Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

Pedro Ricardo Fernandes Batista

Programação ABAP e NEPTUNE para os sistemas SAP

Relatório de Estágio

Orientado por:

Prof. Doutor Célio Gonçalo Cardoso Marques, Instituto Politécnico de Tomar
Mestre André Filipe Marmelo Carrilho, Softinsa, Lda.

Relatório de Estágio
apresentado ao Instituto Politécnico de Tomar
para cumprimento dos requisitos necessários
à obtenção do grau de Mestre
em Analítica e Inteligência Organizacional

"O que hoje está fraco e mau, amanhã, estará muito melhor"

Autor desconhecido.

Resumo

A falta de informação ou a utilização de informação pouco fidedigna pode originar problemas graves para as organizações. A utilização de ferramentas que façam a recolha, tratamento e análise de informação torna-se, por isso, indispensável no mundo empresarial.

As soluções SAP são utilizadas por muitas organizações, não só porque permitem uma melhor gestão dos seus recursos como também permitem uma melhor gestão estratégica da organização, facilitando a tomada de decisão. Estas soluções contém as principais funções comerciais de uma empresa, dividindo-a em módulos: vendas, financeira, logística, entre outros.

O estágio na Softinsa visou a aquisição de novas competências a nível da programação ABAP e Neptune, que permitiram a criação e edição de programas, a implementação de notas e o desenvolvimento de várias aplicações para dispositivos móveis. Todas estes projetos foram desenvolvidos com base em metodologias de desenvolvimento SCRUM e Waterfall.

Neste relatório descrevem-se quatro projetos desenvolvidos pelo estagiário e que traduzem as funções de um consultor SAP ABAP/Neptune. Uma Implementação de Notas que consiste na instalação de um conjunto de correções para melhorar um sistema; a criação e edição de *reports*, onde é feita uma descrição do programa e do seu desenvolvimento; e por fim a criação de duas aplicações para dispositivos móveis usando a plataforma Neptune Software UX, que consiste na criação de soluções SAP para dispositivos móveis.

Com a realização deste estágio, foram adquiridas competências para a realização de tarefas SAP, bem como, conhecimentos a nível da programação que complementaram os conhecimentos adquiridos na instituição de ensino. O estágio permitiu também o desenvolvimento de uma aplicação para dispositivos móveis que venceu um concurso. No futuro, o estagiário pretende focar-se no desenvolvimento de soluções SAP que utilizem tecnologias SAP Leonardo e aprofundar os conhecimentos da programação ABAP e Neptune.

Palavras-chave: SAP, ERP, ABAP, Neptune, SAP Leonardo.

Abstract

The lack of information or the use of unreliable information can cause serious problems for organizations. Therefore, the use of tools that collect, treat and analyze information have become indispensable in the business world.

The SAP solutions are used by many organizations, not only because they allow better management of their resources but also because they allow better strategic management of the organization, making decision making easier. These solutions contain the main business functions of a company, dividing it into modules: sales, financial, logistics, among others.

The internship at Softinsa was focused on the acquisition of new skills in ABAP and Neptune programming, which allowed the creation and editing of programs, the note implementation and the development of various applications for mobile devices. All of these projects were developed based on SCRUM and Waterfall development methodologies.

This report describes four projects developed by the trainee and which translate the functions of an SAP ABAP / Neptune consultant. A Notes Implementation that consists of installing a set of fixes to improve a system; the creation and edition of reports, where a description of the program is made and how it was developed; and finally the creation of two applications for mobile devices using the platform Neptune Software UX, which consists of creating SAP solutions in a mobile.

With the completion of this internship, competences were acquired to perform SAP tasks, as well as programming knowledge that complemented the knowledge acquired in the Higher Education Institution. The internship also allowed the development of a mobile app that beat one contest. In the future, the trainee intends to focus on the development of SAP solutions using SAP Leonardo technologies and to deepen ABAP and Neptune programming knowledge.

Keywords: SAP, ERP, ABAP, Neptune, SAP Leonardo.

Agradecimentos

A realização deste relatório de estágio contou com vários apoios e incentivos que, sem eles, este relatório não se teria tornado realidade.

Ao Doutor Célio Gonçalo Marques, pela sua ajuda e orientação, pelas opiniões e críticas, pela sua paciência e total disponibilidade, e pelos conhecimentos que transmitiu durante o desenvolvimento deste relatório.

Ao meu orientador de estágio na Softinsa, o Mestre André Carrilho, pelo apoio que transmitiu durante todo o estágio, e pela sua disponibilidade para ajudar a solucionar os desafios que foram aparecendo durante o estágio e para a elaboração deste relatório.

Ao Eng.º Nuno Madeira, Diretor do Mestrado de Analítica e Inteligência Organizacional no Instituto Politécnico de Tomar, pela sua disponibilidade e apoio não só na elaboração deste relatório, mas durante todo o Mestrado.

À minha namorada, que esteve comigo em todos os momentos, pela ajuda, apoio e paciência que transmitiu durante todo o desenvolvimento deste relatório.

A todos os meus amigos, em especial, aos membros do Covil, por toda a ajuda e força que me transmitiram nestes meses.

Em último, mas com um papel fulcral, um especial agradecimento aos meus pais, irmão e restante família, por toda a paciência, pelo apoio incansável, incentivo e toda a ajuda que me transmitiram durante todos estes anos, sem eles não teria chegado até aqui e não seria a pessoa que sou hoje.

A eles dedico, incondicionalmente, este trabalho.

Índice

Índice de Figuras

Índice de Tabelas

Lista de Abreviaturas e Siglas

ABAP – *Advanced Business Application Programming*

CRM – *Customer Relationship Management*

CENIT – Centro de Inovação Tecnológica

ERP – *Enterprise Resource Planning*

Key User – Funcionário que executa os testes do lado do cliente

OT – Ordem de Transporte

SAFT-PT – *Standard Audit File for Tax Purpose*

SAP – *Systems, Applications and Products in Data Processing*

Capítulo 1 | Introdução

1.1 Contextualização

Nos dias de hoje, a informação tem um papel fulcral nas organizações, quer na tomada de decisão quer na antecipação de cenários.

Quando se fala de informação, tem que se pensar primeiramente em dados e da maneira como são adquiridos e tratados. Quando uma pessoa visita um *Website* ou faz uma compra, normalmente não pensa que os dados que estão a ser gerados vão ser usados futuramente. Contudo, as organizações armazenam esses dados, para depois serem tratados e analisados.

A publicidade é cada vez mais especifica e dirigida a cada utilizador, por exemplo, se uma pessoa quiser comprar um carro e manifestar esse interesse através de pesquisas, muito provavelmente, vai receber publicidades de *stands* ou oficinas. Isto é possível porque vivemos conectados ao mundo, através do *E-mail*, *Facebook*, *Google*, entre outros.

Uma das empresas que impulsionou muito a área da informação foi a SAP, através da criação de soluções ERP (*Enterprise Resource Planning*) para os seus clientes. O uso de soluções da SAP não só permite às empresas a consulta de informação em tempo real como também permite uma gestão estratégica muito mais eficiente, facilitando a tomada de decisão e prevenindo situações menos positivas para a organização.

Para o desenvolvimento destas soluções, é utilizada a linguagem de programação ABAP. Devido à necessidade dos utilizadores acederem à informação em qualquer lugar e a qualquer altura, foi desenvolvida uma ferramenta de desenvolvimento de aplicações SAP para dispositivos móveis: a *Neptune.* Esta ferramenta permite que seja possível a criação de aplicações móveis, através da linguagem ABAP e de *JavaScript*.

1.2 Objetivos

O objetivo geral do estágio é a execução de tarefas a nível da programação, de forma a que seja possível criar e manter aplicações SAP e aplicações Neptune sobre SAP.

Os objetivos específicos do estágio são:

- Aprendizagem de programação em SAP, com recurso à linguagem ABAP;
- Criação de competências em Neptune e em linguagem de programação *Javascript*;
- Implementação de notas corretivas em máquinas de clientes;
- Desenvolvimento de programas em ABAP;
- Desenvolvimento de aplicações móveis em Neptune.

1.3 Relevância do Estágio

O estágio permitirá a aplicação prática d o conhecimento adquirido pelo estagiário, tanto na Licenciatura de Engenharia Informática, como no Mestrado de Analítica e Inteligência Organizacional, bem como a aquisição de novas competências.

Num nível mais técnico, este estágio permitirá a aprendizagem de duas linguagens de programação: ABAP e Neptune. O estagiário obterá competências para poder criar, corrigir ou melhorar programas ou aplicações para dispositivos móveis, e para realizar tarefas de um Consultor SAP ABAP/Neptune.

A aprendizagem destas linguagens de programação permitirá também a consolidação de conhecimentos a nível de base de dados, como também, de *JavaScript* (utilizada na linguagem de programação Neptune).

Ao fazer o estágio numa empresa com a dimensão da SoftINSA, o estagiário irá conhecer pessoas com uma vasta experiência nesta área, que poderão transmitir-lhe conhecimentos importantes para o seu desenvolvimento profissional.

1.4 Organização do Relatório

O Capítulo 1 diz respeito à introdução do relatório, onde é feita uma contextualização do estágio, a sua finalidade, o local onde foi realizado, os seus objetivos e a relevância do mesmo. Por fim, é apresentada a organização do relatório.

O Capítulo 2 diz respeito à caracterização do meio envolvente, abordando a empresa onde foi realizado o estágio, as áreas que a constituem e o papel do consultor ABAP/Neptune na empresa.

No Capítulo 3 é feita uma introdução aos sistemas de informação, e é explicado em que consiste um ERP. De seguida é feita uma introdução à empresa SAP e às soluções que fornece aos seus clientes. Também são descritas as ferramentas utilizadas neste estágio para o desenvolvimento de soluções SAP, e a necessidade e importância das mesmas. Finalmente é explicado o papel do consultor SAP.

No Capítulo 4 são descritas as metodologias de desenvolvimento de *software*, comparando-se as metodologias ágeis com as metodologias tradicionais. São também apresentadas as descrições das metodologias utilizadas no desenvolvimento de soluções SAP pelos consultores SAP ABAP/Neptune.

No Capítulo 5 é feita a descrição dos projetos que foram realizadas durante o estágio. O mesmo está dividido em quatro projetos. O primeiro projeto diz respeito à Implementação de Notas que consiste num conjunto de correções e melhorias no sistema. O segundo projeto diz respeito à criação de um programa de consulta de dados. O terceiro e o quarto projeto dizem respeito à criação de soluções SAP para dispositivos móveis.

No capítulo 6 são apresentadas as reflexões e considerações finais resultantes do estágio.

Capítulo 2 | Caracterização do Meio Envolvente

2.1 Empresa

O estágio foi realizado na empresa SoftINSA, uma empresa de tecnologias de informação, que permitiu ao estagiário desenvolver as competências adquiridas no Mestrado de Analítica e Inteligência Organizacional e na Licenciatura de Engenharia Informática.

A Softinsa é uma empresa do grupo *International Business Machines* (IBM) e da Viewnext. A IBM é uma empresa de Tecnologias de Informação criada nos Estados Unidos da América no início do século XX. A *Viewnext* está sedeada em Espanha e tem como especialidade o desenvolvimento e gestão de aplicações e infraestrutura.

A empresa SoftINSA foi fundada em 2007, contudo a sua história vem de mais atrás, através da ligação entre a CGI Portugal e a Viewnext (SoftINSA, 2018a).

No ano de 1994 foi criada a empresa CGI Portugal - *Compagnie Generale D'Informatique* em Portugal. Esta era uma empresa francesa pertencente ao grupo IBM, com várias delegações espalhadas pelo Mundo, cingindo-se apenas à comercialização, implementação, desenvolvimento e manutenção de plataformas de gestão de recursos humanos e imobilizado (SoftINSA, 2018a).

No dia 1 de abril de 1998, a INSA, S.A - *Ingenieria de Software Avanzado* (empresa espanhola fundada em 1991 e que também pertence à IBM Espanha), atual *Viewnext*, transforma a CGI Portugal na sua sucursal portuguesa, em Lisboa, tendo, assim, a capacidade e a possibilidade para cobrir uma área maior na península Ibérica (SoftINSA, 2018a).

No início de agosto de 2007 a sucursal portuguesa da *Viewnext* é transformada numa empresa independente, de nome Softinsa - Engenharia de Software Avançado Lda. Esta é detida pela companhia IBM Portugal e pela *Viewnext*, dando continuidade ao trabalho desenvolvido (SoftINSA, 2018a).

A SoftINSA atua no mercado de serviços de gestão e desenvolvimento de aplicações e infraestruturas, sendo especialista nas áreas de *Management Services, IT Managed Services, Cloud, Analytics, Cognitive, Human Capital Solutions, SAP, Site & Facilities, Mobility e Asset Management*. A mesma está no mercado português há cerca de 19 anos e, atualmente, conta com mais de 600 profissionais em tecnologias de informação e com uma

cobertura ibérica de escritórios e centros de inovação (Viseu, Tomar, Salamanca, Cáceres, Réus, Ourense, Almeria e Málaga) (SoftINSA, 2018a).

Os Centros de Inovação Tecnológicas (CENIT´s) são especializados na prestação de serviços e na gestão e desenvolvimento de aplicações, com capacidade regional e global, e integram a rede internacional dos Centros de Inovação da IBM. Estes centros colocam à disposição dos seus clientes as infraestruturas, as ferramentas e os profissionais capazes de responder às mais diversas necessidades na área das tecnologias de informação, oferecendo serviços personalizados, como a gestão remota de infraestruturas e aplicações ou processos (SoftINSA, 2018b).

2.2 Estrutura Organizacional

A Softinsa está organizada hierarquicamente da seguinte forma: Um Diretor Geral (Sérgio Pereira), um Diretor de Estratégia (Henrique Mourisca), um Diretor dos Centros de Entrega (Nuno Dionísio), um Diretor de Desenvolvimento do Negócio (José Caratão), um Diretor do Departamento de *Global Business Services* (Nuno Alves), um Diretor do Departamento de *Global Technology Services* (Francisco Caeiro), um Diretor do Departamento Financeiro (Paulo Gervásio) e, por fim, a Diretora do Departamento de Recursos Humanos (Raquel Jerónimo).

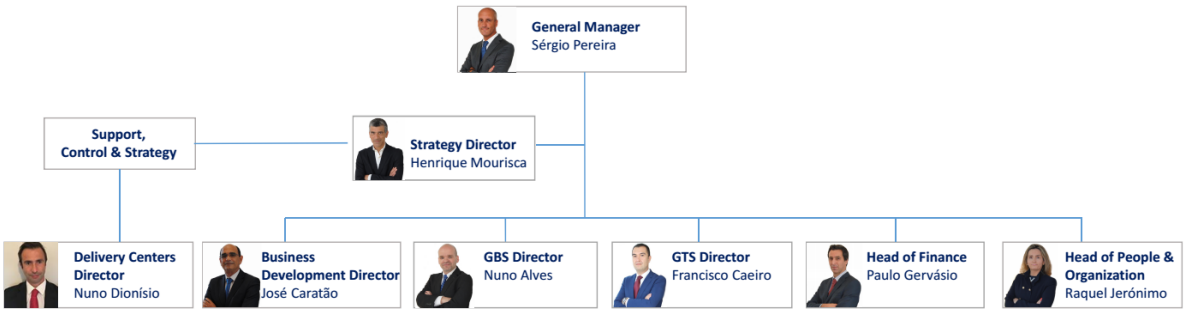


Figura 1 - Organograma Softinsa (SoftINSA, 2018).

Em relação à componente de SAP, a mesma está definida e organizada no departamento de GBS (*Global Business Services*). Em relação à área ABAP, o Henrique Coelho é o gestor da mesma, acumulando funções de *People Manager*. A coordenadora da equipa de SAP no CENIT Tomar é a Ana Lígia. A pessoa que coordena e distribui os

trabalhos a realizar pelos consultores da equipa de ABAP no CENIT de Tomar (*focal point*) é o Carlos Pires. Por fim estão os consultores SAP ABAP/Neptune na qual o estagiário se insere.

2.3 Departamento de Global Business Services (GBS)

A equipa de *Global Business Services* da Softinsa foi criada com o objetivo de oferecer os recursos necessários que atendam aos requisitos dos seus clientes que baseiam o seu ERP em sistemas SAP. Atualmente, a equipa de GBS da Softinsa é constituída por mais de 350 consultores, programadores e analistas, especializados no desenvolvimento, implementação e gestão de soluções tecnológicas (SoftINSA, 2018c).

Esta equipa permite um suporte constante aos seus clientes, nos mais diversos setores de atividade, ajudando os mesmos a atingir os seus objetivos (SoftINSA, 2018c). A equipa de GBS fornece o seu suporte aos seus clientes, independentemente da sua localização, fazendo com que seja possível fazer uma melhor gestão dos recursos que dispõe (DeMent, 2018).

2.4 Consultor do GBS

Numa definição mais geral, um consultor tem como principal função diagnosticar e formular soluções para resolver um problema. O departamento de GBS é dividido pelas seguintes áreas: *Enterprise Applications*, SAP, *Cognitive Automation*, *Applications Services*, *IBM Business & Resource Management*, *Delivery Transformation & Inovation* e AMS. O consultor ABAP/Neptune está inserido na área de SAP.

Um consultor SAP, numa definição mais especifica, fornece suporte tanto no desenvolvimento e implementação, como manutenção e melhorias dos módulos SAP (Finanças, logística, entre outros) com o objetivo de satisfazer as necessidades dos clientes, visando o lucro e desenvolvimento das suas empresas (Adviser, 2018).

Depois do consultor SAP efetuar uma implementação, o mesmo poderá auxiliar o cliente, com suporte local ou remoto, efetuando novos desenvolvimentos de processos, testes e melhorias, mantendo um contato constante com o cliente (Adviser, 2018).

Na Softinsa, o consultor SAP é responsável pela realização de inúmeras tarefas, não só a nível de desenvolvimento, como também na manutenção dos seus sistemas, ao providenciar um suporte constante ao cliente, garantindo o seu bom funcionamento.

Capítulo 3 | Sistemas de Informação

3.1 Sistemas de Informação

Um sistema de informação é um sistema que é constituído por pessoas, procedimentos, dados/informação e componentes TIC (*hardware, software* e comunicações) com o objetivo de recolher, processar, armazenar, analisar e distribuir as informações tratadas para o cliente (Conceito.de 2018).

« Information systems (IS) involve a variety of information technologies (IT) such as computers, software, databases, communication systems, the Internet, mobile devices and much more, to perform specific tasks, interact with and inform various actors in different organizational or social contexts. (…) » (Boell & Cecez-Kecmanovic, 2015)

Um sistema de informação funciona como qualquer outro sistema, sendo composto por *inputs*, onde são inseridos dados ou instruções, e *outputs*, onde são recebidos os relatórios ou cálculos realizados pelo sistema. O sistema de informação processa os dados recebidos e gera informação que é disponibilizada ao utilizador final ou a outros sistemas (Turban, McLean & Wetherbe, 1999). As principais vantagens e desvantagens do uso de sistemas de informação são as seguintes:

Tabela 1 - Vantagens e desvantagens do uso de sistemas de informação (Cajiza, 2015).

Vantagens
Otimiza o fluxo de informação e a qualidade da mesma na organização.
Maior integridade da informação.
Maior segurança no acesso à informação
Otimização do processo de tomada de decisão.
Elimina a redundância de atividades.
Desvantagens
A utilização de sistemas de informação por si só não torna uma empresa verdadeiramente integrada.
Altos custos que não justificam o custo/benefício.
Sistemas de informação com defeito podem levar à circulação de informação errada para outros sistemas, podendo criar problemas não só para a empresa como para os seus clientes.

Os sistemas de informação podem ser classificados em dois grandes tipos: Área Funcional, onde engloba todos os sistemas ligados a *markting*, vendas, produção, recursos humanos, contabilidade, finanças e entre outros; Nível da Gestão onde o seu foco é mais evidenciado a nível da gestão, a nível estratégico, nível conhecimento e nível operacional (Andrade & Morais, 2014).

Existem vários tipos de sistemas de informação, sendo que os mais conhecidos são: ERP, o CRM, o SCM e o SIG. O ERP (*Enterprise Resource Planning*) é um software que integra vários processos e dados da empresa, reunindo os mesmos num único local. O CRM (*Customer Relationship Manager*) é um sistema que tem como principal função automatizar todas as funções que estejam relacionadas com o cliente, permitindo o armazenamento de dados do mesmo, como as suas preferências, histórico de compras, entre outros. O SCM (*Supply Chain Management*) é um sistema de informação que integra os processos relativos aos fornecedores de serviços, produtos, entre outros. O SIG (Sistemas de Informação de Gestão) é um sistema mais direcionados para o apoio à tomada de decisão e atua mais a nível estratégico, operacional e tático (Mesquita, 2016).

3.2 Enterprise Resource Planning (ERP)

Os sistemas ERP (*Enterprise Resource Planning*) são softwares que permitem a gestão empresarial, englobando vários módulos (Figura 2) que suportam diversas áreas funcionais, tais como, planeamento, fabrico, vendas, *marketing*, distribuição, contabilidade, finanças, gestão de recursos humanos, gestão de projetos e gestão de inventários (Rashid, Hossain, & Patrick, 2002).

Figura 2 - Módulos ERP (GrandviewSolutions, 2017).

Uma grande parte dos sistemas ERP apresentam as seguintes características (Wailgum & Perkins, 2018):

- Integração em toda empresa;
- Uma base de dados comum;
- Operações em tempo real;
- Flexibilidade;
- Divisão em módulos.

Os sistemas ERP têm um impacto significativo no mundo empresarial e apresentam várias vantagens, tais como:

- Custos das tecnologias de informação mais baixos: um sistema ERP baixa as despesas relacionadas com tecnologias de informação (infraestruturas, suporte, licenciamento de aplicações, formação);
- Visibilidade: permite acesso a cada processo importante no negócio uma vez que os dados de cada departamento estão facilmente acessíveis;
- Segurança dos dados: como é um sistema que controla toda a informação da organização, é necessário que a segurança seja elevada;
- Eficiência: reduz o tempo e esforço para realizar determinadas atividades;
- Melhora o acesso aos dados: permite que o acesso aos dados seja feito de uma forma mais fácil;

- Melhora a colaboração e o fluxo do trabalho: como o acesso à informação é mais acessível, permite que os funcionários da organização tenham conhecimento em tempo real dos fluxos de trabalho.

Quanto às desvantagens, a que mais se destaca é, sem dúvida, os custos de toda a implementação deste tipo de sistemas (planeamento, testes, configuração, personalização, entre outros). A personalização, caso seja mal executada, pode ter um impacto bastante negativo na atividade da empresa (SelectHub, 2018).

3.3 Systems Applications and Products (SAP)

A empresa SAP (*Systems Applications and Products in Data Processing*) é uma das maiores empresas de software de negócios do mundo. Fundada em 1972, a SAP não só transformou a maneira como as empresas fazem negócios, como também revolucionou o mundo das tecnologias de informação (SAP, s.d.a). Os principais produtos que a SAP desenvolve para os seus clientes são *softwares* ERP.

O SAP ERP é um *software* de planeamento e gestão de recursos empresariais desenvolvido pela empresa alemã SAP SE. Este *software* incorpora as principais funções comerciais de uma organização. O SAP ERP procura contemplar a empresa como um todo, dividindo-a em módulos, onde cada um corresponde a uma área específica, como por exemplo, o módulo SD (*Sales and Distribution*) que contempla a área de Vendas e Distribuição, fazendo a integração das informações para um determinado processo (Guru99, 2018).

Cada programa é executado através de uma transação independente. Estes programas são desenvolvidos em ABAP, uma linguagem de programação, da qual a SAP detém os direitos (SAP, s.d.b).

