeçalho do documento. Este cabeçalho funciona como a fundação do restante processo, sendo que sem este o acesso detalhado das informações de artigos e descontos é impossível.



Figura 62 - Preenchimento do cabeçalho do documento

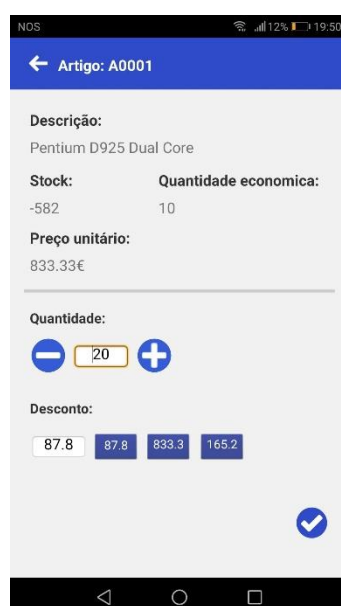


Figura 61 - Seleção de artigo via catálogo

Após o preenchimento do cabeçalho é possível ver o PVP dos artigos pretendidos de uma forma objetiva que corresponde ao cliente selecionado, sendo que também é possível verificar os descontos mais populares para o mesmo, tal como é demonstrado na imagem.

É também possível selecionar o artigo pretendido através de leitor de código de barras de forma a oferecer uma maior flexibilidade de uso da aplicação.

Na página de seleção de artigo o campo de input de quantidade do artigo irá apresentar por *default* o valor de venda mínima suportada pelo artigo selecionado, não deixando que seja inserido um valor inferior. Da mesma forma que o sistema de valor de venda mínima, este também impede que sejam inseridos valores superiores à quantidade de stock existente.



Figura 64 - Seleção de artigo via código de barras



Figura 63 - Seleção dos dados do artigo após leitura do código de barras

Esta flexibilidade oferecida na escolha de artigos e as suas características inerentes de apoio, oferecem uma elevada interatividade e funcionalidade ao utilizador que permite oferecer uma experiência rica.

Após a criação do documento, este é inserido no reportório de documentos do ERP, sendo que não é possível listar de seguida o mesmo na aplicação devido a protocolos de aceitação financeira do mesmo no ERP. Na imagem seguinte vemos o documento criado com sucesso no ERP V10.

Artigo	Arm.	Localização	Lote	Descrição	CIVA	IVA	Pr. Unit.	Desc.	UN	Qtd.	Total Liq.	Projeto
A0001	A1	A1.A.1.001		Pentium D925 Dual Core	20	20,00	833,33	87,80	UN	20,00	1 830,00	
A0003	A1	A1.A.5.001		Computador Pentium III	20	20,00	1 198,00	10,00	UN	25,00	24 259,50	

Figura 65 - Documento criado com sucesso no relatório do ERP V10

O processo de validação da integração de documentos em modo *offline* funcionou através do mesmo método de criação de documento, utilizando a mesma estrutura de pedidos, servindo o exemplo anterior como um meio de validar a funcionalidade.

O conjunto de funcionalidades de suporte e a independência de funcionamento com a versão *desktop* do ERP, permite oferecer o máximo de funcionalidade possível ao utilizador bem como a capacidade de interatividade e personalização de alguns aspetos da aplicação e formulários de criação oferece também uma componente que incorpora na vertente de aplicação rica. O funcionamento lógico contínuo dos menus e a sua simples navegação oferece também uma interface de uso fácil tornando a experiência de uso confortável e promove futuras utilizações.

## 10. Discussão sobre a realização do projeto

A caracterização do artefacto final do projeto como apenas um protótipo, deveu-se à profundidade e complexidade do ERP V10. Tendo sido lançado em 2018 o ERP encontra-se ainda numa fase de desenvolvimento e expansão, sendo a componente Web API umas das componentes que necessita de uma especial atenção, devido à sua subutilização no panorama atual móvel da organização. Através do desenvolvimento deste projeto, foi possível a identificação e otimização de certos componentes da Web API, requisitando assim um estudo extra e constante, não previsto durante todo o desenvolvimento. Tendo já sido referido anteriormente, os pedidos foram complementados através das listas personalizadas, que permitiram criar pedidos que não existiam na Web API ou que se encontravam em desenvolvimento.

Durante o estudo e utilização da Web API foram identificados uma série de pedidos que poderiam ser adicionados ou completados. Uma das principais ausências nos pedidos, foi a de dados estatísticos, sendo que apesar de existirem módulos internos do ERP que contem variadas métricas de avaliação e *performance*, não se encontravam ao alcance da Web API, tendo sido utilizado como recurso, o processamento de dados, provenientes de outros pedidos, no interior da aplicação. Além disto, alguns pedidos do ERP ainda se encontravam em desenvolvimento, tais como a listagem de moedas utilizadas, os países de atividade ou os tipos de documentos disponíveis. Por outro lado, todas as listagens, à exceção dos documentos, necessitaram de um desenvolvimento que permitissem o acesso aos dados mais e detalhados pretendidos pela aplicação.

Na tabela 4 encontram-se disponíveis todos os pedidos que foram criados de forma a conseguir dar resposta a todas as funcionalidades da aplicação.



Tabela 4 - Lista de pedidos criados

<b>Requisito</b>	<b>Pedido</b>	<b>Resolução</b>
Listar Clientes	Existente de forma incompleta.	Criação de novo pedido com recurso a lista personalizada.
Listar Artigos	Existente de forma incompleta	Criação de novo pedido com recurso a lista personalizada
Listar Documentos	Existente de forma incompleta	Completado através de uma lista personalizada que devolve dados complementares aos existentes
Criar documentos	Existente, mas com bug de processamento de dados	Criação de um processamento extra de dados na aplicação de forma a contornar o bug existente.
Listar países de atividade	Inexistente	Criação de novo pedido com recurso a lista personalizada
Listar moeda de cambio	Inexistente	Criação de novo pedido com recurso a lista personalizada
Listar tipo de documentos	Inexistente	Criação de novo pedido com recurso a lista personalizada
Listar diferentes séries	Inexistente	Criação de novo pedido com recurso a lista personalizada
Identificação de diferentes PVP	Inexistente	Criação de novo pedido com recurso a lista personalizada
Identificação de diferentes	Inexistente	Criação de novo pedido com recurso a lista personalizada

famílias/subfamílias de artigos		
Obtenção de imagens de catálogo	Inexistente	Criação de uma diretoria virtual no servidor com acesso direto à diretoria de anexos local do ERP e de um pedido que devolve as mesmas
Obtenção de estatísticas	Inexistente	Construção de estatísticas através da listagem de documentos através de <i>queries</i> diretamente na aplicação
Paginações	Inexistente	Criação de várias <i>stored procedures</i> em conjunto com as respectivas listas personalizadas que criam o pedido necessário

Nesta tabela é possível perceber quais os passos necessários para os diferentes estados em que se encontravam os pedidos necessários ao funcionamento da aplicação.

Analisando a tabela anterior, é de salientar o requisito de acesso a imagens dos artigos de catálogo. Na presente Web API apenas seria possível devolver o nome da imagem com o artigo correspondente, não sendo possível aceder ao caminho da mesma ou até o ficheiro. Para contornar este problema, foi criada uma diretoria virtual diretamente no servidor (máquina pessoal) que aponta diretamente para a pasta de anexos, que contém todas as imagens guardadas através do ERP. Cada imagem é guardada com o seu *guid* entre chavetas como nome (ex. {ea141d20-0077-42d0-ab09-e4c7ca16000f}.png). Na BD está presente uma tabela com a informação anteriormente descrita, que é importada na listagem de artigos e, quando requerido na página de catálogo, recolhe e atribui a imagem ao artigo correto.

Diferenciando de pedidos simples e como explicado também em capítulos anteriores, existiu a necessidade de criar paginações. A implementação deste sistema requereu uma análise de como proceder perante a ausência deste sistema na Web API ou no ERP em geral, pois uma implementação errada poderia levar a uma perda de eficiência geral da aplicação.

Numa primeira fase, foi estudada a hipótese de importar todos os dados da BD para a aplicação, procedendo à correspondente paginação de um pedido que acesse ao ERP e retornasse apenas os dados ainda não importados para aplicação, tornando este sistema um de atualização dos dados. Apesar de ser promissor, este primeiro sistema requeria uma primeira importação demorada de dados que poderia comprometer a integridade dos restantes pedidos, sendo que seria necessário a implementação posterior de temporizadores ou sistemas de análise, que verificasse constantemente a existência de novas instâncias, tornando assim todo o processo demasiado complexo e ineficiente, numa perspetiva de tempo de desenvolvimento. No final foi implementado um sistema de *stored procedures* na BD do ERP que processam todos os dados pretendidos em *back end* e, através do uso das já comprovadas listas personalizadas que permitiram o transporte de parâmetros, possibilitaram o uso de paginação. Além das simples listagens, foram também incorporadas as funcionalidades de filtragem, otimizando assim o uso deste sistema.

Por fim, considera-se a situação da criação de documentos, que está dividida em 3 pedidos *POST*: criar cabeçalho, adicionar linhas e enviar documento final. Durante este processo, o transporte de dados é efetuado com recurso a diversos objetos JSONs, tendo sido aproveitado a utilização desta funcionalidade para efetuar uma otimização geral do processo que conseguisse, numa melhor forma, o processamento dos dados.

Além da Web API, o ERP requereu também um estudo e exploração contínuo durante toda a realização do projeto, devido a este conter vários módulos e camadas de funcionamento. Se analisarmos apenas a simples ação de aceder ao catálogo somos deparados com várias camadas de protocolos que correspondem a perfis de clientes, controlo de stock, processamento de estatísticas ou ligação de contabilidade. Todas estas diferentes camadas representam desafios acrescidos ao desenvolvimento de um protótipo que consiga ir de encontro ao perfil de uma *rich application*. Como mencionado anteriormente, a constrição de tempo levou a que o protótipo apenas representasse uma pequena parte das funcionalidades presente no ERP.

Um dos principais cuidados a ter durante a realização do projeto, foi o de seguir todos os processos lógicos e protocolos de gestão de dados presentes no ERP, tornando a implementação de cada funcionalidade, *per si*, um exercício de gestão de informação. Esta

informação pode ser encontrada de forma prevalente no processo de criação de documentos, onde foi necessário integrar os diferentes pedidos de gestão de informação necessários para a integração de documentos com o ERP. Também, na escolha dos artigos do documento, foi necessária a compreensão da existência de diferentes perfis de clientes que engloba a existência de diferentes preços unitários e descontos associados a cada artigo.

Na sua íntegra, o processo de desenvolvimento deste protótipo foi acompanhado por um processo de estudo, compreensão e análise de todas as infraestruturas envolvidas, tomando partido da vertente móvel para otimizar todos os periféricos intervenientes.

## 11. Conclusões

Apesar de já contar com décadas de conceção, construção e desenvolvimento de sistemas de informação, a área de investigação apresenta ainda algumas dificuldades em tentar recolher e, principalmente, compreender todos os requisitos que consigam dar resposta a um dado problema. Neste contexto, a investigação realizada foi dificultada pela pouca informação relacionada, expressamente para o tema de Desmaterialização de sistemas e mobilidade empresarial, dois temas que, apesar de populares, apresentam-se como partes íntegras de outros temas abordados na área de tecnologias de sistemas da informação, dificultando a tarefa de investigação, obrigando a uma procura mais alargada e generalizada dos tópicos.

A investigação efetuada seguiu o pensamento de reengenharia e implementação de processos, dois temas que aparecem muitas vezes associados com a desmaterialização de processos, oferecendo, assim, uma vista geral do que significa este tema, tal como explicado no capítulo 3. Também explorado desta forma, foi a temática de mobilidade, que complementou a desmaterialização com recurso ao conceito de *Rich Application*, ponto que foi abordado nos capítulos 3 e 6.

Para auxiliar o processo de investigação e compreensão dos temas que compreenderam o trabalho realizado foi utilizado a metodologia de investigação DSR que, durante todas as fases de investigação e desenvolvimento, auxiliou a interpretação do problema abordando a sua relevância através dos ciclos de rigor, relevância e desenho, tal como compreendido no capítulo 2.

O artefacto obtido neste projeto serviu para validar a utilização do DSR, sendo que, através do uso desta metodologia durante o desenvolvimento do trabalho, se obteve um contributo de conhecimento sobre a sua implementação no panorama atual da empresa.

A DSR possibilitou que o seguimento da metodologia Scrum Solo fosse eficaz, permitindo que a organização das sprints fosse facilitada, de uma forma mais eficiente. Por sua vez, esta metodologia prática de desenvolvimento permitiu uma abordagem agilizada, que disponibilizou uma organização eficiente na construção sistemática dos requisitos, mantendo sempre uma visão global do projeto, sem perder de vista a meta final do mesmo.

Seguindo o principal objetivo deste trabalho de dissertação, o artefacto obtido, na forma de um protótipo funcional, permitiu a comprovação do trabalho realizado, dando resposta a

todos requisitos requeridos pelo cliente. À medida que estes foram completados, uma comparação posterior com os processos através do processo Scrum Solo, permitiu que fossem completados de forma a responder ao máximo de cenários possíveis. Destes cenários, a integração com o ERP V10 e os seus subsequentes módulos de atividade foram abordados de forma mais eficiente, depois de uma revisão mais metódica dos processos após o início da fase de realização.

As funcionalidades presentes no protótipo obtido conseguem dar resposta ao tema principal de desmaterialização de processos, tornando o processo anteriormente existente de transporte de dados por via analógica, num processo digital e flexível, sendo que, agora, permite a construção de documentos de venda e criação de clientes em qualquer lugar. Com isto, a mobilidade é também validada graças ao funcionamento online e offline que oferece uma flexibilidade à utilização da aplicação. Esta implementação permitiu, também, um incremento de utilização das funcionalidades da aplicação, fazendo com que, mesmo em situações de falta de rede, seja possível manter o processo aplicado de desmaterialização de processos, não permitindo que haja redundância situacional.

A mobilidade é complementada pela possibilidade de consulta do catálogo de artigo e lista de clientes correspondente à organização, sendo que, através do uso do conceito de *rich application*, promove a utilização e aceitação desta componente do ERP móvel. Além disto, e apesar de a aplicação não conter todas as funcionalidades do ERP, o seguimento deste conceito de *rich application* fez com que todas as funcionalidades aplicadas na aplicação fossem desenvolvidas com o intuito de permitirem ao utilizador realizar a tarefa pretendida com um nível de restrição baixo de processos.

Este tema suportou, também, a potencialização da plataforma Kalipso, sendo uma base de validação da mesma como um ambiente de desenvolvimento capaz de dar resposta às necessidades presentes na Primavera, justificando assim o investimento realizado. Este desenvolvimento demonstrou ainda o potencial de integração presente nesta nova versão do software Primavera, que poderá num futuro próximo ingressar em mais projetos de natureza móvel, tal como explorado anteriormente.

Por fim, foi entregue um conjunto de documentos, tal como previsto nos objetivos de dissertação, que serviram para validar toda a visão, especificação técnica e utilização do

protótipo. Estes documentos validam a credibilidade da potencialidade deste projeto de forma a ser possível ser aplicada uma continuidade do desenvolvimento.

Este protótipo, como referido em capítulos anteriores, foi desenvolvido em ambiente de consultoria, tendo sido apresentado no final do estágio perante os responsáveis e interessados no tema em questão, sendo que, no final, ficou na supervisão do orientador desta dissertação na organização, estando de momento em *stand by* devido a fatores externos, existindo a intenção de servir como meio de promoção da potencialidade de mobilidade, tal como o objetivo inicial do projeto.

---

## 12. Referências

- Guedes, V., Santos, C., & Ferreira, A. (2010). Controlo Interno: Impacto das Novas Tecnologias na Desmaterialização de Processos. 19.
- Ávila, L., Teixeira, L., & Almeida, P. (2012). *Desmaterialização de processos com recurso a tecnologias open-source numa instituição de ensino superior*. 10.
- Ranoya, G. (2012). Tecnologias da desmaterialização. *Novos Olhares*, 13, 22.
- Santos, A. P. dos, & Barbosa, R. R. (2011). Desafios da mobilidade corporativa para a Gestão da Informação e do Conhecimento. *Informação & Sociedade: Estudos*, 21(2).
- Insight, I. T. (2017, Novembro 2). *IT Insight—A era da mobilidade empresarial moderna*. IT Insight. <https://www.itinsight.pt/news/round-table/a-era-da-mobilidade-empresarial-moderna>
- O que é mobilidade empresarial? + 10 benefícios específicos*. (2019, Março 1). Blog da Field Control. <https://fieldcontrol.com.br/blog/processos/o-que-e-mobilidade-empresarial/>
- Santos, A. P. dos. (2011). Impactos da tecnologia móvel e sem fio nos sistemas de informação empresariais sob a perspectiva dos atores sociais: Estudo de caso. *Perspectivas em Ciência da Informação*, 16(3), 275–275.
- March, V., Gu, Y., Leonardi, E., Goh, G., Kirchberg, M., & Lee, B. S. (2011).  $\mu$ Cloud: Towards a New Paradigm of Rich Mobile Applications. *Procedia Computer Science*, 5, 618–624.
- Kozlov, S. (2017, Agosto 11). *Why It's Important for Your Business to Make Your Mobile App Work Offline*. Business 2 Community. <https://www.business2community.com/mobile-apps/important-business-make-mobile-app-work-offline-01897628>



Abolfazli, S., Sanaei, Z., Gani, A., Xia, F., & Yang, L. T. (2014). Rich Mobile Applications: Genesis, taxonomy, and open issues. *Journal of Network and Computer Applications*, 40, 345–362.

Shah, S. (2018, Fevereiro 19). *Why you should develop an offline mobile app*. Digital Doughnut. <https://www.digitaldoughnut.com/articles/2018/february/why-you-should-develop-an-offline-mobile-app>

T. Pagotto, J. A. Fabri, A. Lerario and J. A. Gonçalves (2016). "Scrum solo: Software process for individual development," *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, Las Palmas, pp. 1-6.

Hevner, A. R. (2019). *A Three Cycle View of Design Science Research*. 7.

Design science (methodology). Em *Wikipedia*.

[https://en.wikipedia.org/w/index.php?title=Design\\_science\\_\(methodology\)&oldid=9178430](https://en.wikipedia.org/w/index.php?title=Design_science_(methodology)&oldid=9178430)  
31

Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77.

Hevner, A. R., March, S. T., Park, J., & Ram, S. *Design Science in Information Systems Research*. 33.

Gábor, A. (2007). *Techniques for pagination in SQL*. 14.

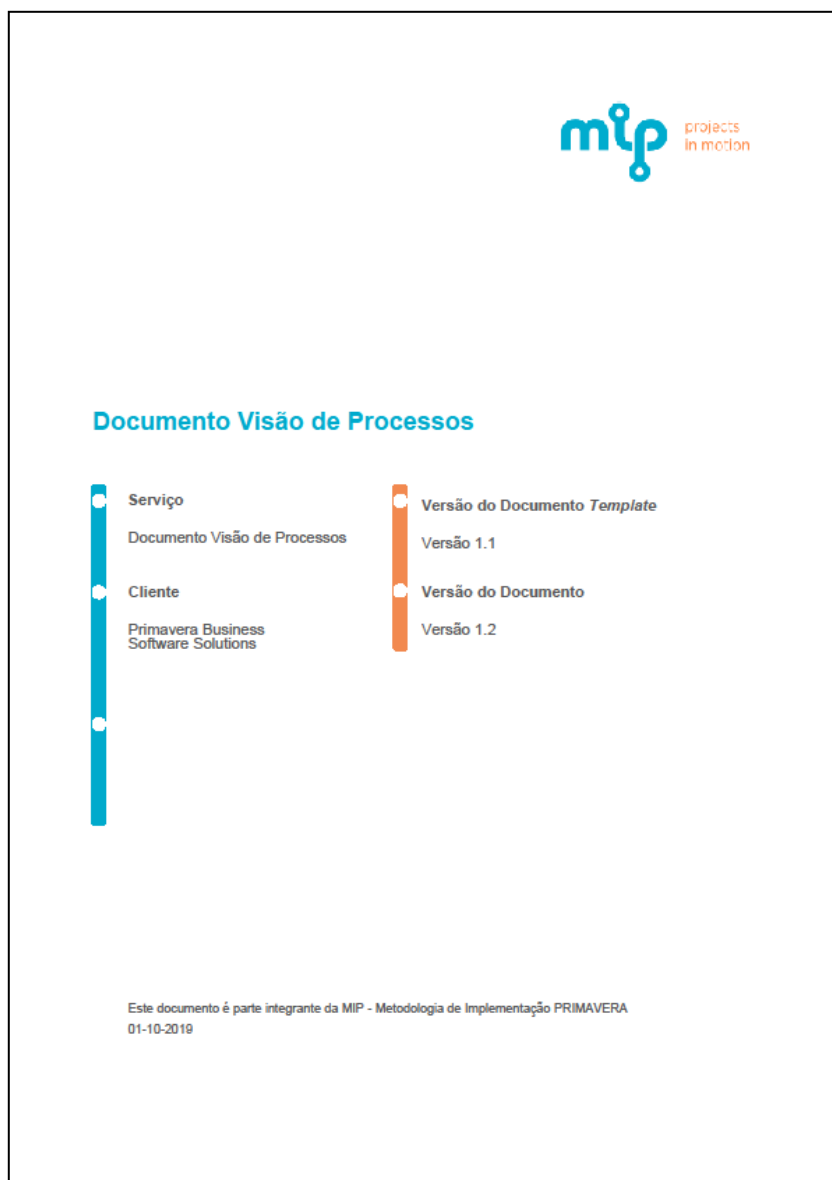
Spaskovsky, P. (2014). Advanced pagination of the result. *AIP Conference Proceedings 1631*.

Diz, J. (2020, Fevereiro 3). Paginação de resultado de consulta SQL. *Porto SQL*.

<https://portosql.wordpress.com/2020/02/03/paginacao-consulta-sql/>

## Anexo 1 – Documento de visão de processos

O seguinte documento em anexo foi realizado durante a fase de análise do projeto e representa a visão do projeto realizado na organização, seguindo o protocolo de MIP presente mesma. Este documento pertence ao grupo de entregáveis, requisitados pela organização de forma a validar o projeto.



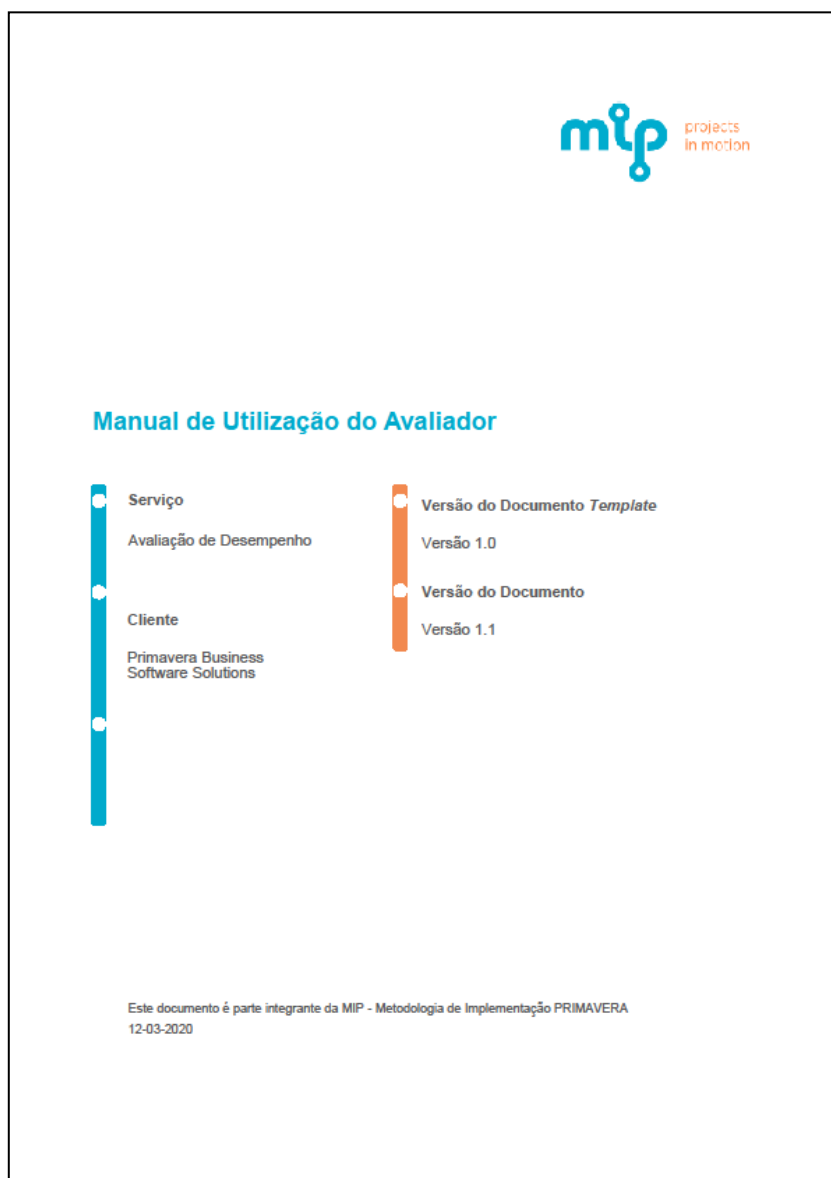
## Anexo 2 – Manual de instalação

O seguinte documento em anexo pertence ao grupo de documentos entregues na fase final da realização do projeto. Este documento apenas foi necessário devido à utilização de um servidor de teste *localhost*, sendo necessário a construção de um guia de instalação/configuração de uma máquina pessoal que consigo suportar as funcionalidades da aplicação. Este documento pertence ao grupo de entregáveis, requisitados pela organização de forma a validar o projeto.



## Anexo 3 – Manual de utilização

O seguinte documento em anexo pertence ao grupo de documentos entregues na fase final da realização do projeto. Este documento serve de um guia de utilização e navegação de todas as páginas, funcionalidades e processos lógicos da aplicação. Este documento pertence ao grupo de entregáveis, requisitados pela organização de forma a validar o projeto.




## Anexo 4 – Documento de especificações técnicas

O seguinte documento em anexo pertence ao grupo de documentos entregues na fase final da realização do projeto. Este documento serve de um guia de especificações técnicas de todas funcionalidades e processos lógicos da aplicação, funcionando como um ponto de orientação requisitado de forma a ser possível o desenvolvimento contínuo da aplicação. Este documento pertence ao grupo de entregáveis, requisitados pela organização de forma a validar o projeto.



## Documento Visão de Processos

	<b>Serviço</b>		<b>Versão do Documento <i>Template</i></b>
	Documento Visão de Processos		Versão 1.1
	<b>Cliente</b>		<b>Versão do Documento</b>
	Primavera Business Software Solutions		Versão 1.2
			



### Histórico de Versões

Consultor	Data	Alterações	Versão Doc.
Rui Silva	01-10-2019	Criação do documento	1.0
Rui Silva	21-10-2019	Alteração de imagens e reformulação de pontos	1.1
Rui Sllva	02-12-2019	Inserção de tópicos back end	1.2

# ÍNDICE

<b>1. ENQUADRAMENTO DO ESTUDO.....</b>	<b>4</b>
1.1. SUMÁRIO EXECUTIVO.....	4
<b>2. VISÃO DA IMPLEMENTAÇÃO .....</b>	<b>5</b>
2.1. ÁREA FUNCIONAL.....	5
2.1.1. Principais Necessidades de Melhoria.....	5
2.1.2. Arquitetura geral da Solução.....	5
2.1.3. Processos Funcionais .....	7
2.2. OBJETIVOS E AÇÕES PROPOSTAS .....	25
<b>CONCLUSÕES .....</b>	<b>26</b>
2.3. RECOMENDAÇÕES FINAIS .....	26
<b>3. VALIDAÇÃO DO CONTEÚDO DESTE DOCUMENTO.....</b>	<b>27</b>
3.1. APROVAÇÃO .....	27





# 1. Enquadramento do Estudo

## 1.1. Sumário executivo

O presente estudo efetuado aos objetivos de negócios da empresa PrimaveraBSS visa sobretudo sistematizar a informação recolhida ao longo do processo de análise de requisitos, realizado no âmbito da implementação da aplicação móvel V10, com base na Metodologia de Implementação PRIMAVERA (M.I.P.).

Este documento deverá ser alvo de uma análise cuidada pela gestão de topo da PrimaveraBSS com vista a ser feita uma validação dos pressupostos sobre os quais assentará todo o processo de desenvolvimento.



## 2. Visão da implementação

Neste capítulo serão representados e descritos os processos de negócio abrangidos pela implementação e que darão resposta aos objetivos de negócio dos colaboradores da PrimaveraBSS e seus requisitos chave.

### 2.1. Área Funcional

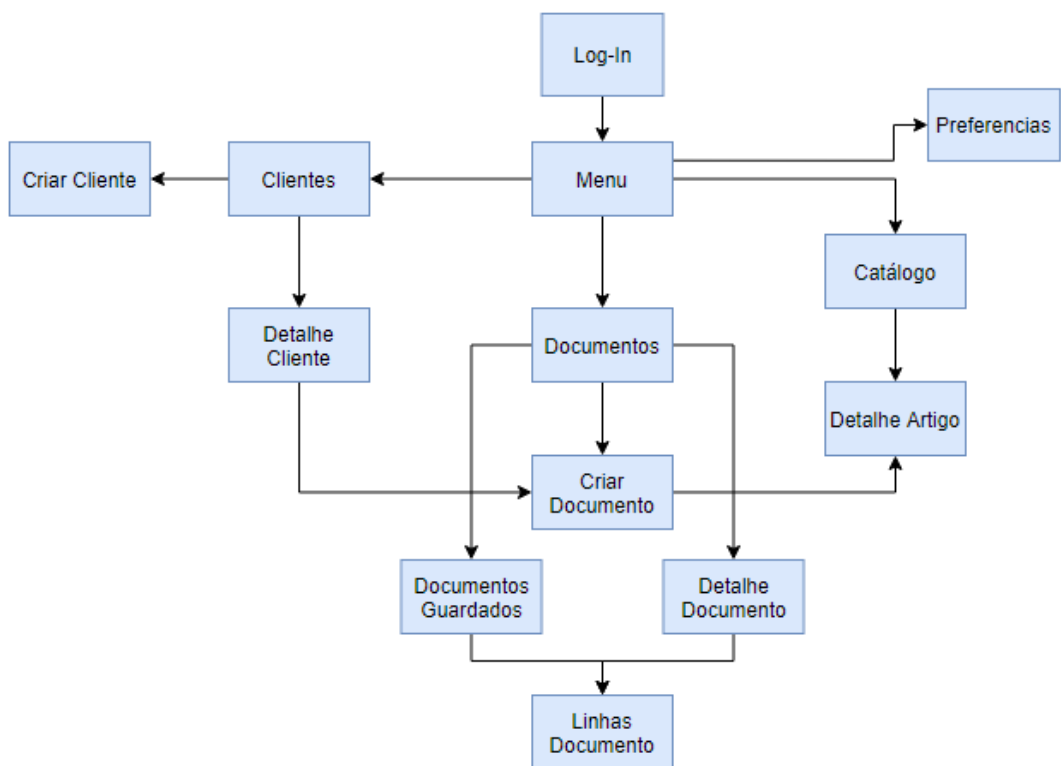
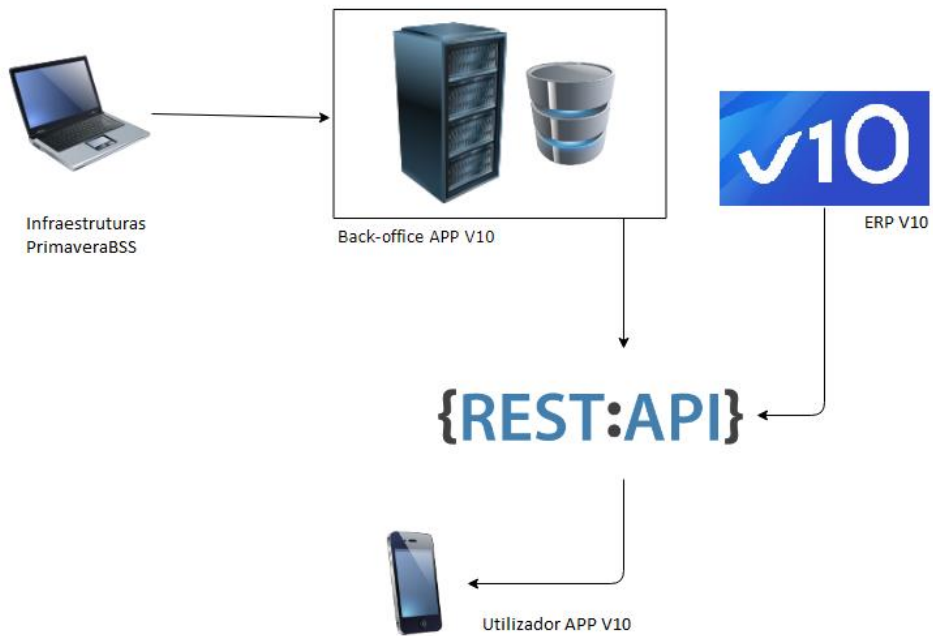
#### 2.1.1. Principais Necessidades de Melhoria

Na sequência do estudo dos processos de negócio praticados pela Primavera e pelos seus colaboradores, ao nível de faturação e gestão de vendas, é identificado o problema principal de mobilidade empresarial, no sentido de que neste momento é impossível a criação e lançamento de faturas por parte dos vendedores no exterior das suas instalações, onde normalmente se encontra a principal ferramenta de faturação e gestão da PrimaveraBSS, o ERP V10. Com isto, pretende-se resolver esta questão através de uma aplicação móvel integrada com o ERP, que permita uma desmaterialização dos processos já referidos.

#### 2.1.2. Arquitetura geral da Solução

A aplicação V10 é constituída por um sistema integrado com foco na mobilidade empresarial. Esta aplicação será suportada pela WEB API e ERP V10 da PrimaveraBSS de forma a responder a todos os requisitos propostos.

O diagrama seguinte apresenta a estrutura da solução de forma esquematizada:



## 2.1.3. Processos Funcionais

### 2.1.3.1. Aplicação móvel

Espera-se desta aplicação um suporte total para os colaboradores da PrimaveraBSS, numa perspetiva de vendedor.

É esperado um funcionamento em Android e iOS.

A solução final será adaptável para qualquer tipo de dispositivo móvel, como smartphone ou tablet (*Responsive*).

### 2.1.3.2. Modo de operação

A aplicação terá dois modos de funcionamento, autenticado *online* e autenticado *offline*. Este funcionamento é esperado de forma a de ir encontra ao facto de manter a mobilidade da solução, evitando a redundância da existência da mesma, fazendo com que não ocorram lapsos de uso devido à ausência de rede ou internet.

Numa primeira utilização da aplicação, o utilizador terá de se autenticar em modo *online* de forma a validar os seus dados. Estes serão guardados internamente no dispositivo, de forma a que, numa situação em que o utilizador se encontre sem rede ou possibilidade de aceder ao modo *online*, consiga efetuar uma validação *offline*.

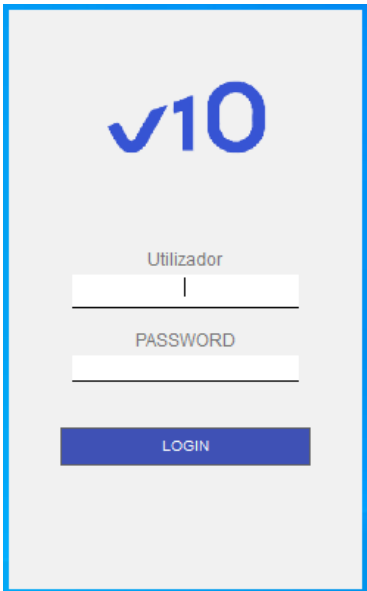
De forma a conseguir esta funcionamento será utilizado uma base de dados interna do dispositivo para guardar os dados de autenticação de um utilizador de forma a poder autenticar o mesmo.

O funcionamento da aplicação será baseado em 4 páginas essenciais:

- Vendas – página de listagem e criação de faturas;
- Catálogo – página de listagem de produtos e serviços;
- Clientes – página de listagem e criação de cliente.

### 2.1.3.3. Log in

Nesta página da aplicação o utilizador terá os campos de utilizador e *password* que terá de preencher para aceder a aplicação em modo *online*.



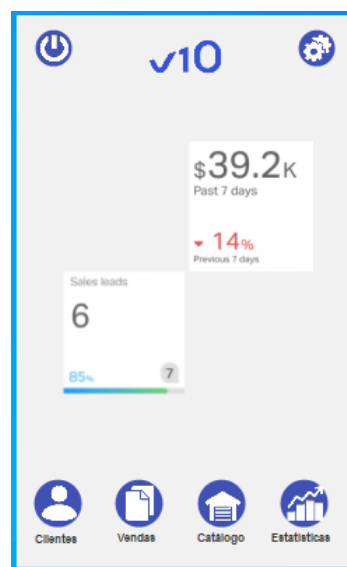
The image shows a login interface for the 'v10' application. At the top, the logo 'v10' is displayed in blue. Below the logo, there are two input fields: the first is labeled 'Utilizador' and contains a vertical cursor; the second is labeled 'PASSWORD'. At the bottom of the form is a blue button with the text 'LOGIN' in white capital letters.

#### 2.1.3.4. Menu

Na página Menu, o utilizador tem presente uma *dashboard* resumo que apresente alguns dados básicos.

Esta página tem presente os botões de navegação para as seguintes páginas:

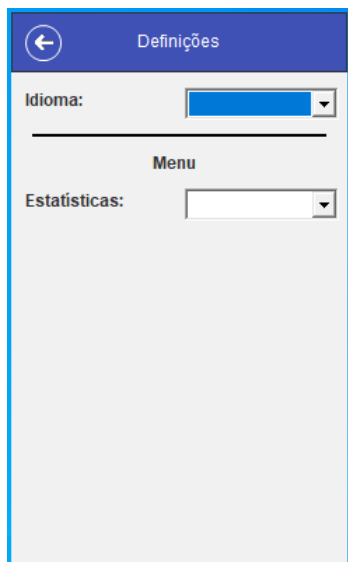
- Clientes;
- Vendas;
- Catálogo;
- Estatísticas.



#### 2.1.3.5. Definições

A página de definições permite ao utilizador alterar os seguintes parâmetros da aplicação:

- Idioma – O idioma será por *default* o idioma que o cliente tiver definido na base de dados.
- Estatísticas – Os valores que serão apresentados no Menu principal.

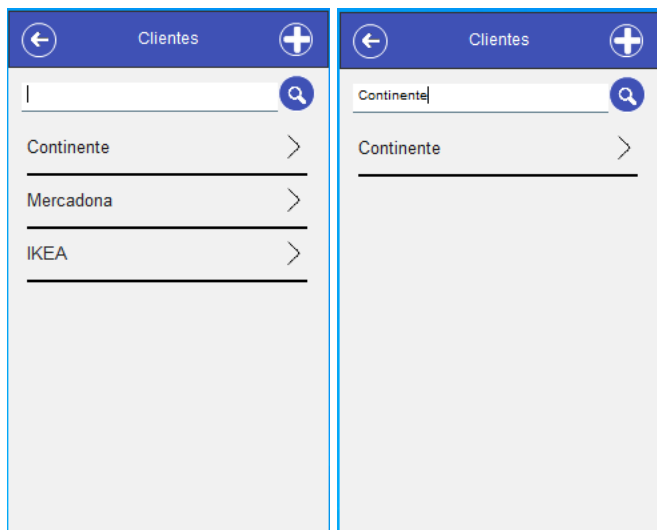


#### 2.1.3.6. Clientes

Na página de clientes, o utilizador poderá aceder a listagem dos clientes que lhe serão visíveis.

##### Requisitos:

- Serão apresentados todos os clientes disponíveis da web api;
- Os clientes são apresentados por ordem alfabética com opção de filtrar através de procura personalizada;
- Assim que se selecionar um cliente é esperado que seja aberta a página de detalhes do mesmo;
- Forma de criar novo clientes (botão) através de uma página de criação de cliente.



### 2.1.3.7. Detalhe de Cliente

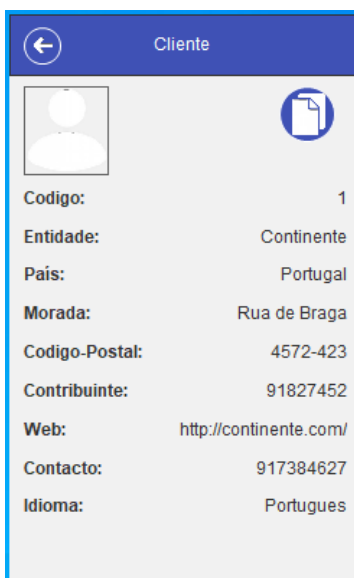
Acedida através da página de Clientes quando selecionado um cliente, esta página permite ao utilizador à informação mais detalhada do cliente que este pretender, apresentando os seguintes dados:

- Código de Cliente
- Nome do Cliente
- Morada
- País
- Código Postal
- N°. Contribuinte
- Email;
- Contacto;
- Idioma.

Através desta página o utilizador pode aceder diretamente à página de criação de uma nova venda, transportando todos os dados necessários do cliente que estiver representado na mesma, auto completando todos campos necessários e relativos ao cliente, presentes na fatura.



Pode também aceder diretamente ao *googlemaps* através da morada e chamadas através do contacto e e-mail.



Codigo:	1
Entidade:	Continente
País:	Portugal
Morada:	Rua de Braga
Codigo-Postal:	4572-423
Contribuinte:	91827452
Web:	<a href="http://continente.com/">http://continente.com/</a>
Contacto:	917384627
Idioma:	Portugues

#### 2.1.3.8. Criar Cliente

Nesta página o utilizador consegue criar um novo cliente, tendo que preencher todos os dados presentes no formulário.

Os dados são:

- Nome do Cliente
- Morada
- País
- Código Postal
- N.º. Contribuinte
- Email;
- Contacto;
- Idioma.

← Criar Clientes

Entidade:

País:

Morada:


Código-Postal:


Contribuinte:

Email:

Contacto:

Idioma:

 Carregue uma fotografia.

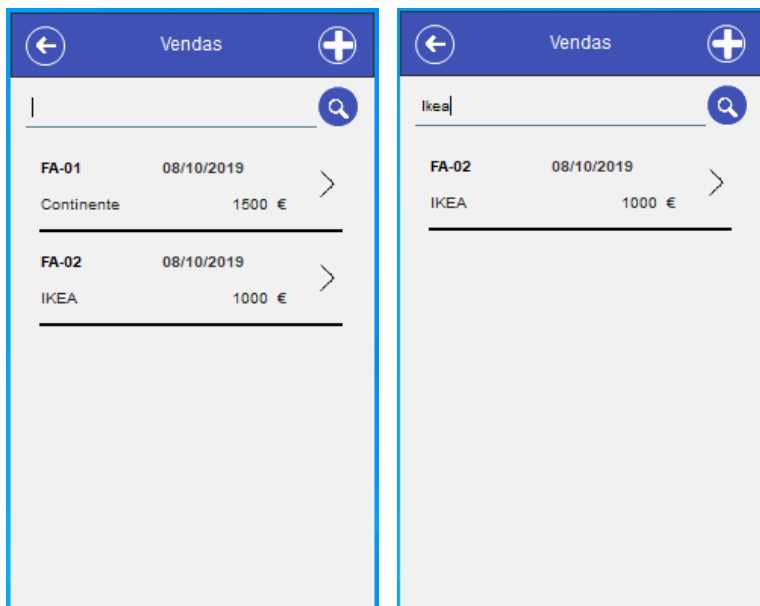


### 2.1.3.9. Vendas

Na página de Vendas, o utilizador poderá aceder a listagem das faturas relativas as suas vendas efetuadas, limitadas a um intervalo temporal definido pelo sistema.

#### Requisitos:

- As faturas são apresentadas por ordem cronológica com opção de filtrar através de procura personalizada;
- Assim que se abrir uma fatura é esperado que seja aberta a página de detalhes da mesmo;
- Forma de criar nova fatura (botão) através de uma página de criação de fatura.
- Na criação de uma nova fatura, poderá ser registado um produto através de recurso a leitor de código de barras;
- Guardar documentos criados na aplicação.



### 2.1.3.10. Detalhe da Fatura

Nesta página, navegando através da seleção de uma fatura na página de Vendas, o utilizador tem acesso aos detalhes de uma fatura. Esta página permite também o download e/ou partilha da fatura através de e-mail.

Esta página apresenta os seguintes dados:

- Documento;
- Entidade;
- País;
- Morada;
- Código Postal;
- N.º. Contribuinte;
- Observações;

Esta página apresenta a seleção de artigos na forma de uma tabela que apresenta todos os artigos que constituem a fatura. Esta tabela apresenta as seguintes colunas:

- Produto;
- Quantidade;
- Valor;

←
Fatura

**Documento:** FA-01

**Entidade:** Continente

**País:** Portugal

**Morada:** Rua de Braga

**Contribuinte:** 91529452

**Observações:**  
Pequena Compra

**Produtos:**

Produto	Desc	Qty.	Valor(€)
A001	Computador	3	1000
A002	Ecrã HD	6	500

Data do documento: 08/10/2019

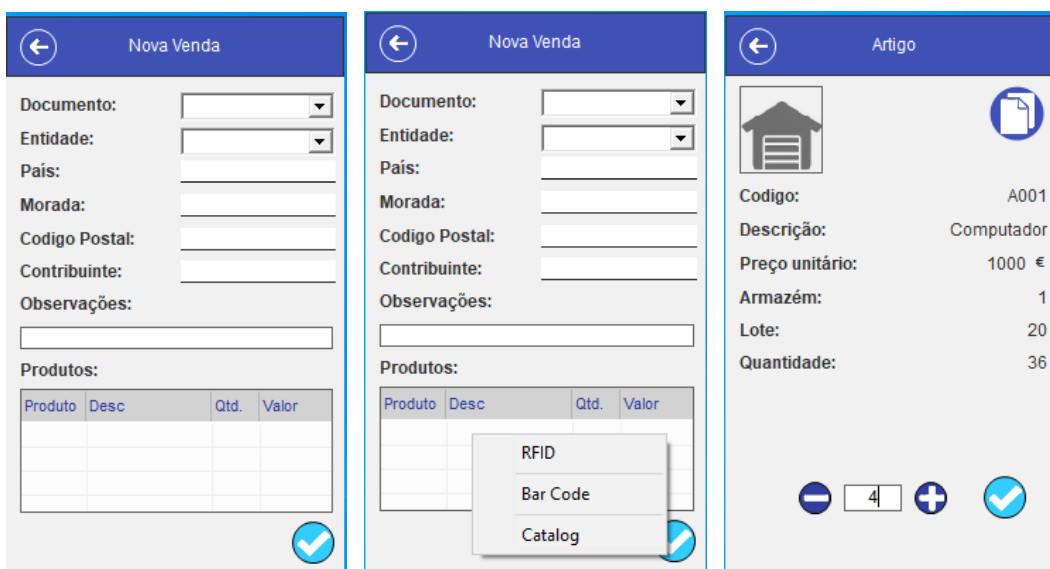
Data de vencimento: 20/10/2019

### 2.1.3.11. Criar Fatura

Em criar fatura, o utilizador consegue criar uma fatura relativa a uma venda preenchendo todos os campos que a constituem:

- Documento;
- Entidade;
- País;
- Morada;
- Código Postal;
- N.º. Contribuinte;
- Observações;

Para seleccionar os produtos, esta página tem presente uma tabela que, após seleccionar a mesma, a aplicação navega para a página do Catálogo podendo este seleccionar todos os produtos pretendidos.

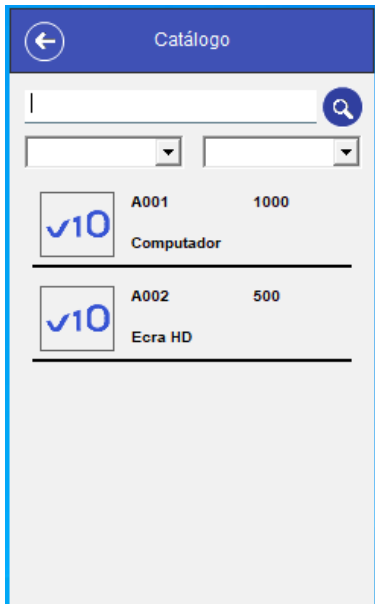


### 2.1.3.12. Catálogo

Na página catálogo, o utilizador poderá aceder a listagem dos produtos que lhe serão visíveis.

#### Requisitos:

- Serão apresentados todos os produtos disponíveis da web api;
- A lista será apresentada inicialmente com todos os produtos disponíveis;
- Os produtos são apresentados por ordem crescente de código com opção de filtrar através de procura personalizada;
- Assim que se abrir um produto é esperado que seja aberta a página de detalhes do mesmo;
- Filtrar artigos por família e subfamília;



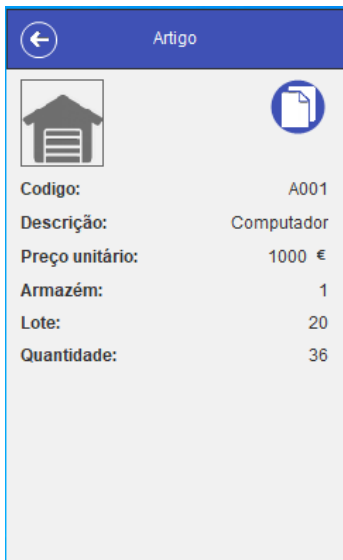
### 2.1.3.13. Detalhe de Artigo

Na página de detalhe de artigo, o utilizador pode aceder aos detalhes de qualquer artigo presente na página Catálogo.

Os dados apresentados são os seguintes:

- Código;
- Descrição;
- Preço unitário;
- Cliente;
- Armazém;
- Localização;
- Lote.

Através desta página o utilizador tem a opção de selecionar o artigo diretamente para a página de criação de Fatura, auto completando assim os seus dados presentes



Artigo	
Código:	A001
Descrição:	Computador
Preço unitário:	1000 €
Armazém:	1
Lote:	20
Quantidade:	36

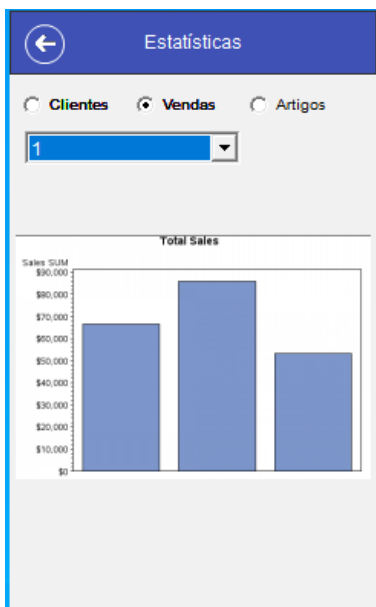
#### 2.1.3.14. Estatísticas

Esta página tem como objetivo apresentar uma *dashboard* com dados relevantes relativo a clientes ou produtos, e mais elaborados do que os presentes na página do Menu.

#### Requisitos:

- Escolha de visualização de estatísticas referentes a clientes, produtos ou vendas;
- Interface simples e de fácil uso e compreensão.

(Em imagem é apresentado apenas um modelo representativo de um gráfico genérico)



## 2.1.4. Backoffice

### 2.1.4.1. Gestão de utilizadores

O **log in** é caracterizado pela seguinte informação:

- Utilizador;
- Password;
- Companhia;
- Instância;
- *Grant type*.

Permissões:

- Necessário estar registado no ERP V10 previamente.
- As informações de Log in são guardadas no dispositivo de forma ao utilizados aceder de forma offline.



#### 2.1.4.2. Gestão de clientes

Para cada cliente são guardadas as seguintes informações:

- Código de entidade;
- Nome;
- Morada;
- Código Postal;
- Contacto telefónico;
- Número de contribuinte;
- País;
- Url;
- Localidade;
- Código Postal Local;
- Número Fax;
- Moeda;
- Descrição;
- Distrito;
- Vendedor.

Permissões:

- Necessário ter permissão de acesso aos clientes no ERP;
- O utilizador consegue adicionar novos clientes;

Requisitos:

- Um utilizador registado consegue criar um novo cliente inserindo as informações base obrigatórias requeridas pelo ERP V10.

### 2.1.4.3. Gestão de Vendas

Para os documentos de venda são guardadas as seguintes informações:

- Documento;
- Entidade;
- Morada;
- Número contabilístico;
- País;
- Observações;
- Data de criação;
- Data de vencimento;
- Valor total;
- Número de documento;
- Série;
- Filial;
- Número de documento Kalipso.

Permissões:

- O utilizador consegue aceder apenas aos documentos de venda relativos aos clientes a que tem acesso;
- O utilizador registado consegue criar um documento simplificado.

Requisitos:

- Após a criação de um documento, este fica em stand by no backoffice à espera de ser aprovado.

#### 2.1.4.4. Gestão de artigos de documento

Os artigos provenientes do ERP associados a documentos existentes são geridos de forma separada dos documentos em si, e são passados com as seguintes informações:

- Código de produto;
- Descrição;
- Código de fatura;
- Código de documento;
- Código de Série;
- Código de entidade;
- Quantidade;
- Tipo de entidade;
- Valor unitário;
- Desconto;
- Armazém;
- Localização;
- Código Kalipso;

#### 2.1.4.5. Gestão de criação de um documento

Cada vez que é aberto a página de criação de um documento na aplicação, é gerado um GUID que prepara a criação de uma instância da base de dados. De seguida é enviado para o ERP o pedido de criação de cabeçalho que é guardado na base de dados através de uma string em formato JSON.

Após a escolha de um artigo é feita a adicionar linha por cada artigo que estiver associado ao GUID do documento gerado. Após este processo é finalizada a criação e enviado para o ERP.