Apesar da SAP ser um *software* muito abrangente, por vezes não dispõe de certas funcionalidades exigidas pelo cliente. Isto leva a que seja necessário o desenvolvimento de soluções personalizadas.

3.3.1 Módulos SAP

A SAP é constituída por vários módulos, tal como foi referido no ponto anterior, sendo que cada módulo é responsável por mais de mil processos de negócio, baseado em práticas do dia-a-dia de cada empresa. O sistema é configurado para atender às necessidades de cada processo, onde mais de 8 mil tabelas administram em tempo real as informações que trafegam pela empresa. Os principais módulos presentes neste sistema são os apresentados na Tabela 2.

Tabela 2 - Módulos de SAP (Cavalcanti, 2011).

Módulos	Descrição
SAP MM	*Material Management*
SAP WM	*Warehouse Management*
SAP FI	*Financial Accounting*
SAP SD	*Sales and Distribution*
SAP HCM	*Human Capital Managgement*
SAP PS	*Project System*
SAP PP	*Production Planning and Control*
SAP CO	*Controlling*
SAP QM	*Quality Management*
SAP PM	*Plant Maintenance*
SAP IS	*Industry Solutions*
SAP RE	*Real Estate*
SAP BW	*Business Warehousing*

3.3.2 Fluxo de Trabalho entre Equipas

Quando a plataforma SAP não possui as funcionalidades que permitam satisfazer as necessidades solicitadas pelos utilizadores, os consultores funcionais elaboram especificações para que os consultores técnicos as possam implementar. Estas especificações podem decorrer de pedidos dos utilizadores e/ou alterações processuais devido a modificações legais ou novas necessidades (Figura 3) (*Workflow* Trabalho - Documentação Interna).

Figura 3 - Fluxo de trabalho entre equipas (Softinsa, 2018).

As especificações técnicas, além de explorarem o problema, podem apresentar trabalho já desenvolvido. As implementações daqui resultantes são feitas através de módulos de função, procedimentos e outros objetos, cujo nome começa obrigatoriamente por "Z" ou "Y" para indicar que não é um programa *standard* da SAP. Num *upgrade* de sistemas por exemplo, a SAP consegue identificar os programas desenvolvidos por terceiros, evitando que os mesmos sejam alterados ou apagados da mesma forma que os programas *standard*. Após ser implementada a especificação técnica, o consultor técnico responsável regista o seu trabalho no documento que recebeu do consultor funcional (*Workflow* Trabalho - Documentação Interna).

A plataforma SAP é modular e o desenvolvimento é feito através da implementação de processos. Alguns conceitos fundamentais são:

- Processo – É uma sequência lógica de procedimentos e fluxos funcionais incluindo a recolha de informação, o seu processamento, e resultados deste;
- Transação – É uma instanciação de um processo que pode resultar em sucesso ou erro. As transações são atómicas;
- Telas – São os ecrãs para o utilizador, podendo ser desenvolvidas ou criadas pelo próprio sistema;
- Programas: são os módulos onde se implementam funções de processamento (*Workflow* Trabalho - Documentação Interna).

3.4 Advanced Business Application Programming (ABAP)

A *Advanced Business Application Programming* (ABAP) é uma linguagem de programação de alto nível, desenvolvida pela SAP. É a principal linguagem utilizada nos produtos mais conhecidos da SAP, como por exemplo o SAP R/3 (Cleverism, 2018). O SAP R/3 é um *software* de gestão de recursos nas empresas.

A utilização de um sistema SAP R/3 é descrita da seguinte forma (Figura 4):

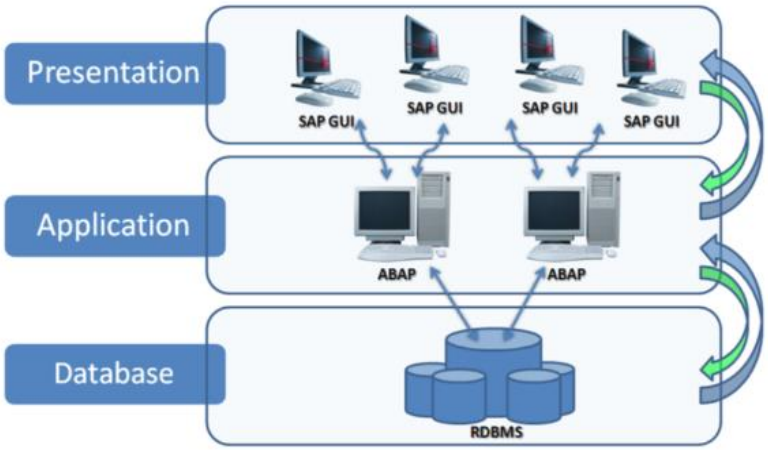


Figura 4 - Arquitetura do sistema ABAP (Pete, 2012).

Na arquitetura do sistema ABAP, a camada de Apresentação (ou Acesso), representa a camada onde o utilizador poderá aceder ao sistema SAP e executar tarefas. Pode ser feito através de um navegador Web, dispositivo móvel, entre outras formas. Todo o processamento central é feito na camada de Aplicação. O Servidor de Aplicação não é só um sistema em si, mas pode conter várias instâncias do sistema de processamento. O servidor comunica com a camada de Base de Dados, que normalmente é mantida num servidor à parte (SAP, s.d.c; Tutorialspoint, 2018a). Os programas ABAP são executados ao nível da camada de Aplicação.

O ABAP é uma linguagem que obriga a conhecimentos avançados de programação e é utilizada pelos programadores da SAP, que desenvolvem o *software*, e por consultores ABAP, que adaptam o *software* às necessidades dos clientes. A sua sintaxe é similar á do COBOL (linguagem de programação orientada para o processamento de base de dados comerciais).

3.5 SAP Fiori

O SAP Fiori é uma ferramenta disponibilizada pela SAP que permite o desenvolvimento de aplicações para *browsers* ou dispositivos móveis (SAP, 2018). O SAP Fiori providência mais de 300 aplicações standard que podem ser usadas em diversas áreas de negócio (Figura 5) (Tutorialspoint, 2018b).

Figura 5 - SAP Fiori (Tutorialspoint, 2018b).

As aplicações em SAP Fiori estão prontas a serem utilizadas em qualquer dispositivo móvel, através da *Internet*, o que permite ao utilizador iniciar um processo num computador e continuar esse processo num telemóvel ou *tablet* (Procensus, s.d.).

O uso de aplicações SAP Fiori tem inúmeras vantagens como:

- Aumentar a produtividade nas empresas, ao simplificar os seus processos;
- Elevada mobilidade;
- Pode ser utilizado em diversos dispositivos, de diferentes formas e tamanhos;
- Aumenta a aceitação por parte dos colaboradores;
- Reduz tempo de formação e adaptação dos utilizadores às aplicações (SAP, 2017).

3.6 Neptune

O *Neptune* é um *add-on* desenvolvido pela empresa Neptune Software que permite o desenvolvimento de aplicações móveis com integração direta nos sistemas SAP (NeptuneSoftware, 2018a). Esta ferramenta usa linguagens de programação ABAP, *Javascript* e CSS para o desenvolvimento de aplicações e devido à sua arquitetura, a mesma permite a diminuição do tempo de desenvolvimento de aplicações (Neptune Software, 2018b).

O *Neptune Software UX Platform* é a única plataforma de desenvolvimento de aplicações com certificação SAP e permite que seja possível otimizar e modernizar os processos SAP e as interfaces dos utilizadores, tendo todos os benefícios do SAP Fiori e ABAP (Neptune Software, 2018c).

O *Neptune UX Platform* permite fazer a cobertura de todo o ciclo de vida de uma aplicação desenvolvida nesta plataforma (Figura 6). O ciclo de vida da aplicação é definido em três fases: Desenvolvimento, Administração e Acesso (Pène & Collin, 2017).

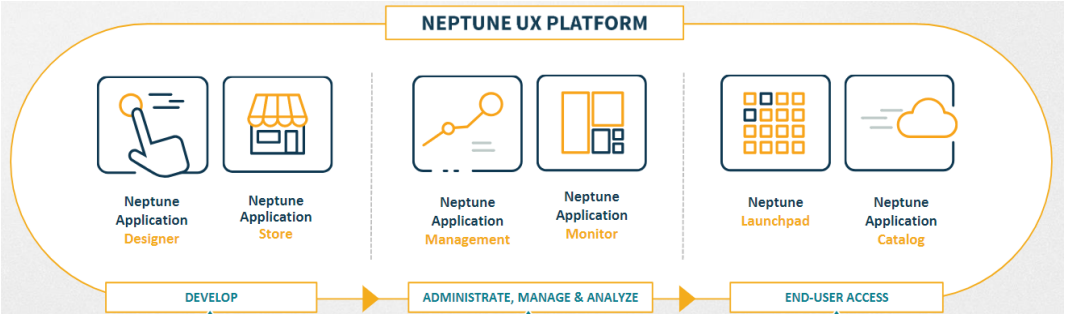


Figura 6 - *Neptune UX Platform* (Pène & Collin, 2017).

Desenvolvimento - O programador pode desenvolver as aplicações de raiz, através de *drag and drop* de objetos para a aplicação. De seguida, o utilizador faz a programação em *Javascript* nesses objetos (Figura 7). O utilizador poderá também ir à *Neptune Application Store* onde pode obter aplicações base e trabalhar nessas mesmas aplicações, poupando assim bastante tempo. Nem sempre é a melhor alternativa a seguir, sendo uma opção que depende bastante do tipo de aplicação e do tempo estipulado para o seu desenvolvimento (Pène & Collin, 2017).

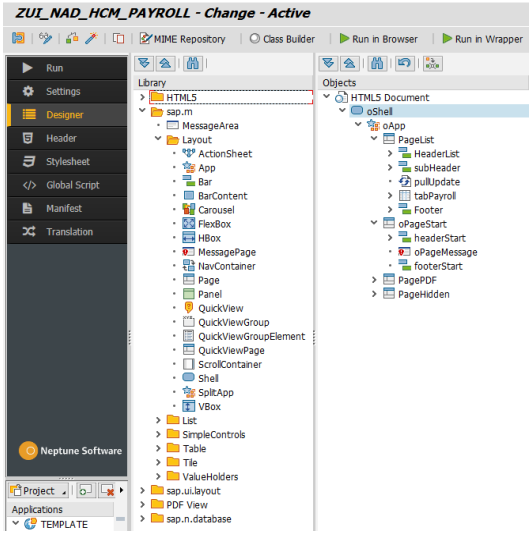


Figura 7 - *Neptune drag and drop* (Buajordet, 2017).

Administração, Análise e Gestão – Permite ao utilizador organizar e gerir as aplicações em todo o seu ciclo de vida (Pène & Collin, 2017).

Acesso - O utilizador final poderá ter acesso às aplicações. As aplicações estão vinculadas ao *Neptune Launchpad* (Figura 8), e o acesso a elas está dependente das autorizações que esse utilizador tem (Pène & Collin, 2017).

Figura 8 - *Launchpad* (Retailsolutions, 2018).

Esta solução sofreu recentemente um *rebranding*, e neste momento, a Neptune Software disponibiliza duas soluções: *Planet 8*, que é usado em SAP (antigo Neptune) e o

Planet 9 que tem como base no uso de API´s. Ambas as soluções são muito semelhantes a nível da arquitetura, permitindo que qualquer programador de *Planet 8* possa desenvolver aplicações em *Planet 9* sem grandes adaptações.

3.7 Implementação de Notas SAP

As notas SAP são correções ou melhorias *standard* (criadas pela SAP) que são implementadas no sistema SAP do cliente. Estas notas incluem uma descrição dos sintomas, causas dos erros, e a forma como deverão ser implementadas no sistema, entre outras informações (Sistemasgrainhill, 2013).

Para ver as notas SAP, o utilizador deve aceder ao Portal de Suporte SAP através dos dados de utilizador e *password*. As notas SAP podem ser do tipo informativo, onde não propõem qualquer tipo de correção, ou podem ser do tipo corretivo, onde propõem correções a nível do código. O código necessário para corrigir o problema é facultado pela nota SAP (Sistemasgrainhill, 2013).

As notas podem conter requisitos Pré e Pós Implementação (antes ou depois de implementar a nota), que consistem num conjunto de operações que o consultor tem de realizar para preparar ou concluir a implementação da nota. A maioria das notas são automáticas, ou seja, são implementadas automaticamente através da transação SNOTE (Sistemasgrainhill, 2013).

Para realizar alterações a nível de código, poderá ser necessário fazer o registo de objetos. Muitas das alterações que o utilizador tem de realizar são em objetos *standard* da SAP, e para o utilizador efetuar alterações desses objetos, o mesmo tem de registar essa alteração. Isto permite que a SAP tenha noção dos objetos *standard* que foram alterados e validar se essas alterações foram bem realizadas. Caso as alterações não sejam válidas, a SAP poderá recusar-se a dar suporte ao objeto em causa ou tópicos que envolvam esse objeto.

As notas SAP são implementadas consoante a versão e configuração do sistema. A implementação de notas erradamente pode afetar o sistema, sendo muito importante a criação de um *backup* antes da sua implementação. Antes de aplicar qualquer nota SAP, é

também necessário verificar a compatibilidade da nota com as configurações do sistema (Sistemasgrainhill, 2013).

As notas podem estar dependentes de outras, ou seja, uma nota só deve ser implementada se as notas das quais está dependente já estejam implementadas no sistema. Para evitar problemas na implementação de notas, é feito um levantamento de todas as notas a implementar, pela ordem certa. Esse levantamento pode ser feito pelo consultor que está no lado do cliente ou pelo programador, dependendo do conhecimento que cada um possui (Sistemasgrainhill, 2013).

3.8 SAP Leonardo

O SAP Leonardo é um sistema que combina um conjunto de tecnologias viradas para o futuro num único serviço. Trata-se de uma ferramenta poderosa que permite não só uma inovação mais rápida no negócio, como também permite criar novos modelos e redefinir os seus negócios (SAP, 2017). Inicialmente, o SAP Leonardo era um serviço que se dedicava a soluções IoT (*Internet of Things*), contudo devido às necessidades do mercado, neste momento é um serviço que suporta um conjunto de outras tecnologias que estão integradas na SAP *Cloud Platform* (Figura 9) (Matsu, 2017).

As três grandes características do SAP Leonardo são:

- Inovação: Oferece novos serviços como *Machine Learning, IoT, Big Data* entre outros, permitindo redefinir os processos de uma empresa, tornando o negócio mais digital e virado para o futuro (SAP, 2017);
- Integração: Desde a plataforma da *Cloud* às aplicações que o constituem, o SAP Leonardo é aberto, extensível e está pronto a ser implementado em qualquer área de negócio (SAP, 2017);
- Escalabilidade: O SAP Leonardo permite uma implementação muito mais rápida através da utilização dos seus serviços. Permite também a automatização e melhoria dos processos que já existem, e o desenvolvimento de novas aplicações, podendo dar origem a novos modelos de negócio (SAP, 2017).

3.8.1 Soluções SAP Leonardo

Neste momento, o SAP Leonardo é uma plataforma que disponibiliza um conjunto de tecnologias que permitem auxiliar as empresas tanto na gestão como no apoio à tomada de decisão. A Figura 9 mostra as tecnologias que constituem o SAP Leonardo (Carey, 2018).

Figura 9 - Tecnologias que constituem o SAP Leonardo (Wong, 2018).

A *Internet* das Coisas (IoT) permite conectar as pessoas e processos, ou seja, conecta não só os produtos e ativos que impulsionam o IoT Industrial como também a conexão de infraestruturas, mercados e pessoas, permitindo uma *Internet of Everything* (SAP, 2017).

A *Machine Learning* permite que os computadores aprendam sem ser necessário efetuar programação. Os computadores adquirem conhecimentos através da análise dos dados recolhidos (SAP, 2017). O uso de *Machine Learning* permite uma inteligência integrada para resolver desafios do negócio, e permite a implementação e treino de modelos de aprendizagem mais complexos (SAP, 2017).

A *Analytics* permite a recolha e análise dos dados que permitem tomar decisões. Esta tecnologia permite a análise dos dados de todo o negócio. A tecnologia *Machine Learning* também está integrada neste sistema. (SAP, 2017).

A *Data Intelligence* permite a inserção dos dados num contexto de negócio. Esta tecnologia usa referências confiáveis e recolhidas em tempo real e desenvolve cenários de tomada de decisão (SAP, 2017).

A *Big Data* permite a gestão de grandes quantidades de dados. Esta tecnologia é responsável por toda a análise dos dados de todo o negócio, permitindo a obtenção de

conhecimento em tempo real e a incorporação de conhecimentos nos processos de negócio (SAP, 2017).

Os serviços *Blockchain* estão integrados nas aplicações do negócio. Esta tecnologia garante uma maior confiança nas transações *peer-to-peer*, uma total visibilidade da proveniência dos bens e histórico de propriedade e maior capacidade de auditoria e menor fraude (SAP, 2017).

3.9 SAP HANA

O SAP HANA (*High-Performance ANalytical Appliance*) é uma aplicação que usa a tecnologia de base de dados *in-memory* que permite o processamento de grandes volumes de dados em tempo real num curto período de tempo (Rouse, 2017; Techopedia, s.d.). O SAP HANA processa os dados que estão armazenados na RAM em vez de ler esses mesmos dados no disco, fazendo com que a leitura de grandes volumes de dados seja feita rapidamente (Farber, et al., 2011; Farber, et al., 2012; Plattner & Zeier, 2011).

O SAP HANA tem uma componente de programação que permite às empresas a criação e execução de aplicações sobre o HANA, como também possibilita a criação de relatórios num intervalo de tempo predefinido. Como o HANA pode ser executado ao mesmo tempo que um aplicativo SAP ERP, os analistas podem aceder aos dados em tempo real e não precisam de esperar pela execução do relatório diário ou semanal (Figura 10) (Farber, et al., 2011; Farber, et al., 2012; Plattner & Zeier, 2011).

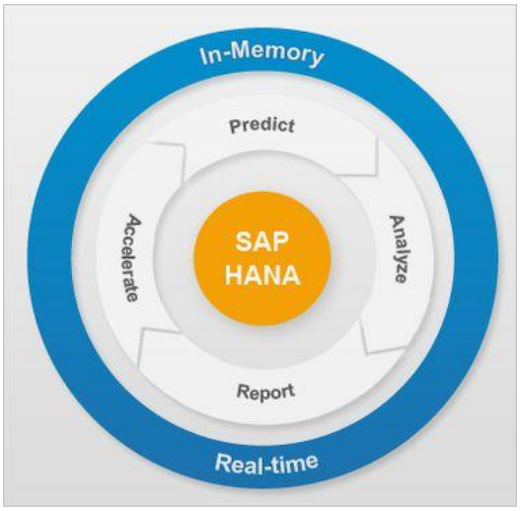


Figura 10 - Funcionamento do SAP HANA (Conquesti, 2018).

Vantagens:

- Por usar tecnologia *in-memory*, o SAP HANA faz um processamento de dados massivo de alta velocidade e permite a análise de todos os dados transacionais;
- A seleção de dados é feita em tempo real, permitindo ao gestor tomar decisões rapidamente;
- O SAP HANA reduz o custo total de propriedade (TCO), pois exige menos testes, manutenção de hardware e reduz os custos numa empresa de tecnologias de informação;
- Ao utilizar soluções inovadoras, o SAP HANA proporciona novos processos e modelos de negócio;
- O SAP HANA pode obter os dados de várias fontes, quer sejam dados SAP ou não SAP;
- É mais fácil fazer a gestão e manipulação dos dados;
- Ajuda a aumentar a receita de uma organização, pois facilita na identificação de oportunidades de vendas lucrativas através da análise dos dados de vendas (STechies, s.d.).

Capítulo 4 | Metodologia

O uso de metodologias de projeto permite que seja mais fácil o planeamento, execução, monitorização e controlo do projeto. Isto permite uma maior eficiência no desenvolvimento de projetos, sendo extremamente vantajoso para as empresas (Abreu, s.d.).

Existem vários tipos de metodologias, e a sua utilização depende de projeto para projeto. Tanto o gestor do projeto como o cliente definem a metodologia a ser utilizada, existindo alguns aspetos a serem levados em consideração como os objetivos do projeto, prazos para a finalização do mesmo, riscos envolvidos, custos, entre outros.

Nos projetos realizados foram utilizadas tanto as metodologias ágeis como as tradicionais, sendo que as metodologias mais utilizadas no desenvolvimento de soluções SAP são o SCRUM e o *WATERFALL* (Abreu, s.d.; Projectbuilder, 2017; Vinicius, 2016).

4.1 Metodologias Ágeis e Tradicionais

Existe uma grande diferença entre as metodologias ágeis e as tradicionais, mas ambas têm como objetivo a otimização dos projetos, sendo que ambas são de igual importância para as empresas (Vinicius, 2016).

As metodologias ágeis focam-se muito no trabalho em equipa, fazendo com que todos os membros da equipa se sintam ligados ao projeto e que sentem que são uma parte importante do mesmo. Como equipa, os mesmos têm liberdade para tomar decisões em conjunto e existe uma grande flexibilidade em efetuar alterações ao projeto, mesmo que o mesmo esteja em fase final (Projectbuilder, 2017).

A comunicação da equipa é fluida e auto organizável, ou seja, a divisão do trabalho é efetuada através do entendimento entre os membros da equipa. Normalmente, com o uso deste tipo de metodologia, as equipas são pequenas e multidisciplinares, fazendo com que exista uma menor formalidade na comunicação (Abreu, s.d.; Projectbuilder, 2017; Vinicius, 2016).

O projeto é entregue em partes de forma contínua, em que essas mesmas partes contêm um conjunto mínimo de funcionalidades, de forma a atender as necessidades do cliente. Ao optar pela entrega em partes, o cliente tem a possibilidade de acompanhar todo o

desenvolvimento do projeto, tendo a oportunidade de ter uma participação mais ativa no desenvolvimento do mesmo (Abreu, s.d.; Projectbuilder, 2017; Vinicius, 2016).

Nas metodologias tradicionais, o planeamento do projeto é feito de forma mais rígida, e as decisões são tomadas numa abordagem *top-down*, focando-se bastante nos padrões de boas práticas. O uso deste tipo de metodologias faz com que seja mais difícil a realização de alterações no projeto, pois existe muita burocracia para fazer essas mesmas alterações (Abreu, s.d.; Projectbuilder, 2017; Vinicius, 2016).

Grande parte da comunicação passa pelo Gestor de Projeto, acabando por ser a figura central de todo o desenvolvimento do projeto, ao fazer a distribuição das tarefas e explicar como estas devem ser feitas, não existindo grande debate das tarefas entre a equipa. O projeto é entregue somente nas fases finais de desenvolvimento (Abreu, s.d.; Projectbuilder, 2017; Vinicius, 2016).

Tabela 3 - Diferenças entre as metodologias tradicionais e as metodologias ágeis (Moura, 2016).