Requisitos:

- Qualquer utilizador registado consiga criar qualquer tipo de documento sobre os seus clientes;

#### **2.1.4.6. Gestão de artigos**

Os artigos são geridos segundo as seguintes informações:

- Código de artigo;
- Descrição;
- Código de barras;
- Armazém;
- Local;
- Quantidade;
- Quantidade económica;
- IVA;
- Foto;
- Família;
- Subfamília;
- Prazo de entrega;
- Última atualização;

#### **2.1.4.7. Gestão de PVP.**

Os PVP's são 6 tipos diferentes de preços unitários que são atribuídos a diferentes clientes, consoante o decidido no ERP.

As informações geridas são:

- Código de artigo;
- PVP1;

- PVP2;
- PVP3;
- PVP4;
- PVP5;
- PVP6.

#### **2.1.4.8. Gestão de Famílias de Artigos**

Cada artigo é caracterizado por uma Família e Subfamília, sendo estas informações utilizadas para a filtragem no catálogo.

- Código de Famílias;
- Código de Subfamília;
- Descrição Família.

#### **2.1.4.9. Gestão de tipo de documento**

Para cada documento é atribuído um tipo, de forma a diferenciar uma venda de uma fatura, de uma devolução, etc.

Os tipos de documentos são organizados por:

- Código do Tipo.

#### **2.1.4.10. Gestão de Series.**

Cada tipo de documento tem atribuído uma série.

As diferentes séries são geridas por:

- Tipo.

#### 2.1.4.11. Gestão de Países

No ERP estão presentes diferentes países que são abrangidos, sendo utilizados para a criação de uma fatura na aplicação.

Os países são organizados por:

- Código;
- Descrição.

## 2.2. OBJETIVOS E AÇÕES PROPOSTAS

Os objetivos da PrimaveraBSS ao optar pela implementação da Solução, são os seguintes:

- Desmaterializar os processos de negócio numa perspetiva de vendedor fora das instalações da empresa;
- Aumentar o nível de integração entre os diferentes componentes do Sistema de Informação de Gestão;
- Tirar partido da mobilidade empresarial presente no panorama atual do mercado;
- Integração da aplicação móvel com o ERP Primavera.



# Conclusões

## 2.3. Recomendações Finais

A Estratégia de Sistemas de Informação da PrimaveraBSS de curto e médio prazo, deverá passar pela implementação das ações anteriormente propostas e pela otimização do funcionamento dos vários componentes da Aplicação móvel V10, assim como dos desenvolvimentos específicos descritos neste documento.

Estas permitirão organizar a informação de uma forma simples, flexível e eficiente. A integração dos diferentes módulos permitirá evitar a duplicação de tarefas e poupar desta forma bastante tempo, para além de evitar erros.

Com base no atual Documento de Visão será definido na fase seguinte um plano detalhado das tarefas a realizar calendarizado no tempo e que deverá assegurar o cumprimento dos prazos de implementação propostos.

## 3. Validação do Conteúdo deste Documento

Este documento está inserido na M.I.P. e a sua apresentação é exigida pela PRIMAVERA Business Software Solutions, S.A. aos seus Parceiros, sempre que seja efetuado um projeto de implementação das soluções PRIMAVERA. Este documento pode ser alterado mediante apresentação de sugestões específicas para o efeito, sendo considerado aprovado na ausência de sugestões no período de 5 dias úteis após disponibilização.

### 3.1. Aprovação

De acordo com a M.I.P., declaro que este documento sintetiza a realidade atual e as premissas de evolução ao nível do Sistemas de Informação de Gestão que são, neste momento, essenciais para a nossa Organização.

<b>Cliente</b>	Primavera Business Software Solutions
<b>Responsável pela Validação do Conteúdo deste Documento</b>	
<b>Data</b>	26-02-2020

WWW.PRIMAVERABSS.COM



## Manual de Instalação

•	<b>Serviço</b>
	Manual de Instalação
•	<b>Cliente</b>
	Primavera Business Software Solutions
•	

•	<b>Versão do Documento <i>Template</i></b>
	Versão 1.0
•	<b>Versão do Documento</b>
	Versão 0.0



**Histórico de Versões**

<b>Consultor</b>	<b>Data</b>	<b>Alterações</b>	<b>Versão Doc.</b>
Rui Silva	26-02-2020	Criação de documento	1.0

# ÍNDICE

<b>1. ENQUADRAMENTO</b> .....	<b>4</b>
1.1. SUMÁRIO EXECUTIVO.....	4
<b>2. OBJETIVO</b> .....	<b>5</b>
<b>3. INSTALAÇÃO</b> .....	<b>5</b>
3.1. INSTALAÇÃO DE NOVOS MÓDULOS/ APLICAÇÕES PRIMAVERA.....	<b>ERRO! MARCADOR NÃO DEFINIDO.</b>
3.2. CRIAÇÃO DE CAMPOS E TABELAS DO UTILIZADOR .....	<b>ERRO! MARCADOR NÃO DEFINIDO.</b>
3.2.1. Criar campo <nome_campo> na tabela <nome_tabela > .....	<b>Erro! Marcador não definido.</b>
3.2.2. Criação da tabela de <nome_tabela>.....	<b>Erro! Marcador não definido.</b>
3.3. REGISTO DE COMPONENTES ACTIVEX .....	<b>ERRO! MARCADOR NÃO DEFINIDO.</b>
3.4. PROJETOS VBA .....	<b>ERRO! MARCADOR NÃO DEFINIDO.</b>
3.5. CONFIGURAR AS FUNÇÕES E MENUS DE UTILIZADOR .....	<b>ERRO! MARCADOR NÃO DEFINIDO.</b>
3.6. CRIAÇÃO DE MAPAS DE UTILIZADOR.....	<b>ERRO! MARCADOR NÃO DEFINIDO.</b>
3.7. BINÁRIOS E DLLS.....	<b>ERRO! MARCADOR NÃO DEFINIDO.</b>



# 1. Enquadramento

## 1.1. Sumário Executivo

Este projeto surgiu com o intuito de explorar a potencialidade de mobilidade do ERP V10, tentando dar resposta à existência de processo redundantes ainda presentes no processo de criação de documentos de venda.

Este trabalho visa a exploração de uma aplicação que proporcione mobilidade ao ERP V10 com recurso à WebApi. A exploração destes recursos tem também como intenção a sua potencialização de forma a tornar os processos internos da organização mais eficientes.

A realização deste projeto pode também ser utilizado para uma análise aprofundada das capacidades das funcionalidades da presente WebApi do ERP, ajudando a perceber erros ou funcionalidades em falta que possa auxiliar o desenvolvimento das próprias infraestruturas do *software*.



## 2. Objetivo

A aplicação V10 tem como objetivo disponibilizar ferramentas que permitem ao utilizador efetuar as seguintes operações:

- Excluir a necessidade de meios intermédios de criação de um documento de venda;
- Criar mobilidade ao ERP V10;
- Criar documentos de venda de forma externa às instalações do ERP;
- Aceder ao catálogo de forma externa às instalações do ERP;
- Aceder à lista de clientes de forma externa às instalações do ERP;

## 3. Instalação

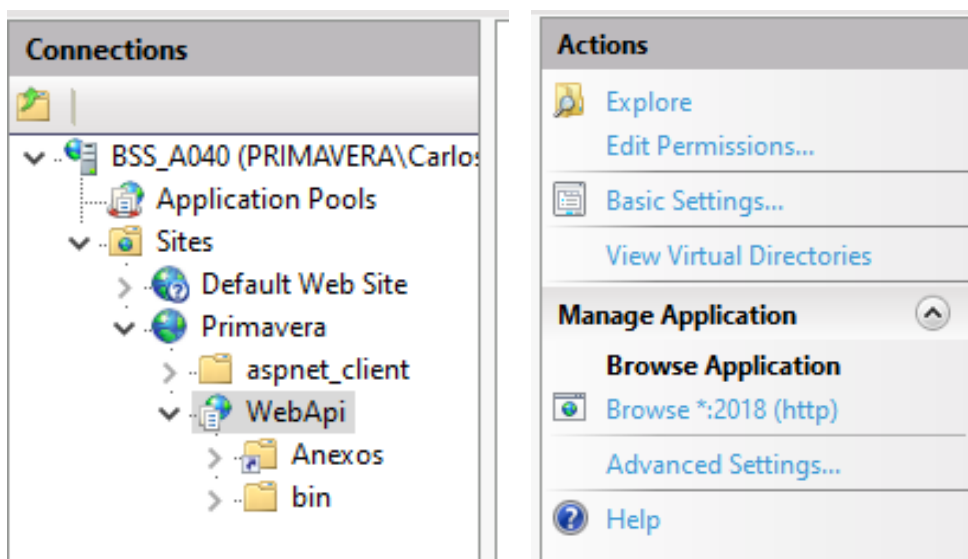
### 3.1. Configuração do IIS e Port

Em primeiro lugar é necessário verificar se o IIS está instalado na máquina. Caso não esteja instalado é possível efetuar o download no website da Microsoft.

Após a instalação do módulo WepApi no ERP V10, o site do mesmo deverá já aparecer no IIS.

Caso esteja a ser utilizado a máquina pessoal como servidor poderá ser necessário criar uma exceção da Port requisitada pelo IIS na *firewall*. Após criar esta exceção é necessário que o dispositivo móvel que contem a aplicação esteja conectado à mesma rede interna da Primavera (PRIWIFI)

A Port necessária para o *Login* da aplicação poderá ser verificado no IIS. De forma a que o dispositivo consiga aceder à máquina é necessário verificar o IPV4 através da linha de comando recorrendo ao comando *ipconfig*. O conjunto de Url com a port deverá ser idêntico a: 172.16.3.105:2018.



### 3.2. Instalação da WebApi

De forma inicial é necessário que o utilizador tenha uma conta válida no sistema ERP V10.

Caso o módulo WebApi V10 não tenha sido instalado durante a primeira instalação do ERP V10, é necessário aceder às configurações de aplicações no *PRIMAVERA Windows Services Management*, seleccionar este módulo e instalar o mesmo.

PRIMAVERA Windows Services v2.0

- PRIMAVERA Deployment Center v2.00
  - Configuração de Aplicações
    - Instalações Disponíveis
    - Instalação nos Postos
    - Histórico de Instalações
    - Instalações Ignoradas
  - PRIMAVERA Maintenance Plans Service

**PRIMAVERA**

PRIMAVERA Deployment Center v2.00 | Configuração de Aplicações

Descrição Geral

Indique quais as aplicações e os respectivos módulos que deseja incluir no processo de pesquisa do serviço

PRIMAVERA PROFESSIONAL - ERP v10.00

[Voltar à Lista de Aplicações](#) [Actualizar Módulos da Aplicação](#) [Guardar Configuração](#)

Configure os módulos a incluir nas pesquisas desta aplicação e os seus parâmetros de instalação

Módulo

- PRIMAVERA PROFESSIONAL - Internos v10.00
- PRIMAVERA PROFESSIONAL - Inventário v10.00
- PRIMAVERA PROFESSIONAL - Mobile v10.00
- PRIMAVERA PROFESSIONAL - Plataforma e Administrador v10.00
- PRIMAVERA PROFESSIONAL - Produção v10.00
- PRIMAVERA PROFESSIONAL - Projetos e Serviços v10.00
- PRIMAVERA PROFESSIONAL - Recursos Humanos v10.00
- PRIMAVERA PROFESSIONAL - Serviços Técnicos v10.00
- PRIMAVERA PROFESSIONAL - Tesouraria v10.00
- PRIMAVERA PROFESSIONAL - Transações Eletrónicas v10.00
- PRIMAVERA PROFESSIONAL - Vendas v10.00
- PRIMAVERA PROFESSIONAL - WebApi v10.00
- PRIMAVERA PROFESSIONAL - Windows Scheduler v10.00

Parâmetros de Instalação

### 3.3. Base de dados

A aplicação tira partido da função de listas do ERP sendo que estas listas apesar de permitir uma maior flexibilidade no acesso à base de dados, faz com que uma lista criada por um utilizador apenas ira afetar a sua aplicação.

Para este efeito foi criada uma *script* que cria as listas necessárias para o bom funcionamento dos processos da aplicação.

Esta *script* encontra-se anexada a este documento.

### 3.4. Acesso aos anexos

Para aceder às imagens dos artigos é necessário criar uma diretoria virtual no Site da WebApi no IIS da seguinte forma:

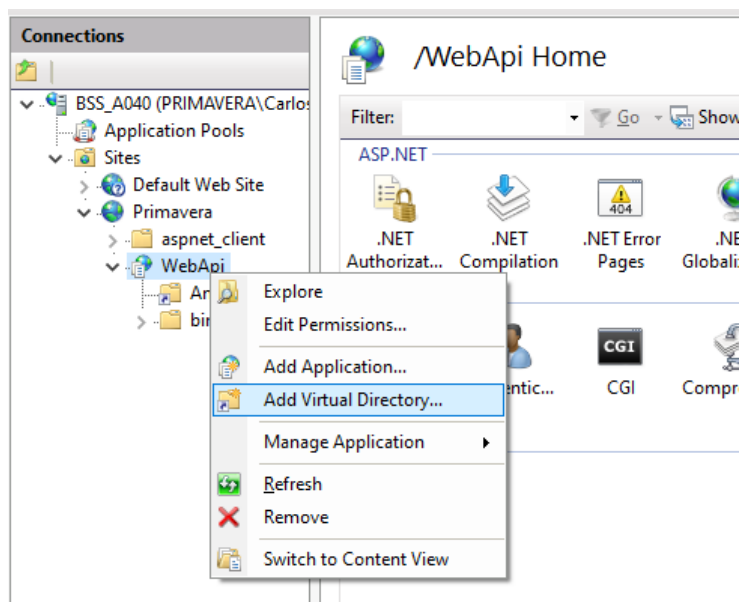


Figura 1 - Selecionar a opção de adicionar diretoria virtual



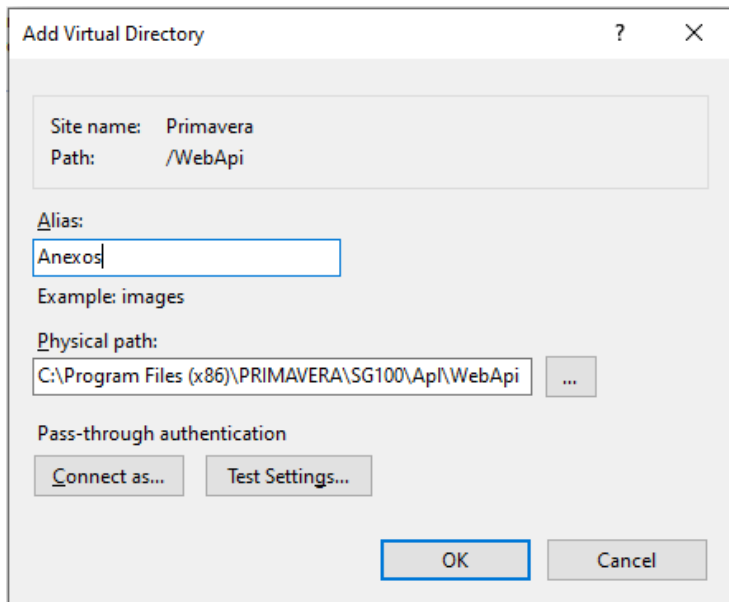


Figura 2 - Colocar o nome da diretoria e o path da WebApi

## Manual de Utilização do Avaliador

	<b>Serviço</b>		<b>Versão do Documento <i>Template</i></b>
	Avaliação de Desempenho		Versão 1.0
	<b>Cliente</b>		<b>Versão do Documento</b>
	Primavera Business Software Solutions		Versão 1.1
			

### Histórico de Versões

Consultor	Data	Alterações	Versão Doc.
Rui Silva	26-02-2020	Criação de Documento	1.0
Rui Silva	12-03-2020	Atualização de imagens	1.1

# ÍNDICE

<b>1.</b>	<b>LOGIN</b> .....	<b>4</b>
<b>2.</b>	<b>MENU INICIAL</b> .....	<b>5</b>
<b>3.</b>	<b>CLIENTES</b> .....	<b>6</b>
3.1.	LISTA DE CLIENTES.....	6
3.2.	DETALHE CLIENTE .....	7
3.3.	CRIAR CLIENTE.....	8
<b>4.</b>	<b>CATÁLOGO</b> .....	<b>9</b>
4.1.	LISTA DE ARTIGOS .....	9
4.2.	DETALHE DE ARTIGO .....	10
<b>5.</b>	<b>DOCUMENTOS</b> .....	<b>11</b>
5.1.	LISTA DE DOCUMENTOS .....	11
5.2.	DETALHE DE DOCUMENTOS .....	12
5.3.	CRIAR DOCUMENTO .....	13
5.4.	DOCUMENTOS GUARDADOS .....	15
<b>6.</b>	<b>PREFERÊNCIAS</b> .....	<b>16</b>
6.1.	MENU .....	16
6.2.	LOGIN .....	17

## 1.Login

Para aceder à aplicação é necessário fazer o login com as credenciais do ERP V10(nome de utilizador, password e companhia) e o *Url* e *Port* correspondente ao servidor.



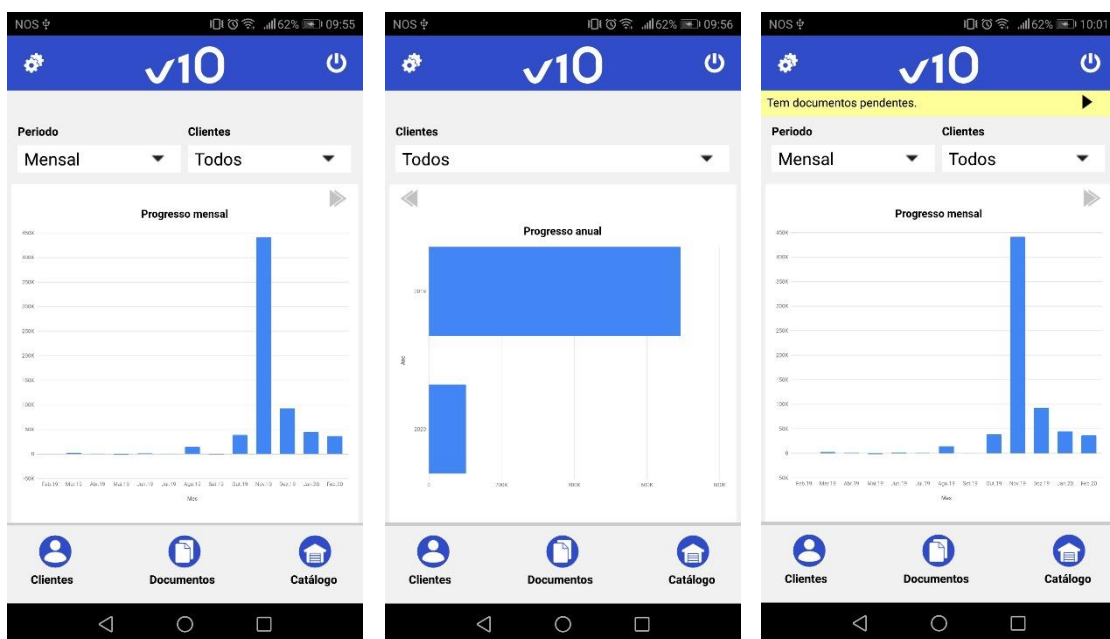
The screenshot shows a mobile application interface for logging into the V10 ERP system. The background is blue with a white 'V10' logo. Below the logo, there are several input fields with labels: 'Utilizador' (containing 'Carlos'), 'Password' (containing masked characters), 'Url' (containing '172.16.3.44'), 'Port' (containing '2018'), and 'Companhia' (containing 'DEMOV10'). At the bottom of the form is a blue button labeled 'LOGIN'. The top of the screen shows a status bar with 'NOS', signal strength, Wi-Fi, 83% battery, and the time '09:39'. The bottom of the screen shows the Android navigation bar.

Estas informações serão pedidas apenas no primeiro *Login* a não ser que o utilizador faça *logoff*, caso contrário o acesso será feito automaticamente cada vez que seja efetuado o acesso à aplicação.

## 2. Menu Inicial

O menu inicial disponibiliza o centro principal de navegação da aplicação, estando dividido em duas páginas que disponibilizam ao utilizador informações estatísticas sobre o panorama de vendas.

As estatísticas são disponibilizadas em gráficos de barras que podem ser manipulados com recurso a combobox que permite ao utilizador seleccionar a periodicidade e companhia que pertence que seja demonstrada.

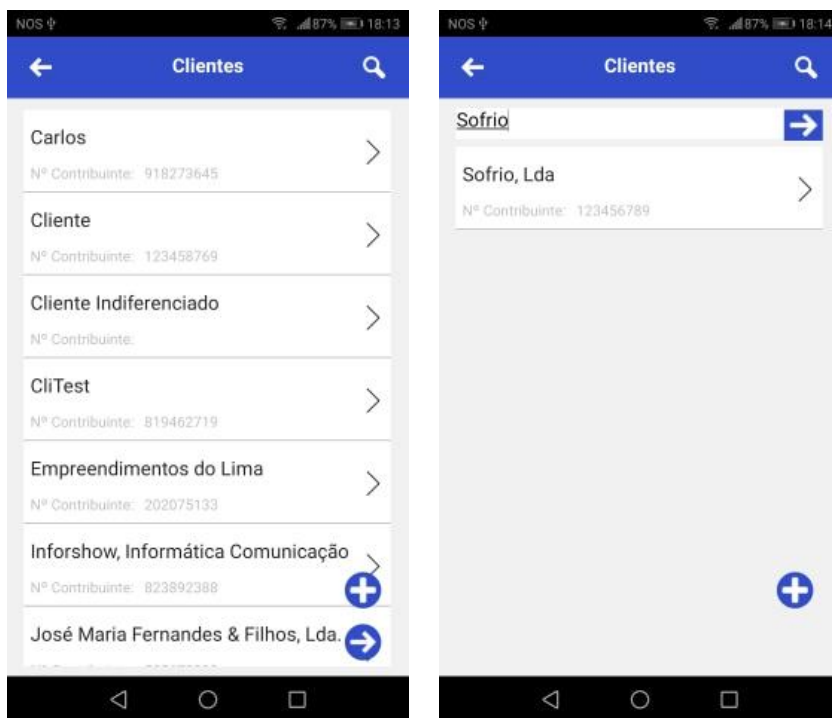


No cenário de existirem documentos guardados na aplicação será demonstrada uma barra de aviso amarela de forma a chamar o utilizador a atenção para que estes não sejam esquecidos.

## 3. Clientes

### 3.1. Lista de clientes

Através do ícone de Clientes presente na página de Menu é possível aceder á listagem de clientes da companhia associada ao *LogIn*. Nesta Listagem é possível filtrar os clientes por nome através do ícone “*search*” presente no canto superior direito.



### 3.2. Detalhe cliente

Selecionando um cliente a partir da lista de clientes é possível verificar as informações detalhadas do cliente selecionado. Através desta página é possível navegar para a página de criação de documento selecionando o ícone no canto superior direito.

Selecionando a Morada é possível aceder á mesma no googlemaps de forma direta. Através do contacto é possível efetuar uma chamada direta e/ou selecionando o email é possível aceder diretamente ao email.





### 3.3. Criar cliente

Através do ícone “+” presente na lista de clientes é possível aceder á página de criação de cliente. Após o preenchimento dos dados estes são enviados diretamente para o ERP V10.

The image displays two sequential screenshots of a mobile application interface for creating a new client. Both screens have a blue header with a back arrow and the text "Novo Cliente".

**Screen 1 (1/2):** This screen contains the following input fields:

- Cliente: \***: A text input field.
- Nome: \***: A text input field.
- Contribuinte: \***: A text input field.
- Morada:**: A text input field.
- Localidade:**: A text input field.
- Codigo-Postal:**: A text input field.

A blue button labeled "Seguinte" is positioned at the bottom of the form area. The page indicator "1/2" is located in the bottom right corner.

**Screen 2 (2/2):** This screen contains the following input fields:

- Pais: \***: A dropdown menu.
- Distrito:**: A text input field.
- Telefone: \***: A text input field.
- Fax:**: A text input field.
- Endereço Web:**: A text input field.
- Moeda: \***: A dropdown menu.

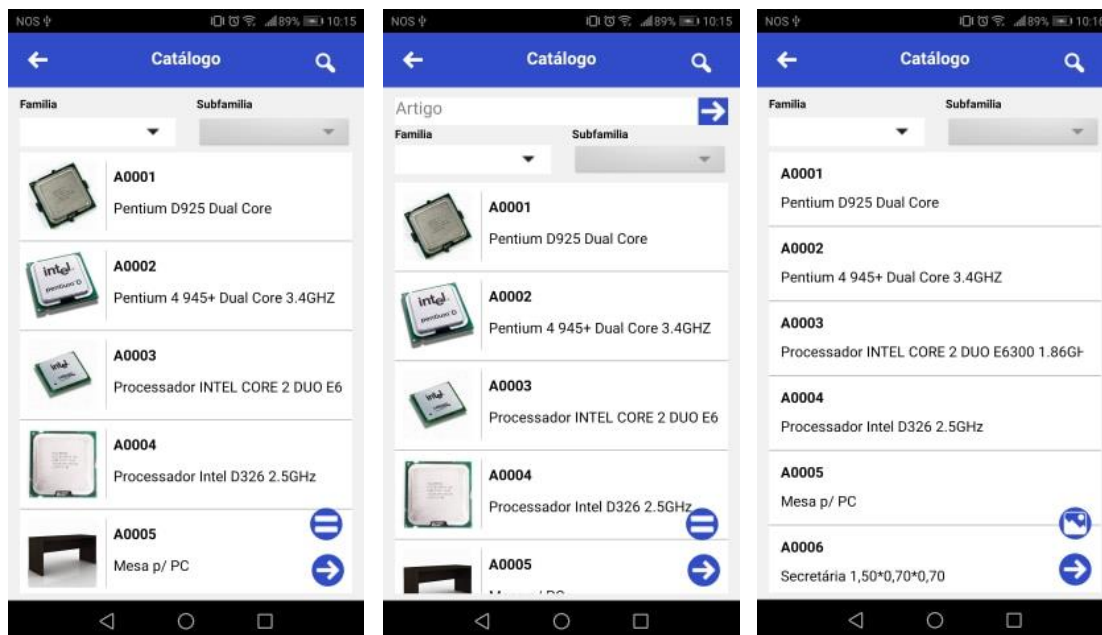
Two blue buttons, "Voltar" and "Guardar", are located at the bottom of the form area. The page indicator "2/2" is located in the bottom right corner.

## 4. Catálogo

### 4.1. Lista de artigos

Através do ícone de Catálogo presente na página de Menu é possível aceder á listagem de artigos da companhia associada ao *LogIn*. Nesta Listagem é possível filtrar os artigos por código de artigo, descrição, família e subfamília através do ícone “search” presente no canto superior direito.

A lista de artigos é apresentada em várias páginas, variando de forma dinâmica dependendo da quantidade de artigos presentes na base de dados. É possível também alternar entre uma lista simples e lista com imagens.