Metodologias Tradicionais	**Metodologias Ágeis**
Planeamento rígido.	Maior liberdade no planeamento das ações.
Resistência a mudanças.	Flexibilidade e uma postura positiva diante da necessidade de mudanças (mesmo em fases finais do projeto).
Decisões tomadas numa abordagem *top-down*.	Liberdade para tomar decisões em equipa.
Forte centralização em torno da figura do Gestor de Projeto.	Responsabilidade compartilhada entre todos os membros da equipa e espírito de colaboração.
Uma liderança que monopoliza toda a comunicação já que está preocupada com o comando e o controlo detalhado das ações.	Comunicação fluída e livre entre os membros da equipa.
Líderes indicam “O que fazer”, “Como”, em vez de dizer “Porquê”	Equipas auto-organizáveis; a divisão do trabalho é resultado do entendimento do projeto e de um consenso entre a equipa.
Problemas geralmente escalados até à gestão.	Atuação conjunta da equipa para a resolução de problemas.
O planeamento costuma prever um trabalho extenso, com a entrega do produto somente nos estágios finais do cronograma (o que leva a muitos conflitos com o cliente).	Entregas de partes do projeto de forma contínua e incremental (iterações), a fim de se obter um rápido feedback do cliente acerca do andamento do projeto.
Uma longa fase de análise; em muitos casos, parte da equipa é deixada de lado nestes estágios iniciais (já que considera que tais membros ingressarão apenas na fase de execução).	Reuniões diárias entre a equipa; o intuito está em discutir o que será feito naquele momento, revendo o planeamento a médio e curto prazo, além de prováveis impedimentos.

Um grande foco na criação de documentos.	Embora existam documentos e se estimule a criação dos mesmos, existe um pragmatismo maior.
Um maior foco nos processos do que no produto esperado.	Menos formalidade e maior ênfase em se chegar ao produto esperado.
Maior envolvimento do cliente em estágios iniciais, com um certo relaxamento de postura assim que o projeto tenha início.	Participação ativa do cliente, inclusive enquanto o projeto está a ser implementado.
Foco na "antecipação" (tarefa difícil num ambiente sempre sujeito a mudanças repentinas).	Ênfase na "adaptação" ("jogo de cintura").

4.2 Metodologia SCRUM

A Metodologia SCRUM é uma metodologia ágil para gestão de projetos e desenvolvimento de software (Scruminc, 2018). No SCRUM, os projetos são divididos em ciclos, denominados por *Sprint*. Cada *sprint* representa um conjunto de atividades que têm de ser desempenhadas para a realização do projeto (Rouse, 2017). Esta Metodologia enquadra-se muito numa expressão popular (bastante utilizada no mundo da programação) que é "*Divide to Conquer*", que basicamente consiste na divisão de um problema em muitos problemas mais pequenos e de fácil resolução. Desta forma, é mais fácil organizar e distribuir trabalho, como também aumenta o rendimento da equipa.

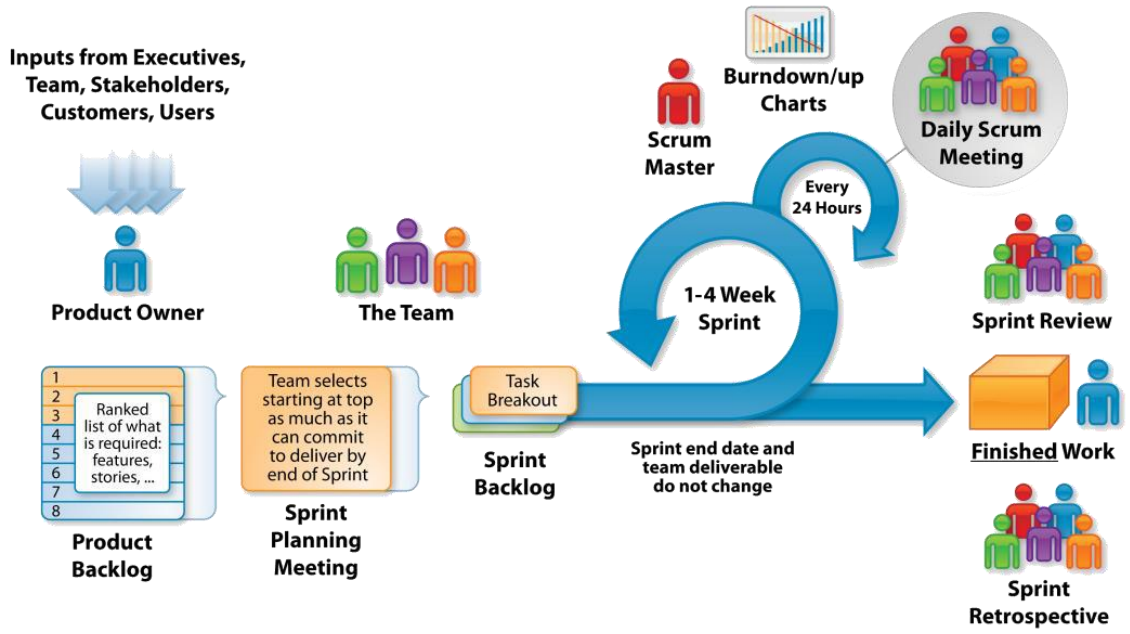


Figura 11 - Diagrama de funcionamento da metodologia SCRUM (Kisspng, 2017).

4.2.1 Componentes da Metodologia SCRUM

A Metodologia SCRUM é constituída por dois artefactos (SCRUM *Artifacts*), três papeis (SCRUM Roles) e quatro tipos de reuniões (Scruminc, 2018). Os artefactos são "ferramentas" (métodos) que apoiam na resolução de um problema. Os papeis são as funções dentro do SCRUM e por fim existem quatro tipos de reuniões, cada reunião com um foco específico.

SCRUM Artifacts

Os dois tipos de SCRUM Artifacts que existem são o *Backlog* do Produto e o *Sprint Backlog*. O *Backlog* do Produto (*Product Backlog*) é uma lista de tarefas associadas ao projeto (Albarqi & Rizwan, 2018). Esta lista pode conter melhorias, características, defeitos, entre outros, e cada tarefa tem um ID único, consoante a sua categoria (melhoria, defeito, entre outros). O *Backlog* do Produto não precisa de estar completo no início do projeto pois o mesmo cresce durante o desenvolvimento do produto. O conteúdo desta lista é definido pelo dono do produto (*Product Owner*) e pelo Gestor do Projeto (Desenvolvimentoagil, 2014).

O *Sprint Backlog* contém todas as tarefas a serem desenvolvidas pela equipa até ao final da *Sprint* (Albarqi & Rizwan, 2018). Os itens do *Sprint Backlog* são extraídos do *Product Backlog* com base nas prioridades definidas pelo *Product Owner*. Cabe à equipa determinar os itens a desenvolver. Uma vez escolhidos, a equipa têm de se comprometer a finalizá-los até ao final do tempo definido (Desenvolvimentoagil, 2014).

SCRUM Roles

Os três papeis (roles) que existem no SCRUM são: *Product Owner*, SCRUM *Master* e os trabalhadores que constituem as equipas. O Dono do Produto (*Product Owner*) é o membro mais importante da equipa de SCRUM que tem como responsabilidade facilitar a comunicação entre a equipa e o cliente. É ele que define as prioridades no *Backlog* do Produto (Almutairi & Qureshi, 2015; Ashraf & Aftab, 2017).

O SCRUM *Master* é um líder, e tem a responsabilidade de verificar se a equipa está a trabalhar conforme o planeado, tendo também a responsabilidade de remover possíveis obstáculos que possam aparecer durante a realização do projeto (Scruminc, 2018).

Os trabalhadores que constituem as equipas desempenham um papel fundamental. Devem ser polivalentes e capazes de se organizarem sem intervenção superior. O número de membros da equipa deve variar entre quatro e nove (Scruminc, 2018).

SCRUM Meeting

Existem quatro tipos de reuniões na metodologia SCRUM: O Planeamento da *Sprint* (*Sprint Planning*) onde a equipa decide quais são as tarefas que devem ser realizadas na *sprint* e a duração das mesmas (Scruminc, 2018).

No SCRUM Diário (*Daily* SCRUM) é feito um relatório diariamente, onde se debate o ponto de situação da *sprint*, o que foi feito ontem, o que vai ser feito hoje e o que será feito amanhã.

A Revisão da *Sprint* (*Sprint Review*) é realizada no final de cada *sprint*, em que é apresentado o que foi realizado durante a *sprint* e se obtém o *feedback* do produto realizado (Desenvolvimentoagil, 2014).

A *Sprint* Retrospetiva (*Sprint Retrospective*) é realizada no final de cada *sprint* para reunir o *feedback* da equipa e verificar o que funcionou bem e possíveis (Scruminc, 2018).

Tabela 4 - Vantagens e desvantagens da metodologia SCRUM (Chandana, 2018).

Vantagens
Uso eficaz do tempo e dinheiro.
Grandes projetos são divididos em *sprints*, permitindo uma gestão mais fácil.
Os desenvolvimentos são testados durante a revisão da *sprint.*
Funciona bem para projetos de desenvolvimento rápido.
O SCRUM, como metodologia ágil, adota *feedback* dos clientes e *stakeholders*.
O esforço individual de cada membro da equipa é visível durante as reuniões diárias.
Desvantagens
O SCRUM pode levar a falhas nos prazos, por não ter uma data de entrega fixa.
Se os membros da equipa não forem comprometidos e cooperativos, o projeto pode fracassar.
Adotar o SCRUM em grandes equipas é um desafio.
O desenvolvimento do projeto pode apenas ter sucesso se os membros forem experientes.
As reuniões diárias podem aumentar a frustração de membros da equipa.
Se algum membro da equipa sair a meio do projeto pode ter um enorme impacto negativo no projeto.

4.3 Metodologia *WATERFALL*

Uma metodologia tradicional bastante usada para o desenvolvimento de *software* é o *Waterfall*. É uma das metodologias mais antigas e tem como grande foco o planeamento de todo o projeto nos estágios iniciais (Baba & Nonyelum, 2017; Munassar & Govardhan, 2010). Como tal, este modelo é estático, contém uma calendarização de todo o projeto, sendo possível fazer o desenvolvimento do mesmo de uma forma linear e sequencial, pois só se inicia a atividade seguinte depois de terminada a atividade anterior (Adenowo & Adenowo, 2013).

Como é feito um planeamento intensivo logo de início, esta metodologia funciona bem em desenvolvimentos onde o controlo de qualidade é uma preocupação constante, sendo muito utilizada em projetos governamentais e em grandes empresas (Baba & Nonyelum, 2017; Munassar & Govardhan, 2010) (Figura 12).

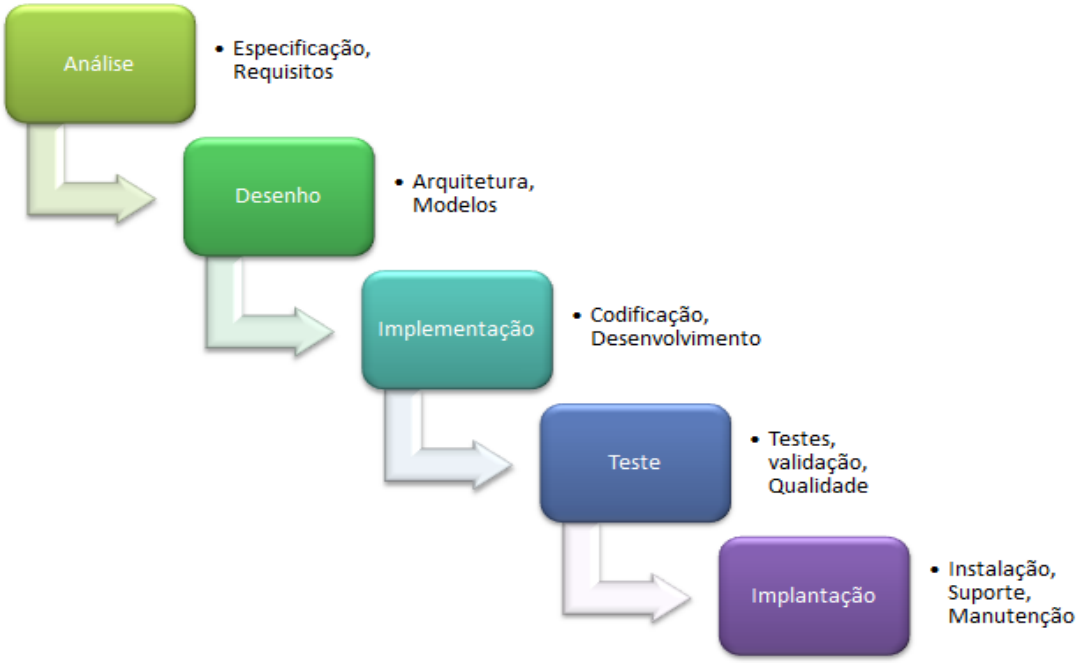


Figura 12 - Diagrama de funcionamento da metodologia *Waterfall* (Universo-Projeto, 2014).

A Metodologia *Waterfall* é constituída por cinco fases: Análise, *Design*, Implementação, Testes e Manutenção (Bassil, 2012). Este tipo de Metodologia normalmente tem objetivos distintos para cada fase. Quando é terminada uma fase, é iniciada a próxima fase e não existe oportunidade para voltar à fase anterior. Esta metodologia suporta uma abordagem estruturada e focalizada no processo (Adenowo & Adenowo, 2013). Fowler e

Scott (1999) destacam que existem transferências entre fases e a existência de *backflows*, mas os mesmos devem ser evitados. Durante o desenvolvimento, se for detetado um erro numa fase que já tenha sido concluída, normalmente não existe oportunidade de "voltar" à fase anterior (Adenowo & Adenowo, 2013).

4.3.1 Fases da Metodologia *WÁTERFALL*

A fase de Análise, conhecida como Requisitos de *Software* é quando se faz uma análise e descrição do comportamento do projeto a ser desenvolvido.

Nesta fase, os analistas de sistemas e o cliente definem os requisitos funcionais e não funcionais do projeto. Normalmente, os requisitos funcionais são representados através de *Use Cases* que descrevem as interações dos utilizadores com o *software*. Os requisitos não funcionais consistem nos critérios, restrições, limitações e requisitos ao projeto e execução do projeto.

A fase de *Design* consiste no processo de planeamento da solução. Nesta fase, os consultores e *designers* definem o plano para uma solução, que inclui o algoritmo, arquitetura, estrutura base de dados e interface gráfica para o utilizador.

A fase de Implementação consiste na realização das especificações do projeto num programa executável, base de dados e, *website* através de programação. É nesta fase que o código é escrito e compilado numa aplicação e é criada a base de dados (Bassil, 2012).

A fase de Testes consiste na verificação e validação da solução desenvolvida. É necessário saber se atende aos requisitos e especificações que foram definidas e se cumpre o seu propósito. É nesta fase onde são feitos muitos testes para depurar bugs e falhas no sistema, de forma a corrigir e melhorar o mesmo, de acordo com o pretendido (Institute of Electrical and Electronics Engineers, 1991).

A fase de Manutenção é o processo de modificação da solução após a entrega, para corrigir erros, melhorar o desempenho e qualidade e refinar outros aspetos importantes do funcionamento geral. Poderão ser feitas melhorias à solução, com a definição de novos requisitos, aumentando as funcionalidades da solução (Stellman & Greene, 2005).

Tabela 5 - Vantagens e desvantagens da metodologia *Waterfall* (Alshamrani & Bahattab, 2015).

Vantagens
Fácil de entender e implementar.
Amplamente utilizada e conhecida.
É definido antes da realização do design e é realizado o design antes de desenvolver.
Funciona bem em produtos maduros.
Minimiza a sobrecarga de planeamento.
As fases são processadas e finalizadas uma de cada vez.
Desvantagens
Todos os requisitos devem ser conhecidos antecipadamente.
Inflexível.
Voltar atrás e corrigir é difícil quando a aplicação esteja em fase de testes.
Apenas uma entrega no final do desenvolvimento.
O cliente pode não ser claro sobre o que ele quer e o que é necessário.
Os clientes podem ter poucas oportunidades de pré-visualizar o sistema, podendo ser demasiado tarde para corrigir possíveis problemas.
Não é uma metodologia muito usada em projetos complexos ou projetos orientados a objetos.
Altos níveis de risco e incertezas, sendo que pequenas alterações podem causar grandes problemas.

Esta metodologia deve ser utilizada quando a qualidade é mais importante do que o custo ou prazos ou quando os requisitos são claros e fixos. Também é uma boa metodologia para ser utilizada caso o projeto consista numa nova versão de um produto existente ou quando se pretende passar o produto para outra plataforma.

Capítulo 5 | Apresentação e descrição das atividades realizadas

5.1 Atividades Realizadas

Durante o estágio foram realizados três tipos de tarefas: implementação de notas; criação e edição de *reports* usando programação ABAP; e o desenvolvimento de aplicações em *Neptune*.

Neste capítulo são apresentados exemplos de quatro projetos desenvolvidos durante o estágio, onde é explicado em que consiste cada projeto, os seus objetivos, e os procedimentos necessários para a sua realização.

Vão ser apresentadas dois projetos baseados em programação *Neptune*, ambos provas de conceito, contudo, um deles foi destinado a um cliente, enquanto que o outro foi destinado a um concurso que a Softinsa venceu.

5.2 Caso 1 | Implementação de Notas

5.2.1 Problema

O Cliente A necessitava que fosse implementado um conjunto de Notas SAFT-PT (*Standard Audit File for Tax Purpose*) para corrigir anomalias no sistema. Para a realização desta tarefa foi estipulado um prazo e foram facultadas as informações da Máquina e VPN do cliente, os dados de autenticação para que o programador acedesse à Máquina e VPN, bem como ao portal (OSS Note) onde é possível analisar as notas e fazer o registo das chaves de objetos. A metodologia utilizada para o desenvolvimento desta tarefa foi o Waterfall.

5.2.2 Análise de Requisitos

O programador acedeu à Máquina do cliente onde foi verificar os dados da máquina, mais propriamente os módulos que a mesma possui, bem como a *release* e o nível do *Support Package* (Figura 13). De acordo com as especificações do cliente, o módulo que iria ser trabalhado era o de Logística, logo o módulo que teve que ser analisado com mais detalhe foi o SAP_APPL.

Komponente	Release	SP-Level	Support Package	Kurzbeschreibung der Komponente
SAP_BS_FND	747	0010	SAPK-74710INSAPBSFND	SAP Business Suite Foundation
WEBCUIF	747	0010	SAPK-74710INWEBCUIF	SAP Web UI Framework
MDG_APPL	617	0010	SAPK-61710INMDGAPPL	MDG Applications 607
SAP_APPL	617	0010	SAPKH61710	Logística e contabilidade
SAP_FIN	617	0010	SAPK-61710INSAPFIN	SAP_FIN

Figura 13 - Informações da máquina do cliente.

De seguida, o programador foi ao portal da SAP (OSS Note), onde pesquisou e analisou a *collective note* de SAFT, que continha a listagem de todas as notas a serem implementadas, consoante as versões das máquinas em questão (Figura 14). Nesta *collective note*, a listagem contém 138 notas.

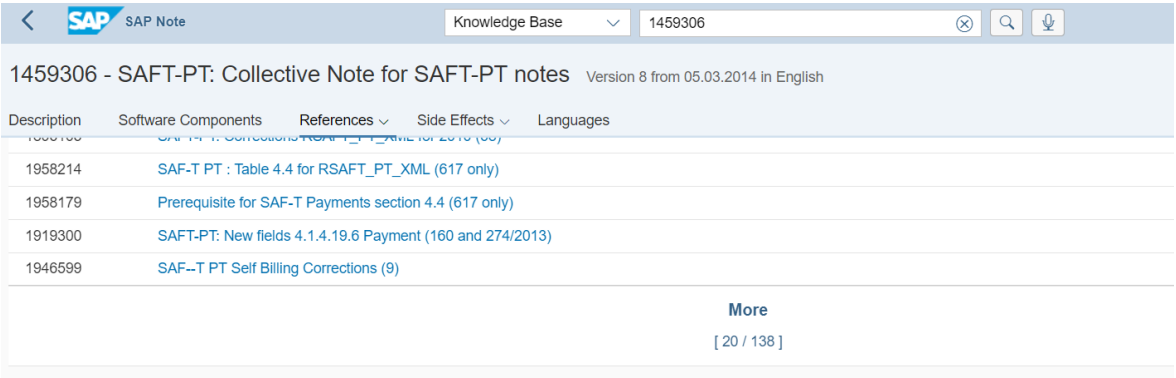


Figura 14 - *Collective Note* do SAFT.

Através da transação SNOTE o programador foi verificar se as notas que estavam na *collective note* se apresentaram implementadas (Figura 15). Caso não estivessem, o programador teria que adicionar a informação dessa nota a uma lista que vai conter, por ordem, todas as notas a serem implementadas (Figura 16). O programador deve inserir na lista não só o número da nota, como a descrição, data da última atualização, versão da nota e a indicação se contém passos manuais.

Nota	Versão	Texto breve	Componente	Status proces	Estado implementação	Nome usuário
2200208	2	SAFT-PT: country is not filled in Ship-to for some deliverie	XX-CSC-PT-LO	em processamento	Completamente implementado	SOFTINSA07
2210956	10	SAFT-PT: deliveries missing in SAFT report	XX-CSC-PT-LO	em processamento	Completamente implementado	SOFTINSA07
2253863	2	SAFT PT : Adjusting the BAdI implementation FIEU_SAFT_XMLGEN	XX-CSC-PT-LO	em processamento	Completamente implementado	SOFTINSA07

Figura 15 - Validação das Notas.

Quando a listagem das notas a serem implementadas ficou preenchida, foi necessário preparar um documento onde foi registada toda a implementação, mais propriamente, todas as notas implementadas, tanto os passos pré e pós Implementação bem como a implementação da Nota.

OSS Notes	Descrição	Released on	Versão	Passos manuais	Dados de Implementação						
					Pedido	Request	Dat	User	Releas	Component
2210956	SAFT-PT: deliveries missing in SAFT report	22.04.2016	Version 10	Sim						XX-CSC-PT-LO
2200208	SAFT-PT: country is not filled in Ship-to for some deliveries	30.11.2015	Version 2	Não						XX-CSC-PT-LO
2253863	SAFT PT : Adjusting the BAdI implementation FIEU_SAFT_XMLGEN_PTMOG	13.03.2017	Version 2	Não						XX-CSC-PT-LO
2306720	SAFT-PT: Issues on missing delivery documents, CustomerId and Quantity in MovementOfGoods section	16.05.2016	Version 6	Sim						XX-CSC-PT-LO
2322043	SAFT PT: Issues with fields Line number, period, MovementStatus and MovementStatusDate in XML file for MovementOfGoods section	27.06.2016	Version 4	Sim						XX-CSC-PT-LO
2298384	New method in BADI SIPT_BADI_SAFT_MOVE_GOODS to change the signature information for extrernal delivery documents	20.06.2016	Version 1	Não						XX-CSC-PT-LO
2330087	SAFT-PT:BAdI to change the signature info for external delivery documents in MOG section	20.06.2016	Version 1	Não						XX-CSC-PT-LO
2330820	SAFT-PT: Issue with MovementStartTime in MOG	25.07.2016	Version 2	Não						XX-CSC-PT-LO
2333238	SAFT-PT: Issue with unit price and movement status date in MOG section	11.01.2017	Version 3	Sim						XX-CSC-PT-LO

Figura 16 - Exemplo de listagem de Notas a serem implementadas.

5.2.3 Resolução

O programador abriu a nota a ser implementada no portal da SAP (Figura 17) e verificou se tinha passos manuais para realizar, entre outras informações.

2210956 - SAFT-PT: deliveries missing in SAFT report Version 10 from 22.04.2016 in English

Component: XX-CSC-PT-LO
Priority: Correction with medium priority
Category: Program error
Release Status: Released for Customer
Corrections: 10
Manual Activities: 1
Prerequisites: 6
SAP Note/KBA Number

Description | Software Components | Corrections | Support Packages | Attachments | Languages

Symptom

SAFT-PT: You are creating SAFT XML file for country Portugal but some deliveries are missing that need to be reported in the XML file and ProductCode is not filled with leading zeros

Figura 17 - Verificação da existência de passos manuais na Nota.

Caso a nota tenha passos manuais para realizar, os mesmos estão descritos no final da página que contem a informação da mesma (Figura 18).