## 4.2. Detalhe de artigo

NOS 62% 09:59

← Artigo: A0001



**Descrição:**  
Pentium D925 Dual Core

**Familia:** H01      **SubFamilia:** H01.001

**Armazém:** A1      **Local:** A1.A.1.001

**Preço unitário:**  
1000€

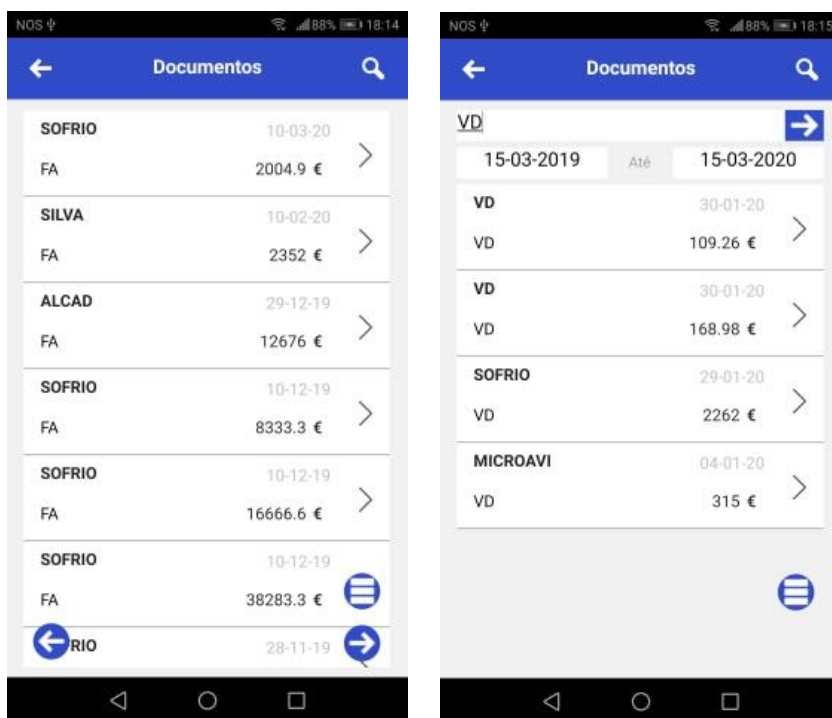
**IVA:** 20      **Prazo de entrega:** 15

**Quantidade:** -203      **Quantidade economica:** 10

## 5. Documentos

### 5.1. Lista de documentos

Através do ícone de Documentos presente na página de Menu é possível aceder á listagem de documentos da companhia associada ao *LogIn*. Nesta Listagem é possível filtrar os documentos de forma cronológica, tipo de documento e entidade através do ícone “*search*” presente no canto superior direito.



## 5.2. Detalhe de documentos

Ao seleccionar um documento apresenta na lista de documentos é possível aceder aos seus detalhes. É possível também partilhar e efetuar o download do documento seleccionado.

The image displays four screenshots of a mobile application interface, arranged in a 2x2 grid, showing document details and line items.

**Top Left Screenshot: Documento**

Header: Documento

Fields:

- Nº: 18
- Documento: FA, Serie: A
- Entidade: SOFRIO
- Contribuinte: 123456789
- Morada: AV. DO ETERNO GELO, 88888
- País: PT
- Criação: 30-01-20, Vencimento: 16-03-20

Buttons: Linhas (with arrow), Download (with download icon)

**Top Right Screenshot: Linhas do documento**

Header: Linhas do documento

<b>A0001</b>	2634 €	⋮
Qtd: 3Und	Dsc: 0%	
<b>M0001</b>	319.2 €	⋮
Qtd: 8Und	Dsc: 76.61%	

Total: 2953.2 €

**Bottom Left Screenshot: Linhas do documento**

Header: Linhas do documento

Fields:

- Código: A0001
- Descrição: Pentium D925 Dual Core
- Armazém: A1, Local: A1
- Preço unitário: 878€, IVA: 21%
- Quantidade: 3Und, Desconto: 0%
- Saida: 30-01-20, Entrega: 30-12-10

**Bottom Right Screenshot: Documentos**

Header: Documentos

Search: [Magnifying Glass]

SILVA	10-02-20	>
FA	2352 €	>
SOFRIO	05-02-20	>
FA	1760 €	>
[Modal: Novo documento / Documentos guardados]		
SOFRIO	30-01-20	>
FA	2953.2 €	>
VD	30-01-20	>
VD	168.98 €	>
SOFRIO	30-01-20	[Menu]

### 5.3. Criar documento

Através da página de listagem de documentos é possível aceder à página de criação de documentos. Durante a criação do documento o utilizador pode escolher entre guardar ou enviar o documento. Para guardar o utilizador tem de seleccionar o ícone presente na página de cabeçalho do documento no canto superior direito.

A criação de documento é distribuída em 3 fases:

- Cabeçalho;
- Linhas;
- Seleção de artigo.

Nas páginas de cabeçalho e linhas o utilizador pode escolher inserir as informações de forma manual ou recorrendo as informações presentes na base de dados.

Se o utilizador escolher seleccionar os artigos recorrendo às informações presentes no dispositivo pode escolher através do catálogo ou através de uma leitura de código de barras recorrendo ao ícone presente na página de linhas no canto superior direito.

Acedendo ao artigo através de página de criação de documento, o detalhe de artigo é apresentado apenas com as informações relevantes e escolha de quantidade e desconto que o utilizador escolha. Nesta página são apresentados os 3 descontos mais utilizados para o produto seleccionado.

Caso o utilizador esteja a aceder à aplicação de forma *offline* apenas terá a opção de guardar o documento.

NOS 62% 09:59

**Novo Documento**

**Documento:\***  
FA

**Serie:\***  
T

**Entidade:\***  
SOFRIO

**Contribuinte:**  
123456789

**Morada:**  
AV. DO ETERNO GELO, 88888

**Pais:**  
PT

**Linhas** → ✓

NOS 62% 10:00

**Linhas do documento**

<b>A0001</b>	<b>16666.6 €</b>	
Qtd: 20Und	Dsc: 87.8%	

**Total:** 2033.3252 € +

NOS 67% 11:58

**← Artigo: A0001**

**Descrição:**  
Pentium D925 Dual Core

**Stock:** -203      **Quantidade economica:** 10

**Preço unitário:**  
833.33€

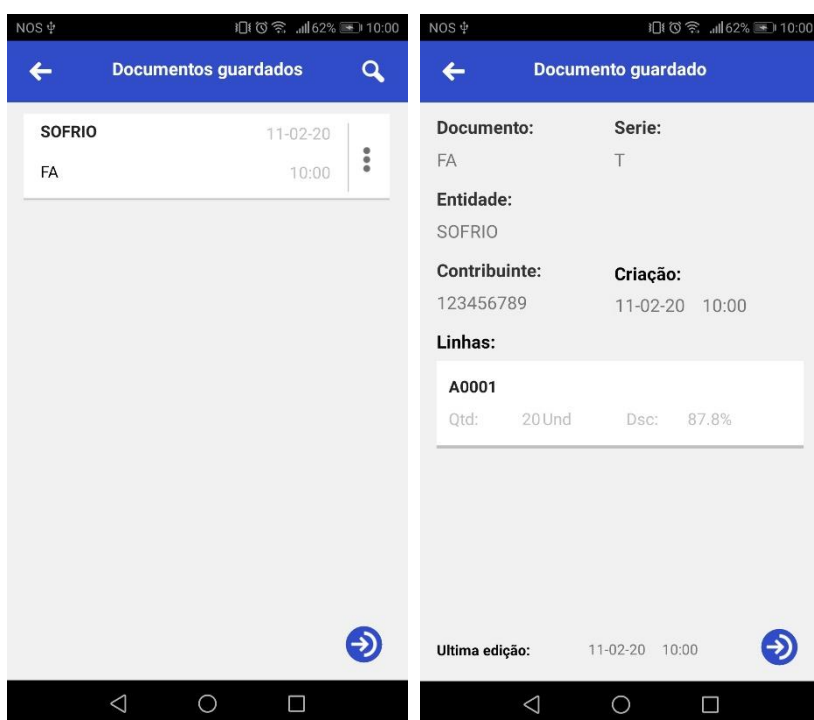
**Quantidade:**  
- 10 +

**Desconto:**  
10 87.8 833.3 83.33

✓

## 5.4. Documentos guardados

Através da página de listagem de documentos é possível aceder à página de documentos guardados. Nesta página o utilizador pode integrar os documentos que escolheu guardar na aplicação com o ERP V10.





## 6. Preferências

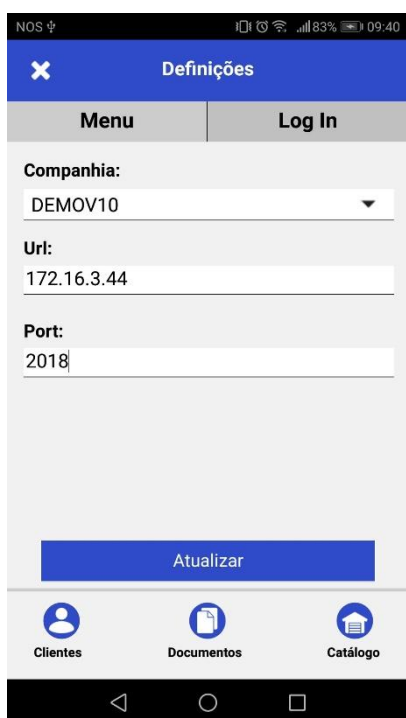
### 6.1. Menu

Nesta página é possível escolher as preferencias gerais da aplicação e importar dados do ERP de forma manual.



## 6.2. LogIn

Nesta página é possível escolher as preferencias de LogIn da aplicação. Quando um destes dados é atualizado, a aplicação efetua um LogIn automático de forma a atualizar os dados da aplicação.



## Especificação Técnica de Desenvolvimentos



### Serviço

Especificação Técnica de  
Desenvolvimentos

### Cliente

Primavera Business  
Software Solutions



### Versão do Documento *Template*

Versão 1.0

### Versão do Documento

Versão 1.1



<b>Histórico de Versões</b>			
<b>Consultor</b>	<b>Data</b>	<b>Alterações</b>	<b>Versão Doc.</b>
Rui Silva	04-03-2020	Criação do documento	1.0
Rui Silva	13-03-2020	Atualização de código	1.1

# ÍNDICE

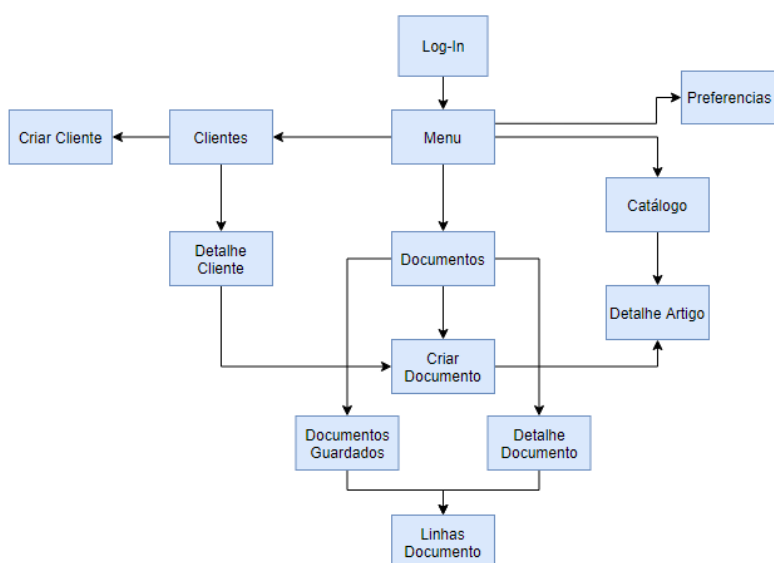
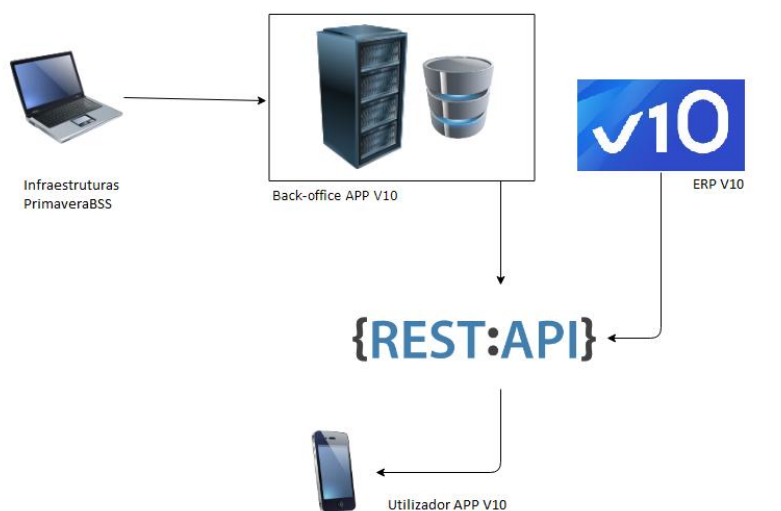
<b>1. ESPECIFICAÇÃO TÉCNICA</b>	<b>4</b>
1.1. BASE DE DADOS	5
1.1.1. Tabela: Artigo	5
1.1.2. Tabela: Artigo_Fatura	6
1.1.3. Tabela: Artigo_imagens	6
1.1.4. Tabela: Artigo_pvp	7
1.1.5. Tabela: Clientes	7
1.1.6. Tabela: Companhias	8
1.1.7. Tabela: Doc_Save	8
1.1.8. Tabela: Familias_artg	8
1.1.9. Tabela: Fatura	9
1.1.10. Tabela: Json_Doc	9
1.1.11. Tabela: log_in	10
1.1.12. Tabela: Moedas	10
1.1.13. Tabela: Novo_Doc	10
1.1.14. Tabela: Pais	11
1.1.15. Tabela: Series	11
1.1.16. Tabela: SubFamilias_artg	11
1.1.17. Tabela: Tipo_document	11
1.2. BACKEND	12
1.2.1. Form: Login	12
1.2.2. Form: Form_Menu	15
1.2.3. Form: Form_Clientes	29
1.2.4. Form: Form_Cliente_Detalhe	32
1.2.5. Form: Criar_Cliente	33
1.2.6. Form: Form_Documentos	35
1.2.7. Form: Form_Documento_Detalhe	38
1.2.8. Form: Form_Criar_Documento	41
1.2.9. Form: Documentos_guardados	50
1.2.10. Form: Form_Documento_Detalhe_guardado	52
1.2.11. Form: Form_Catalogo	54
1.2.12. Form: Form_Artigo_Detalhe	56

# 1. Especificação Técnica

Este manual visa o suporte de desenvolvimento contínuo da aplicação V10 através da ferramenta de desenvolvimento mobile Kalipso.

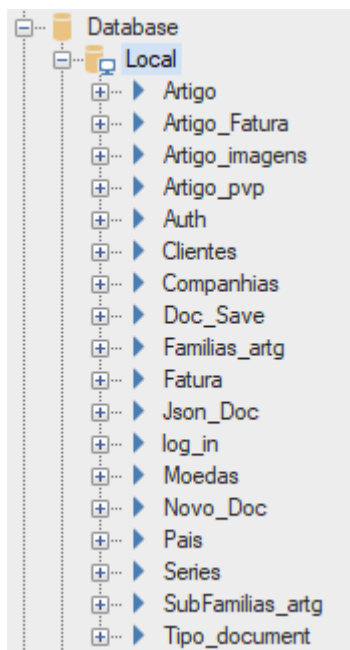
Neste manual serão expostas todas as componentes lógicas que constituem a aplicação de forma a compreender a sua estrutura.

## 1.1. Estrutura Global da aplicação

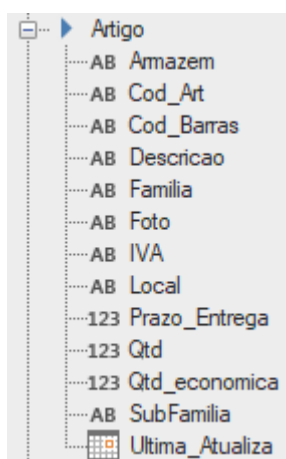


## 1.2. Base de Dados

Para este projeto é utilizada apenas a base dados local que funcionará como armazenamento interno do dispositivo de suporte para a aplicação.

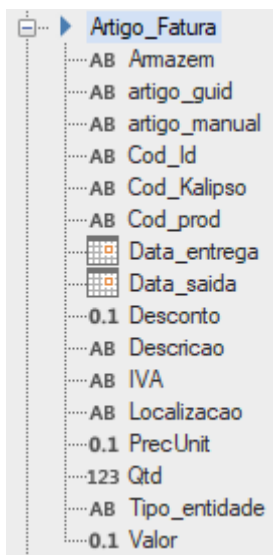


### 1.2.1. Tabela: Artigo



A tabela “Artigo” é utilizada para o armazenamento dos artigos de catálogo do ERP. Esta base de dados é utilizada para listar o catálogo na aplicação. O armazenamento corresponde a todas as informações necessárias do artigo.

### 1.2.2. Tabela: Artigo\_Fatura

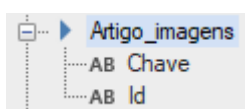


The screenshot shows a tree view of the 'Artigo\_Fatura' table structure. The table name is expanded, showing a list of fields with their data types and constraints. The fields are: 'Amazem' (AB), 'artigo\_guid' (AB), 'artigo\_manual' (AB), 'Cod\_Id' (AB), 'Cod\_Kalipso' (AB), 'Cod\_prod' (AB), 'Data\_entrega' (Data type with a calendar icon), 'Data\_saida' (Data type with a calendar icon), 'Desconto' (0.1), 'Descricao' (AB), 'IVA' (AB), 'Localizacao' (AB), 'PrecUnit' (0.1), 'Qty' (123), 'Tipo\_entidade' (AB), and 'Valor' (0.1).

Field Name	Data Type / Constraint
Amazem	AB
artigo_guid	AB
artigo_manual	AB
Cod_Id	AB
Cod_Kalipso	AB
Cod_prod	AB
Data_entrega	Data
Data_saida	Data
Desconto	0.1
Descricao	AB
IVA	AB
Localizacao	AB
PrecUnit	0.1
Qty	123
Tipo_entidade	AB
Valor	0.1

A tabela “Artigo\_Fatura” faz parte do conjunto de tabelas que constituem os documentos de venda presentes no ERP. Esta tabela armazena todos os artigos presentes nos documentos, sendo que estes são relacionados na aplicação através de “Cod\_Id”.

### 1.2.3. Tabela: Artigo\_imagens



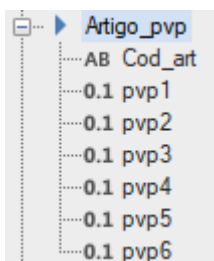
The screenshot shows a tree view of the 'Artigo\_imagens' table structure. The table name is expanded, showing a list of fields with their data types and constraints. The fields are: 'Chave' (AB) and 'Id' (AB).

Field Name	Data Type / Constraint
Chave	AB
Id	AB

A tabela “Artigo\_imagens” armazena as informações de imagens associadas aos artigos. O “Id” corresponde ao *guid* associado ao nome de cada imagem que é associado através da “Chave” que corresponde ao código de artigo.



#### 1.2.4. Tabela: Artigo\_pvp



AB Cod_art
0.1 pvp1
0.1 pvp2
0.1 pvp3
0.1 pvp4
0.1 pvp5
0.1 pvp6

O “Artigo\_pvp” armazena todos os diferentes pvp’s que poderão ser associados a cada artigo. Estes pvp’s são utilizados consoante o cliente requerido.

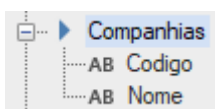
#### 1.2.5. Tabela: Clientes



AB CodPostal
AB Cont_Telf
AB Descricao
AB Distrito
AB Fax
AB id
AB Local_codPostal
AB Localidade
AB Moeda
AB Morada
AB Nome
AB NumContrib
AB Pais
AB Vendedor
AB Web

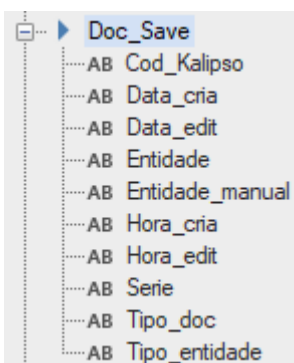
A tabela “Clientes” é utilizada para o armazenamento de clientes do ERP. Esta base de dados é utilizada para listar os clientes na aplicação. O armazenamento corresponde a todas as informações necessárias do cliente.

### 1.2.6. Tabela: Companhias



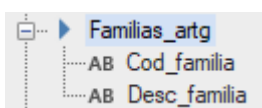
“Companhias” serve para armazenar todas as entidades associadas a um dado utilizador no ERP. Esta tabela serve para a listagem em combos presentes na aplicação.

### 1.2.7. Tabela: Doc\_Save



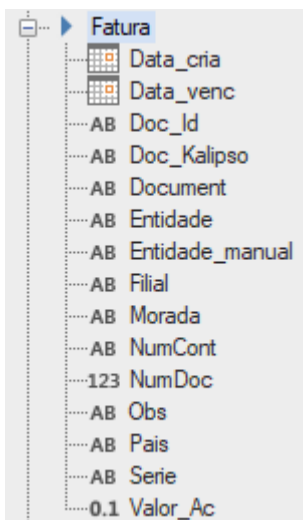
A tabela “Doc\_Save” armazena documentos caso o utilizador pretenda guardar um documento antes de o enviar. Esta tabela é utilizada como um suporte direto ao funcionamento *offline* da aplicação.

### 1.2.8. Tabela: Familias\_artg



“Familias\_artg” serve para armazenar todas as famílias que caracterizam um artigo no ERP. Esta tabela serve para a listagem em combos presentes na aplicação.

### 1.2.9. Tabela: Fatura



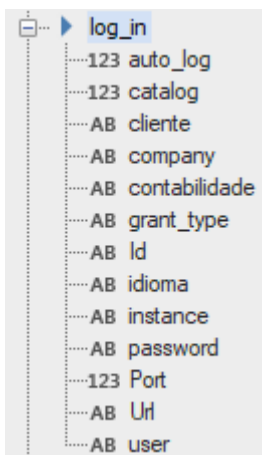
A tabela “Fatura” faz parte do conjunto de tabelas que constituem os documentos de venda presentes no ERP. Esta tabela armazena todos os cabeçalhos dos documentos, sendo que estes são relacionados com os artigos correspondentes através de “Cod\_Id” na aplicação.

### 1.2.10. Tabela: Json\_Doc



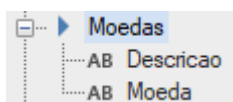
A tabela “Json\_Doc” serve para armazenar o Json construído através da resposta do pedido de “FillRelatedData” de forma a completar a criação de um documento de venda.

### 1.2.11. Tabela: log\_in



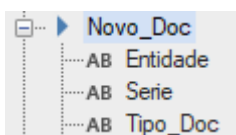
A tabela “log\_in” armazena todas as informações de log\_in de um utilizador. Apenas um registo é criado após cada entrada na aplicação. “Url” e “Port” são utilizadas também para a criação de variáveis globais de forma a completar todos os pedidos na aplicação.

### 1.2.12. Tabela: Moedas



“Moedas” serve para armazenar todas as diferentes Moedas presentes no ERP. Esta tabela serve para a listagem em combos presentes na aplicação.

### 1.2.13. Tabela: Novo\_Doc



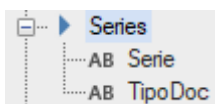
Esta tabela serve como apoio para a criação do Json que efetua o pedido de “FillRelatedData”.

#### 1.2.14. Tabela: Pais



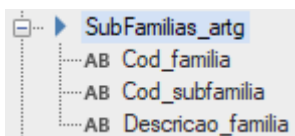
“Pais” serve para armazenar todas os diferentes países presentes no ERP. Esta tabela serve para a listagem em combos presentes na aplicação.

#### 1.2.15. Tabela: Series



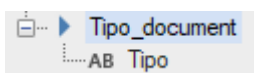
“Series” serve para armazenar todas os diferentes series associadas a cada tipo de documento. Esta tabela serve para a listagem em combos presentes na aplicação.

#### 1.2.16. Tabela: SubFamilias\_artg



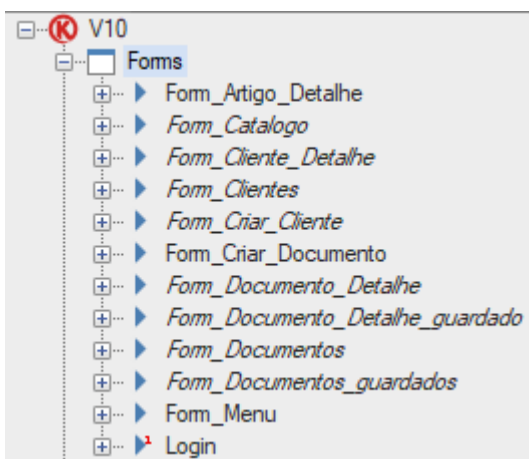
“SubFamilias\_artg” serve para armazenar todas as diferentes subfamílias associadas a cada família que caracteriza um artigo. Esta tabela serve para a listagem em combos presentes na aplicação.

#### 1.2.17. Tabela: Tipo\_document



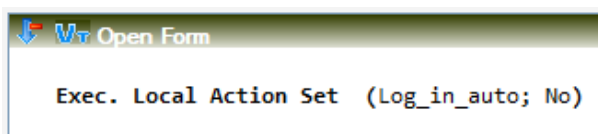
“Tipo\_document” serve para armazenar todas os diferentes tipos de documentos que caracterizam todos os documentos. Esta tabela serve para a listagem em combos presentes na aplicação.

## 1.3. BackEnd

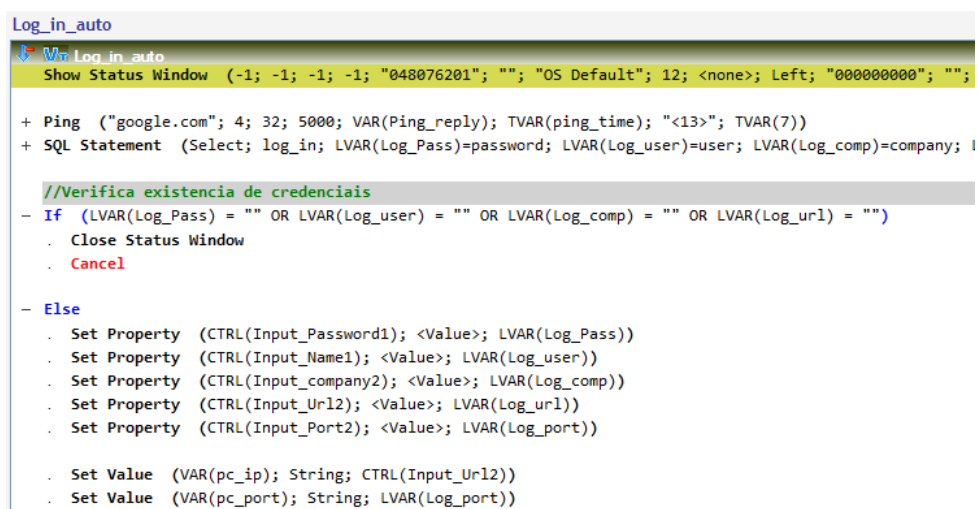


### 1.3.1. Form: Login

#### 1.3.1.1. Form

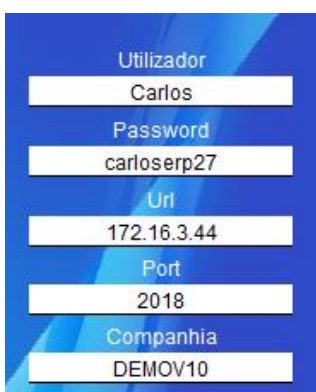


Durante o acesso à aplicação é efetuada a navegação através do *form* “Login” que, ao abrir, executa a *Local Action Set* “Log\_in\_auto”.