2210956 - SAFT-PT: deliveries missing in SAFT report Version 10 from 22.04.2016 in English

Description | Software Components | Corrections | Support Packages | Attachments | Languages

Manual Activities

```
-----------------------------------------------------------------------
|Manual Activity                                                      |
-----------------------------------------------------------------------
|VALID FOR                                                            |
|Software Component   SAP_APPL                    SAP Application   |
| Release 600          SAPKH60001 - SAPKH60028                        |
| Release 602          Until SAPKH60218                               |
| Release 603          Until SAPKH60317                               |
| Release 604          SAPKH60401 - SAPKH60418                        |
| Release 605          Until SAPKH60515                               |
| Release 606          SAPKH60601 - SAPKH60616                        |
| Release 616          Until SAPKH61609                               |
| Release 617          SAPKH61701 - SAPKH61711                        |
| Release 618          Until SAPK-61802INSAPAPPL                      |
-----------------------------------------------------------------------

Please implement the manual steps that are attached in the document  '2210956_premanualsteps.docx' before implementing the correction instructions of the note
```

Figura 18 - Passos manuais de uma Nota.

Na Figura 18, pode-se verificar que os passos manuais são válidos apenas para certas máquinas. A máquina utiliza o *Support Package* SAPKH61710 (Figura 13), e como se pode verificar, na linha onde o Componente é o 617, o *Support Pack*age está entre o SAPKH61701 e SAPKH61711, logo, os passos manuais são válidos para a máquina.

Neste exemplo os passos manuais estão contidos num documento em anexo. Ao fazer o download do ficheiro, o programador conseguiu visualizar as alterações que tinha que realizar (Figura 19).

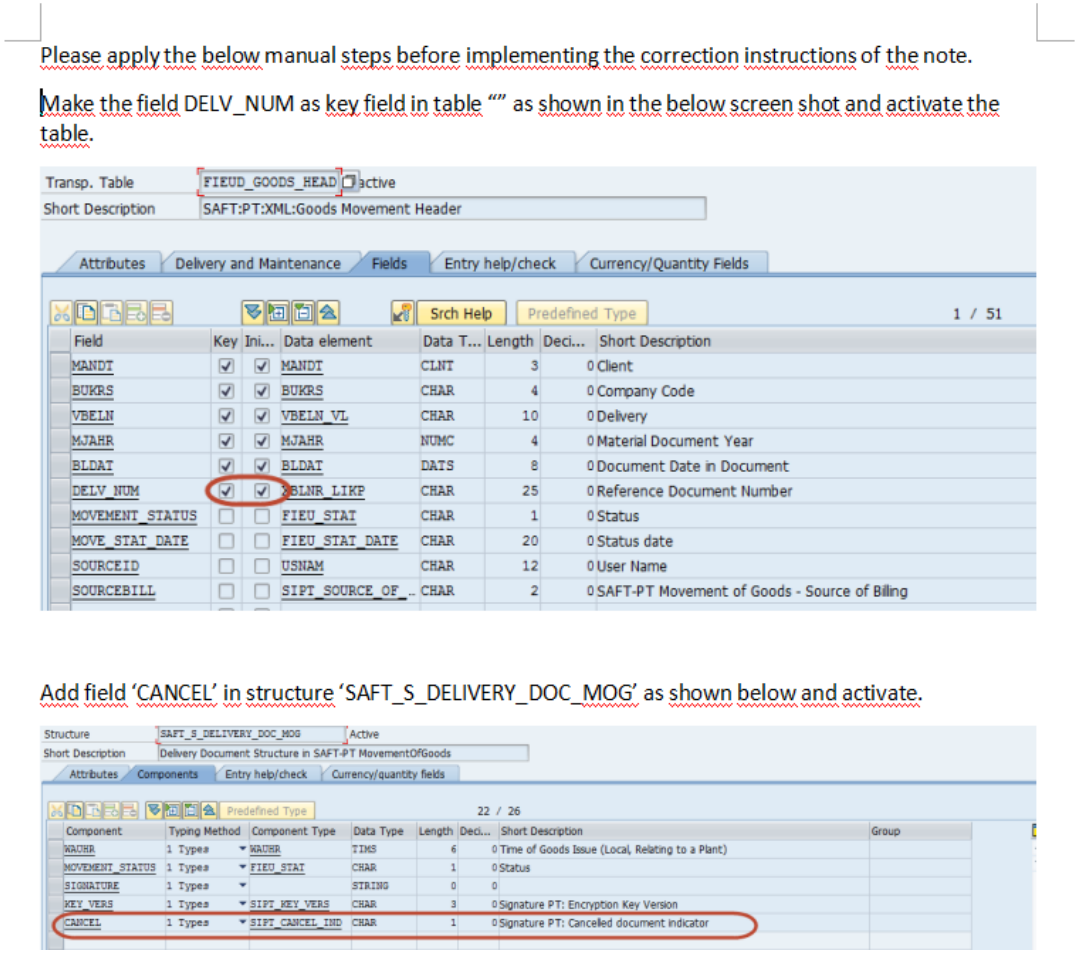


Figura 19 - Detalhes dos passos manuais de uma Nota.

O programador teve que aceder à transação SE11 (onde os objetos de dicionário de dados são mantidos) e inserir no campo "*Tab*. Banco de Dados" (*Database Table*) o nome "FIEUD_GOODS_HEAD" e entrar nessa tabela (Figura 20). Contudo para efetuar a alteração, o mesmo teve que fazer o registo de objetos no portal da SAP (Figura 21).

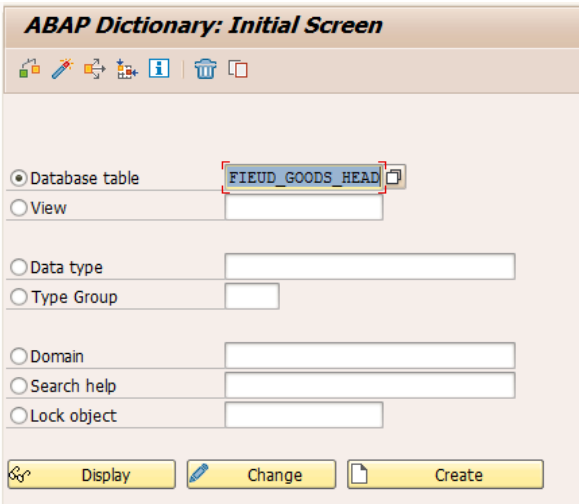


Figura 20 - Acesso aos campos de uma tabela.

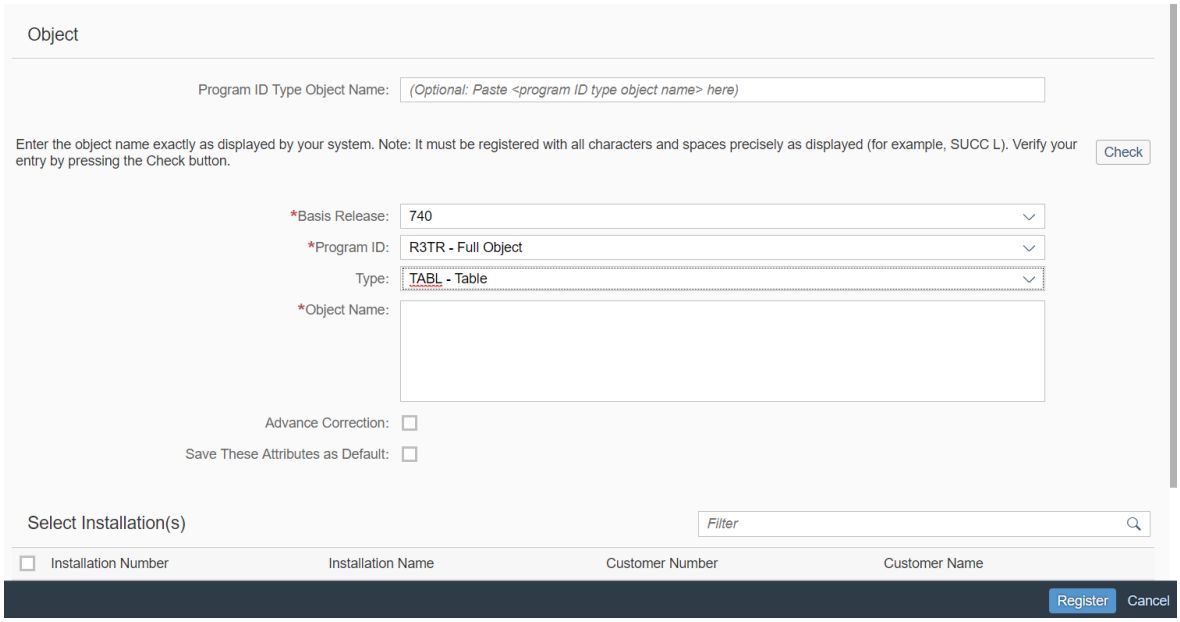


Figura 21 - Zona de registo de objetos.

O programador ao fazer o registo, gerou um número (Figura 22). Antes de prosseguir, o programador teve que registar os dados do registo de objetos que efetuou num documento à parte (Figura 23).

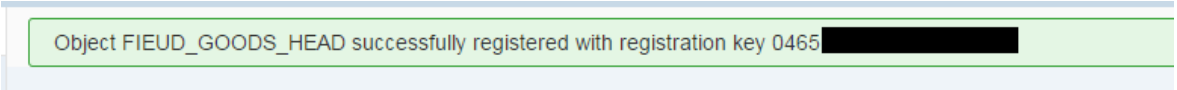


Figura 22 - Chave do objeto gerado.

ID	Instalation	SAP Release	OSS Note	Object Ctg	Object Type	Object name	Registration Key
		740	2210956	R3TR	TABL	FIEUD_GOODS_HEAD	0465[redacted]
		740	2210956	R3TR	TABL	FIEUD_GOODS_ITEM	1392[redacted]
		740	2210956	R3TR	TABL	SAFT_S_DELIVERY_DOC_MOG	4085[redacted]
		740	2210956	R3TR	TTYP	SAFT_TT_DELIVERY_DOC_MOG	1790[redacted]

Figura 23 - Tabela onde são guardados os registos de objetos.

O programador inseriu a chave do registo de objetos no *popup* que apareceu quando o mesmo tentou efetuar alterações na tabela FIEUD_GOODS_HEAD (Figura 24).

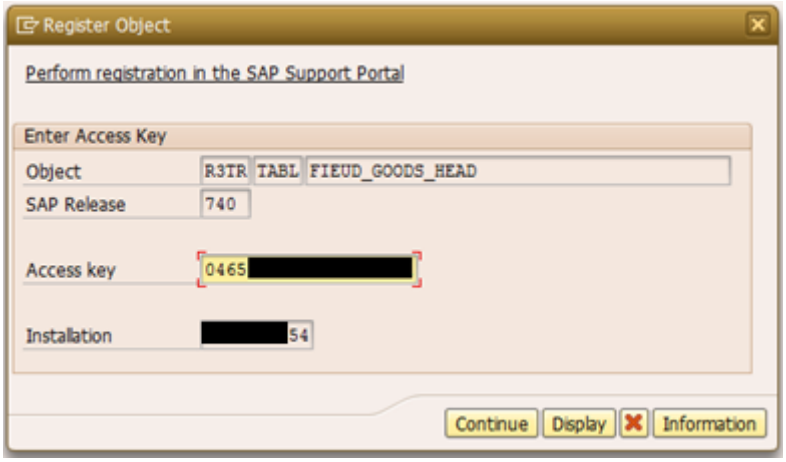


Figura 24 - Inserção da chave gerada no registo.

Depois de inserir a chave, o programador pôde efetuar as alterações necessárias. De acordo com os passos manuais, o programador teve que alterar o estado da *checkbox* do campo "DELV_NUM" (Figura 25). Ao efetuar essa alteração, o programador guardou as alterações.

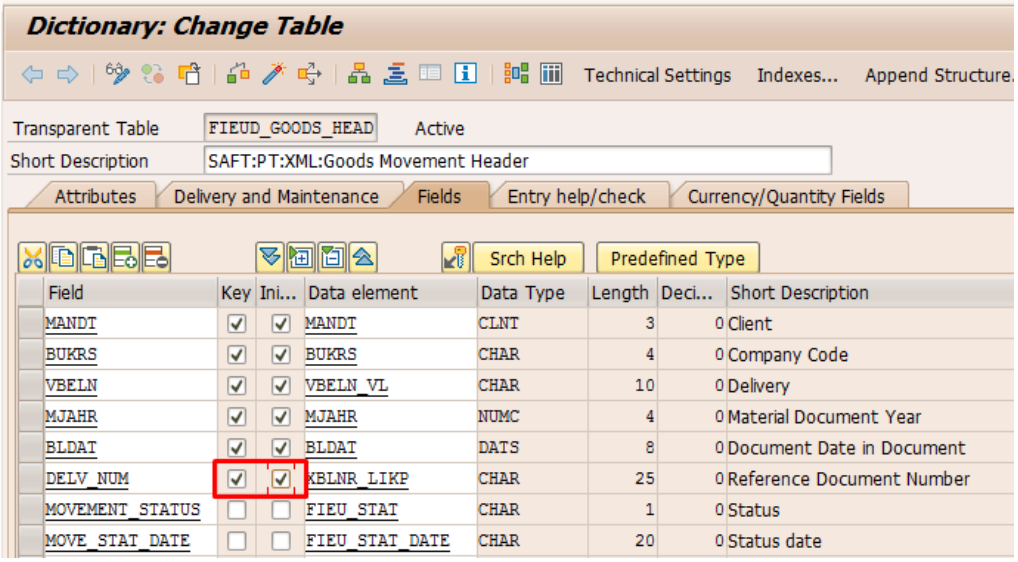


Figura 25 - Alterações realizadas com base na Nota.

O programador também poderia inserir um novo campo na estrutura. Para isso, entrava na estrutura pretendida (Figura 26) (neste caso não pediu registo de objetos porque já não era a primeira vez que o utilizador alterava esta estrutura) e inseria o novo campo (Figura 27).

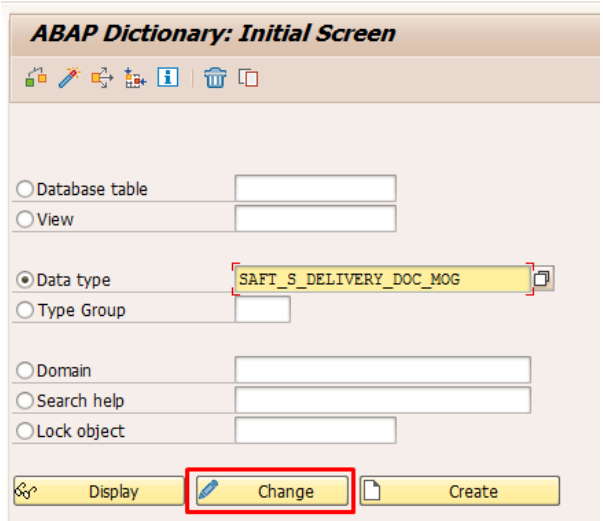
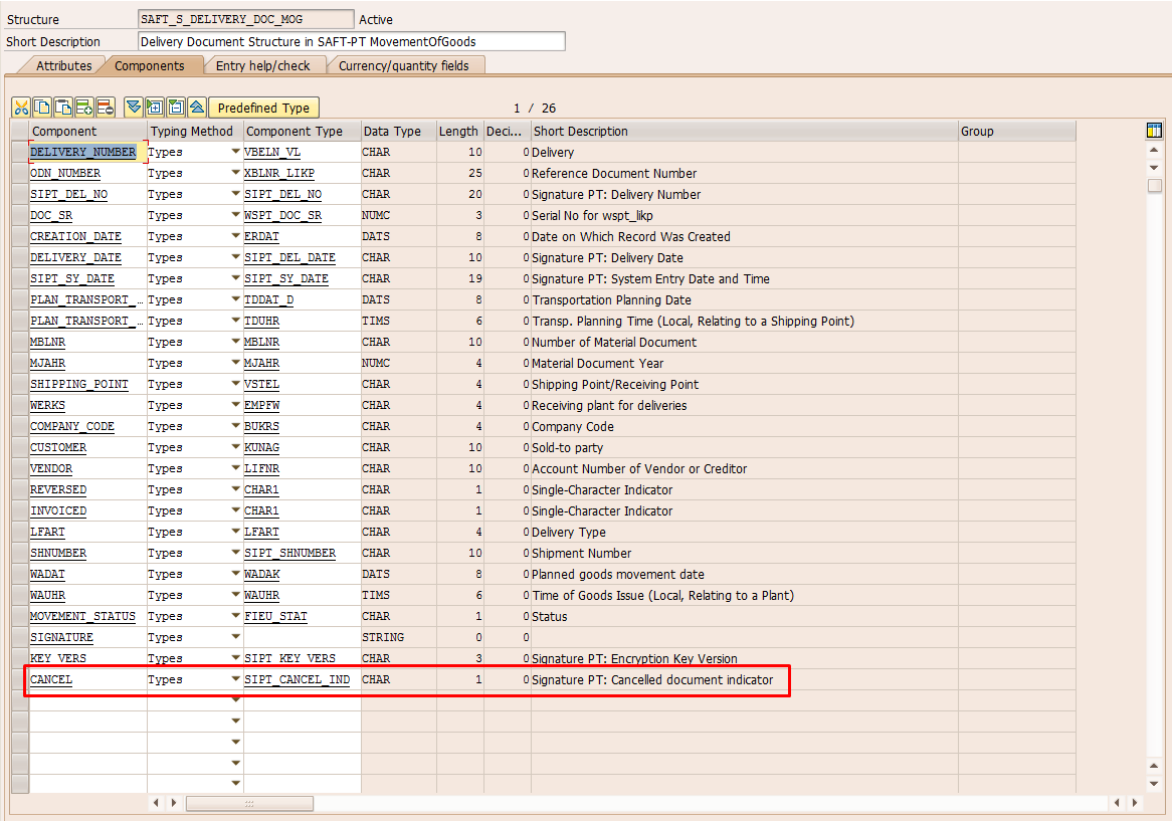


Figura 26 - Acesso à estrutura que é necessário alterar.



Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description	Group
DELIVERY_NUMBER	Types	VBELN_VL	CHAR	10	0	Delivery	
ODN_NUMBER	Types	XBLNR_LIKP	CHAR	25	0	Reference Document Number	
SIPT_DEL_NO	Types	SIPT_DEL_NO	CHAR	20	0	Signature PT: Delivery Number	
DOC_SR	Types	WSPT_DOC_SR	NUMC	3	0	Serial No for wspt_likp	
CREATION_DATE	Types	ERDAT	DATS	8	0	Date on Which Record Was Created	
DELIVERY_DATE	Types	SIPT_DEL_DATE	CHAR	10	0	Signature PT: Delivery Date	
SIPT_SY_DATE	Types	SIPT_SY_DATE	CHAR	19	0	Signature PT: System Entry Date and Time	
PLAN_TRANSPORT_...	Types	TDDAT_D	DATS	8	0	Transportation Planning Date	
PLAN_TRANSPORT_...	Types	TDUHR	TIMS	6	0	Transp. Planning Time (Local, Relating to a Shipping Point)	
MBLNR	Types	MBLNR	CHAR	10	0	Number of Material Document	
MJAHR	Types	MJAHR	NUMC	4	0	Material Document Year	
SHIPPING_POINT	Types	VSTEL	CHAR	4	0	Shipping Point/Receiving Point	
WERKS	Types	EMPFW	CHAR	4	0	Receiving plant for deliveries	
COMPANY_CODE	Types	BUKRS	CHAR	4	0	Company Code	
CUSTOMER	Types	KUNAG	CHAR	10	0	Sold-to party	
VENDOR	Types	LIFNR	CHAR	10	0	Account Number of Vendor or Creditor	
REVERSED	Types	CHAR1	CHAR	1	0	Single-Character Indicator	
INVOICED	Types	CHAR1	CHAR	1	0	Single-Character Indicator	
LFART	Types	LFART	CHAR	4	0	Delivery Type	
SHNUMBER	Types	SIPT_SHNUMBER	CHAR	10	0	Shipment Number	
WADAT	Types	WADAK	DATS	8	0	Planned goods movement date	
WAUHR	Types	WAUHR	TIMS	6	0	Time of Goods Issue (Local, Relating to a Plant)	
MOVEMENT_STATUS	Types	FIEU_STAT	CHAR	1	0	Status	
SIGNATURE	Types		STRING	0	0		
KEY_VERS	Types	SIPT_KEY_VERS	CHAR	3	0	Signature PT: Encryption Key Version	
CANCEL	Types	SIPT_CANCEL_IND	CHAR	1	0	Signature PT: Cancelled document indicator	

Figura 27 - Linha adicionada na estrutura.

Depois de realizar todos os passos pré-implementação, o programador acedeu à transação SNOTE para implementar a nota (Figura 28). O botão que está sinalizado na Figura 28 foi onde o utilizador clicou para implementar a nota.

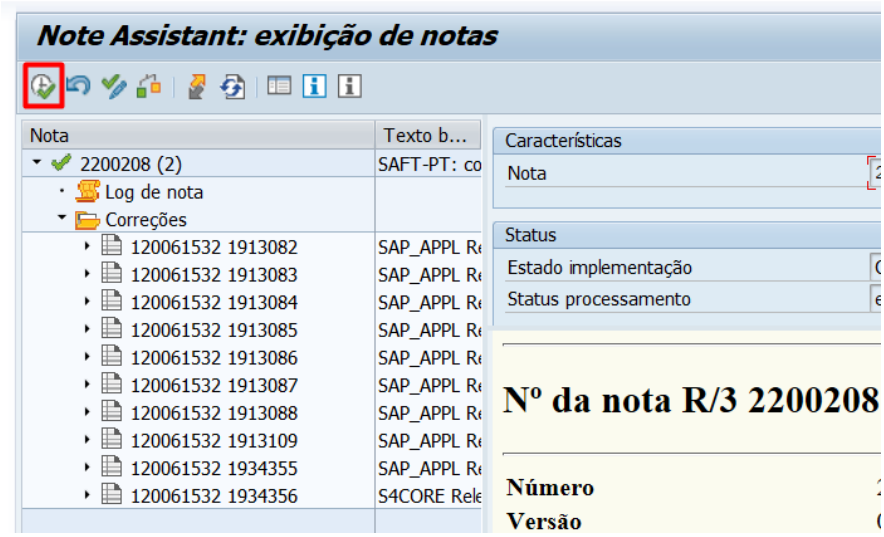


Figura 28 - Transação para a implementação de Notas.

Ao dar início à implementação da nota, surgiu um ecrã com todos os objetos que iriam ser alterados com a implementação da nota (Figura 29). Nesse ecrã poderá ser possível validar se existe algum problema na implementação (Status verde está tudo bem, amarelo contém avisos, vermelho contém erros). Depois de confirmadas as alterações, aparece um novo ecrã (Figura 30) com os objetos que necessitam de ser ativados.

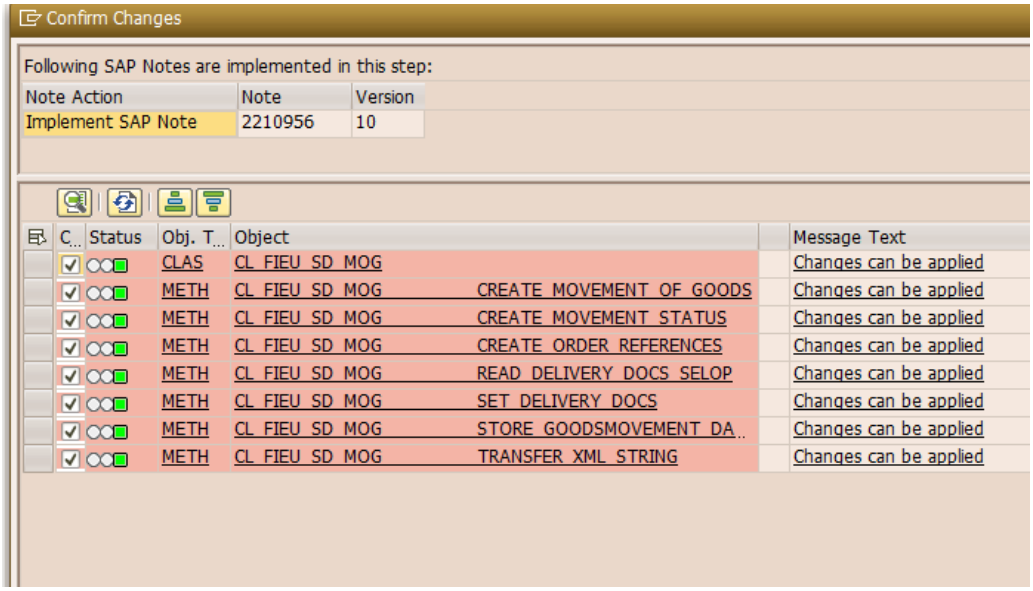


Figura 29 - Verificação do estado dos objetos a serem implementados.

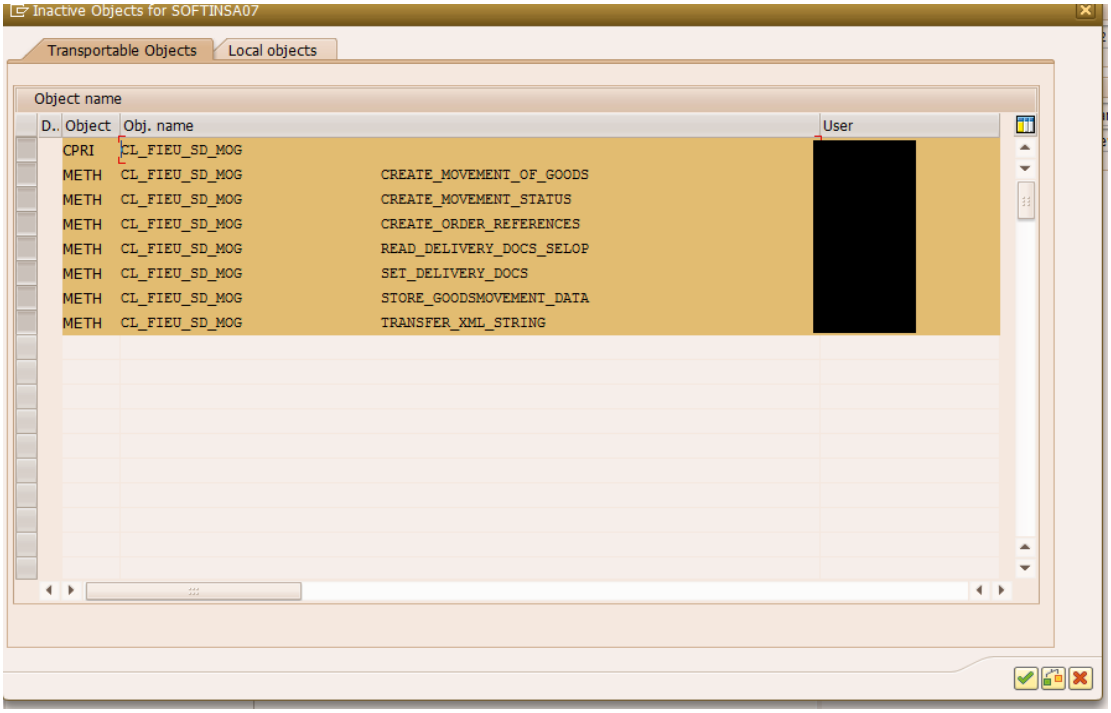


Figura 30 - Ativação dos objetos.

Por fim surge a informação de que a nota foi implementada com sucesso ou os erros de implementação (Figura 31).

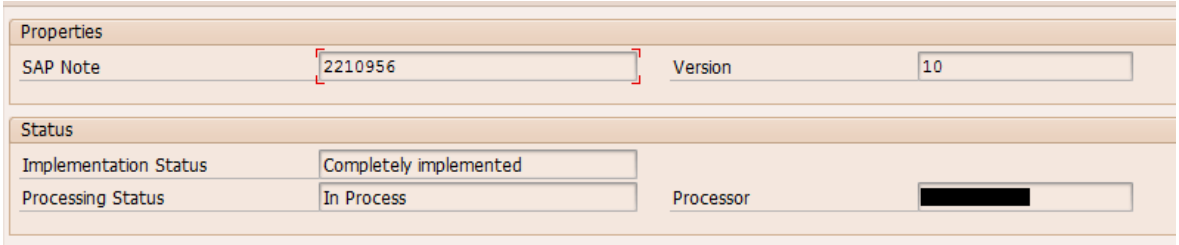


Figura 31 - Informação de que a Nota foi implementada.

Caso a nota tenha passos pós-implementação, é necessário efetuar esses passos antes de implementar outra nota.

Este processo deve ser repetido até serem implementadas todas as notas da listagem.

5.2.4 Conclusão

Depois da implementação das notas, o *Key User* do lado do cliente foi informado de que as alterações foram realizadas e que as mesmas foram guardadas nas ordens de transporte. É facultada a lista de todas as notas implementadas e o registo de objetos que foi necessário efetuar.

O documento de implementação, que contém todos os *prints* da implementação ficou guardado num repositório, pois caso exista algum problema com essa implementação, será sempre possível verificar as alterações efetuadas.

5.3 Caso 2 | Criação de um Report

5.3.1 Problema

O segundo projeto consiste na criação de um programa que permita a consulta de dados dos documentos de uma empresa. O programa tem que apresentar um ecrã de seleção, onde poderá ser feita a filtragem dos valores a apresentar na lista. Ao apresentar a lista, deverá ser possível apresentar os detalhes do documento e a data. A metodologia utilizada para o desenvolvimento desta tarefa foi o Waterfall.

5.3.2 Análise de Requisitos

O programa apresentava as seguintes especificações:

1 – Deverá ser possível filtrar documentos por: Empresa, Número de Documento, Exercício e Data do Documento;

2 – Poderá ser possível limitar o número de linhas da lista a apresentar.

3 – Só depois de uma confirmação é que é possível gerar a lista;

4 – Ao selecionar o Número do Documento na lista, o utilizador deverá ser redirecionado para a transação FB03 (transação usada para consultar os dados de um documento) com os dados do documento.

5 – Ao selecionar a data, é apresentado ao utilizador um *popup* informativo.

5.3.3 Resolução

Foi criado um *Report* (Programa) com o nome Z_ALV1 com três *includes* dentro desse programa (Figura 32):

Include Z_ALV1_TOP – Onde foram definidas todas as variáveis, bem como as tabelas e estruturas internas.

Include Z_ALV1_SCREENS – Onde foram definidos os campos de seleção do ecrã inicial;

Include Z_ALV1_FORMS – Onde foi programado grande parte do código ABAP, dividido em sub-rotinas.

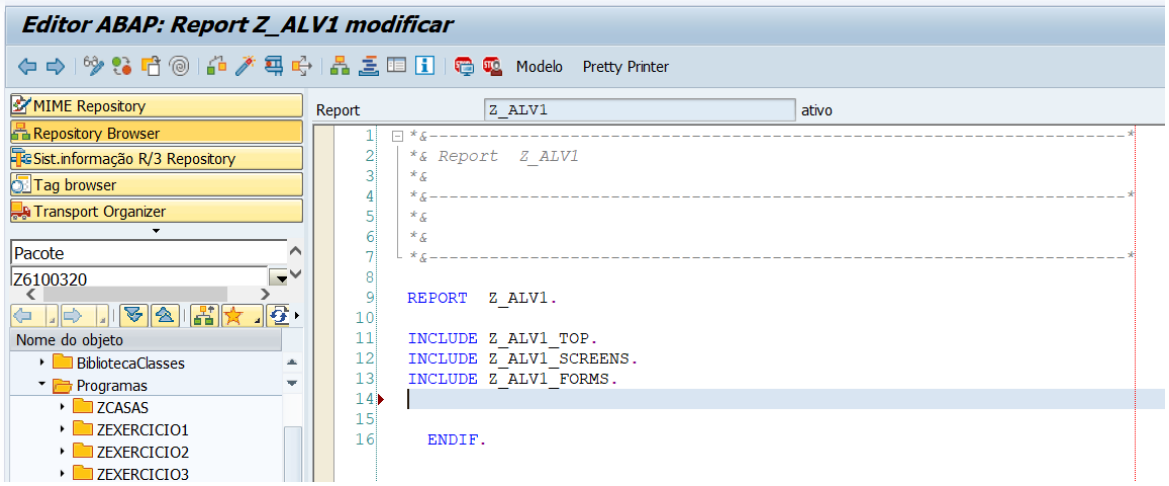


Figura 32 - Estrutura base do programa Z_ALV1.

Foram definidas variáveis, tabelas, estruturas, parâmetros, entre outros, dentro do Include Z_ALV1_TOP à medida que o programa foi desenvolvido, de acordo com as necessidades do programador (Figura 33).

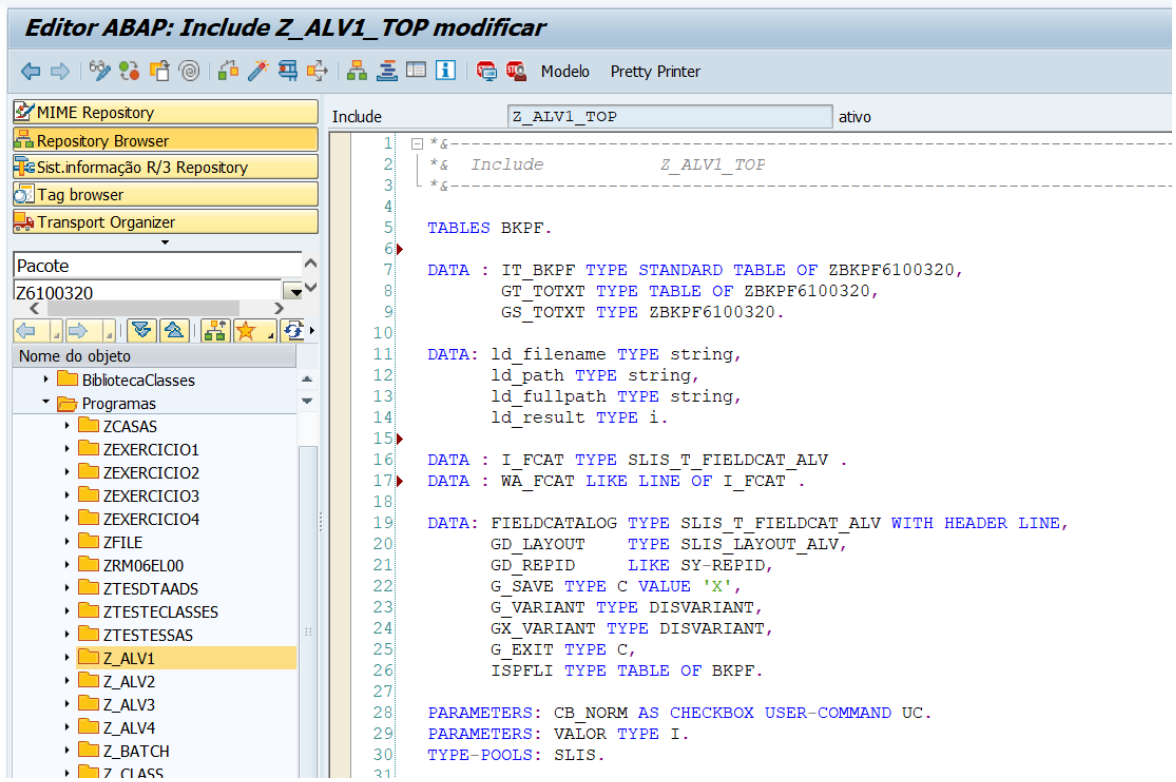


Figura 33 - Include Z_ALV1_TOP.

No Include Z_ALV1_SCREENS foram definidos todos os campos de seleção. Como os dados estão guardados na tabela BKPF (tabela onde são guardados os dados de cabeçalho dos documentos financeiros), é necessário consultar essa tabela para obter os nomes dos campos a usar (Figura 34).

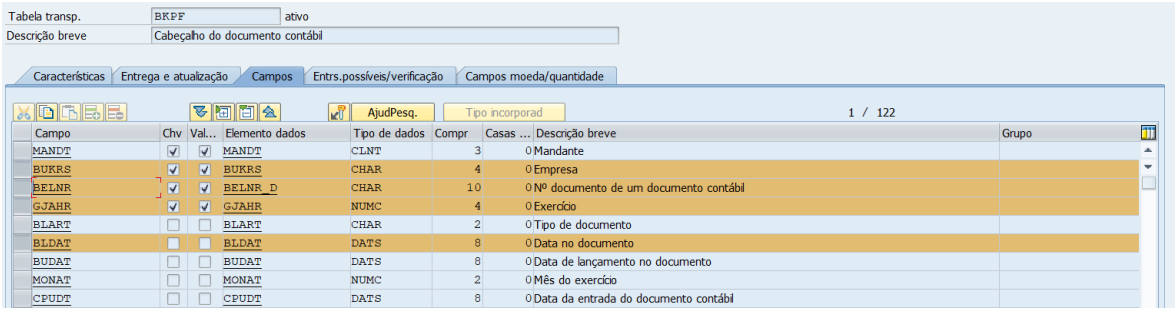


Figura 34 - Campos a usar no ecrã de seleção.

Como já eram conhecidos os campos, foi possível criar o campo de seleção (Figura 35). Estes campos de seleção estão programados de forma a que seja possível introduzir um intervalo de valores ou uma listagem.

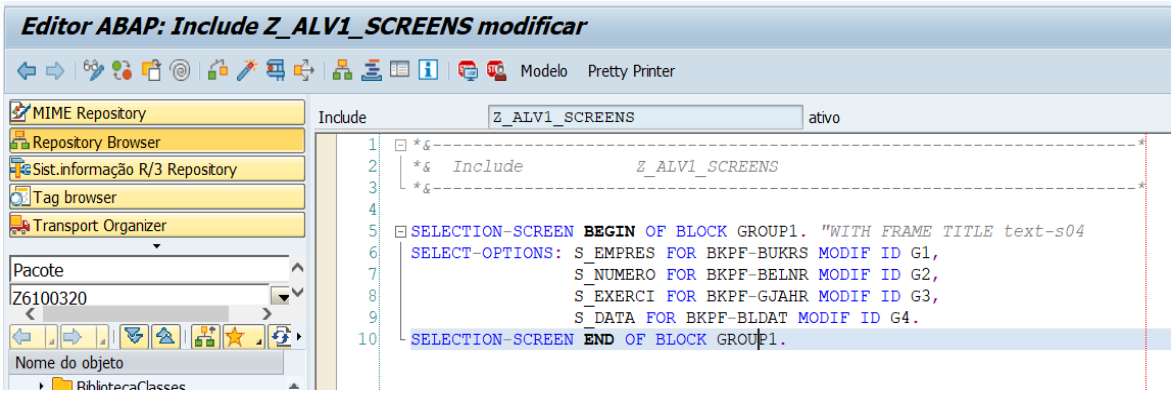


Figura 35 - Include preenchido com os campos de seleção a usar.

Foi feito um s*elect* à tabela BKPF da base de dados para se obter a listagem completa de documentos. Essa listagem foi guardada dentro de uma tabela interna (IT_BKPF) e este *select* foi restringido pelas condições definidas pelo utilizador final no campo de seleção (Figura 36).

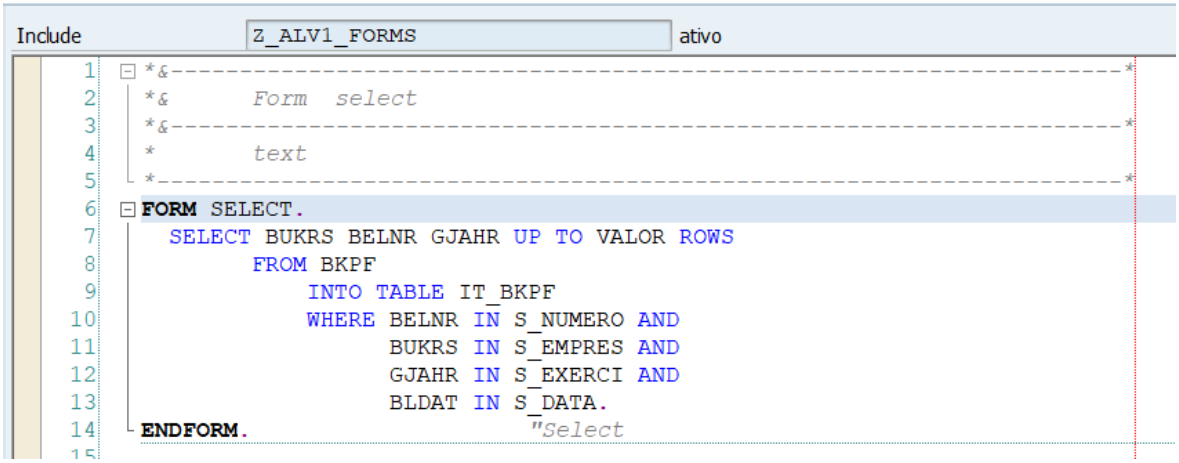


Figura 36 - *Select* à tabela BKPF.

Depois da obtenção dos dados, foi necessário criar a listagem. Como tal foi criado um *Fieldcatalog* dinâmico, que define a listagem (Figura 37). Esta parte de código fica definida na sub-rotina "*FIELDCATALOG*".

```
Include          Z_ALV1_FORMS          ativo
130  *       text
131  *---------------------------------------------------------------------
132  FORM BUILD_FIELDCATALOG.
133    CALL FUNCTION 'REUSE_ALV_FIELDCATALOG_MERGE'
134      EXPORTING
135        I_PROGRAM_NAME         = SY-REPID
136  *     I_INTERNAL_TABNAME     = 'IT_BKPF'
137        I_STRUCTURE_NAME       = 'ZBKPF6100320'
138  *     I_CLIENT_NEVER_DISPLAY = 'X'
139        I_INCLNAME             = SY-REPID
140  *     I_BYPASSING_BUFFER     =
141  *     I_BUFFER_ACTIVE        =
142      CHANGING
143        CT_FIELDCAT            = I_FCAT
144      EXCEPTIONS
145        INCONSISTENT_INTERFACE = 1
146        PROGRAM_ERROR          = 2
147        OTHERS                 = 3.
148    IF SY-SUBRC <> 0.
149  * Implement suitable error handling here
150    ENDIF.
151
152    LOOP AT I_FCAT INTO WA_FCAT. "MUDAR NOMES MENU
153      CASE WA_FCAT-FIELDNAME.
154        WHEN 'BELNR'.
155          WA_FCAT-HOTSPOT = 'X'.
156          MODIFY I_FCAT FROM WA_FCAT.
157        WHEN 'BLDAT'.
158          WA_FCAT-HOTSPOT = 'X'.
159          MODIFY I_FCAT FROM WA_FCAT.
160        WHEN 'ZCHECK'.
161          WA_FCAT-NO_OUT = 'X'.
162      ENDCASE.
163      MODIFY I_FCAT FROM WA_FCAT.
164    ENDLOOP.
165
166  ENDFORM.                    "BUILD_FIELDCATALOG
```

Figura 37 - Programação do *Fieldcatalog*.

É necessário a implementação de uma sub-rotina responsável para mostrar todos os valores que estão guardados na tabela interna na listagem. Esta sub-rotina faz a ligação do Fieldcatalog, dos dados do programa e das funcionalidades do utilizador (Figura 38).

```
Include          Z_ALV1_FORMS                    ativo
41  *&---------------------------------------------------------------------*
42  *       text
43  *----------------------------------------------------------------------*
44  FORM DISPLAY_ALV_REPORT .
45    GD_REPID = SY-REPID.
46    CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
47   EXPORTING
48  *   I_INTERFACE_CHECK                 = ' '
49  *   I_BYPASSING_BUFFER                = ' '
50  *   I_BUFFER_ACTIVE                   = ' '
51      I_CALLBACK_PROGRAM                = GD_REPID
52      I_CALLBACK_PF_STATUS_SET          = 'PF_STATUS'
53      I_CALLBACK_USER_COMMAND           = 'USER_COMMAND'
54  *   I_CALLBACK_TOP_OF_PAGE            = ' '
55  *   I_CALLBACK_HTML_TOP_OF_PAGE       = ' '
56  *   I_CALLBACK_HTML_END_OF_LIST       = ' '
57      I_STRUCTURE_NAME                  = 'I_FCAT'
58  *   I_BACKGROUND_ID                   = ' '
59  *   I_GRID_TITLE                      =
60      IS_LAYOUT                         = GD_LAYOUT
61  *   I_GRID_SETTINGS                   =
62      IT_FIELDCAT                       = I_FCAT   "--->1-fieldcatalog[]
63      IS_VARIANT                        = G_VARIANT
64    TABLES
65      T_OUTTAB                          = IT_BKPF
66   EXCEPTIONS
67     PROGRAM_ERROR                     = 1
68     OTHERS                            = 2
69          .
70    IF SY-SUBRC <> 0.
71      MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
72      WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
73    ENDIF.
74  ENDFORM.                    "DISPLAY_ALV_REPORT
```

Figura 38 - Programação do *Display* .

Por fim, foram programados os campos ativos da listagem (Requisitos 4 e 5), onde foram feitas as configurações do utilizador. Para se cumprir o requisito número 4, foi chamada a transação e enviados para a mesma os valores da Empresa, Número de Documento e Exercício. De acordo com o requisito número 5, foram criadas mensagens com o valor da data (Figura 39).

```
248     WHEN '&IC1'.
249       CASE RS_SELFIELD-FIELDNAME.
250         WHEN 'BELNR'.
251           READ TABLE IT_BKPF INTO GS_TOTXT INDEX RS_SELFIELD-TABINDEX.
252         SET PARAMETER ID 'BUK' FIELD GS_TOTXT-BUKRS.
253         SET PARAMETER ID 'BLN' FIELD GS_TOTXT-BELNR.
254         SET PARAMETER ID 'GJR' FIELD GS_TOTXT-GJAHR.
255
256         CALL TRANSACTION 'FB03' AND SKIP FIRST SCREEN.
257         WHEN 'BLDAT'.
258           CALL FUNCTION 'POPUP_TO_INFORM'
259           EXPORTING
260             TITEL = 'Data'
261             TXT1  = 'A DATA ESCOLHIDA FOI!'
262             TXT2  = RS_SELFIELD-VALUE.
263       ENDCASE.
264   ENDCASE.
265 ENDFORM.  "User_command
```

Figura 39 - Configurações dos campos ativos para os utilizadores.

Para cumprir o requisito número 3 foi posta a condição no início do programa. Caso o utilizador não selecione a *checkbox*, o programa não faz nenhuma instrução (campo cb_norm). No final o corpo do programa fica definido conforme a Figura 40. Caso o cb_norm seja selecionado, é executado um conjunto de sub-rotinas que são responsáveis por realizar a seleção dos dados e apresentar esses mesmos dados no ecrã.