Começa-se por ser executado um “Ping” que verifica a ligação à rede. Após esta confirmação é efetuado um select à tabela “log\_in” que retira o “utilizador”, “password”, “url”, “port” e “companhia” para respetivas variáveis.

É efetuado um ciclo “if” que verifica se estas variáveis contem valores. Caso seja verdade o log in é cancelado e é apenas disponibilizado o *form* “Login” ao utilizador de forma a este inserir os respetivos dados.



A screenshot of a login form with a blue background. The form contains five input fields, each with a label above it: 'Utilizador' (containing 'Carlos'), 'Password' (containing 'carloserp27'), 'Uri' (containing '172.16.3.44'), 'Port' (containing '2018'), and 'Companhia' (containing 'DEMOV10').

Caso existam dados nas variáveis, o valor de estas são atribuídas aos campos do formulário. São também atribuídos os valores de “url” e “port” a variáveis globais que serão utilizadas em todos os restantes pedidos na aplicação.

```
. //Verifica conexão à rede
- . If (VAR(Ping_reply) = 0)
. . Message Box ("Aviso!"; "Não se encontra ligado à rede de momento, poderá não ter acesso
. . Close Status Window
. . Show Form (Form_Menu; 0)

- . Else
+ . . SQL Statement (Select; log_in; LVAR(username)=user; LVAR(password)=password; LVAR(compar
+ . . Http Request (Run Locally; 0; TVAR(var_json); TVAR(header); TVAR(status); TVAR(text); Ye
+ . . JSON Get Value (TVAR(var_json); "access_token"; No; VAR(answer_token); LVAR(keep_log))

. . Close Status Window
. . Show Form (Form_Menu; 0)

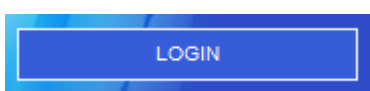
. End If
End If
```

Após a atribuição dos valores é efetuado um novo ciclo “if” que valida a ligação à rede do “Ping” anteriormente executado. Caso não haja ligação é demonstrada uma caixa de texto que informa o utilizador da situação e é efetuada a navegação para o formulário “Form\_Menu” sem que seja retornado um token.

Caso exista ligação é efetuado o pedido a WebApi do token com as respetivas credenciais. O token retornado do pedido é posteriormente guardado numa variável global de forma a ser utilizado pelos restantes pedidos na aplicação.

Por fim é efetuada a navegação para o formulário “Form\_Menu”.

#### 1.3.1.2. Botão: LOGIN



Após serem inseridos os dados pedidos no formulário o utilizador deverá seleccionar o botão LOGIN.

```
Click
//Verifica a existencia de ligação à rede
Show Status Window (-1; -1; -1; -1; "048076201"; ""; "OS Default"; 12; <none>; Left; "
+ Ping ("google.com"; 4; 32; 5000; VAR(Ping_reply); TVAR(ping_time); "<13>"; TVAR(7))

- If (VAR(Ping_reply) = 0)
. Close Status Window
. Message Box ("Atenção"; "Não tem de momento acesso à rede, tente mais tarde."; Infor
. Show Form (Form_Menu; 0)

- Else
. Set Value (VAR(pc_ip); String; CTRL(Input_Ur12))
. Set Value (VAR(pc_port); String; CTRL(Input_Port2))
+ . SQL Statement (Delete; log_in; 1=1)
+ . SQL Statement (Insert; log_in; No; Yes; company=CTRL(Input_company2); grant_type="p
+ . SQL Statement (Select; log_in; LVAR(username)=user; LVAR(password)=password; LVAR(c
+ . Http Request (Run Locally; 15000; TVAR(var_json); TVAR(header); TVAR(status); TVAR(
+ . JSON Get Value (TVAR(var_json); "access_token"; No; VAR(answer_token); LVAR(keep_lo

. Close Status Window
- . If (TVAR(status) = 200)
. . Show Form (Form_Menu; 0)
- . Else
. . Message Box (""; "Ocorreu um erro, tente mais tarde."; Information; OK; OK; )
. End If

End If
```

Ao seleccionar o botão através da ação “Click” é efetuado um “Ping” que verifica a ligação a rede. Para validar isto é efetuado um ciclo “if” que caso não haja ligação é executado o processo anteriormente descrito no processo do “Form”. Caso exista uma ligação são atribuídos os valores dos inputs Port e Url do formulário às respetivas variáveis globais de forma a serem usadas nos restantes pedidos.



É efetuada um delete á tabela de “Log\_in” de forma a eliminar qualquer dado já existente pois apenas é pretendido que exista um registo por cada login. São depois inseridos os dados dos inputs na mesma tabela criando assim um registo.

É efetuado um *select* à mesma tabela, retirando todos os dados necessários para o pedido de log in à WebApi. É de seguida efetuado o pedido à WebApi e retirado o token da resposta do mesmo, que é posteriormente guardado na respetiva variável global de forma a ser utilizada pelos restantes pedidos da aplicação.

Após o pedido é validada a resposta de status do mesmo. Através de um novo ciclo “if” é validada a resposta que caso seja “200” é efetuada a navegação para o formulário “Form\_Menu”. Caso seja diferente de “200” é apresentada uma caixa de texto a informar o utilizador que algo correu mal com o pedido e é cancelado todo o processo.

## 1.3.2. Form: Form\_Menu

### 1.3.2.1. Form

```
W Open Form
Go To Plane (2)
Show Status Window (-1; -1; 30; 30; "048076201"; ""; "OS Default"; 12; <none>; Left; "00000000");

//Executar principais REQUESTs à web api
Exec. Local Action Set (Principais_Pedidos; No)

//Consulta preferencias do user
+ SQL Statement (Select; log_in; LVAR(conta_log)=contabilidade; LVAR(cliente_log)=cliente; LVAR(idic
+ Refresh Control (CTRL(Combo_entidades); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_entidades2); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_entidades1); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_tipo_mensal); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_set_cliente); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_set_contabilidade); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_set_idioma); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_set_company); Refresh List; Refresh Value)
```

Devido a este formulário conter mais do que um *plane* é, em primeiro lugar, efetuada a navegação para o primeiro plane pretendido, que neste caso será o 2. Após isto é executada a *Local Action Set* “Principais\_Pedidos” e um *Select* à tabela log\_in que retira os dados de preferências do utilizador. Por fim são efetuados os devidos *refreshs* a todas as combos presentes no formulário.

```

//Artigos Request e PVP
+ SQL Statement (Delete; Artigo_pvp; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_pvp); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip); "\v
+ JSON Import to Table (Text; TVAR(answer_pvp); Artigo_pvp)

//Artigos Imagens
+ SQL Statement (Delete; Artigo_imagens; 1=1)
+ SQL Statement (Select; Artigo_imagens; TVAR(Check_art_img)=COUNT(Id); ; No; Yes)
+ If (TVAR(Check_art_img) = 0)

//Clientes Request
+ SQL Statement (Delete; Clientes; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_client); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip);
+ JSON Import to Table (Text; TVAR(answer_client); Clientes)

//Documentos Request
+ SQL Statement (Delete; Fatura; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_sales); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip);
+ JSON Import to Table (Text; TVAR(answer_sales); Fatura)

//Artigos de Fatura Request
+ SQL Statement (Delete; Artigo_Fatura; FIELD(Artigo_Fatura,Cod_Kalipso) = '')
+ Http Request (Run Locally; 0; TVAR(answer_sales_artg); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_
+ JSON Import to Table (Text; TVAR(answer_sales_artg); Artigo_Fatura)

```

```

//Inserir Combo Familias
+ SQL Statement (Delete; Familias_artg; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_pvp); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip);
+ JSON Import to Table (Text; TVAR(answer_pvp); Familias_artg)

```

```

//Inserir Combo Sub-Familias
+ SQL Statement (Delete; SubFamilias_artg; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_pvp); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip);
+ JSON Import to Table (Text; TVAR(answer_pvp); SubFamilias_artg)

```

```

//Combo Países
+ SQL Statement (Delete; Pais; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_client); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_i
+ JSON Import to Table (Text; TVAR(answer_client); Pais)

```

```

//Combo Moedas
+ SQL Statement (Delete; Moedas; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_items); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip)
+ JSON Import to Table (Text; TVAR(answer_items); Moedas)

```

```

//Preenche Base de Dados para o Combo de Tipo de Documentos
+ SQL Statement (Delete; Tipo_document; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_items); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip)
+ JSON Get Value (TVAR(answer_items); "DataSet"; Yes; TVAR(22); TVAR(21))
+ JSON Import to Table (Text; TVAR(22); Tipo_document)

```

```

//Preenche Base de Dados para o Combo de Series
+ SQL Statement (Delete; Series; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_client); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip);
+ JSON Import to Table (Text; TVAR(answer_client); Series)

```

```

//Listar Companhias
+ SQL Statement (Delete; Companhias; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_items); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip);
+ JSON Import to Table (Text; TVAR(answer_items); Companhias)

```

A *local action set* “Principais\_Pedidos” executa todos os pedidos necessários para o funcionamento da aplicação. Todos os blocos de pedido seguem a mesma estrutura básica de apagar quaisquer dados que ainda se encontrem nas tabelas seguido pela execução do pedido ao *endpoint* correspondente à WebApi terminando com a importação dos dados do Json da resposta diretamente para a tabela correspondente.

Apenas os blocos de pedido de “Artigos Imagens” e “Tipo de Documentos” apresentam ações extra. No “Artigos Imagens” antes de ser efetuado o pedido à WebApi é primeiro verificado se existe imagens no dispositivo, apenas no caso de não existir é que o pedido é efetuado. Por último no “Tipo de Documento” é efetuado um *Get Value* extra que serve para uma navegação mais aprofundado do Json devido a este conter um array.

Nesta *local action set* são utilizados vários pedidos com recurso às listas personalizáveis do ERP V10 na tabela seguinte são mostradas as correspondentes.

Bloco	Lista no ERP
Artigos Request e PVP (PVP)	PvP_linhas
Artigos Imagens	Get_Anexos
Clientes Request	Clientes
Documentos Request	Documentos
Artigos de Fatura Request	Linhas_Doc
Inserir Combo Familias	Familias_artg
Inserir Combo Sub-Familias	SubFamilias_artg
Combo Países	Países
Combo Moedas	Moeda
Combo Series	Series

Após o formulário abrir é executado um segundo conjunto de código que começa por atribuir os valores provenientes da tabela Log\_in às combos e inputs correspondentes preenchendo assim as preferências de utilizador no plane de Definições.

```

M After Open Form
//Preencher as preferencias do user
Set Selection (CTRL(Combo_entidades2); By Returned Value; LVAR(cliente_log))
Set Selection (CTRL(Combo_entidades1); By Returned Value; LVAR(cliente_log))
Set Selection (CTRL(Combo_entidades); By Returned Value; LVAR(cliente_log))

Set Selection (CTRL(Combo_tipo_mensal); By Returned Value; LVAR(conta_log))

Set Selection (CTRL(Combo_set_cliente); By Returned Value; LVAR(cliente_log))
Set Selection (CTRL(Combo_set_idioma); By Returned Value; LVAR(idio_log))
Set Selection (CTRL(Combo_set_contabilidade); By Returned Value; LVAR(conta_log))

Set Selection (CTRL(Combo_set_company); By Returned Value; LVAR(company_log))
Set Value (CTRL(Input_Url_update); String; LVAR(url_log))
Set Value (CTRL(Input_Port_update); String; VAR(pc_port))
Set Selection (CTRL(Radio_list); By Returned Value; LVAR(catalog_log))

```

Após a verificação das preferências são desenhados os gráficos de estatísticas presentes no plane 1 e 2. Em primeiro lugar é verificado a opção de contabilidade, estando distribuída por mensal, trimestral e semestral.

```

//Verificar preferencias contabilisticas do user
+ SQL Statement (Select; log_in; LVAR(check_cont)=contabilidade; ; No; Yes)
- If (LVAR(check_cont) = 1)
. Exec. Local Action Set (Estatisticas_mensais_last12; No)
- Else
. If (LVAR(check_cont) = 2)
. . Exec. Local Action Set (Estatisticas_trimestrais; No)
- . Else
. . Exec. Local Action Set (Estatisticas_semestrais; No)
. End If
End If

```

Consoante a preferência presente na tabela e validando através de um ciclo “if” a devida *local action set* é executada.

```
//Desenhar os graficos
Exec. Local Action Set (Dados_estatisticos_graph2; No)

+ Refresh Control (CTRL(HTMLviewer_chart); Refresh Value)

+ SQL Statement (Select; Doc_Save; TVAR(count_save_docs)=COUNT(Cod_Kalipso); ; No; Yes)
- If (TVAR(count_save_docs) = 0)
. Set Property (CTRL(Label_avisos); Visible; No)
. Set Property (CTRL(Image_avisos); Visible; No)
. Set Property (CTRL(Shape_avisos); Visible; No)
. Set Property (CTRL(Label_avisos1); Visible; No)
. Set Property (CTRL(Image_avisos1); Visible; No)
. Set Property (CTRL(Shape_avisos1); Visible; No)

- Else
. Set Property (CTRL(Label_avisos); Visible; Yes)
. Set Property (CTRL(Image_avisos); Visible; Yes)
. Set Property (CTRL(Shape_avisos); Visible; Yes)
. Set Property (CTRL(Label_avisos1); Visible; Yes)
. Set Property (CTRL(Image_avisos1); Visible; Yes)
. Set Property (CTRL(Shape_avisos1); Visible; Yes)

End If
```

Após a execução da *local action set* (LAS) pretendida é executada a LAS “Dados\_estatisticos\_graph2” que corresponde ao gráfico presente no plane 3.

Após isto é executado um *select* à tabela “Doc\_Save” de forma a verificar a existência de documentos de venda guardados no armazenamento do dispositivo. Caso exista documentos são disponibilizados auxiliares visuais que alertem o utilizador para tal.

```
Close Status Window
- If (CTRL(Combo_set_contabilidade) = "" OR CTRL(Combo_set_idioma) = "" OR CTRL(Combo_set_cliente) = "")
. Message Box (""; "Ainda não tem as suas preferencias definidas. Pretende efetuar esta ação agora?"; Infoc
- . If (TVAR(count_save_docs) = 7)
. . Go To Plane (4)
- . Else
. End If

- Else
End If
```

Por fim é verificado se o utilizador tem ou não preferências já definidas (código auxiliar para o primeiro log in) que caso não tenha é disponibilizado uma mensagem de texto que, caso o utilizador aceite, navega o mesmo para o plane de Definições.

A LAS “Estatistica\_mensais\_last12” tem como objetivo a construção de um Json a exportar com os dados de vendas dos últimos 12 meses. Este processo é iniciado com a construção do início do Json através da variável JsonString. São também criadas variáveis temporárias que contem a data atual do dispositivo, ano anterior (4 e 2 dígitos) e o mesmo atual.

```
Wn Estatisticas_mensais_last12
Set Value (TVAR(JsonString); String; "[['Mes','Vendas']")
Set Value (TVAR(Data_atual); String; DateSys())
Set Value (TVAR(Ano_inicial); Numeric; (Year(DateSys(), 4) - 1))
Set Value (TVAR(Ano_inicial_por_2); Numeric; (Year(DateSys(), 2) - 1))
Set Value (TVAR(Mes_inicial); Numeric; Month(DateSys()))

//Listar últimos 12 meses
- For Each (FOR; Numeric; TVAR(Count); 0; TVAR(Count) < 12; TVAR(Count) + 1)
. . If (TVAR(Mes_inicial) = 13)
. . . Set Value (TVAR(Ano_inicial); Numeric; Year(DateSys(), 4))
. . . Set Value (TVAR(Mes_inicial); Numeric; 01)
. . . Set Value (TVAR(Ano_inicial_por_2); Numeric; (Year(DateSys(), 2)))
. . Else
. . . End If
+ . If (TVAR(Mes_inicial) = 01)
+ . If (Length(TVAR(Mes_inicial)) = 1)
+ . SQL Statement (Select; Fatura; TVAR(Valor_vendas)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" +
+ . If (TVAR(Valor_vendas) = "")
. . Set Value (TVAR(JsonString); String; TVAR(JsonString) + "," + '[' + TVAR(Mes_nome) + "." + TVAR(Ano_inicial_
. Set Value (TVAR(Mes_inicial); Numeric; TVAR(Mes_inicial) + 1)
+ End For Each
```

É executado depois um ciclo “for each” que avalia todos os valores dos últimos 12 meses segundo a data atual. É verificado o valor do mês, que caso seja igual a 13 é revertido para 1.

```

- . If (TVAR(Mes_inicial) = 01)
- . . Set Value (TVAR(Mes_nome); String; "Jan")
- . Else
- . . If (TVAR(Mes_inicial) = 02)
- . . . Set Value (TVAR(Mes_nome); String; "Feb")
- . . . Else
- . . . . If (TVAR(Mes_inicial) = 03)
- . . . . . Set Value (TVAR(Mes_nome); String; "Mar")
- . . . . . Else
- . . . . . . If (TVAR(Mes_inicial) = 04)
- . . . . . . . Set Value (TVAR(Mes_nome); String; "Abr")
- . . . . . . . Else
- . . . . . . . . If (TVAR(Mes_inicial) = 05)
- . . . . . . . . . Set Value (TVAR(Mes_nome); String; "Mai")
- . . . . . . . . . Else
- . . . . . . . . . . If (TVAR(Mes_inicial) = 06)
- . . . . . . . . . . . Set Value (TVAR(Mes_nome); String; "Jun")
- . . . . . . . . . . . Else
- . . . . . . . . . . . . If (TVAR(Mes_inicial) = 07)
- . . . . . . . . . . . . . Set Value (TVAR(Mes_nome); String; "Jul")
- . . . . . . . . . . . . . Else
- . . . . . . . . . . . . . . If (TVAR(Mes_inicial) = 08)
- . . . . . . . . . . . . . . . Set Value (TVAR(Mes_nome); String; "Ago")
- . . . . . . . . . . . . . . . Else
- . . . . . . . . . . . . . . . . If (TVAR(Mes_inicial) = 09)
- . . . . . . . . . . . . . . . . . Set Value (TVAR(Mes_nome); String; "Set")
- . . . . . . . . . . . . . . . . . Else
- . . . . . . . . . . . . . . . . . . If (TVAR(Mes_inicial) = 10)

```

Dentro do ciclo “for each” é executado uma susceção de ciclos “if” que analisa todos os valores de mês que por sua vez atribui a abreviatura do mês correspondente.

```

- . If (Length(TVAR(Mes_inicial)) = 1)
- . . Set Value (TVAR(Mes_inicial); String; "0" + TVAR(Mes_inicial))
- . Else
- . . End If

```

É também executado um novo ciclo “if” que caso um valor de um certo mês seja de apenas um dígito, é acrescentado um 0 de forma a ficar, por exemplo, 1 = 01, de forma a criar uma conformidade com os resultados das datas obtidas.

```

//Listar o mes atual
Set Value (TVAR(Mes_atual); String; Month(DateSys()))

+ If (TVAR(Mes_atual) = 01)
+ If (Length(TVAR(Mes_atual)) = 1)
+ SQL Statement (Select; Fatura; TVAR(Valor_vendas)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + C

- If (TVAR(Valor_vendas) = "")
- . Set Value (TVAR(Valor_vendas); Numeric; 0)
- Else
- . End If

Set Value (TVAR(JsonString); String; TVAR(JsonString) + "[" + TVAR(Mes_nome) + ". " + Year(DateSys(), 2)
Set Value (TVAR(JsonString); String; TVAR(JsonString) + "]")

Set Value (TVAR(Data_draw); String; JSONEncode(TVAR(JsonString)))
+ HTML Execute JScript (CTRL(HTMLViewer_chart); "drawStuff(" + TVAR(Data_draw) + ");"; TVAR(Data_draw))

```

Após o ciclo “for each” é executado o mesmo processo para o mês atual. Para finalizar é fechado o Json através da variável JsonString executando por fim o HTML *Execute JScript* que envia o valor da variável para o HTML presente no plane 2.

```

+ . . . Set Value (TVAR(Json_string); String; "[['Semestre','Vendas'],")
+ . . . Set Value (TVAR(Data_atual); String; DateSys())

+ . . . SQL Statement (Select; Fatura; TVAR(count_sem_1)=COUNT(Doc_Id); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidades) + '%" AND "
+ . . . SQL Statement (Select; Fatura; TVAR(count_sem_2)=COUNT(Doc_Id); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidades) + '%" AND "

- . . . If (TVAR(Data_atual) > Year(DateSys(), 4) + "0101 AND" + TVAR(Data_atual) < Year(DateSys(), 4) + "0630")
. . . . Set Value (TVAR(Ano_anterior); Numeric; Year(DateSys(), 4) - 1)
. . . . Set Value (TVAR(Ano_anterior_2); Numeric; Year(DateSys(), 2) - 1)

+ . . . SQL Statement (Select; Fatura; TVAR(Valor_sem_1)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidades) + '%" AN
+ . . . SQL Statement (Select; Fatura; TVAR(Valor_sem_ant)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidades) + '%"

- . . . If (TVAR(Valor_sem_1) = "")
. . . . Set Value (TVAR(Valor_sem_1); Numeric; 0)
- . . . Else
. . . . End If
- . . . If (TVAR(Valor_sem_ant) = "")
. . . . Set Value (TVAR(Valor_sem_ant); Numeric; 0)
- . . . Else
. . . . End If

. . . Set Value (TVAR(Json_string); String; TVAR(Json_string) + "["Semestre2" + "." + TVAR(Ano_anterior_2) + "," + TVAR(Valor_sem_ant) + "

- . . . Else
- . . . . If (TVAR(Data_atual) > Year(DateSys(), 4) + "0701" AND TVAR(Data_atual) < Year(DateSys(), 4) + "1231")
+ . . . . SQL Statement (Select; Fatura; TVAR(Valor_sem_1)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidades) + "
+ . . . . SQL Statement (Select; Fatura; TVAR(Valor_sem_2)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidades) + "

+ . . . . If (TVAR(Valor_sem_1) = "")
+ . . . . If (TVAR(Valor_sem_2) = "")

. . . . Set Value (TVAR(Json_string); String; TVAR(Json_string) + "["Semestre1" + "." + Year(DateSys(), 2) + "," + TVAR(Valor_sem_1) + "

- . . . Else
. . . . End If
End If

Set Value (TVAR(Data_draw); String; JSONEncode(TVAR(Json_string)))
+ HTML Execute JScript (CTRL(HTMLViewer_chart); "drawStuff(" + TVAR(Data_draw) + ");"; TVAR(Data_draw))

```

A LAS “Estatísticas\_semestrais” começa como a anterior, criando o início do Json da mesma forma. Desta vez apenas a data atual é definida numa variável temporária. São executados 2 *selects* que efetuam um *count* ao número de documento

Após isto é executado um ciclo “if” que efetua a construção dos valores do semestre atual e anterior. Executando por fim a HTML *Execute JScript* com o Json criado.



```

Vn Estatisticas Trimestrais
Set Value (TVAR(Json_string); String; "[['Trimestre','Vendas'],")

Set Value (TVAR(Data_atual); String; DateSys())
Set Value (TVAR(Ano_anterior); Numeric; Year(DateSys(), 4) - 1)
Set Value (TVAR(Ano_anterior2); Numeric; Year(DateSys(), 2) - 1)

- If (TVAR(Data_atual) > Year(DateSys(), 4) + "0101" AND TVAR(Data_atual) < Year(DateSys(), 4) + "0331")
+ . SQL Statement (Select; Fatura; TVAR(valor_tri_ant2)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%"
+ . SQL Statement (Select; Fatura; TVAR(valor_tri_ant3)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%"
+ . SQL Statement (Select; Fatura; TVAR(valor_tri_ant4)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%"
+ . SQL Statement (Select; Fatura; TVAR(valor_tri_1)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + C

- . If (TVAR(valor_tri_ant2) = "")
. Set Value (TVAR(valor_tri_ant2); Numeric; 0)
- . Else
. End If
+ . If (TVAR(valor_tri_ant3) = "")
+ . If (TVAR(valor_tri_ant4) = "")
+ . If (TVAR(valor_tri_1) = "")

. Set Value (TVAR(Json_string); String; TVAR(Json_string) + "["Trimestre2" + "." + TVAR(Ano_anterior2) + "

- Else

- . If (TVAR(Data_atual) > Year(DateSys(), 4) + "0401" AND TVAR(Data_atual) < Year(DateSys(), 4) + "0630")
+ . . SQL Statement (Select; Fatura; TVAR(valor_tri_ant3)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" +
+ . . SQL Statement (Select; Fatura; TVAR(valor_tri_ant4)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" +
+ . . SQL Statement (Select; Fatura; TVAR(valor_tri_1)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + C1
+ . . SQL Statement (Select; Fatura; TVAR(valor_tri_2)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + C1

+ . . If (TVAR(valor_tri_ant3) = "")
+ . . If (TVAR(valor_tri_ant4) = "")
+ . . If (TVAR(valor_tri_1) = "")
+ . . If (TVAR(valor_tri_2) = "")

. . Set Value (TVAR(Json_string); String; TVAR(Json_string) + "["Trimestre3" + "." + TVAR(Ano_anterior2) + "

- . Else
- . . If (TVAR(Data_atual) > Year(DateSys(), 4) + "0701" AND TVAR(Data_atual) < Year(DateSys(), 4) + "0930")
+ . . . SQL Statement (Select; Fatura; TVAR(valor_tri_ant4)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%"
+ . . . SQL Statement (Select; Fatura; TVAR(valor_tri_1)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" +
+ . . . SQL Statement (Select; Fatura; TVAR(valor_tri_2)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" +
+ . . . SQL Statement (Select; Fatura; TVAR(valor_tri_3)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" +

+ . . . If (TVAR(valor_tri_ant4) = "")
+ . . . If (TVAR(valor_tri_1) = "")
+ . . . If (TVAR(valor_tri_2) = "")
+ . . . If (TVAR(valor_tri_3) = "")

. . . Set Value (TVAR(Json_string); String; TVAR(Json_string) + "["Trimestre4" + "." + TVAR(Ano_anterior2) + "

- . . Else

```

No LAS “Estatisticas\_trimestrais” tal como as 2 anteriores é iniciado da mesma forma com a variável `Json_string` e datas correspondentes. Seguindo o trimestre atual, são analisados os últimos 3 trimestres anteriores a serem executados. Isto é executado através de um ciclo “if” que executa, por cada um, um *select* com o intervalo de datas que correspondem a um dado trimestre. Caso o valor seja vazio o valor do trimestre é atribuído a 0.

```

- . . . . . If (TVAR(Data_atual) > Year(DateSys(), 4) + "1001" AND TVAR(Data_atual) < Year(DateSys(), 4) +
+ . . . . . SQL Statement (Select; Fatura; TVAR(valor_tri_1)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + "
+ . . . . . SQL Statement (Select; Fatura; TVAR(valor_tri_2)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + "
+ . . . . . SQL Statement (Select; Fatura; TVAR(valor_tri_3)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + "
+ . . . . . SQL Statement (Select; Fatura; TVAR(valor_tri_4)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + "

+ . . . . . If (TVAR(valor_tri_1) = "")
+ . . . . . If (TVAR(valor_tri_2) = "")
+ . . . . . If (TVAR(valor_tri_3) = "")
+ . . . . . If (TVAR(valor_tri_4) = "")

. . . . . Set Value (TVAR(Json_string); String; TVAR(Json_string) + "["Trimestre1" + "." + Year(DateS

- . . . . . Else
. . . . . End If
. . . . . End If
. . . . . End If
End If

Set Value (TVAR(Data_draw); String; JSONEncode(TVAR(Json_string)))
+ HTML Execute JScript (CTRL(HTMLviewer_chart); "drawStuff(" + TVAR(Data_draw) + ");"; TVAR(Data_draw))

```

Por fim é executado o HTML *Execute JScript* como previamente.

```

Dados estatísticos graph2
+ SQL Statement (Select; Fatura; TVAR(valor_ano1)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidade
- If (TVAR(valor_ano1) = "" OR TVAR(valor_ano1) < 0)
. Set Value (TVAR(valor_ano1); Numeric; 0)
- Else
End If

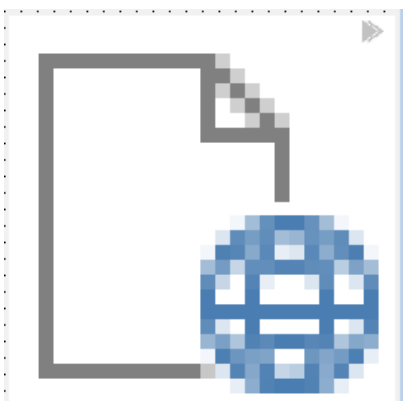
Set Value (TVAR(Ano_anterior); Numeric; Year(DateSys(), 4) - 1)
+ SQL Statement (Select; Fatura; TVAR(valor_ano2)=SUM(Valor_Ac); FIELD(Fatura, Entidade) + " LIKE '%" + CTRL(Combo_entidade
- If (TVAR(valor_ano2) = "" OR TVAR(valor_ano2) < 0)
. Set Value (TVAR(valor_ano2); Numeric; 0)
- Else
End If

Set Value (TVAR(Data_draw); String; JSONEncode(["[Ano', 'Valor', { role: 'style' }], [" + TVAR(Ano_anterior) + "', " +
+ HTML Execute JScript (CTRL(HTMLviewer_chart2); "drawGraph(" + TVAR(Data_draw) + ");"; TVAR(Data_draw))

```

A última LAS utilizada neste Form é a “Dados\_estatísticos\_graph2” que corresponde ao HTML presente no plane3 e executa os valores contabilísticos do ano atual e anterior. O funcionamento segue a mesma lógica das LAS anteriores.

### 1.3.2.2. HTMLViewer



O “HTMLViewer” funciona através de componente normal de html sendo que a variável é executada através de uma *function*.

```
<script type="text/javascript">

google.charts.load('current', {packages:['bar']});
google.charts.setOnLoadCallback(drawStuff);

function drawStuff(ArrayData) {

var data = google.visualization.arrayToDataTable(ArrayData);

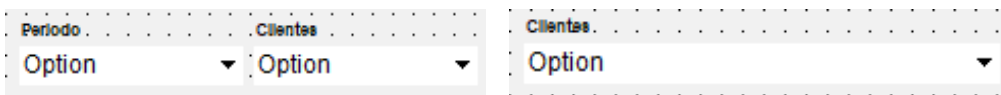
var options = {
  chart: {},
  bars: 'vertical',
  legend: {position: 'none'},
  vAxis: { textPosition: 'none' }
};

var chart = new google.charts.Bar(document.getElementById('barchart_material'));
chart.draw(data, google.charts.Bar.convertOptions(options));

}

</script>
```

### 1.3.2.3. Combos



As combos apresentam as opções de visualização dos HTMLViewer's sendo que alterar o seu valor é executada a LAS correspondente ao valor.

Alterando a seleção da combo “Período” irá ser executada a LAS contábilística correspondente. Caso seja a cliente irá ser executada as mesmas LAS mas com o cliente correspondente.

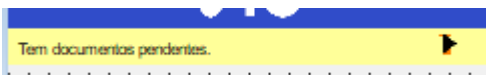
```

W Selection Change
- If (CTRL(Combo_tipo_mensal) = 1)
  . Exec. Local Action Set (Estatisticas_mensais_last12; No)
- Else
  . If (CTRL(Combo_tipo_mensal) = 2)
    . . Exec. Local Action Set (Estatisticas_trimestrais; No)
  . Else
    . . Exec. Local Action Set (Estatisticas_semensais; No)
  . End If
End If

+ Refresh Control (CTRL(HTMLViewer_chart); Refresh Value)

```

#### 1.3.2.4. Label de aviso



A label de aviso funciona como um auxiliar visual para avisar o utilizador sendo que quando este esteja visível pode ser selecionado.

Ao selecionar este navega para o “Form\_Documentos\_guardados” sendo que ao regressar ao “Form\_Menu” uma nova contagem de documentos guardados é efetuada um *select* de forma a prever se é ainda necessário apresentar o aviso.

```

W Click
Show Form (Form_Documentos_guardados; 0; (LVAR(Menu); String; "1"))
+ SQL Statement (Select; Doc_Save; TVAR(count_save_docs)=COUNT(Cod_Kalipso); ; No; Yes)
- If (TVAR(count_save_docs) = 0)
  . Set Property (CTRL(Label_aviso); Visible; No)
  . Set Property (CTRL(Image_aviso); Visible; No)
  . Set Property (CTRL(Shape_aviso); Visible; No)
  . Set Property (CTRL(Label_aviso1); Visible; No)
  . Set Property (CTRL(Image_aviso1); Visible; No)
  . Set Property (CTRL(Shape_aviso1); Visible; No)
- Else
  . Set Property (CTRL(Label_aviso); Visible; Yes)
  . Set Property (CTRL(Image_aviso); Visible; Yes)
  . Set Property (CTRL(Shape_aviso); Visible; Yes)
  . Set Property (CTRL(Label_aviso1); Visible; Yes)
  . Set Property (CTRL(Image_aviso1); Visible; Yes)
  . Set Property (CTRL(Shape_aviso1); Visible; Yes)
End If

```

Este processo de contagem é também executado na navegação de cada página.

### 1.3.2.5. Label\_Log\_out



Ao selecionar este label é apresentado uma caixa de texto que questiona o utilizador se quer realmente efetuar logou.

Caso queiró é efetuado o log out para o formulário “Login” executando também a LAS “Delete\_log\_in”.

```
W Click
Message Box ("Atenção"; "Deseja sair da aplicação?"; Question; YES - NO; NO; TVAR(0))

- If (TVAR(0) = 7)
  . Exec. Local Action Set (Delete_log_in; No)
  . Show Form (Login; 0)
- Else
  . Cancel
End If

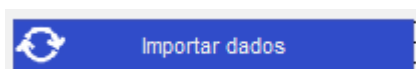
W Delete log in
Show Status Window (-1; -1; -1; -1; "048076201"; ""; "05 De

+ SQL Statement (Delete; log_in; 1=1 )
+ SQL Statement (Delete; Clientes; 1=1 )
+ SQL Statement (Delete; Fatura; 1=1 )
+ SQL Statement (Delete; Artigo; 1=1 )
+ SQL Statement (Delete; Artigo_Fatura; 1=1 )
+ SQL Statement (Delete; Artigo_pvp; 1=1 )
+ SQL Statement (Delete; Moedas; 1=1 )
+ SQL Statement (Delete; Pais; 1=1 )
+ SQL Statement (Delete; Series; 1=1 )
+ SQL Statement (Delete; SubFamilias_artg; 1=1 )
+ SQL Statement (Delete; Tipo_document; 1=1 )
+ SQL Statement (Delete; Artigo_imagens; 1=1 )

Close Status Window
```

Esta LAS apenas executa o *Delete* a todas as tabelas presentes no dispositivo de forma a preparar o mesmo para um novo login.

### 1.3.2.6. Import\_Data

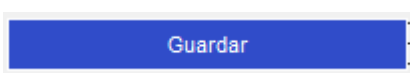


Este botão serve para forçar uma atualização de todos os pedidos da aplicação. Ao clicar é executada a LAS “Principais\_Pedidos” que efetua todos os pedidos necessários. Antes de efetuar este pedido é efetuado um “Ping” de forma a verificar a ligação à rede

sendo que caso seja negativo é apresentada uma caixa de texto que avisa o utilizador e cancela todo o processo.

```
Wtr Click
+ Ping ("google.com"; 4; 32; 5000; TVAR(ping_reply); TVAR(ping_time); "<13>"; TVAR(ping_time))
- If (TVAR(ping_reply) = 0)
. Message Box ("Atenção"; "Não se encontra liga a rede de momento. Verifique a sua conexão!"; Info)
- Else
. Show Status Window (-1; -1; -1; -1; "048076201"; "A Sincronizar Dados"; "OS Default"; 12; <none>;
. Exec. Local Action Set (Principais_Pedidos; No)
. Close Status Window
End If
```

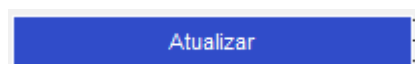
### 1.3.2.7. Save\_Menu



Este botão serve para atualizar as preferências na tabela “log\_in” consoante os valores que as combos e radio apresentam. Após guardar os valores é efetuada a navegação para o Plane 2.

```
Wtr Click
Show Status Window (-1; -1; -1; -1; "255249189"; "A preparar as suas preferencias";
+ SQL Statement (Update; log_in; No; Yes; catalog=CTRL(Radio_list); cliente=CTRL(Combo
//Consulta preferencias do user
+ SQL Statement (Select; log_in; LVAR(conta_log)=contabilidade; LVAR(cliente_log)=clie
//Preencher as preferencias do user
Set Selection (CTRL(Combo_entidades); By Returned Value; LVAR(cliente_log))
Set Selection (CTRL(Combo_entidades1); By Returned Value; LVAR(cliente_log))
Set Selection (CTRL(Combo_entidades2); By Returned Value; LVAR(cliente_log))
Set Selection (CTRL(Combo_tipo_mensal); By Returned Value; LVAR(conta_log))
Set Selection (CTRL(Combo_set_cliente); By Returned Value; LVAR(cliente_log))
Set Selection (CTRL(Combo_set_contabilidade); By Returned Value; LVAR(conta_log))
Set Selection (CTRL(Combo_set_idioma); By Returned Value; LVAR(idio_log))
+ If (CTRL(Combo_tipo_mensal) = 1)
Exec. Local Action Set (Dados_estatisticos_graph2; No)
+ Refresh Control (CTRL(HTMLViewer_chart); Refresh Value)
+ Refresh Control (CTRL(HTMLViewer_chart2); Refresh Value)
Close Status Window
Go To Plane (2)
```

### 1.3.2.8. Save\_Login



Este botão serve para a atualização das informações de “Url”, “Port” e “companhia”. Após serem alterados os valores nas combos, e selecionado este botão, são

atualizados os valores na tabela “log\_in” e efetua um novo login de forma a atualizar as informações correspondentes.

```

W Click
Show Status Window (-1; -1; -1; -1; "255249189"; "A preparar as suas preferencias"; "OS Default"
+ SQL Statement (Update; log_in; No; Yes; cliente=CTRL(Combo_set_cliente); contabilidade=CTRL(Comb

//Consulta preferencias do user
+ SQL Statement (Select; log_in; LVAR(company_log)=company; LVAR(url_log)=Url; FIELD(log_in,Id) =

- If (LVAR(company_log) = CTRL(Combo_set_company))
. Close Status Window
. Go To Plane (2)
- Else
+ . SQL Statement (Update; log_in; No; Yes; company=CTRL(Combo_set_company); FIELD(log_in,Id) = "
. Close Status Window
. Show Form (Login; 0)
End If

- If (LVAR(url_log) = CTRL(Input_Url_update))
. Close Status Window
. Go To Plane (2)
- Else
+ . SQL Statement (Update; log_in; No; Yes; Url=CTRL(Input_Url_update); FIELD(log_in,Id) = "1"
. Close Status Window
. Show Form (Login; 0)
End If

Close Status Window

```

### 1.3.3. Form: Form\_Clientes

#### 1.3.3.1. Form

```

W Open Form
Show Status Window (-1; -1; 30; 30; "048076201"; ""; "OS Default"; 12; <none>; Left; "00000000"; ""; "OS Default"; 12; <none>
Go To Plane (1)

Get Property (CTRL(Input_Search); <Value>; LVAR(insert_search))

//Tratamento dos dados de envio
- If (LVAR(insert_search) = "")
. Set Value (LVAR(insert_search); String; "%25")
- Else
End If
Set Value (LVAR(offset_rows); Numeric; 0)

//Get Value Count de clientes
+ Http Request (Run Locally; 0; TVAR(answer_count); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip); "WebAp
Set Value (LVAR(cliente_count); String; TVAR(answer_count))
+ JSON Get Value (LVAR(cliente_count); "Data"; Yes; TVAR(answer_items); TVAR(1))
Set Value (TVAR(answer_items); String; Left(TVAR(answer_items), Length(TVAR(answer_items)) - 1))
Set Value (TVAR(answer_items); String; Right(TVAR(answer_items), Length(TVAR(answer_items)) - 1))
+ JSON Get Value (TVAR(answer_items); "Column1"; Yes; LVAR(cliente_count); TVAR(1))

```

Ao navegar para o “Form\_Clientes”, e como é um formulário com mais do que um plane, é efetuada a navegação para o plane 1. É efetuada de seguida a recolha dos dados necessários para o pedido de *count* ao erp que é feito a seguir de forma a confirmar o número total de clientes existentes para esta companhia. O valor obtido é guardado em uma variável local.

```

//Envio dos dados
+ SQL Statement (Delete; Clientes; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_items); TVAR(header); TVAR(status); TVAR(text); Yes; "http:/
+ JSON Import to Table (Text; TVAR(answer_items); Clientes)
+ SQL Statement (Select; Clientes; LVAR(cliente_count_BD)=COUNT(id); ; No; Yes)

- If (LVAR(cliente_count) > 15)
. Set Property (CTRL(Image_right1); Visible; Yes)
- Else
. Set Property (CTRL(Image_right1); Visible; No)
End If

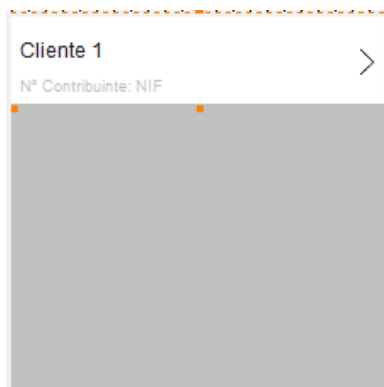
Set Value (LVAR(10); Numeric; 0)
+ Refresh Control (CTRL(Looper_list); Refresh List; Don't Refresh Sub-Lists; Thread Mode)

```

Recolhidos os dados, é enviado o primeiro pedido de paginação de clientes, retornando 15 cliente de cada vez. Caso o número total de clientes seja menor ou igual a 15 são desativados os botões de navegação da página.

É efetuado o refresh do *Looper* e atribuído o valor á variável de disponibilização de filtro.

#### 1.3.3.2. Looper\_list



Este *Looper* é conectado á tabela clientes e lista todos apresentando o nome e número de contribuinte. Através deste *Looper* é possível aceder ao detalhe de cada cliente.

```
"SELECT id, Nome, NumContrib FROM Clientes"
```

#### 1.3.3.1. Image right / Image Left



Nestes botões é possível efetuar a navegação entre paginações, incrementando (neste caso) o pedido de 15 em 15. São criadas 2 variáveis de navegação que efetuam a



leitura do número de clientes presentes na base de dados após cada pedido efetuando um incremento consoante este número.

```
W Click
Set Property (CTRL(Image_left1); Visible; Yes)

//Nova Paginação
+ SQL Statement (Delete; Clientes; 1=1)
  Set Value (LVAR(offset_rows); Numeric; LVAR(offset_rows) + 15)
+ Http Request (Run Locally; 0; TVAR(answer_items); TVAR(header); TVAR(status); TVAR(text); Y
+ JSON Import to Table (Text; TVAR(answer_items); Clientes)

+ SQL Statement (Select; Clientes; LVAR(Count_mais)=COUNT(id); ; No; Yes)
  Set Value (LVAR(cliente_count_BD); Numeric; LVAR(cliente_count_BD) + LVAR(Count_mais))

- If (LVAR(cliente_count_BD) = LVAR(cliente_count))
  . Set Property (CTRL(Image_right1); Visible; No)
- Else
  End If

+ Refresh Control (CTRL(Looper_list); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
```

É verificado se a paginação alcançou o número máximo de clientes presente no erp. Caso se verifique este caso é desativado o botão direito.

O Botão esquerdo funciona da mesma forma, mas de sentido inverso.

#### 1.3.3.2. Filtro Looper



Este botão força um pedido de paginação após o utilizador escolher efetuar este com recurso ao input de filtro que permite procurar clientes através do seu nome. Este funciona da mesma forma que os botões anteriores.

#### 1.3.3.3. Image\_goto

Esta imagem presente em cada linha do *looper* serve para efetuar a navegação para



o formulário “Form\_Cliente\_Detalhe” transportando o id do cliente para o mesmo.

```
W Click
Get Property (CTRL(Image_goto); 0; Hidden Value; TVAR(Id))
Show Form (Form_Cliente_Detalhe; 0; (LVAR(inID); String; TVAR(Id)))
```

### 1.3.4. Form: Form\_Cliente\_Detalhe

#### 1.3.4.1. Form

```
WV Open Form
Go To Plane (1)
+ Refresh (Refresh List; Yes; Thread Mode)

WV Close Form

WV After Open Form
//Preenche dados do cliente
+ SQL Statement (Select; Clientes; CTRL(Label_cod2)=id; CTRL(Label_entid

WV On Rotation
```

Ao navegar para o formulário “Form\_Cliente\_Detalhe” e com o Id do cliente proveniente do formulário anterior é efetuado um *Select* que preenche os dados do formulário com o cliente correspondente.

#### 1.3.4.2. Label morada/mail



Esta *label* em conjunto com o ícone de *maps* permite a navegação para a morada no *googlemaps*.

```
WV Click
Set Value (TVAR(string_maps); String; CTRL(Label_morada2))
Set Value (TVAR(string_maps); String; Replace(TVAR(string_maps), " ", "+"))
Set Value (TVAR(string_maps); String; Replace(TVAR(string_maps), ",", ""))
+ Shell Execute ("https://www.google.com/maps/search/?api=1&query=" + CTRL(Label_morada2); ""; "Open"; "")
```

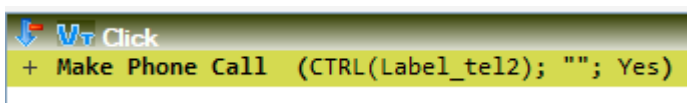
Retirando o valor e dados da morada do atual cliente e tratando a *string* da devida forma é possível efetuar um *Shell execute* para a api *googlemaps* que permite navegar para essa dada morada.

O email funciona da mesma forma, retirando o valor do *label* correspondente que a partir de um *Shell execute* permite a ligação direta de serviços de email do dispositivo.

#### 1.3.4.3. Label contacto



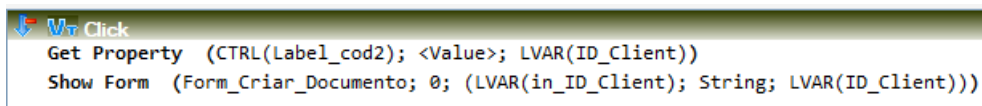
Ao seleccionar o contacto telefónico de um cliente será executada a ação de “*make a phonecall*” que irá abrir automaticamente o telefone do dispositivo pronto a efetuar a chamada.



#### 1.3.4.4. Image Vendas

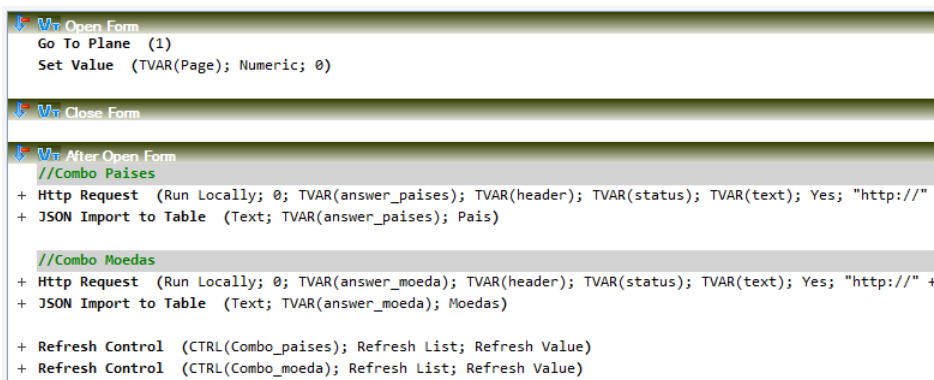


Este ícone permite uma navegação direta para o formulário “Form\_Criar\_Documento”, transportando o id do cliente atual para o mesmo.



### 1.3.5. Form: Criar\_Cliente

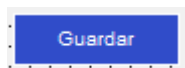
#### 1.3.5.1. Form



Após a navegação para este formulário é criada a variável “Page” que servirá para controlar a navegação entre os 2 planes que este formulário constitui.

Após a sua abertura, irão ser executados os pedidos de listar países e moedas de forma a popular as combos respetivas. Um *refresh* é posteriormente executado as mesmas.

#### 1.3.5.2. ImageButton\_Save



Após o preenchimento dos dois planes do formulário é necessário selecionar este botão que irá efetuar o pedido *Post* necessário para a conclusão da criação de um cliente.

```
Click
//Verifica se os campos obrigatorios foram preenchidos
Show Status Window (-1; -1; -1; -1; "255249189"; ""; "OS Default"; 12; <none>; Left
- If (CTRL(Combo_países) = "" OR CTRL(Input_telef) = "" OR CTRL(Combo_moeda) = "")
  . Close Status Window
  . Message Box ("Aviso"; "Tem de preencher todos os campos obrigatorios presentes r
  . Cancel
- Else
+ . SQL Statement (Insert; Clientes; No; Yes; CodPostal=CTRL(Input_codpost); Cont_Te
+ . JSON Export from Table (Text; ANSI/BINARY; TVAR(var_json); "SELECT id, Nome, Mor
+ . Http Request (Run Locally; 0; TVAR(7); TVAR(header); TVAR(status); TVAR(text); \
  . Return To Form (Form_Clientes)
End If

Close Status Window
```

De forma a criar o Json necessário para o pedido é primeiramente inserida na tabela os dados necessários para o mesmo, sendo posteriormente exportados para a sua construção.

Anteriormente ao pedido é verificado se todos os dados obrigatórios estão preenchidos. Este comportamento é também implementado no botão "Seguinte" do primeiro plane.

## 1.3.6. Form: Form\_Documentos

### 1.3.6.1. Form

```
W Open Form
Go To Plane (1)

Set Value (LVAR(3); Numeric; 0)
Show Status Window (-1; -1; 30; 30; "048076201"; ""; "OS Default"; 12;

Set Value (LVAR(doc_count_BD); Numeric; 0)
Looper Add Line (CTRL(Looper_list); 0)

//Filtro de data
Set Property (CTRL(Input_data_f); <Value>; DateSys())
```

Após a navegação do Form\_Documentos é definida o valor da data atual do dispositivo e adicionado no input de data\_f, servindo como data final *standart* da lista.

```
W After Open Form
Set Value (LVAR(Day); String; Day(DateSys()))
Set Value (LVAR(Month); String; Month(DateSys()))
Set Value (LVAR(Year); Numeric; Year(DateSys(), 4) - 1)
Set Property (CTRL(Input_data_i); <Value>; LVAR(Year) + LVAR(Month) + LVAR(Day))

Get Property (CTRL(Input_Search); <Value>; LVAR(insert_search))
Get Property (CTRL(Input_data_f); <Value>; LVAR(insert_dataf))
Get Property (CTRL(Input_data_i); <Value>; LVAR(insert_datai))
Set Value (LVAR(insert_dataf); String; Decode(LVAR(insert_dataf)))
Set Value (LVAR(insert_datai); String; Decode(LVAR(insert_datai)))

//Tratamento dos dados de envio
- If (LVAR(insert_search) = "")
. Set Value (LVAR(insert_search); String; "%25")
- Else
End If
Set Value (LVAR(offset_rows); Numeric; 0)
```

De seguida é definida a data inicial de filtragem que por *default* será a data anual menos 1 ano. São também recolhidos os dados necessário para os pedidos de paginação seguintes.

```

//Get Value Count de documentos
+ Http Request (Run Locally; 0; TVAR(answer_count); TVAR(header); TVAR(status); TVAR(text);
Set Value (LVAR(doc_count); String; TVAR(answer_count))
+ JSON Get Value (LVAR(doc_count); "Data"; Yes; LVAR(doc_count); TVAR(1))
Set Value (LVAR(doc_count); String; Left(LVAR(doc_count), Length(LVAR(doc_count)) - 1))
Set Value (LVAR(doc_count); String; Right(LVAR(doc_count), Length(LVAR(doc_count)) - 1))
+ JSON Get Value (LVAR(doc_count); "Column1"; Yes; LVAR(doc_count); TVAR(1))

//Envio dos dados
+ SQL Statement (Delete; Fatura; 1=1)
+ Http Request (Run Locally; 0; TVAR(answer_count); TVAR(header); TVAR(status); TVAR(text);
+ JSON Import to Table (Text; TVAR(answer_count); Fatura)
+ SQL Statement (Select; Fatura; LVAR(doc_count_BD)=COUNT(Doc_Id); ; No; Yes)

```

O primeiro pedido é o pedido *count* que é feito de forma a confirmar o número total de documentos existentes para esta companhia. O valor obtido é guardado em uma variável local.

```

- If (LVAR(doc_count) > 25)
. Set Property (CTRL(Image_right1); Visible; Yes)
- Else
. Set Property (CTRL(Image_right1); Visible; No)
End If

+ Refresh Control (CTRL(Looper_list); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
Close Status Window

```

Por fim é verificado se o número de documento é igual ou inferior a 25, que caso se verifique são desativados os botões de navegação.