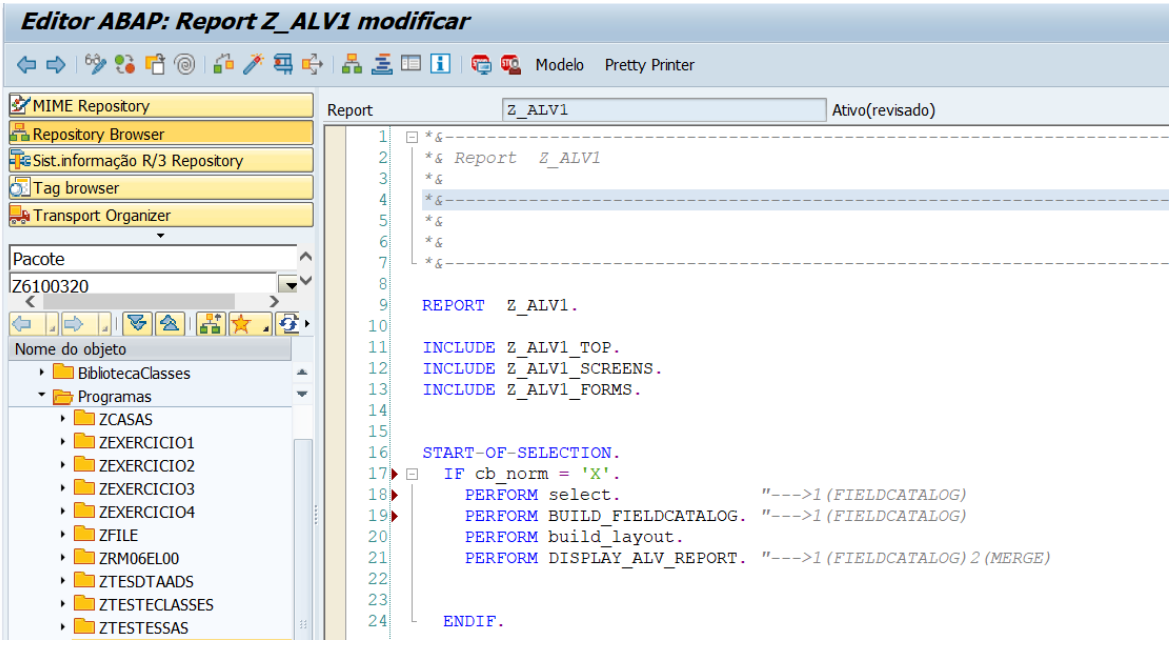


Figura 40 - Estrutura base do programa.

5.3.4 Transações Principais

As transações principais foram a SE38 para criar o programa, a SE11 para fazer consulta de campos à tabela BKPF e a transação SE37 para analisar os módulos de função utilizados para o *FIELDCATALOG* e para o *DISPLAY*.

5.3.5 Conclusão

Foram desenvolvidas todas as funcionalidades que estavam planeadas para o programa. Este programa permite desde a criação de campos de seleção que possibilitam a filtragem da informação a apresentar no ecrã, à visualização de outras informações de um item da lista que seja selecionado (Figura 41).

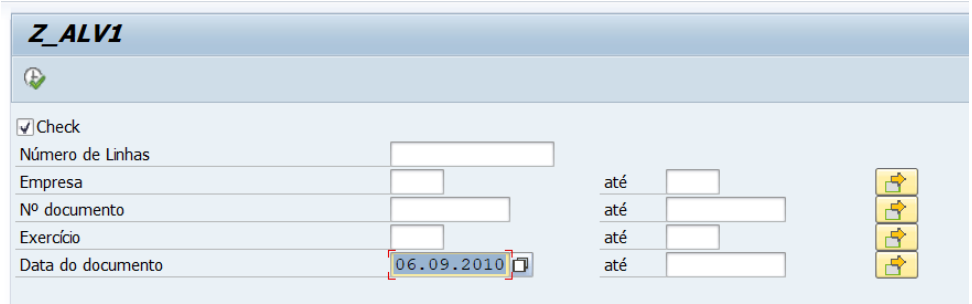
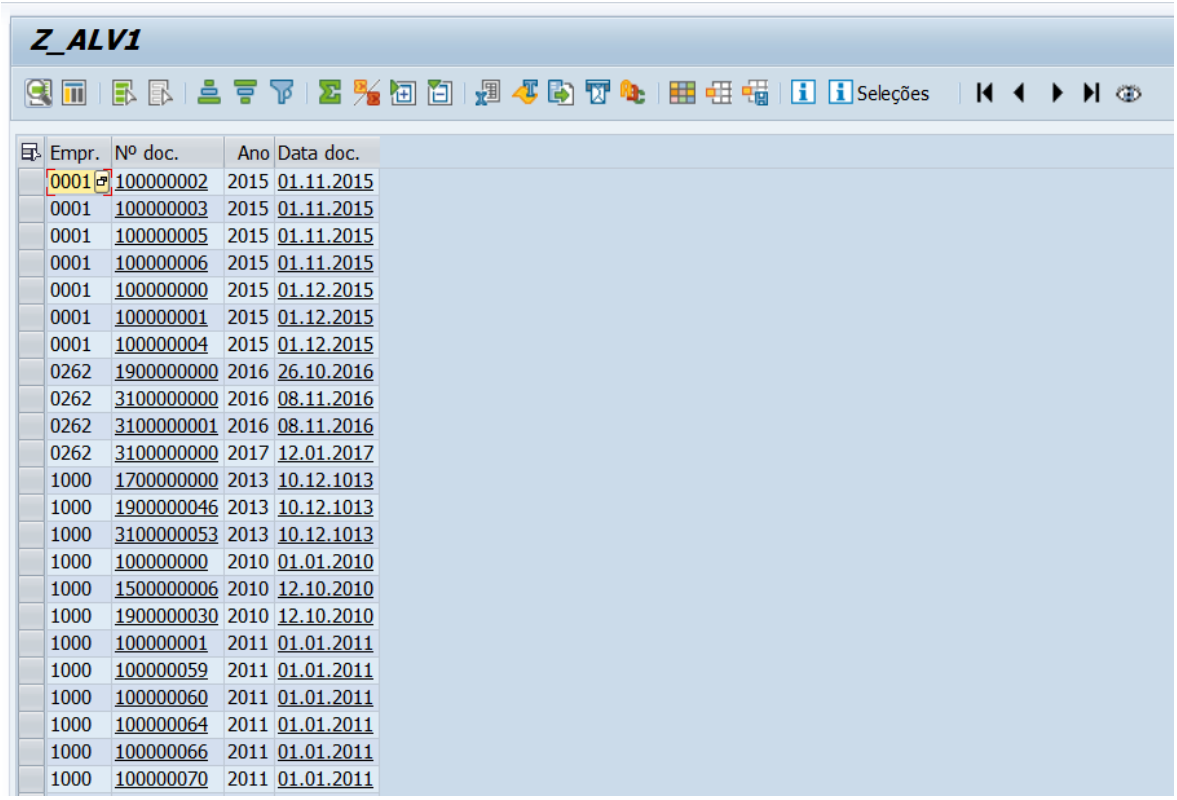


Figura 41 - Primeiro ecrã do programa.

Depois do utilizador ter inserido os filtros desejados, foi executada uma pesquisa, que gerou uma listagem com os documentos (Figura 42).



Empr.	Nº doc.	Ano	Data doc.
0001	100000002	2015	01.11.2015
0001	100000003	2015	01.11.2015
0001	100000005	2015	01.11.2015
0001	100000006	2015	01.11.2015
0001	100000000	2015	01.12.2015
0001	100000001	2015	01.12.2015
0001	100000004	2015	01.12.2015
0262	1900000000	2016	26.10.2016
0262	3100000000	2016	08.11.2016
0262	3100000001	2016	08.11.2016
0262	3100000000	2017	12.01.2017
1000	1700000000	2013	10.12.1013
1000	1900000046	2013	10.12.1013
1000	3100000053	2013	10.12.1013
1000	100000000	2010	01.01.2010
1000	1500000006	2010	12.10.2010
1000	1900000030	2010	12.10.2010
1000	100000001	2011	01.01.2011
1000	100000059	2011	01.01.2011
1000	100000060	2011	01.01.2011
1000	100000064	2011	01.01.2011
1000	100000066	2011	01.01.2011
1000	100000070	2011	01.01.2011

Figura 42 - Listagem dos documentos.

Ao selecionar o número de um documento, o utilizador foi encaminhado para a transação FB03 onde lhe foi permitido visualizar mais informações acerca do documento em causa (Figura 43).

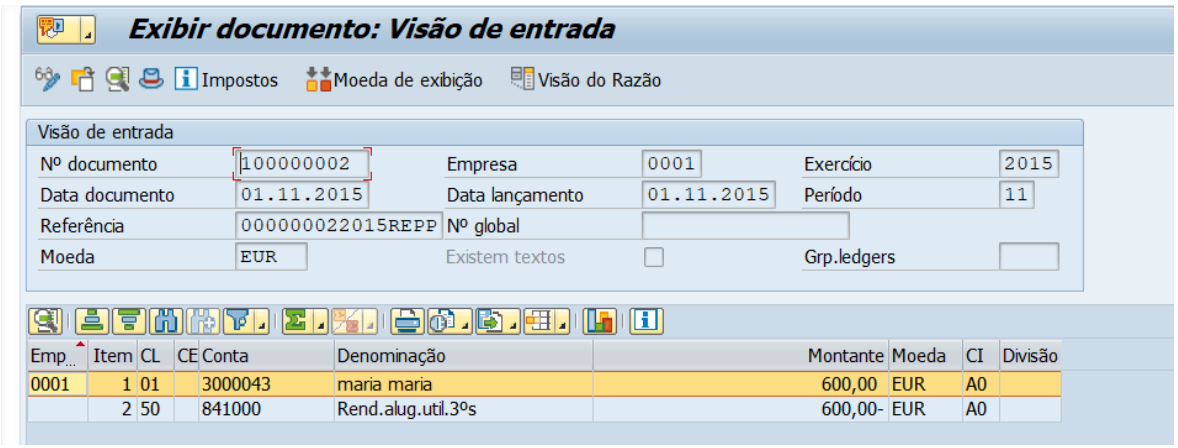


Figura 43 - Transação FB03 com os dados do documento.

Ao selecionar a data abriu-se um *popup* informativo com o texto "A data escolhida foi:" e surgiu a data desse documento (Figura 44).

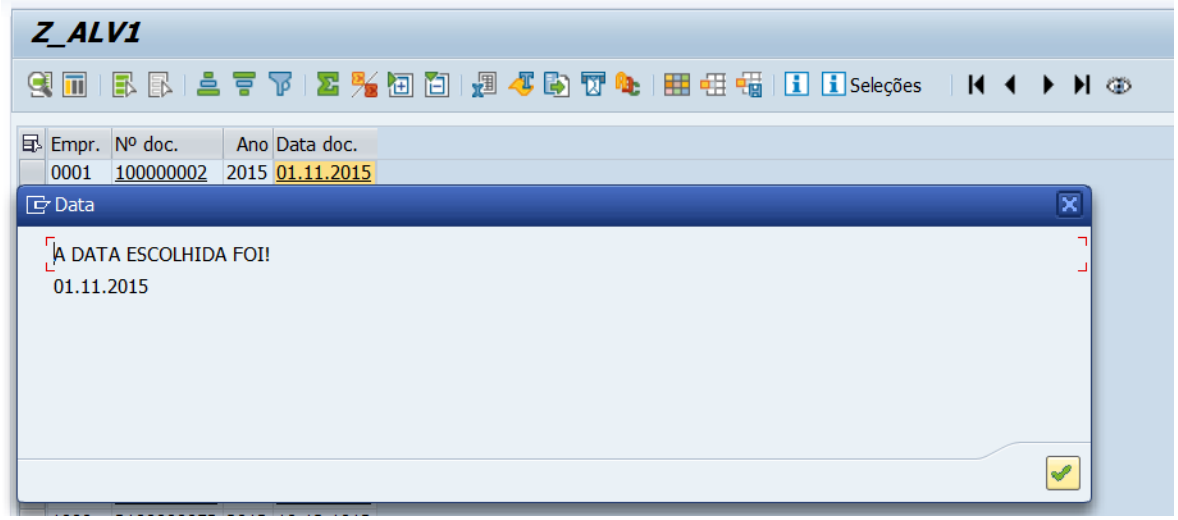


Figura 44 - *Popup* que aparece após clicar na data da lista.

5.4 Caso 3 | DMG Report

5.4.1 Problema

A Softinsa participou num concurso organizado pela Empresa C, que consistia na apresentação de uma ideia de uma aplicação a ser utilizada num dispositivo móvel dessa mesma empresa (Batista, Marques & Carrilho, 2018). O objetivo do concurso era obter ideias que potencializassem as funcionalidades do dispositivo móvel. A metodologia utilizada para o desenvolvimento desta tarefa foi o SCRUM.

A Softinsa venceu o concurso, sendo desafiada a desenvolver uma prova de conceito dessa mesma ideia. A prova de conceito consistia numa aplicação de reporte de danos em materiais num armazém. O nome da aplicação é *DMG Report (Damage Report)*.

5.4.2 Análise de Requisitos

A aplicação apresentava as seguintes especificações:

1 - A aplicação deverá apresentar no início uma listagem com todos os registos de danos em materiais;

2 - Deverá ser possível pesquisar um determinado registo através do seu identificador. Essa pesquisa poderá ser realizada através da introdução de texto manual ou através da leitura de código de barras;

3 - Ao selecionar um item numa lista, deverão aparecer os detalhes desse registo;

4 - Deverá ser possível visualizar a informação detalhada do registo através de um ficheiro PDF, podendo realizar-se o *download* para o dispositivo;

5 - Deverá ser possível criar novos registos, com as seguintes informações: Número de Material, Tipo de Material, Grupo de Materiais, Descrição dano no Material, Fotografia do Dano;

6 - O utilizador poderá criar um ficheiro PDF com a informação de vários registos, podendo definir quais são os que devem constar nesse documento.

5.4.3 Resolução

Começou-se pela criação de um novo programa. Esta ação foi feita na transação */n/neptune/designer* (Figura 45). O nome da aplicação teve de começar pela letra Z. As aplicações *standard* criadas pela SAP podem começar por qualquer letra. Para sinalizar que a aplicação não é *standard*, é obrigatório que o nome da aplicação comece por Z).

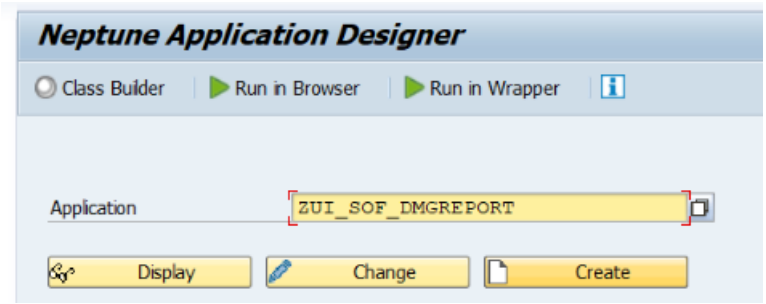


Figura 45 - Criação de uma nova aplicação.

Depois do programador criar o programa foi necessário inserir uma descrição do mesmo, bem como, vincular-lhe uma classe (Figura 46). Caso a classe ainda não exista, o programador poderá criar uma nova classe através da transação SE24 (Figura 47).

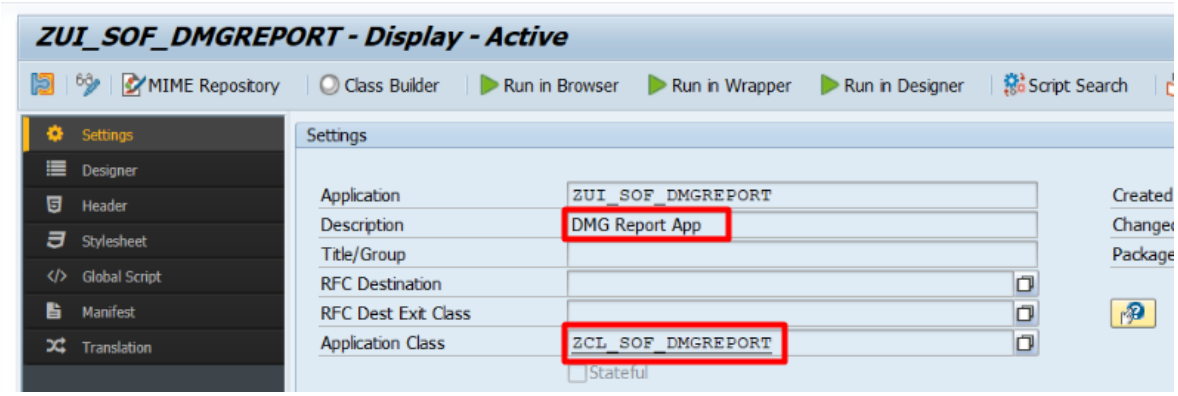


Figura 46 - Edição das configurações da aplicação.

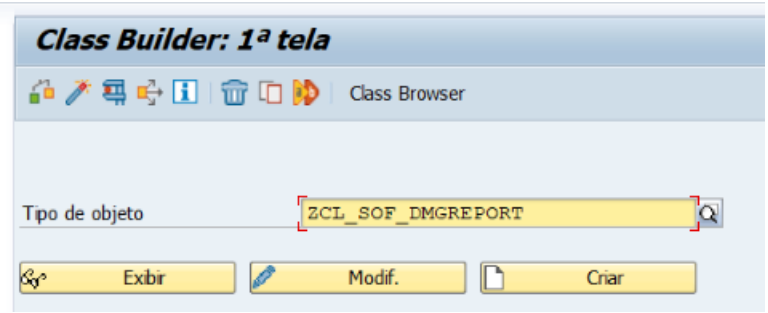


Figura 47 - Criação de uma nova classe.

Para que a classe funcionasse conforme as exigências do Neptune, foi necessário definir uma *interface* (Figura 48). Depois de definida, foi gerado automaticamente um conjunto de métodos que podem ser programados consoante as necessidades do utilizador (Figura 49).

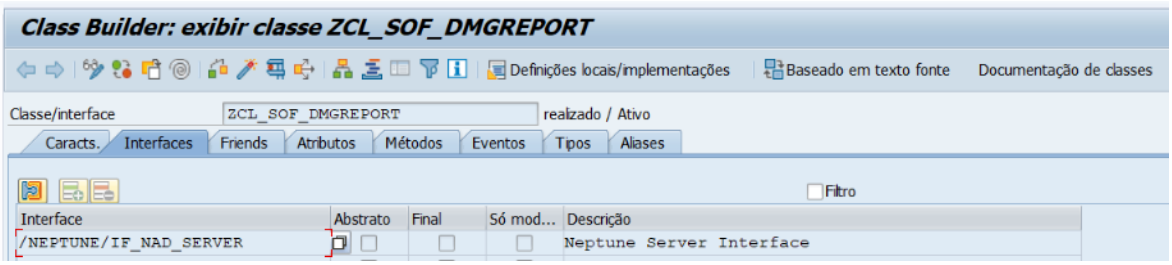


Figura 48 - Definição da *interface* da classe.

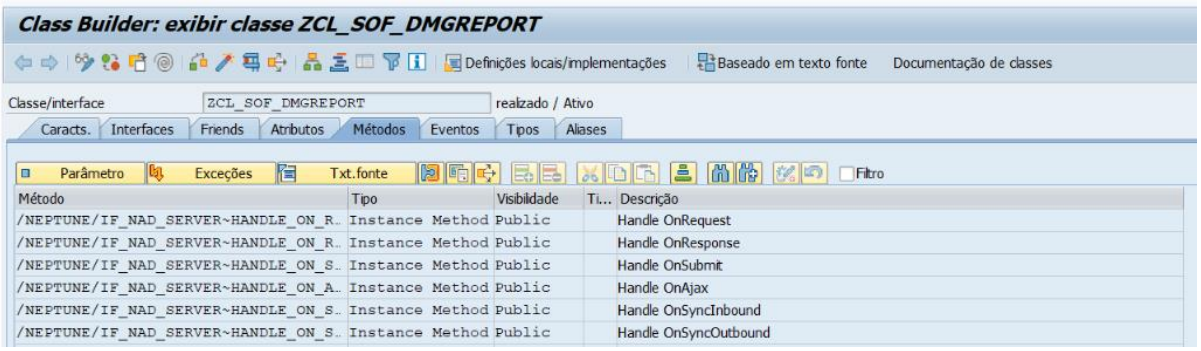


Figura 49 - Métodos gerados após a introdução da *interface*.

É programado um objeto (*Init* em *Javascript*) para ser executado no início da aplicação. Quando a aplicação é executada, são carregados os dados da mesma que são apresentados no ecrã. Para isso, é necessário realizar uma chamada à base de dados através de código *javascript*. Esse código vai ser programado no *Init* (Figura 50).

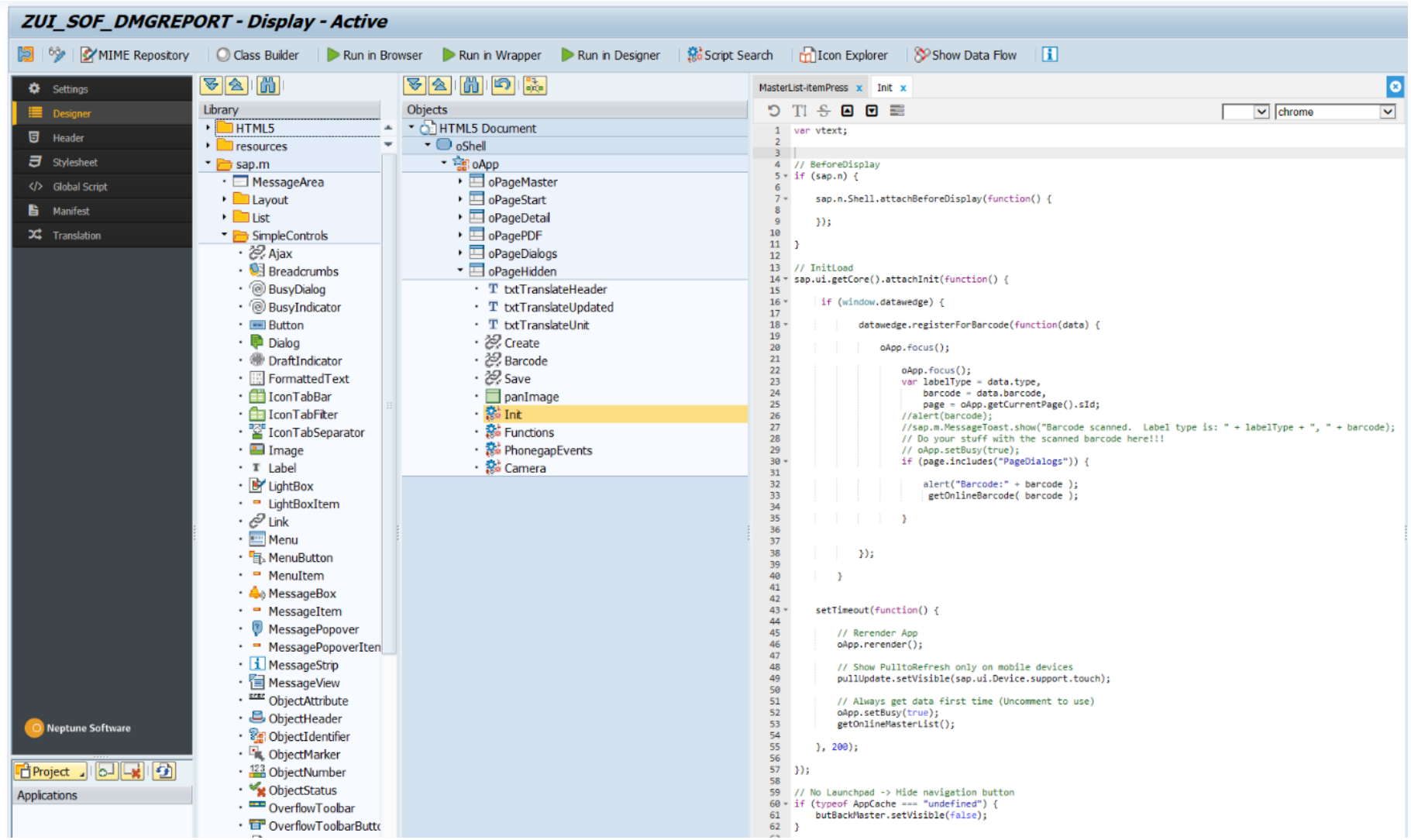


Figura 50 - Código *Javascript* que é executado inicialmente.