### 1.3.6.2. Image\_search



Este botão serve para abrir as opções do filtro da listagem de documentos. Ao selecionar este botão é executado o código que torna visível todas as opções e ajusta a posição e tamanho do *looper*.

```
W Click
- If (LVAR(10) = 0)
. Set Value (LVAR(10); Numeric; 1)

. Set Property (CTRL(Looper_list); Height; 298)
. Set Property (CTRL(Looper_list); Y; 90)

. Set Property (CTRL(Input_Search); Visible; Yes)
. Set Property (CTRL(DEL-Shape_line); Visible; Yes)
. Set Property (CTRL(Image_calendar); Visible; Yes)
. Set Property (CTRL(Input_data_i); Visible; Yes)
. Set Property (CTRL(Input_data_f); Visible; Yes)
. Set Property (CTRL(Label_ate1); Visible; Yes)

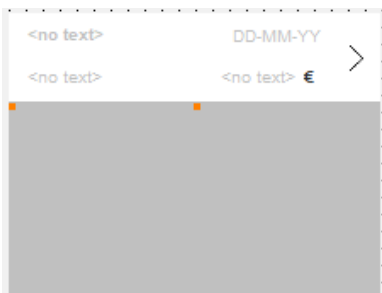
- Else
. Set Value (LVAR(10); Numeric; 0)

. Set Property (CTRL(Input_Search); Visible; No)
. Set Property (CTRL(DEL-Shape_line); Visible; No)
. Set Property (CTRL(Image_calendar); Visible; No)
. Set Property (CTRL(Input_data_i); Visible; No)
. Set Property (CTRL(Input_data_f); Visible; No)
. Set Property (CTRL(Label_ate1); Visible; No)

. Set Property (CTRL(Looper_list); Height; 343)
. Set Property (CTRL(Looper_list); Y; 45)

End If
```

### 1.3.6.3. Looper List



Este *looper* é conectado à tabela de documentos e apresenta as informações básicas dos mesmos. O *looper* apresenta o tipo documento, entidade, valor total e data do mesmo. Através deste *looper* é possível aceder ao detalhe de cada documento.

```
"SELECT Document, Entidade, Data_cria, Valor_Ac, Doc_Id
FROM Fatura
ORDER BY Data_cria DESC"
```

#### 1.3.6.4. Image right/ Image Left



Nestes botões é possível efetuar a navegação entre paginações, incrementando (neste caso) o pedido de 25 em 25. São criadas 2 variáveis de navegação que efetuam a leitura do número de documentos presentes na base de dados após cada pedido efetuando um incremento consoante este número.

```

Click
Set Property (CTRL(Image_left1); Visible; Yes)

//Nova Paginação
+ SQL Statement (Delete; Fatura; 1=1)
  Set Value (LVAR(offset_rows); Numeric; LVAR(offset_rows) + 25)
+ Http Request (Run Locally; 0; TVAR(answer_count); TVAR(header); TVAR(status); TVAR(text); Ye
+ JSON Import to Table (Text; TVAR(answer_count); Fatura)

+ SQL Statement (Select; Fatura; LVAR(Count_mais)=COUNT(Doc_Id); ; No; Yes)
  Set Value (LVAR(doc_count_BD); Numeric; LVAR(doc_count_BD) + LVAR(Count_mais))

- If (LVAR(doc_count_BD) = LVAR(doc_count))
  . Set Property (CTRL(Image_right1); Visible; No)
- Else
  End If

+ Refresh Control (CTRL(Looper_list); Refresh List; Don't Refresh Sub-Lists; Thread Mode)

```

É verificado se a paginação alcançou o número máximo de documentos presente no erp. Caso se verifique este caso é desativado o botão direito.

O Botão esquerdo funciona da mesma forma, mas de sentido inverso.

#### 1.3.6.5. Filtro Looper

Este botão força um pedido de paginação após o utilizador escolher efetuar este com



recurso ao input de filtro que permite procurar clientes através do seu nome e intervalo de datas. Este funciona da mesma forma que os botões anteriores.



Os inputs de tipo data funcionam com recurso ao sistema android que apresenta o intervalo de datas em formato de calendário.

### 1.3.7. Form: Form\_Documento\_Detalhe

#### 1.3.7.1. Form

```
VT Open Form
Set Value (LVAR(99); Numeric; 1)
Go To Plane (1)

VT Close Form

VT After Open Form
//Preenche dados do documento de venda
+ SQL Statement (Select; Fatura; CTRL(input_data)=Data_cria; CTRL(input_data1)=Data_venc; CTRL(Label_doc2)=ID)
+ SQL Statement (Select; Clientes; CTRL(Label_numcont2)=NumContrib; FIELD(Clientes,id) = CTRL(Label_ent2); I)
+ Refresh Control (CTRL(Looper_list_arts); Refresh List; Don't Refresh Sub-Lists; Thread Mode)

+ SQL Statement (Select; Fatura; CTRL(Label_total3)=Valor_Ac; FIELD(Fatura,Doc_Id) = LVAR(inID); No; Yes)
```

Ao navegar para este form é efetuada a navegação para o plane 1 pois este form contem mais que um plane. De seguida é efetuado o preenchimento do documento que foi selecionado no form “Form\_Documentos” através de transporte do id do documento listado.

#### 1.3.7.2. Image Share



Através deste ícone é possível abrir o menu de partilha do dispositivo, permitindo partilhar o documento presente por vários meios disponíveis.

```
VT Click
+ Http Request (Run Locally; 0; TVAR(answer_file); TVAR(header); TVAR(status); TVAR(text); Yes; "h
+ File Save Content (DFOLDER + "\Anexos\DocumentoVenda.pdf"; TVAR(answer_file); ANSI/BINARY; Yes)
+ Run Process (Don't run on Windows; Run on Android; "android.intent.action.SEND"; DFOLDER + "\Ane
```

É efetuado primeiramente um pedido à WebApi de forma a retornar o documento em formato PDF.

### 1.3.7.3. Image Download



Ao selecionar este ícone é feito o mesmo pedido de formato PDF à WebApi tal como o ícone de partilha anterior, diferenciando no sentido que ao invés de partilhar o documento é guardado diretamente no dispositivo.

```
W Click
+ Http Request (Run Locally; 0; TVAR(answer_file); TVAR(header); TVAR(stat
+ File Open (1; PFOLDER + "\\temp.pdf"; Write; ANSI/BINARY)
+ File Write (1; No; "<13><10>"; TVAR(answer_file))
+ File Close (1)
+ Shell Execute (PFOLDER + "\\temp.pdf"; ""; "Open"; "")
```

### 1.3.7.4. Looper list artg



No plane 2 é disponibilizado a lista de artigos associados ao documento presente, sendo apresentados dentro de um *looper* que se encontra ligado à tabela “Artigo\_Fatura”. Através deste *looper* é possível aceder ao detalhe de cada artigo.

```
"SELECT Cod_prod, Valor, Qtd, Desconto FROM Artigo_Fatura WHERE Cod_Id LIKE '%' + LVAR(inID) + '%'"
```

## 1.3.8. Form: Form\_Criar\_Documento

### 1.3.8.1. Form

```
W Open Form
Go To Plane (1)
Set Value (LVAR(Nav_Plane); Numeric; 1)

- If (VAR(Ping_reply) = 0)
. Message Box ("Atenção"; "Não se encontra ligado a rede de momento, qualquer
. Set Property (CTRL(Image_complete_send); State; Disabled)
. Set Property (CTRL(Image_complete_send); Background Color; "192192192")
- Else
. Set Property (CTRL(Image_complete_send); State; Editable)
. Set Property (CTRL(Image_complete_send); Background Color; "048076201")
End If
```

Ao navegar para este form é verificada a ligação à rede. Caso não haja ligação é mostrada uma caixa de texto que avisa o utilizador da situação e é desativado o botão de submissão do documento.

É também criada a variável “Nav\_Plane” de forma a auxiliar a navegação entre planes, contendo o valor do plane atual. (ex. Plane 1 -> Nav\_Plane = 1).

```
Set Value (LVAR(Json_FilledData); String; "")

//Documento a editar ou nao
- If (LVAR(Editar_doc) = 1)
. Set Value (CTRL(Label_Title); String; "Editar Documento")

. Set Property (CTRL(Combo_serie1); State; Editable)
. Exec. Local Action Set (Enter_Doc_editar; No)

. //Transporte de campos de documento já existente para editar
+ . SQL Statement (Select; Doc_Save; TVAR(select_entidade)=Entidade; TVAR(select_ser:
. Set Selection (CTRL(Combo_doc1); By Returned Value; TVAR(select_doc))
+ . Refresh Control (CTRL(Combo_serie1); Refresh List; Refresh Value)
. Set Selection (CTRL(Combo_serie1); By Returned Value; TVAR(select_serie))
. Set Selection (CTRL(Combo_entidade1); By Returned Value; TVAR(select_entidade))
+ . SQL Statement (Select; Clientes; CTRL(Input_pais1)=Pais; CTRL(Input_mor1)=Morada;

- Else
. Set Value (CTRL(Label_Title); String; "Novo Documento")
. Set Property (CTRL(Combo_serie1); State; Disabled)
. Exec. Local Action Set (Enter_Doc; No)
End If
```

De seguida é criada a variável local “Json\_FilledData” que será utilizada para efetuar o primeiro pedido de criação de documento à WebApi.

Após isto é verificado se a página de criação de documento está a ser acedida com a intenção de editar um documento já existente ou se é intenção criar um novo. Caso seja para editar é compreendido que seja já transportado o Id do Documento de forma a efetuar um *Select* à tabela “Doc\_Save”. É também executada a LAS “Enter\_Doc\_editar”, caso não seja para editar é executada a LAS “Enter\_Doc”.

```

Wm Enter_Doc
//Cria GUID de novo Documento de Venda e cria nova instancia na Base de dados
Set Value (LVAR(GUID); String; GetGUID())
+ SQL Statement (Insert; Fatura; No; Yes; Doc_Kalipso=LVAR(GUID))
+ SQL Statement (Insert; Json_Doc; No; Yes; Kalipso_doc=LVAR(GUID))

//Preenche Base de Dados para o Combo de Tipo de Documentos
+ SQL Statement (Select; Tipo_document; TVAR(Chack_tipo)=Tipo; ; No; Yes)
- If (TVAR(Chack_tipo) = 0)
+ . Http Request (Run Locally; 0; TVAR(answer_type); TVAR(header); TVAR(status); TVAR(
+ . JSON Get Value (TVAR(answer_type); "DataSet"; Yes; TVAR(20); TVAR(21))
- Else
End If

//Preenche Base de Dados para o Combo de Series
+ SQL Statement (Select; Series; TVAR(Check_serie)=Serie; ; No; Yes)
- If (TVAR(Check_serie) = 0)
+ . SQL Statement (Delete; Series; 1=1)
+ . JSON Import to Table (Text; TVAR(20); Tipo_document)
+ . Http Request (Run Locally; 0; TVAR(answer_serie); TVAR(header); TVAR(status); TV
+ . JSON Import to Table (Text; TVAR(answer_serie); Series)
- Else
End If

+ Refresh Control (CTRL(Combo_doc1); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_entidade1); Refresh List; Refresh Value)

```

Caso o utilizador pretenda criar um novo documento será executada a LAS “Enter\_Doc”. Esta LAS começa por criar uma nova *guid* que irá associar ao novo documento criado, criando de seguida uma instância na tabela “Fatura” e “Json\_Doc”. São também efetuados os pedidos à WebApi para o preenchimento das combos com os dados de tipo de documentos e series.

```

Wm Enter_Doc_editar

//Preenche Base de Dados para o Combo de Tipo de Documentos
+ SQL Statement (Select; Tipo_document; TVAR(Check_tipo)=Tipo; ; No; Yes)
+ If (TVAR(Check_tipo) = 0)

//Preenche Base de Dados para o Combo de Series
+ SQL Statement (Select; Series; TVAR(Check_serie)=Serie; FIELD(Serie,TipoDoc) = CTRL(Combo_doc1); No; Yes)
+ If (TVAR(Check_serie) = 0)

+ Refresh Control (CTRL(Combo_doc1); Refresh List; Refresh Value)
+ Refresh Control (CTRL(Combo_entidade1); Refresh List; Refresh Value)

+ Refresh Control (CTRL(Looper_list_arts); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
+ SQL Statement (Select; Artigo_Fatura; TVAR(art_cod)=SUM(Value); FIELD(Artigo_Fatura,Cod_Kalipso) = LVAR(GUID); No; Yes)
Set Property (CTRL(Label_total3); <Value>; TVAR(art_cod))

```

A LAS “Enter\_Doc\_editar” efetua as mesmas ações da LAS anterior retirando a criação de uma nova *guid* e acrescentando um *Select* que retira os dados todos existentes do atual documento para o preenchimento dos campos correspondentes.

```

Vt After Open Form
//Verifica se a página esta a ser acedida através da página Detalhe de Cliente
- If (LVAR(in_ID_Cliente) = "")
- Else
. //Preenche campos presentes do Documento com os dados do cliente
+ . SQL Statement (Select; Clientes; CTRL(Combo_entidade1)=id; CTRL(Input_pais1)=
End If

```

Após a abertura do form é verificado se este também esta a ser acedido através do form “Form\_Cliente\_Detalhe” que compreendendo o Id proveniente do mesmo, é efetuado um *Select* com os dados correspondentes.

### 1.3.8.2. Image Save



Caso o utilizador pretenda apenas guardar o documento no dispositivo, pode fazê-lo selecionando este ícone que irá guardar. É verificado primeiro se este documento já existe.

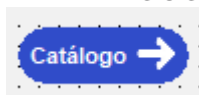
Caso não exista é verificado se a entidade foi inserida de forma manual no formulário.

```

Vt Click
- If (CTRL(Combo_doc1) = "" OR CTRL(Combo_serie1) = "")
. Message Box ("Aviso"; "Tem de preencher o tipo documento, serie e a entidade para selecionar um artigo."; Exclam
- Else
. //Validação da VAR Json
+ . SQL Statement (Select; Doc_Save; TVAR(0)=COUNT(Cod_Kalipso); FIELD(Doc_Save,Cod_Kalipso) = LVAR(GUID); No; Yes)
- . If (TVAR(0) = 0)
- . . If (LVAR(Manual_auto) = 1)
- . . . //Inser e cria "FillRelatedData"
+ . . . SQL Statement (Insert; Doc_Save; No; Yes; Cod_Kalipso=LVAR(GUID); Data_cria=DateSys(); Data_edit=DateSys().
+ . . . JSON Export from Table (Text; ANSI/BINARY; TVAR(Json_Venda); "SELECT Entidade, Serie, Tipo_Doc, TipoEntida
- . . . Return To Form (Form_Documentos)
- . . Else
- . . . //Inser e cria "FillRelatedData"
+ . . . SQL Statement (Insert; Doc_Save; No; Yes; Cod_Kalipso=LVAR(GUID); Data_cria=DateSys(); Data_edit=DateSys().
+ . . . JSON Export from Table (Text; ANSI/BINARY; TVAR(Json_Venda); "SELECT Entidade, Serie, Tipo_Doc, TipoEntida
- . . . Return To Form (Form_Documentos)
- . . End If
- . Else
+ . . SQL Statement (Update; Doc_Save; No; Yes; Data_edit=DateSys(); Entidade=CTRL(Combo_entidade1); Hora_edit=Time!
- . . Return To Form (Form_Documentos_guardados)
- . End If
End If

```

### 1.3.8.3. Image Linhas



Após o preenchimento do cabeçalho do documento o utilizador deverá seleccionar as linhas do mesmo. Esta ação é feita através da seleção do botão “Catálogo” que irá mostrar um *Pop-up* ao utilizador de forma a dar a escolha de inserir os artigos manualmente ou por catálogo.

```
Click
Show Popup ("Ver catálogo"; "Inserir Manualmente")
- If (POPUP = 1)
  . Set Value (LVAR(POPUP_var); Numeric; 1)
- . If (VAR(Ping_reply) = 0)
  . . Exec. Local Action Set (Criar_Doc_offline; No)
- . Else
  . . Exec. Local Action Set (Criar_Doc_online; No)
  . End If

- Else If (POPUP = 2)
  . Set Value (LVAR(POPUP_var); Numeric; 2)
- . If (VAR(Ping_reply) = 0)
  . . Exec. Local Action Set (Criar_Doc_offline; No)
- . Else
  . . Exec. Local Action Set (Criar_Doc_online; No)
  . End If

End If
```

Em ambos existe o cenário de o dispositivo estar ligado à rede ou não. Caso não haja ligação é executada a LAS “Criar\_Doc\_offline”, caso contrário é executada a “Criar\_Doc\_online”.

```
Criar_Doc_offline
+ SQL Statement (Update; Fatura; No; Yes; Entidade=CTRL(Input_entidade_man); Entidade_manual=LVAR(Manual_auto); FIELD(Fat
+ SQL Statement (Select; Fatura; TVAR(0)=Entidade_manual; FIELD(Fatura,Doc_Kalipso) = LVAR(GUID); No; Yes)
//SIM
- If (TVAR(0) = 1)
- . If (CTRL(Combo_doc1) = "" OR CTRL(Combo_serie1) = "" OR CTRL(Input_entidade_man) = "")
  . . Message Box ("Aviso"; "Tem de preencher o tipo documento, serie e a entidade para seleccionar um artigo."); Exclama
- . Else
  . //Validação da VAR Json
+ . . SQL Statement (Select; Doc_Save; TVAR(0)=COUNT(Cod_Kalipso); FIELD(Doc_Save,Cod_Kalipso) = LVAR(GUID); No; Yes)
```

A versão offline do criar doc é feito um *Update* do documento atual com os dados correspondentes e é verificado se a entidade foi inserida de forma manual ou a através de combo. Caso tenha sido inserido de forma manual é primeiramente verificado se os inputs obrigatórios estejam preenchidos, pois caso não estejam o processo é cancelado.

Caso contrário estes dados são guardados na tabela correspondente.

```

- . . . If (TVAR(0) = 0)
- . . . //Inser e cria "FillRelatedData"
- . . . . If (LVAR(Manual_auto) = 1)
+ . . . . . SQL Statement (Insert; Doc_Save; No; Yes; Cod_Kalipso=LVAR(GUID); Data_cria=DateSys(); Entida
+ . . . . . JSON Export from Table (Text; ANSI/BINARY; TVAR(Json_Venda); "SELECT Entidade, Serie, Tipo_Do

- . . . . Else
+ . . . . . SQL Statement (Insert; Doc_Save; No; Yes; Cod_Kalipso=LVAR(GUID); Data_cria=DateSys(); Entida
+ . . . . . JSON Export from Table (Text; ANSI/BINARY; TVAR(Json_Venda); "SELECT Entidade, Serie, Tipo_Do

. . . . End If
. . . . Set Value (LVAR(Nav_Plane); Numeric; 2)
. . . . Go To Plane (2)

- . . . Else
. . . . Set Value (LVAR(Nav_Plane); Numeric; 2)
. . . . Go To Plane (2)
+ . . . Refresh Control (CTRL(Looper_list_arts); Refresh List; Don't Refresh Sub-Lists; Thread Mode)

. . End If
. End If

```

Caso a entidade tenha sido inserida através de combo é criada a estrutura necessária para o pedido de "FillRelatedData" na tabela "Doc\_Save".

```

- Else
- //NÃO
- . If (CTRL(Combo_doc1) = "" OR CTRL(Combo_serie1) = "" OR CTRL(Combo_entidade1) = "")
- . . Message Box ("Aviso"; "Tem de preencher o tipo documento, serie e a entidade para selecionar um

- Else
- //Validação da VAR Json
+ . . SQL Statement (Select; Doc_Save; TVAR(0)=COUNT(Cod_Kalipso); FIELD(Doc_Save,Cod_Kalipso) = LVAR(

- . . If (TVAR(0) = 0)
- . . . //Inser e cria "FillRelatedData"
+ . . . SQL Statement (Insert; Doc_Save; No; Yes; Cod_Kalipso=LVAR(GUID); Data_cria=DateSys(); Entida
+ . . . JSON Export from Table (Text; ANSI/BINARY; TVAR(Json_Venda); "SELECT Entidade, Serie, Tipo_Do

. . . . Set Value (LVAR(Nav_Plane); Numeric; 2)
. . . . Go To Plane (2)

- . . Else
. . . Set Value (LVAR(Nav_Plane); Numeric; 2)
. . . Go To Plane (2)
+ . . Refresh Control (CTRL(Looper_list_arts); Refresh List; Don't Refresh Sub-Lists; Thread Mode)

. . End If
. End If

End If

```

```

+ SQL Statement (Update; Fatura; No; Yes; Entidade=CTRL(Input_entidade_man); Entidade_manual=LVAR(Manual_auto); FIELD(Fatura,Doc
+ SQL Statement (Select; Fatura; TVAR(0)=Entidade_manual; FIELD(Fatura,Doc_Kalipso) = LVAR(GUID); No; Yes)

//Entidade inserida manualmente?
//SIM
- If (TVAR(0) = 1)
+ . SQL Statement (Select; Fatura; TVAR(1)=Entidade; FIELD(Fatura,Doc_Kalipso) = LVAR(GUID); No; Yes)
+ . Http Request (Run Locally; 0; TVAR(Json_answer_1); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip); "v
. //Validar entidade inserida
- . If (TVAR(Json_answer_1) = "true")
- . Else
. . Message Box ("Atenção"; "A entidade que inseriu não existe, insira uma entidade válida."; Exclamation; OK; OK; )
. . Cancel
. End If

```

Em situação com acesso à rede é feita a atualização aos campos respetivos da tabela “Fatura”. Após é verificado também se a entidade foi inserida manualmente.

Caso tenha sido inserida de forma manual é feito um pedido à WebApi de forma a validar se a entidade existe no ERP. Caso não exista é mostrada uma caixa de texto de forma a indicar a situação ao utilizador.

```

- . . If (CTRL(Combo_doc1) = "" OR CTRL(Combo_serie1) = "" OR CTRL(Input_entidade_man) = "")
. . . Message Box ("Aviso"; "Tem de preencher o tipo documento, serie e a entidade para seleccionar um artigo."; Exclamation; OK; OK; )
- . . Else
. . . //Validação da VAR Json
+ . . . SQL Statement (Select; Json_Doc; LVAR(Json_FilledData)=COUNT(Json_doc); FIELD(Json_Doc,Kalipso_doc) = LVAR(GUID); No; Yes)
- . . . If (LVAR(Json_FilledData) = 0)
. . . . //Inserir e cria "FillRelatedData"
+ . . . . SQL Statement (Delete; Novo_Doc; 1=1)
+ . . . . SQL Statement (Insert; Novo_Doc; No; Yes; Entidade=CTRL(Input_entidade_man); Serie=CTRL(Combo_serie1); Tipo_Doc=CTRL(Combo_doc
+ . . . . JSON Export from Table (Text; ANSI/BINARY; TVAR(Json_Venda); "SELECT Entidade, Serie, Tipo_Doc, TipoEntidade FROM Novo_Doc WHE
+ . . . . Http Request (Run Locally; 0; TVAR(Json_answer_1); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip); "WebAp
. . . . //Retirar atributos do objeto "TaxFree" do JSON
. . . . Set Value (TVAR(StringCount); Numeric; Length(TVAR(Json_answer_1)))
. . . . Set Value (TVAR(First); Numeric; Find(TVAR(Json_answer_1), "" + DQUOTE + "TaxFree" + DQUOTE + ":{", 1) + 10)
. . . . Set Value (TVAR(Second); Numeric; Find(TVAR(Json_answer_1), "},"), TVAR(First)) - 1)
. . . . Set Value (TVAR(New_answer); String; Left(TVAR(Json_answer_1), TVAR(First)) + DQUOTE + DQUOTE + Right(TVAR(Json_answer_1), TVA
. . . . Set Value (LVAR(Json_FilledData); String; TVAR(New_answer))
+ . . . . SQL Statement (Update; Json_Doc; No; Yes; Json_doc=LVAR(Json_FilledData); FIELD(Json_Doc,Kalipso_doc) = LVAR(GUID))
+ . . . . If (LVAR(POPUP_var) = 1)
- . . . . Else
+ . . . . If (LVAR(POPUP_var) = 1)
. . . . End If
. End If

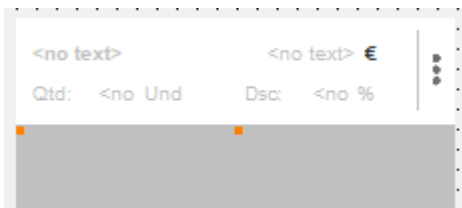
```

Após isto é verificado se os inputs obrigatórios estão preenchidos. Caso esteja tudo preenchido é criado o Json para o “FillRelatedData” que é posteriormente enviado com o pedido à WebApi que cria o cabeçalho.



Neste caso o Json é retornado com um componente extra chamado de “TaxFree” que entra em conflito com os restantes pedidos. De forma a contornar isto é feito o tratamento do mesmo de forma a corrigir o problema.

#### 1.3.8.4. Looper list arts



No plane 2 é disponibilizado um *looper* que apresenta uma lista que servirá para mostrar os artigos seleccionados. Cada linha do *looper* apresenta a opção de editar ou eliminar o artigo correspondente.

```
"SELECT Cod_prod, Valor, Qtd, Desconto, artigo_guid FROM Artigo_Fatura WHERE Cod_Kalipso LIKE '%" + LVAR(GUID) + "%'"
```

#### 1.3.8.5. Image Cod bars



A partir deste ícone o utilizador consegue listar o artigo a partir de um leitor de código de barras.

```
W Click  
+ Barcode Scan (TVAR(art_cod); TVAR(art_type); 5000; Yes)  
  
//Indica que navega para a pagina Catalogo atraves da pagina de criação de documento  
Set Value (LVAR(venda); String; "2")  
Show Form (Form_Artigo_Detalhe; 0; (LVAR(venda); String; LVAR(venda)); (LVAR(Art_cod); String  
  
//Refresh de controlos da pagina de criação do documento  
+ Refresh Control (CTRL(Looper_list_arts); Refresh List; Don't Refresh Sub-Lists; Thread Mode)  
+ SQL Statement (Select; Artigo_Fatura; TVAR(art_cod)=SUM(Valor); FIELD(Artigo_Fatura,Cod_Kalip  
Set Property (CTRL(Label_total3); <Value>; TVAR(art_cod))
```

A partir do código de barras obtido é feito um *Select* automático que irá procurar o artigo pretendido.

### 1.3.8.6. Image Add



É através deste ícone que o utilizador navega para o form “Form\_Catalogo”. Após a navegação é feito um ciclo “for each” que calcula o valor total dos artigos de forma a apresentar no fundo do form.

```
Click
//Indica que navega para a pagina Catalogo atraves da pagina de criação de documento
Set Value (LVAR(venda); String; "1")
Show Form (Form_Catalogo; 0; (LVAR(venda); String; LVAR(venda) ); (LVAR(GUID_in); String; LVAR(GUID)); (LVAR(En

//Refresh de controlos da pagina de criação do documento
+ Refresh Control (CTRL(Looper_list_arts); Refresh List; Don't Refresh Sub-Lists; Thread Mode)

- For Each (Select; Artigo_Fatura; TVAR(codigo_for)=Cod_prod; FIELD(Artigo_Fatura,Cod_Kalipso) = LVAR(GUID); No)
. Set Value (TVAR(Valor); Numeric; 0)

+ . SQL Statement (Select; Artigo_Fatura; TVAR(Valor)=Valor; FIELD(Artigo_Fatura,Cod_Kalipso) = LVAR(GUID) AND F:
+ . SQL Statement (Select; Artigo_Fatura; TVAR(Desconto)=Desconto; FIELD(Artigo_Fatura,Cod_Kalipso) = LVAR(GUID)

. Set Value (TVAR(Desconto); Numeric; TVAR(Desconto) / 100)
. Set Value (TVAR(Valor_desc); Numeric; TVAR(Valor) * TVAR(Desconto))
. Set Value (TVAR(Valor); Numeric; TVAR(Valor) - TVAR(Valor_desc))

. Set Value (TVAR(Valor_total); Numeric; TVAR(Valor_total) + TVAR(Valor))

+ End For Each

Set Property (CTRL(Label_total3); <Value>; TVAR(Valor_total))
```

### 1.3.8.7. Looper list arts man / Image Add



O conjunto destes 3 inputs, botão de adicionar e *looper* constrói o mecanismo de adicionar artigos de forma manual.