A comunicação entre o Neptune e o ABAP é feita através de códigos AJAX. De acordo com a Figura 51, quando é despoletado um código AJAX, é feito no ABAP a comparação entre o nome do AJAX que está a ser chamado e os AJAX existentes no sistema. Se for encontrado, é despoletado todo o código dentro desse AJAX (Figura 51).

Método /NEPTUNE/IF_NAD_SERVER~HANDLE_ON_AJAX ativo

```
1   METHOD /NEPTUNE/IF_NAD_SERVER~HANDLE_ON_AJAX.
2
3     CASE AJAX_ID.
4
5       WHEN 'GET_DATA'.
6
7         CALL METHOD GET_DATA.
8
9       WHEN 'SMART_FORM'.
10
11        CALL METHOD ME->SMART_FORM
12          IMPORTING
13            E_PDF = GS_PDF.
14
15
16      WHEN 'SMART_FORM2'.
17
18 **        CALL METHOD GET_IMAGE.
19
20        CALL METHOD TRATAR_IMAGEM.
21
22        CALL METHOD ME->SMART_FORM2
23          IMPORTING
24            E_PDF = GS_PDF.
25
26
27
28      WHEN 'CREATE'.
29        CALL METHOD SAVE( SERVER ).
30        CALL METHOD CREATE( AJAX_VALUE ).
31
32      WHEN 'BARCODE'.
33
34        CALL METHOD BARCODE( AJAX_VALUE ).
35
36    ENDCASE.
37
38  ENDMETHOD.
```

Figura 51 - Método utilizado para verificar o AJAX a ser utilizado.

O código ABAP baseia-se na seleção dos dados a apresentar na aplicação (Figura 52). Essa informação é guardada em tabelas internas.

```
Método    GET_DATA                                                    ativo
1   METHOD GET_DATA.
2
3     SELECT *
4       FROM ZDMG_TABLE
5       INTO CORRESPONDING FIELDS OF TABLE IT_DAMAGE.
6
7     SELECT MATNR BARCODE DESCRIPTION DATECREATE AS DATACREARTED
8       FROM ZDMG_TABLE
9       INTO CORRESPONDING FIELDS OF TABLE IT_SELECT.
10
11
12  ENDMETHOD.
```

Figura 52 - Método que é executado no início do programa.

O programador ao definir um código AJAX, também define que informações é que devem ser enviadas para o ABAP e o que deve ser recebido depois da execução do AJAX (Figura 53). Na Figura 53, o utilizador definiu que informações é que teve que enviar para o ABAP ao selecionar as *checkbox* na coluna Send e que informações é que queria retornar ao selecionar as checkbox na coluna *Receive*.

A programação em Neptune é feita com base em três mecanismos: no *drag and drop* de objetos para a aplicação, desde botões, tabelas, listas, *labels*, entre outros; programação em *javascript*, onde é feita definida a função dos objetos (adicionados anteriormente); e programação ABAP que contém os métodos onde é feita a seleção dos dados a apresentar. Na Figura 54, é possível visualizar o código *javascript* de uma lista quando é clicado um item dessa mesma lista.

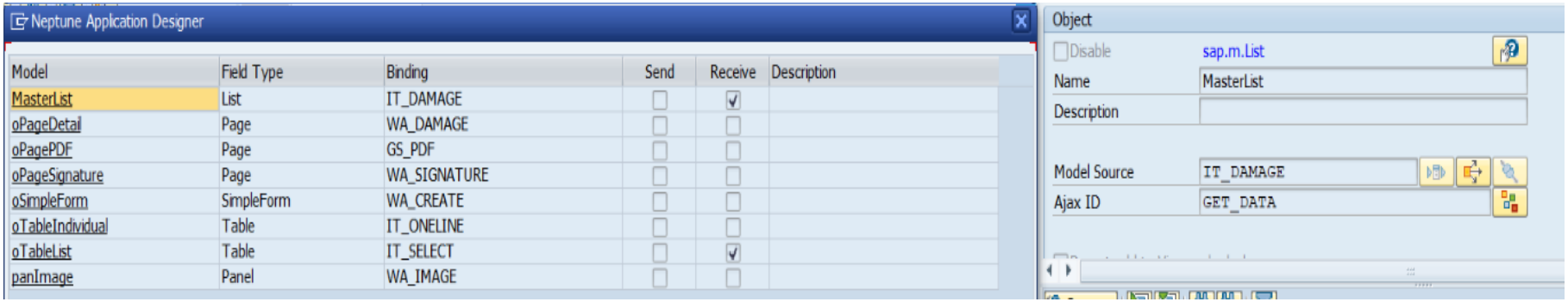


Figura 53 - O que é recebido e enviado pelo AJAX.

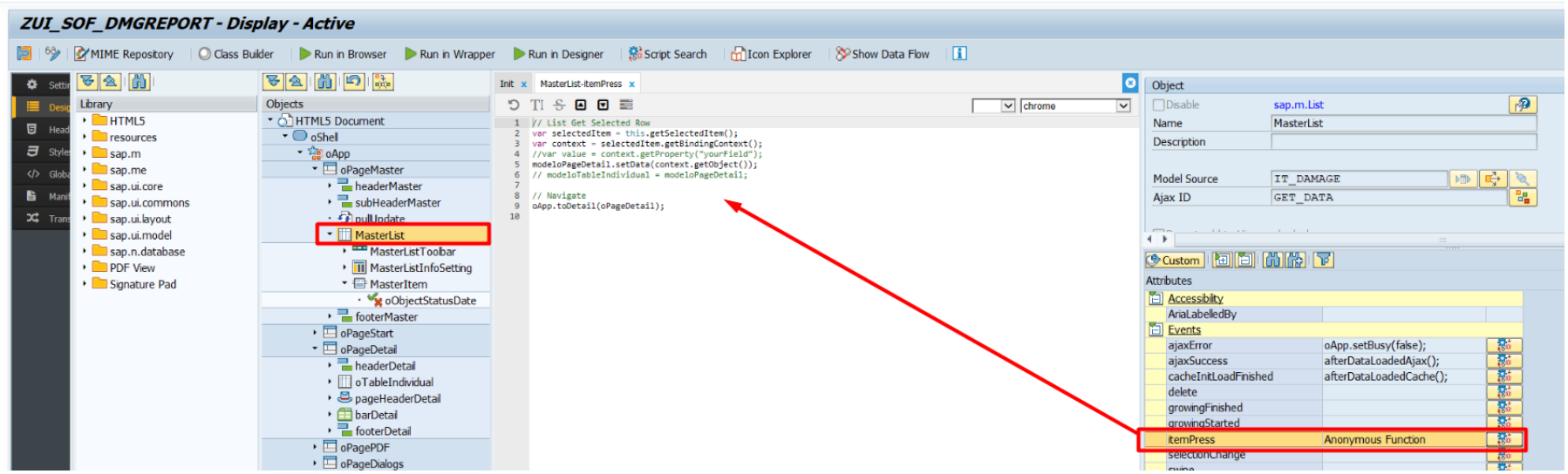


Figura 54 - Programação de um objeto.

5.4.4 Conclusão

Quando o desenvolvimento da aplicação foi concluído foram efetuados inúmeros testes e corrigidos alguns erros. Após a fase de testes, foi enviado a prova de conceito para a Empresa C para os mesmos poderem testar a aplicação com os seus dispositivos.

Na página inicial aparece a listagem de todos os registos efetuados pelo utilizador. Como se pode observar na Figura 55, o utilizador pode fazer a pesquisa pelo registo, podendo fazer pesquisa por leitura de código de barras ou por material. Ao selecionar uma linha da lista, o mesmo acede os detalhes do registo (Figura 56), podendo também atualizar a lista. O utilizador poderá ordenar a lista por data ou número de registo, criar um PDF com um conjunto de registos e criar um registo de dano de material.

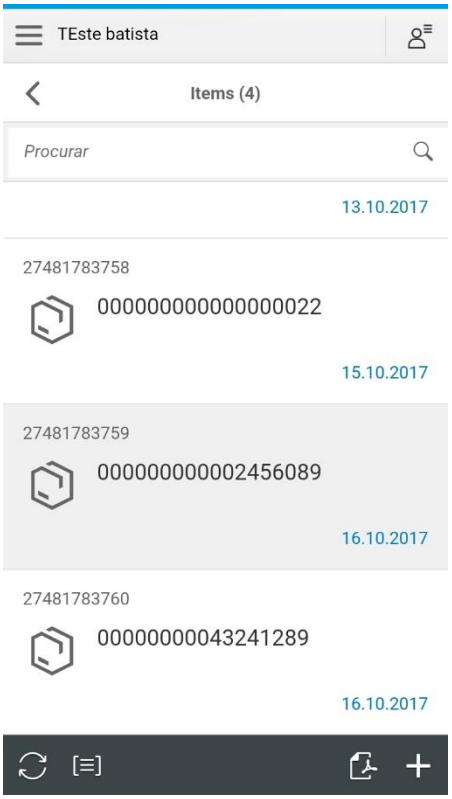


Figura 55 - Ecrã inicial da aplicação.

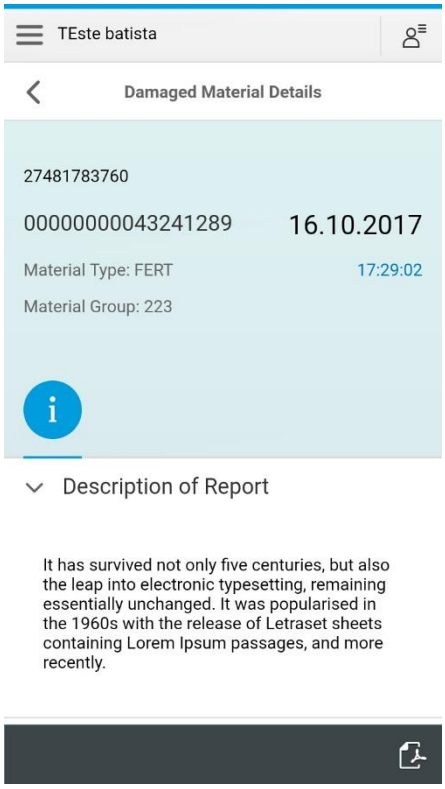


Figura 56 - Detalhes do registo.

Dentro dos detalhes de um registo, o utilizador pode visualizar as informações do registo em formato PDF, podendo fazer o download do mesmo (Figura 57).

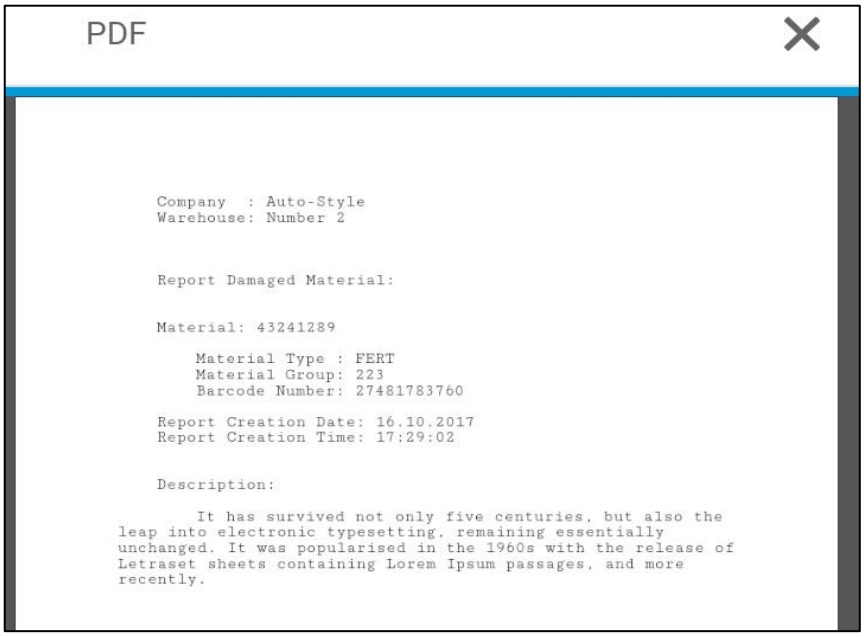


Figura 57 - Visualização da informação num ficheiro PDF.

Caso o utilizador queira criar um documento PDF com as informações de vários registos, primeiro tem de selecionar os registos que pretende (Figura 58) e só depois de confirmar a sua seleção, é que é apresentado o PDF com os detalhes de vários registos (Figura 59).

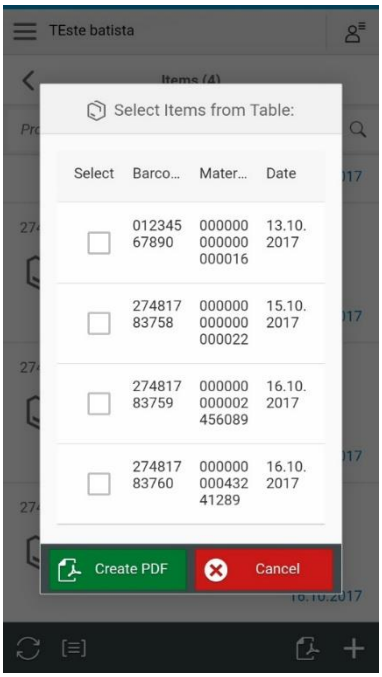


Figura 58 - Seleção dos registos a apresentar.

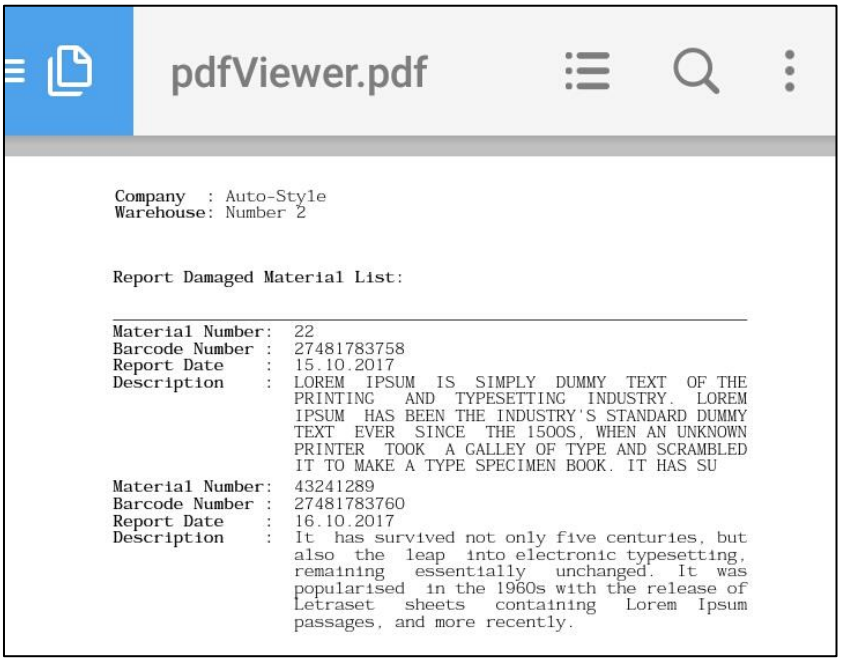


Figura 59 - Documento PDF criado com informação dos registos selecionados.

O utilizador poderá criar novos registos de quebras de material, podendo inserir as informações do material em causa como o número de material, tipo de material, grupo do material (Figura 60), como também pode inserir uma descrição do dano e uma foto do mesmo (Figura 61).

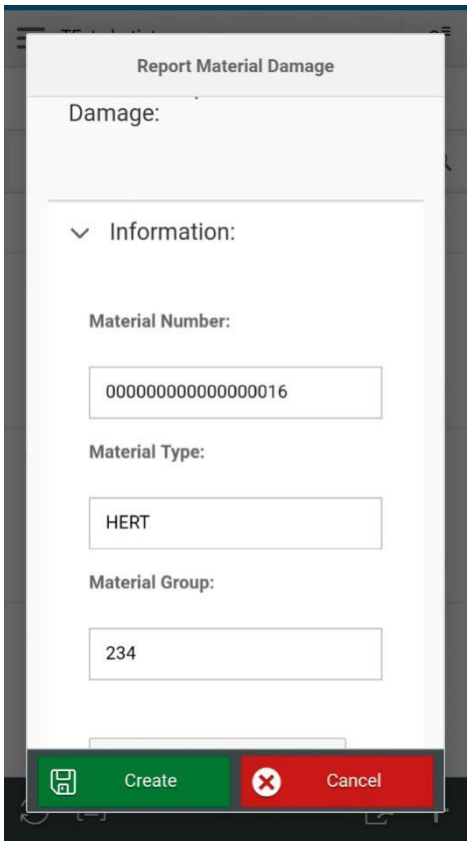


Figura 60 - Campos onde são inseridos os dados do material.

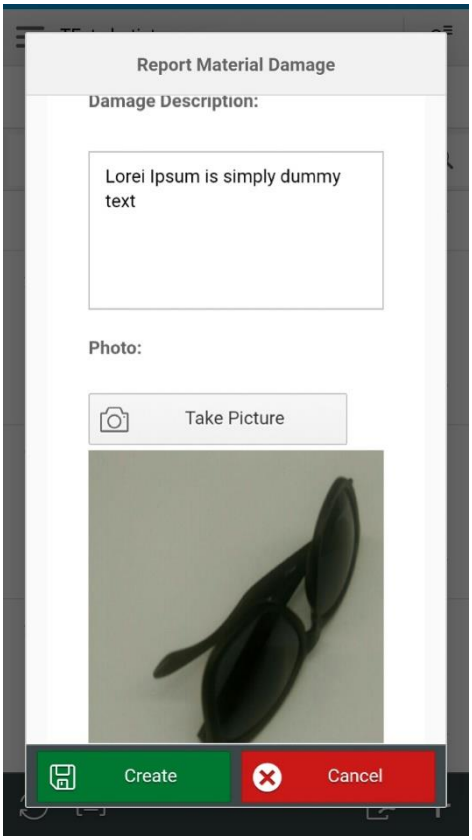


Figura 61 - Campos onde são inseridos os detalhes e foto do material danificado.

5.5 Caso 4 | Receção de Mercadorias

5.5.1 Problema

Foi solicitada a criação de uma aplicação para gerir as mercadorias na Empresa D. Para auxiliar a receção de mercadorias, existem documentos que contêm uma listagem de todos os *itens* pedidos pela empresa. O objetivo foi criar uma aplicação para dispositivos móveis que facilitasse a inserção e contagem das quantidades dos materiais recebidos, tornando todo o processo mais rápido e eficiente. A metodologia utilizada para o desenvolvimento desta tarefa foi o SCRUM.

5.5.2 Análise de Requisitos

A aplicação apresentava as seguintes especificações:

1 – No ecrã inicial, o utilizador poderá fazer a pesquisa de documentos de pedidos de compras através da leitura do número de material ou código EAN (*European Article Number*). A pesquisa poderá ser feita através de leitura de código de barras ou através do preenchimento manual do campo de pesquisa.

2 – O utilizador poderá restringir a visualização dos documentos consoante a data de cada um, ou seja, ele poderá definir que só quer que sejam apresentados documentos a partir de uma certa data. Por defeito, mostra os documentos dos últimos 90 dias.

3 – O utilizador poderá atualizar a listagem dos documentos apresentados.

4- É apresentado uma lista com as seguintes informações: Número de Pedido, Número de Fornecedor, Tipo de Documento, Data Criação do Documento, Grupo de Mercadorias, Grupo de Compradores, Nome do Fornecedor, Valor Total e a Moeda. Ao selecionar um documento dessa lista, o utilizador é encaminhado para a página dos detalhes desse mesmo documento.

5 – Na página dos detalhes deverão ser apresentadas mais informações do documento (campos extra) como uma listagem de todos os itens que compõe esse documento.

6 – Esta listagem deverá conter o nome do item, bem como o número de material e quantidade que já foi recebida. Cada item da lista deverá conter um botão (X) que permite eliminar a quantidade inserida daquele item. Se o utilizador clicar num item, o mesmo deverá ser encaminhado para uma página onde poderá inserir quantidades recebidas do item selecionado.

7 – Na página dos detalhes do item deverá ser apresentado o número de material e a ordem pedida do item em causa, como também deverá ser possível inserir quantidade recebida, a unidade, o lote e a data de validade.

8 – Depois de inserir os valores necessários na página de detalhes do item, e confirmar essa receção, o utilizador é encaminhado para a página dos detalhes do documento de pedido de compras.

9 – Depois de efetuar as alterações necessárias, o utilizador poderá gravar as quantidades recebidas, aparecendo uma listagem com todos os itens desse documento. Nessa listagem faz a comparação da quantidade pedida e da quantidade recebida. O utilizador deverá ter a opção de definir que a receção do material está completa, mesmo que a quantidade pedida não seja igual à quantidade recebida.

10 – Ao confirmar as alterações, as mesmas têm de ser guardadas no sistema, surgindo uma informação que as mesmas foram guardadas com sucesso ou da existência de erros.

5.5.3 Conclusão

De acordo com as especificações fornecidas, quando a aplicação é executada surge um ecrã com a listagem dos pedidos de compra (Figura 62). Neste ecrã, o utilizador pode consultar os pedidos ou pesquisar um pedido específico. Ao selecionar um item da listagem, o utilizador é encaminhado para os detalhes do mesmo.

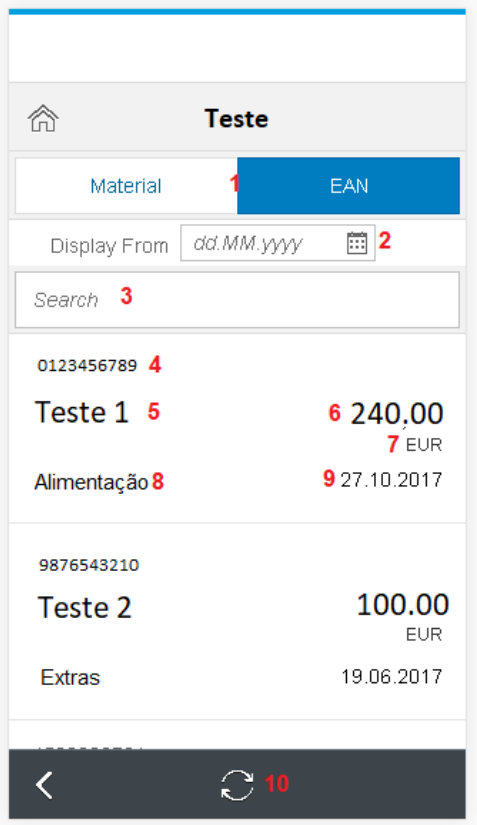


Figura 62 - Ecrã inicial da aplicação.

Legenda da Figura 62:

1 - Filtrar a pesquisa por número de material ou EAN;

2 - Restringir os documentos a apresentar através da data (mostrar a partir de);

3 - Campo para inserir o valor a pesquisar;

4 - Número do documento;

5 - Nome do Fornecedor;

6 - Valor do Pedido de Compra;

7 - Unidade;

8 - Grupo de Compradores;

9 - Data de criação do Documento;

10 - Atualização da Listagem.