O botão *Add* adiciona os dados do artigo através dos inputs atribuindo um *guid* a cada um. Após isto é efetuado um *refresh* que atualiza o *looper* presente, e apagando os dados anteriormente inseridos.

```
"SELECT Cod_prod, Qtd, Desconto, artigo_guid FROM Artigo_Fatura WHERE Cod_Kalipso LIKE '%" + LVAR(GUID) +
"%"
```

### 1.3.8.8. Image Complete Send



Após a conclusão do preenchimento dos campos do documento o utilizador deverá selecionar este botão. Após carregar no botão é verificado novamente se os inputs obrigatórios se encontram preenchidos.

```
W Click
- If (CTRL(Combo_doc1) = "" OR CTRL(Input_Numcont1) = "")
  . Message Box ("Aviso"; "Tem de preencher todos os campos obrigatorios presentes no formulário."; Information; OK; OK; )
  . Cancel
- Else
+ . SQL Statement (Select; Artigo_Fatura; TVAR(prod_present)=COUNT(Cod_Kalipso); FIELD(Artigo_Fatura,Cod_Kalipso) = LVAR(GUID); N
- . If (TVAR(prod_present) = 0)
  . Message Box ("Aviso"; "Não tem Produtos selecionados para este documento."; Information; OK; OK; )
  . Cancel
- . Else
+ . SQL Statement (Select; Json_Doc; TVAR(Json_enviar)=Json_doc; FIELD(Json_Doc,Kalipso_doc) = LVAR(GUID); No; Yes)

  //Loop verifica e insere "Adicionalinha"
- . . . For Each (Select; Artigo_Fatura; TVAR(select_doc)=Cod_prod; TVAR(select_qtd)=Qtd; TVAR(select_armazem)=Armazem; TVAR(select
+ . . . . If (TVAR(select_manual) = 1)

+ . . . . Http Request (Run Locally; 0; TVAR(Json_answer_2); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip);
  . . . . //Retirar atributos do objeto "TaxFree" do JSON
  . . . . Set Value (TVAR(StringCount); Numeric; Length(TVAR(Json_answer_2)))
  . . . . Set Value (TVAR(First); Numeric; Find(TVAR(Json_answer_2), "" + DQUOTE + "TaxFree" + DQUOTE + ":{", 1) + 10)
  . . . . Set Value (TVAR(Second); Numeric; Find(TVAR(Json_answer_2), ",", TVAR(First)) - 1)
  . . . . Set Value (TVAR(New_answer_2); String; Left(TVAR(Json_answer_2), TVAR(First)) + DQUOTE + DQUOTE + Right(TVAR(Json_answer
  . . . . Set Value (TVAR(Json_enviar); String; TVAR(New_answer_2))

+ . . . . End For Each
```

Após isto é efetuado um *count* á tabela de “Artigo\_Fatura” aos artigos que correspondem ao documento criado. Após isto é feito um ciclo “*for each*” que efetua o pedido de “Adicionalinha” a cada um dos artigos presentes.

No final é feito um último *Post* com a resposta dos últimos dois pedidos efetuados de forma a concluir a criação do documento.

```
  . . . //POST de "CreateDocument"
+ . . . Http Request (Run Locally; 0; TVAR(Json_answer_3); TVAR(header); TVAR(status); TVAR(text); Y
  . . . Return To Form (Form_Documentos)

. End If
End If
```

## 1.3.9. Form: Documentos\_guardados

### 1.3.9.1. Form

```
W Open Form
Go To Plane (1)

Show Status Window (-1; -1; 30; 30; "048076201"; ""; "OS Default"; 12; <none>; Left; "00

Set Value (LVAR(10); Numeric; 0)
Looper Add Line (CTRL(Looper_list); 0)

//Filtro de data
Set Property (CTRL(Input_data_i); <Value>; "20190101")
Set Property (CTRL(Input_data_f); <Value>; DateSys())

+ Refresh Control (CTRL(Looper_list); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
```

Ao navegar para este form é efetuado o mesmo processo de form “Form\_Documentos”.

### 1.3.9.2. Image Search



O processo utilizado para este ícone é o mesmo indicado no form “Form\_Documentos”.

### 1.3.9.3. Image Send



Selecionando este ícone é executado o código de envio de todos os documentos guardados na tabela “Doc\_Save”. Inicialmente é mostrada uma caixa de texto para o utilizador validar que realmente quer enviar todos os documentos guardados.

```
W Click
Message Box (""; "Deseja integrar todos os documentos?"; Question; YES - NO; YES; TVAR(0))

- If (TVAR(0) = 7)
. Show Status Window (-1; -1; -1; -1; "255249189"; ""; "OS Default"; 12; <none>; Left; "00000000"; ""; "OS Default"; 12; <none>)
+ . SQL Statement (Select; Doc_Save; TVAR(Count_docs)=COUNT(Cod_Kalipso); ; No; Yes)

- . For Each (Select; Doc_Save; TVAR(Select_codKalip)=Cod_Kalipso; ; No)
. //Inserir e cria "FillRelatedData"
+ . . JSON Export from Table (Text; ANSI/BINARY; TVAR(Json_Venda); "SELECT Entidade, Serie, Tipo_doc, Tipo_entidade FROM Doc_Save
+ . . Http Request (Run Locally; 0; TVAR(Json_answer_1); TVAR(header); TVAR(status); TVAR(text); Yes; "http://" + VAR(pc_ip); "We

. //Retirar atributos do objeto "TaxFree" do JSON
. Set Value (TVAR(answer_file); Numeric; Length(TVAR(Json_answer_1)))
. Set Value (TVAR(First); Numeric; Find(TVAR(Json_answer_1), "" + DQUOTE + "TaxFree" + DQUOTE + ":", 1) + 10)
. Set Value (TVAR(Second); Numeric; Find(TVAR(Json_answer_1), ",", TVAR(First)) - 1)
. Set Value (TVAR(New_answer); String; Left(TVAR(Json_answer_1), TVAR(First)) + DQUOTE + DQUOTE + Right(TVAR(Json_answer_1),
. Set Value (LVAR(10); String; TVAR(New_answer))

+ . . SQL Statement (Update; Json_Doc; No; Yes; Json_doc=LVAR(10); Kalipso_doc=TVAR(Select_codKalip); FIELD(Json_Doc,Kalipso_doc)

. //Loop verifica e insere "Adicionalinha"
+ . . SQL Statement (Select; Json_Doc; TVAR(Json_enviar)=Json_doc; FIELD(Json_Doc,Kalipso_doc) = TVAR(Select_codKalip); No; Yes)
+ . . For Each (Select; Artigo_Fatura; TVAR(select_doc)=Cod_prod; TVAR(select_qtd)=Qtd; TVAR(select_armazem)=Armazem; TVAR(select
```

Após confirmar a intenção do utilizador é feito um ciclo “for each” que efetua o mesmo processo de adiciona linha realizado no form de criar documento.

É verificado se o artigo foi inserido de forma manual. Caso tenha sido é verificado se este existe no ERP e caso exista são pedidos os valores de armazém, local e preço unitário através da WebApi de forma a criar o Adiciona Linha.

```
- . . . . . If (TVAR(select_manual) = 1)
. . . . . //Artigo Inserido manualmente
+ . . . . . Http Request (Run Locally; 0; TVAR(Json_answer_2); TVAR(header); TVAR(status); TVAR(text); Yes; "h

- . . . . . If (TVAR(Json_answer_2) = "true")
. . . . . //Verifica se o artigo inserido existe e retira os seus atributos
. . . . . Set Value (TVAR(GetAtributos1); String; "[" + DQUOTE + "ArmazemSugestao" + DQUOTE + "]")
+ . . . . . Http Request (Run Locally; 0; TVAR(Json_answer_2); TVAR(header); TVAR(status); TVAR(text); Yes;
. . . . . Set Value (TVAR(GetAtributos2); String; "[" + DQUOTE + "LocalizacaoSugestao" + DQUOTE + "]")
+ . . . . . Http Request (Run Locally; 0; TVAR(Json_answer_3); TVAR(header); TVAR(status); TVAR(text); Yes;

. . . . . Set Value (TVAR(Json_answer_2); String; Replace(TVAR(Json_answer_2), "[", ""))
. . . . . Set Value (TVAR(Json_answer_2); String; Replace(TVAR(Json_answer_2), "]", ""))
. . . . . Set Value (TVAR(Json_answer_3); String; Replace(TVAR(Json_answer_3), "[", ""))
. . . . . Set Value (TVAR(Json_answer_3); String; Replace(TVAR(Json_answer_3), "]", ""))
+ . . . . . JSON Get Value (TVAR(Json_answer_2); "Valor"; Yes; TVAR(select_armazem); TVAR(Count))
+ . . . . . JSON Get Value (TVAR(Json_answer_3); "Valor"; Yes; TVAR(select_local); TVAR(Count))

. . . . . //Retira o array da resposta e de seguida o item de preço
+ . . . . . SQL Statement (Select; Json_Doc; TVAR(Json_doc)=Json_doc; FIELD(Json_Doc,Kalipso_doc) = TVAR(Se
+ . . . . . Http Request (Run Locally; 0; TVAR(Json_answer_1); TVAR(header); TVAR(status); TVAR(text); Yes;
+ . . . . . JSON Get Value (TVAR(Json_answer_1); "Linhas"; Yes; TVAR(Json_answer_1); TVAR(Count))
+ . . . . . JSON Get Array Item (TVAR(Json_answer_1); 1; TVAR(Json_answer_1); TVAR(select_qtd))
+ . . . . . JSON Get Value (TVAR(Json_answer_1); "PrecUnit"; No; TVAR(select_precUnit); TVAR(Count))
```

Após o tratamento da *string* de Json da devida forma, é enviado o pedido final de criação de Documento.

Caso o envio seja bem-sucedido é eliminado o mesmo documento do armazenamento do dispositivo.

```
- . . . . Else
. . . . . Message Box ("Atenção"; "O artigo: " + TVAR(select_doc) + " não existe, insira um artigo vál
. . . . . Cancel

. . . . . End If
- . . . . Else
. . . . . End If

+ . . . . Http Request (Run Locally; 0; TVAR(Json_answer_2); TVAR(header); TVAR(status); TVAR(text); Yes; "f

. . . . //Retirar atributos do objeto "TaxFree" do JSON
. . . . Set Value (TVAR(answer_file); Numeric; Length(TVAR(Json_answer_2)))
. . . . Set Value (TVAR(First); Numeric; Find(TVAR(Json_answer_2), "" + DQUOTE + "TaxFree" + DQUOTE + ":{'
. . . . Set Value (TVAR(Second); Numeric; Find(TVAR(Json_answer_2), "}," , TVAR(First)) - 1)
. . . . Set Value (TVAR(New_answer_2); String; Left(TVAR(Json_answer_2), TVAR(First)) + DQUOTE + DQUOTE +
. . . . Set Value (TVAR(Json_enviar); String; TVAR(New_answer_2))

+ . . . . End For Each
. . . . //POST de "CreateDocument"
+ . . . . Http Request (Run Locally; 0; TVAR(Json_answer_3); TVAR(header); TVAR(status); TVAR(text); Yes; "http

- . . . . If (TVAR(status) = 204)
+ . . . . . SQL Statement (Delete; Doc_Save; FIELD(Doc_Save,Cod_Kalipso) = TVAR(Select_codKalip))
+ . . . . . Refresh Control (CTRL(Looper_list); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
- . . . . Else
. . . . . End If
```

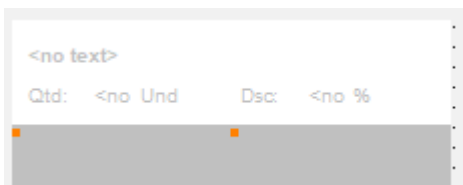
### 1.3.10. Form: Form\_Documento\_Detalhe\_guardado

#### 1.3.10.1. Form

```
After Open Form
//Preenche dados do documento de venda
+ SQL Statement (Select; Doc_Save; CTRL(input_data)=Data_cria; CTRL(Label_doc2)=Tipo_doc; CTRL(Label_ent2)=Entidade; (
+ SQL Statement (Select; Clientes; CTRL(Label_numcont2)=NumContrib; FIELD(Clientes,id) = CTRL(Label_ent2); No; Yes)
+ Refresh Control (CTRL(Looper_list_arts); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
```

Ao navegar para este form, e compreendido o id do documento guardado, são preenchidos os dados correspondentes do documento nos *labels* correspondentes.

### 1.3.10.2. Looper list arts save



Este *looper* apresenta os artigos correspondentes do atual documento.

```
"SELECT Cod_prod, Qty, Desconto FROM Artigo_Fatura WHERE Cod_Kalipso LIKE '%" + LVAR(inID) + "%'"
```

### 1.3.10.3. Image send save



Este ícone apresenta o mesmo código que o botão correspondente do form "Form\_Documentos\_Guardados" mas orientado para apenas um documento.

### 1.3.11. Form: Form\_Catalogo

#### 1.3.11.1. Form

```
Wtr Open Form
Show Status Window (-1; -1; 30; 30; "048076201"; ""; "OS Default"; 12; <none>; Left; "00000000

+ SQL Statement (Select; log_in; TVAR(get_list)=catalog; ; No; Yes)
+ If (TVAR(get_list) = 1)

//Get Value Count de artigos
+ Http Request (Run Locally; 0; TVAR(answer_count); TVAR(header); TVAR(status); TVAR(text); Yes;
Set Value (LVAR(count_artigos); String; TVAR(answer_count))
+ JSON Get Value (TVAR(answer_count); "Data"; Yes; LVAR(Item_count); TVAR(1))
Set Value (LVAR(Item_count); String; Left(LVAR(Item_count), 17))
Set Value (LVAR(Item_count); String; Right(LVAR(Item_count), 16))
+ JSON Get Value (LVAR(Item_count); "Column1"; Yes; LVAR(Item_count); TVAR(1))
+ SQL Statement (Select; Artigo; LVAR(Item_count_BD)=COUNT(Cod_Art); ; No; Yes)

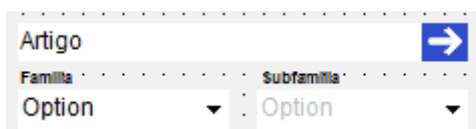
Looper Add Line (CTRL(ListView_catalog); 0)
Set Value (LVAR(search); Numeric; 0)

+ Refresh Control (CTRL(Combo_fam); Refresh List; Refresh Value)
Set Selection (CTRL(Combo_fam); By Index; 1)
+ Refresh Control (CTRL(Combo_fam_sub); Refresh List; Refresh Value)
+ Refresh Control (CTRL(ListView_catalog); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
+ Refresh Control (CTRL(ListView_list); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
```

Ao entrar neste form é efetuado uma validação às preferências do utilizador que irá demonstrar a lista de artigos em forma de lista simples ou lista com imagens. Estas diferentes listas encontram-se em *planes* diferentes.

É feito um pedido à WebApi que retorna o número total de artigos presentes no ERP para este dado utilizador.

#### 1.3.11.2. Filtro Looper



Este botão força um pedido de paginação após o utilizador escolher efetuar este com recurso ao input de filtro que permite procurar artigos através do seu código, família e subfamília. Este funciona da mesma forma que os pedidos iniciais efetuados no formulário.



### 1.3.11.3. List Catalog



Este *looper* é conectado à tabela de “Artigos” e apresenta as informações básicas dos mesmos. O *looper* apresenta o código de artigo e descrição do mesmo. Através deste *looper* é possível aceder ao detalhe de cada artigo.

```
"SELECT Artigo.Cod_Art, Artigo.Descricao, Artigo.Familia,  
CASE WHEN IFNULL(Artigo_imagens.Id,0)=0 THEN '' + PFOLDER + "\v10-icon.png' ELSE '' + DFOLDER + "\Anexos\  
|| Artigo_imagens.Id END  
FROM Artigo  
LEFT JOIN Artigo_imagens ON Artigo.Cod_Art=Artigo_imagens.Chave"
```

### 1.3.11.4. Image Right/Left



Através destes botões é possível navegar entre as várias páginas do catálogo. Cada vez que é selecionado este botão é executada uma nova paginação com recurso a uma lista criada no ERP que utiliza uma stored procedure.

```
Click  
Set Property (CTRL(Image_left); Visible; Yes)  
Set Property (CTRL(Image_left1); Visible; Yes)  
  
//Nova Paginação  
+ SQL Statement (Delete; Artigo; 1=1)  
Set Value (LVAR(offset_rows); Numeric; LVAR(offset_rows) + 25)  
+ Http Request (Run Locally; 0; TVAR(answer_items); TVAR(header); TVAR(status); TVAR(text); Yes;  
+ JSON Import to Table (Text; TVAR(answer_items); Artigo)  
  
+ SQL Statement (Select; Artigo; TVAR(count_mais)=COUNT(Cod_Art); ; No; Yes)  
Set Value (LVAR(Item_count_BD); Numeric; LVAR(Item_count_BD) + TVAR(count_mais))  
  
- If (LVAR(Item_count_BD) = LVAR(Item_count))  
  . Set Property (CTRL(Image_right); Visible; No)  
  . Set Property (CTRL(Image_right1); Visible; No)  
- Else  
End If  
  
+ Refresh Control (CTRL(ListView_catalog); Refresh List; Don't Refresh Sub-Lists; Thread Mode)
```

Após cada pedido são atualizadas as informações necessários e é efetuado um *refresh* à lista.

## 1.3.12. Form: Form\_Artigo\_Detalhe

### 1.3.12.1. Form

```
W Open Form
Show Status Window (-1; -1; -1; -1; "255249189"; ""; "OS Default"; 12; <none>; Let

//Preenche dados do artigo
- If (LVAR(venda) = 2 )
. //Se barcode
+ . SQL Statement (Select; Artigo; LVAR(select_artcod_local)=Cod_Art; CTRL(Label_de

- Else
. //Se venda/catalogo
+ . SQL Statement (Select; Artigo; LVAR(select_artcod_local)=Cod_Art; CTRL(Label_de
. Exec. Local Action Set (Fotos; No)

End If

Set Value (CTRL(Label_Title); String; "Artigo: " + LVAR(select_artcod_local))
Set Value (CTRL(Input_qtd); String; CTRL(Label_qtd_eco2_venda))
```

A navegação para este form compreende que seja transportado o código do artigo a partir da página de catálogo. Primeiramente é verificado se este está a ser acedido através da página de scanner *barcode* ou do catálogo.

Caso esteja a ser acedido através de *barcode*, o mesmo é transportado por variável e é utilizado para apresentar as informações do artigo. Caso seja a partir de catálogo é executado o mesmo processo, mas através do código de artigo. É também executada a LAS “Fotos”

```
W Fotos

+ SQL Statement (Select; Artigo_imagens; TVAR(Select_art_img)=Id; FIELD(Artigo_imagens,Ci

- If (LVAR(select_artcod_local) = "" OR LVAR(select_artcod_local) = 0)
- Else
. Set Property (CTRL(Image); <Value>; DFOLDER + "\Anexos\" + TVAR(Select_art_img))
End If
```

Esta LAS associa o código do artigo à fotografia correspondente.

```

After Open Form
//Existe Rede ou não?
- If (VAR(Ping_reply) = 0)
. //Não
- . If (LVAR(venda) = 1 OR LVAR(venda) = 2)
+ . . SQL Statement (Select; Artigo_pvp; TVAR(preco_label)=pvp1; FIELD(Artigo_pvp,Cod_art) = LVAR(inID); No; Yes)
. . Set Property (CTRL(Label_preco2); <Value>; TVAR(preco_label) + "€")
. . Set Property (CTRL(Label_preco3); <Value>; TVAR(preco_label) + "€")

. . Go To Plane (2)

- . Else
+ . . SQL Statement (Select; Artigo_pvp; TVAR(preco_label)=pvp1; FIELD(Artigo_pvp,Cod_art) = LVAR(inID); No; Yes)
. . Set Property (CTRL(Label_preco2); <Value>; TVAR(preco_label) + "€")
. . Go To Plane (1)

. End If

```

Após a abertura do form é verificada a existência de ligação à rede. Caso não haja é preenchido por *default* o label de pvp com o pvp1.

```

- Else
. //Sim
. //Caso a página esteja a ser acedida através da criação de Documento de venda, serão disponibiliza
- . If (LVAR(venda) = 1 OR LVAR(venda) = 2)
. . //Cria GUID do artigo
- . . If (LVAR(Edit_art) = 1)
. . . Set Value (CTRL(Input_qtd); String; LVAR(qtd_edit))
. . . Set Value (CTRL(Input_desconto); String; LVAR(desc_edit))
- . . Else
. . . Set Value (LVAR(Fatura_Guid); String; GetGUID())
+ . . . SQL Statement (Insert; Artigo_Fatura; No; Yes; artigo_guid=LVAR(Fatura_Guid); Cod_Kalipso=L
. . . End If

. . //Desconto
+ . . SQL Advanced (-1; "SELECT Desconto, COUNT(Desconto) AS Value FROM Artigo_Fatura WHERE Descont
+ . . SQL Advanced (-1; "SELECT Desconto, COUNT(Desconto) AS Value FROM Artigo_Fatura WHERE Descont
+ . . SQL Advanced (-1; "SELECT Desconto, COUNT(Desconto) AS Value FROM Artigo_Fatura WHERE Descont
. . Set Property (CTRL(Button_desconto_1); Text; LVAR(30))
. . Set Property (CTRL(Button_desconto_2); Text; LVAR(31))
. . Set Property (CTRL(Button_desconto_3); Text; LVAR(32))

. . //Retira o array da resposta e de seguida o item de preço
+ . . SQL Statement (Select; Json_Doc; TVAR(Json_doc)=Json_doc; FIELD(Json_Doc,Kalipso_doc) = LVAR(G
+ . . Http Request (Run Locally; 0; TVAR(Json_answer); TVAR(header); TVAR(status); TVAR(text); Yes;
+ . . JSON Get Value (TVAR(Json_answer); "Linhas"; Yes; TVAR(List); TVAR(Count))
+ . . JSON Get Array Item (TVAR(List); 1; TVAR(select_doc); TVAR(select_qtd))
+ . . JSON Get Value (TVAR(select_doc); "PrecUnit"; No; LVAR(Valor_monetario); TVAR(Count))
. . Set Property (CTRL(Label_preco3); <Value>; LVAR(Valor_monetario) + "€")

. . Go To Plane (2)

```

Caso haja acesso à rede é verificado se este está a ser acedido através de criação ou edição de documento, caso seja verdade é preenchido os dados de acordo.

Para o pvp é enviado um pedido à WebApi com as informações da entidade do cliente que retorna o pvp correspondente à mesma.

Para os efeitos de edição ou adição de artigos no contexto de documento o form navega sempre para o *plane 2*.

```
- . Else
+ . . SQL Statement (Select; Artigo_pvp; TVAR(preco_label)=pvpl; FIELD(Artigo_pvp,Cod_art) = LVAR(inID); No; Yes)
. . Set Property (CTRL(Label_preco2); <Value>; TVAR(preco_label) + "€")
. . Go To Plane (1)

. End If
End If

Close Status Window
```

Na situação do form estar apenas a ser acedido a partir do form catálogo são apenas preenchidos os dados correspondentes nas *labels* respetivas.

#### 1.3.12.2. Image Complete



Após o utilizador preencher a quantidade e desconto do artigo selecionado, deverá clicar neste botão. Este botão verifica se o utilizador inseriu quantidade maior que 0 e verifica se o artigo em questão esta a ser editado ou adicionado.

Caso esteja a ser editado, todas as informações inseridas sofrerão um *Update* devido.

```
WT Click
//Associa o Codigo de artigo e Quantidade selecionada para variaveis a ser transportadas
Get Property (CTRL(Input_qtd); <Value>; LVAR(qtd))
+ If (LVAR(qtd) = 0)

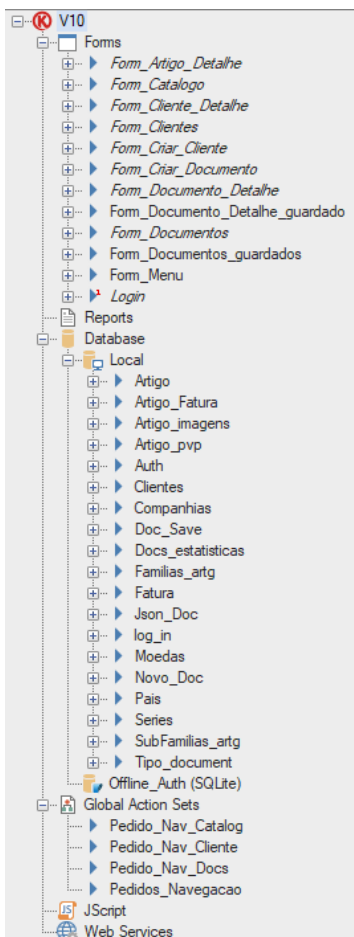
//Verifica se este produto esta a ser editado ou é novo
- If (LVAR(Edit_art) = 1)
+ . SQL Statement (Update; Artigo_Fatura; No; Yes; Desconto=CTRL(Input_desconto); Qtd=LVA
. Set Value (TVAR(Valor); Numeric; LVAR(Valor_monetario) * LVAR(qtd))
+ . SQL Statement (Update; Artigo_Fatura; No; Yes; Valor=TVAR(Valor); FIELD(Artigo_Fatura

- Else
. Set Value (TVAR(Valor); Numeric; LVAR(Valor_monetario) * LVAR(qtd))
+ . SQL Statement (Update; Artigo_Fatura; No; Yes; Armazem=CTRL(Label_arma2); Cod_prod=LVA

End If

Return To Form (Form_Criar_Documento)
```

## 1.4. Kalipso by Sysdev



### About

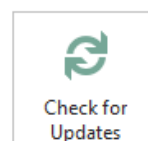
PRODUCT INFORMATION

WHAT'S NEW



Kalipso is a product developed by SYSDEV Mobile Solutions, Lda.

Licensed to: PRIMAVERA BSS  
License Type: Professional  
Serial Number: O1083217715  
License expires: 31/12/9999  
Version: 4.1.0 200127



[www.sysdevmobile.com](http://www.sysdevmobile.com)

[sales@sysdevmobile.com](mailto:sales@sysdevmobile.com)

[support@sysdevmobile.com](mailto:support@sysdevmobile.com)