As implementações anteriores dizem respeito às especificações 1, 2, 3 e 4.

Nos detalhes do pedido de compra é possível consultar mais informações acerca deste, bem como, os itens que constituem o pedido em causa. Ao expandir a *Tab Information* (Figura 63), é possível consultar mais informação acerca do pedido (Figura 64).

Em relação à listagem dos itens (Figura 63), é apresentado o nome do item, número de material, e a quantidade recebida. O utilizador pode eliminar a quantidade recebida do material, ao carregar no Botão X.

Ao selecionar um item da listagem, o utilizador é encaminhado para uma página onde poderá inserir o valor da quantidade recebida daquele material.

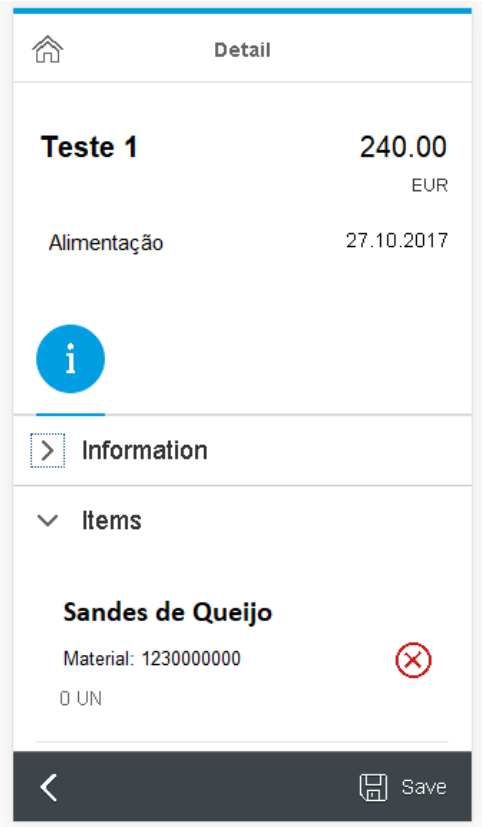


Figura 63 - Página de detalhes do pedido.

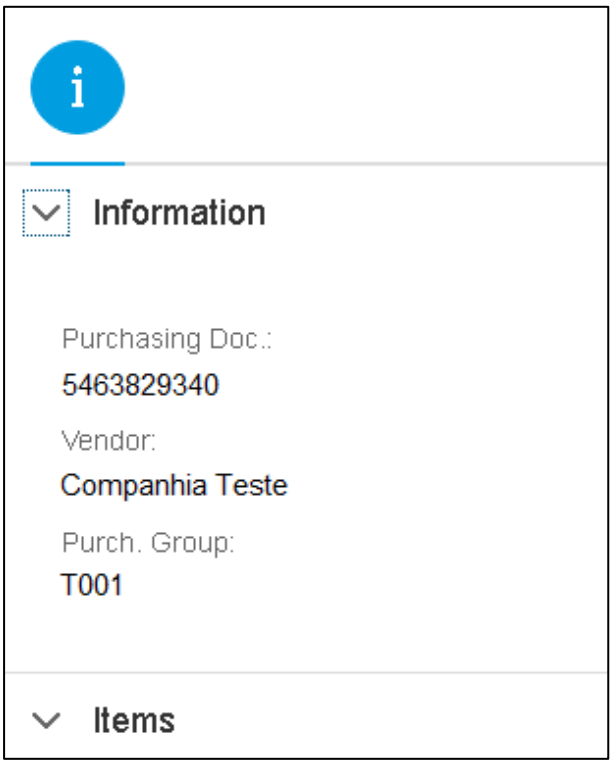


Figura 64 - Informações extra acerca do pedido.

As implementações anteriores dizem respeito às especificações 5 e 6.

Na página dos detalhes do item é possível consultar o número do material bem como a quantidade pedida, existindo a possibilidade de registar a entrada de mercadorias. O utilizador poderá definir a quantidade de entrada, bem como as unidades de medida, o lote e a data de validade (Figura 65).

Depois de registada uma entrada de mercadorias, esse registo aparece numa lista no final do ecrã (Figura 66). É possível introduzir mais do que um registo para o mesmo material, existindo a possibilidade de pertencerem a lotes diferentes. Depois de efetuar todas as alterações necessárias, o utilizador tem de voltar para a página dos detalhes do pedido de compra (pressionando o Botão "<").

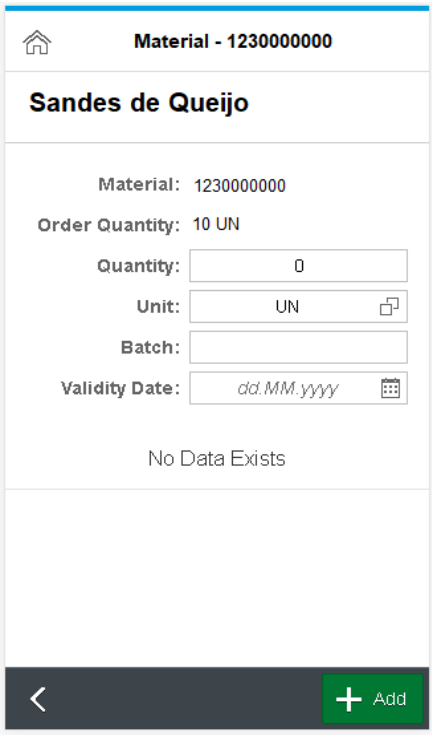


Figura 65 - Página de detalhes do item.

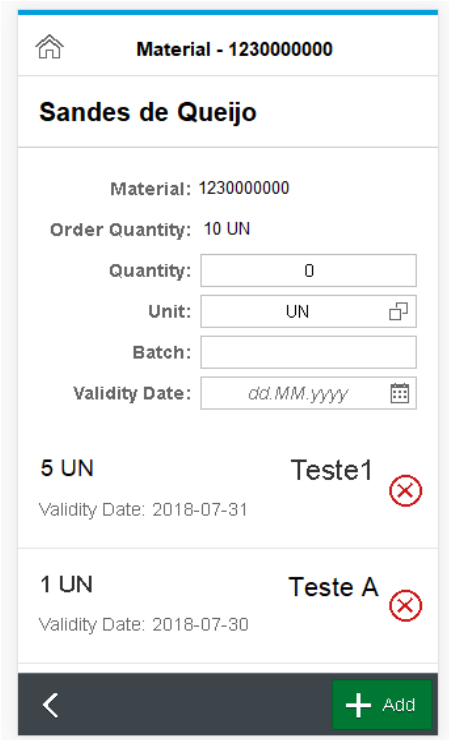


Figura 66 - Ecrã após o registo de entrada de mercadorias.

As implementações anteriores dizem respeito às especificações 7 e 8.

O registo da entrada de mercadorias de um certo material é visível na listagem dos itens (Figura 67).

Quando o utilizador já efetuou todos os registos, o mesmo deverá guardar essas alterações no sistema, ao clicar no Botão *Save* (Figura 68). Depois de o fazer, vai surgir no ecrã um *popup* com a listagem de todos os itens do documento, bem como a quantidade pedida e recebida. O utilizador tem a possibilidade de definir se a receção do material está completa, mesmo que a quantidade pedida e recebida não sejam iguais (ao preencher a *checkbox Complete*).

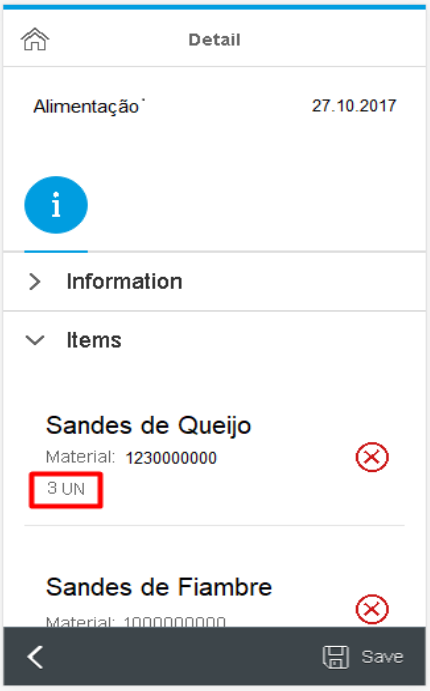


Figura 67 - Alteração das quantidades de um item.

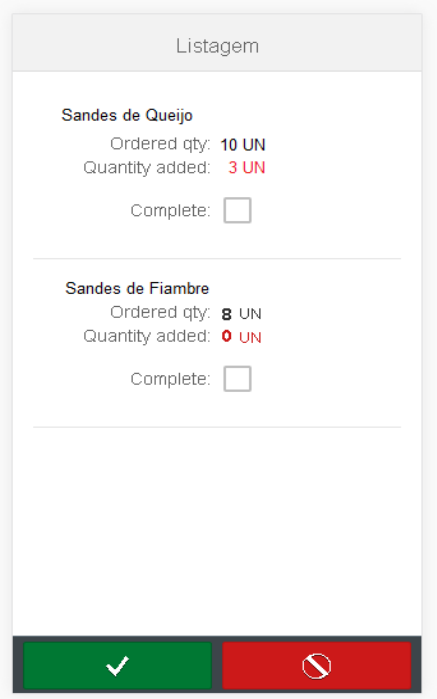


Figura 68 - Listagem para comparar as quantidades.

Ao confirmar todas as alterações, é feita a gravação de todos os registos no sistema e o utilizador é informado da finalização do processo.

Capítulo 6 | Conclusões

Os avanços tecnológicos permitiram que atualmente seja possível a consulta e análise dos dados das organizações em tempo real, tornando a sua gestão mais eficiente. Uma das empresas que ajudou a impulsionar o mundo da informação foi a SAP, através da criação de sistemas de informação de apoio á gestão.

Através da realização deste estágio foram adquiridas competências a nível da programação, de forma a que ser possível a criação e manutenção tanto de aplicações SAP como de aplicações Neptune sobre SAP.

Atualmente, a SAP é uma das maiores empresas no ramo dos sistemas ERP. Estes sistemas cuidam de todas as operações diárias de uma empresa, em todas as áreas, sendo sistemas que garantem performance, mas também segurança e confidencialidade dos dados. Num mundo onde predominam os dispositivos móveis, a necessidade de efetuar consultas ou executar certas tarefas a qualquer hora e em qualquer lugar levou a Neptune a criar uma interface de desenvolvimento de aplicações para dispositivos móveis com as tecnologias dos sistemas SAP.

A SAP disponibiliza aos seus clientes sistemas que facilitam a gestão do negócio, contudo, existem situações em que as funcionalidades standard da SAP não colmatam as necessidades do cliente. Como tal, é necessário o desenvolvimento de soluções mais especificas para as empresas, ou a realização de melhorias e adaptações aos programas já existentes.

Para auxiliar o planeamento e organização dos projetos, foram utilizadas metodologias de desenvolvimento. Cada metodologia tem os seus pontos fortes e fracos, e a sua escolha depende da tarefa, das suas especificações, do tempo de realização, entre outros fatores. Nos projetos descritos foram utilizadas as metodologias SCRUM e Waterfall.

Neste estágio foram implementados mais de vinte projetos, onde se destacam a implementação de notas, a criação e edição de *reports* e o desenvolvimento de aplicações para dispositivos móveis.

A Implementação de Notas é uma tarefa que requer uma análise não só da versão do sistema como também das notas, individualmente. O objetivo das notas é a correção de

possíveis erros que o sistema possa conter, para melhorar os processos. Como tal, a implementação de notas é uma tarefa que tem de ser realizada com rigor e com cuidado, visto que qualquer falha no desenvolvimento dos passos manuais ou da implementação da nota pode trazer problemas mais tarde.

A base dos programas SAP baseia-se na linguagem de programação ABAP e da sua vasta base de dados. É através da linguagem ABAP que o programador tem a possibilidade de efetuar a seleção e tratamento dos dados, entre outras funcionalidades.

A criação e edição de *reports* (programas) é uma tarefa que o consultor ABAP executa regularmente. Com as exigências do mercado e com a necessidade que as empresas possuem em melhorar e otimizar os seus processos, é preciso a criação de novos programas ou o melhoramento dos existentes para tornar a gestão mais eficaz e eficiente. A grande maioria das tarefas são realizadas com base em requisitos impostos pelo cliente, contudo, o programador tem de possuir um sentido crítico, sendo muito importante colocar questões acerca deste desenvolvimento e sugerir alternativas que permitam a resolução de problemas ou que tragam mais benefícios para a organização (cliente).

A criação de soluções SAP usando a plataforma Neptune Software UX foi o foco central do estágio. Soluções SAP num "mundo *mobile"* ainda são novidade, mas as mesmas já provaram que são recursos muitos vantajosos para as empresas. O uso desta plataforma de desenvolvimento permite a criação de aplicações com menores prazos e custos em comparação com o desenvolvimento de programas comuns, devido à sua arquitetura de desenvolvimento (*drag and drop*). As funcionalidades que são pedidas são em função dos dispositivos que o cliente vai adquirir e utilizar (pode ser necessário moldar as aplicações aos dispositivos). Como as aplicações vão ser utilizadas em dispositivos móveis, as mesmas têm de ser fáceis de utilizar e com menor número de iterações.

Uma vez que a plataforma Neptune Software UX é recente, ainda existe grande margem para progressão. É necessário acompanhar esse mesmo crescimento, realizando um estudo constante da ferramenta e das suas funcionalidades. De futuro também é importante uma melhoria na utilização das linguagens de programação ABAP e Javascript que são a base das aplicações em Neptune.

A otimização de código é outro fator de extrema importância, visto que o desempenho das aplicações é uma das características chave das mesmas. Este trabalho dá a conhecer uma área das tecnologias que é desconhecida para alguns estudantes. Permite não só mostrar os conhecimentos que são utilizados na área de SAP desde Base de Dados, *Javascript* entre outros, como também dá a conhecer algumas das metodologias que podem ser utilizadas no mercado de trabalho.

Referências Bibliográficas

Abreu, A. (s.d.). Metodologia de projetos: ágil ou tradicional? Entenda a diferença. Consultado a 24 de agosto de 2018, de *Salpinx*: https://www.salpinx.com.br/metodologia-de-projetos//

Adenowo, A., & Adenowo, B. (2013). Software engineering methodologies: a review of the *waterfall* model and object-oriented approach. *International Journal of Computer Science Issues*, 4 (7), 427-434.

Adviser. (25 de janeiro de 2018). Conheça a função de um consultor SAP. Consultado a 1 de setembro de 2018, de *Adviser*: https://adviserit.com.br/conheca-funcao-de-um-consultor-sap/

Albarqi, A., & Rizwan, Q. (2018). The proposed l-Scrum ban methodology to improve the efficiency of agile software development. *International Journal of Information Engineering and Electronic Business*, 10 (3), 23-25.

Almutairi, A., & Qureshi, R. (2015). The proposal of scaling the roles in Scrum of Scrums for distributed large projects. *Modern Education and Computer Science*, 8, 68-74.

Alshamrani, A., & Bahattab, A. (2015). A comparison between three sdlc models *waterfall* model, spiral model, and incremental/iterative model. *International Journal of Computer Science Issues*, 12 (1), 106-111.

Andrade, C.; Morais, J. (14 de março de 2014). Classificação dos Sistemas de Informação. Consultado a 15 de julho de 2018, de *Prezi*: https://prezi.com/eitp4jcyks3b/classificacao-dos-sistemas-de-informacao/

Ashraf, S., & Aftab, S. (2017). IScrum: an improved Scrum process model. *Modern Education and Computer Science*, 9 (8), 16-24.

Baba, A., & Nonyelum, O. (2017). BOEHM-*waterfall* methodology: issues and challenges. *International Journal of Computer Science and Information Technology & Security*, 7 (4), 49-51.

Batista, P., Marques, C. G., & Carrilho, A. (2018). Criação e Desenvolvimento de uma app de reporte de danos em materiais de armazém para sistemas SAP. *Revista Superavit: Revista de Gestão & Ideias* (artigo submetido).

Bassil, Y. (2012). A simulation model for the *waterfall* software development life cycle. *International Journal of Computer Science Issues*, 2 (5), 742-749.

Boell, S., & Cecez-Kecmanovic, D. (2015). What is an Information System? *Proceedings of the 48th Hawaiian International Conference on System Sciences* (HICSS) (pp. 4959-4968). Kauaii: IEEE Computer Society.

Bragg, J. (9 de junho de 2017). SAP Leonardo – What is it and what we now know after Sapphire 2017. Consultado a 4 de setembro de 2018, de *Mindset*: https://www.mindsetconsulting.com/sap-leonardo-what-is-it-and-what-we-now-know-after-sapphire-2017/

Buajordet, A. (2017). Using the Neptune Application Designer. Consultado a 21 de julho de 2018, de *Community*: https://community.neptune-software.com/documentation/using-the-neptune-application-designer

Cajiza, E. (20 de maio de 2015). Vantagens e desvantagens de um Sistema de Informação. Consultado a 25 de maio de 2018, de *Diversidades e Vivencias*: http://diversidadesevivencias.blogspot.com/2015/05/vantagens-e-desvantagens-de-um-sistema.html

Carey, S. (7 de junho de 2018). What is SAP Leonardo? Everything you need to know about SAP's Leonardo platform. Consultado a 8 de setembro de 2018, de *Computerworld*: https://www.computerworlduk.com/cloud-computing/everything-you-need-know-about-saps-new-look-leonardo-platform-3659277/~

Cavalcanti, N. (1 de março de 2011). Os principais módulos sap e seus relaciomentos. Consultado a 3 de setembro de 2018 de *Nat solutions*: http://www.blog.natsolutions.com.br/10/

Chandana. (6 de julho de 2018). Scrum Project Management – Pros and Cons. Consultado a 16 de julho de 2018, de *Simplilearn*: https://www.simplilearn.com/scrum-project-management-article

Cleverism. (2018). ABAP. Consultado a 14 de julho de 2018, de *Cleverism*: https://www.cleverism.com/skills-and-tools/abap/

Conceito.de (2018). Conceito de Sistemas de Informação. Consultado a 14 de julho de 2018, de *Conceito*.de: https://conceito.de/sistema-de-informacao

Conquesti. (2018). SAP® Hana Foundation. Consultado a 9 de setembro de 2018 de *Conquesti*: https://ead.conquesti.com.br/loja/catalogo/cqt800e-sap-hana-foundation_20/

DeMent, B. (2018). Global Business Services (GBS). Consultado a 1 de setembro de 2018, de *Scottmadden Management Consultants*: https://www.scottmadden.com/insight/global-business-services-gbs/

Desenvolvimentoagil. (2014). SCRUM. Consultado a 5 de agosto de 2018 de *Desenvolvimentoagil*: https://www.desenvolvimentoagil.com.br/scrum/

Farber, F., Cha, S., Primsch, J., Bornhovd, C., Sigg, S., & Wolfgang, L. (2011). SAP hana database – data management for modern business applications. *SIGMOD Record*, 45–51.

Farber, F., May, N., Lehner, W., Grobe, P., Muller, I., Rauhe, H., & Dees, J. (2012). The SAP Hana database–an architecture overview. *IEEE Data Engineering Bulletin*, 35 (1) 28-33.

Fowler, M., & Scott, K. (1999). *UML distilled a brief guide to the standard object modelling language*. Boston: Pearson Education, Inc,.

Genevois, M. (2018 de maio de 28). SAP Lodo neu 2011. Consultado a 10 de agosto de 2018 de *Agridées*: https://www.agridees.com/interview/marc-genevois/sap-logo-neu-2011-2000px-png/

GrandviewSolutions. (2 de setembro de 2017). Let’s talk – ERP. Consultado a 25 de julho de 2018, de *GrandviewSolutions*: http://grandviewcorp.com/lets-talk-erp/

Guru99. (2018). What is SAP? Definition of SAP ERP Software. Consultado a 24 de junho de 2018, de *guru*: https://www.guru99.com/what-is-sap-definition-of-sap-erp-software.html

Kisspng. (2017). Scrum Agile software development Kanban Computer Software - organization chart. Consultado a 20 de Agosto de 2018, de *Kisspng*: https://www.kisspng.com/png-scrum-agile-software-development-kanban-computer-s-804958/preview.html

Institute of Electrical and Electronics Engineers [IEEE] (1991). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. In *IEEE Std 610* (pp.1-217). Piscataway, Nova Jersey: IEEE.

Marques, C. G. (2007). Sociedade Digital. In *Actas da IX Jornadas da Escola Superior de Educação de Beja, Cultura Digital*. Beja: Escola Superior de Educação de Beja, Instituto Politécnico de Beja.

Marques, C. G. (2018). Conceitos e estratégias para a gestão de um negócio online. In C. G. Marques, A. Nata, G. Brás, & S. P. Gonçalves (Orgs.), *Atas da XIII Semana de Gestão - Indústria 4.0, Desafios e Oportunidades* (pp. 73-86). Tomar: Instituto Politécnico de Tomar.

Matsu, C. (15 de setembro de 2017). SAP Leonardo: a aposta da SAP para levar tecnologias emergentes a seus clientes. Consultado a 12 de Julho de 2018, de *Computerworld*: https://computerworld.com.br/2017/09/15/sap-leonardo-aposta-da-sap-para-levar-tecnologias-emergentes-seus-clientes/~

Mesquita, R. (26 de Agosto de 2016). *O que é sistema de informação e quais são as suas características?* Consultado de *Saia do lugar*: http://saiadolugar.com.br/sistema-de-informacao/

Moura, D. (27 de julho de 2016). Abordagem Tradicional x Abordagem Ágil para Desenvolvimento de Software. Consultado a 13 de agosto de 2018, de *Linkedin*: https://pt.linkedin.com/pulse/abordagem-tradicional-x-%C3%A1gil-para-desenvolvimento-de-software-moura

Munassar, N., & Govardhan, A. (2010). A comparison between five models of software engineering. *International Journal of Computer Science Issues*, 7 (5), 94–101.

Neptune Software. (2018a). About Neptune Software and what it does for you. Consultado a 27 de agosto de 2018 de *Neptune Software*: https://www.neptune-software.com/about/

Neptune Software. (2018b). Documentation Neptune Digital Experience Platform. Consultado a 12 de agosto de 2018, de *Community*: https://community.neptune-software.com/documentation

Neptune Software. (2018c). Start Connecting with Your Neptune Community. Consultado a 4 de setembro de 2018, de *Community*: https://community.neptune-software.com/

Pène, E., & Collin, J. (2017). How Neptune Software can accelerate Fiori & SAPUI5 developments? In *SAP Inside Track Paris*. Consultado em 14 de Setembro https://www.sap.com/documents/2017/12/848d6dc3-e57c-0010-82c7-eda71af511fa.html

Pete. (12 de Fevereiro de 2012). Beginners Guide to ABAP – Module 1 – *SAP System Architecture*. Consultado a 5 de setembro de 2018, de SAP Training HQ: http://www.saptraininghq.com/sap-system-architecture/

Plattner, H., & Zeier, A. (2011). *In-Memory Data Management: An Inflection Point for Enterprise Applications*. Germany: Springer.

Procensus. (s.d.). FIORI. Consultado a 30 de agosto de 2018, de *Procensus*: http://www.procensus.pt/wp-content/uploads/2016/09/sap-fiori_PROCENSUS.pdf

Projectbuilder. (30 de maio de 2017). Gestão de projetos: Ágil ou tradicional? Entenda as diferenças. Consultado a 12 de agosto de 2018 de *Projectbuilder*: https://www.projectbuilder.com.br/blog/gestao-de-projetos-agil-ou-tradicional-entenda-as-diferencas/

Purestorage. (2018). The Road to SAP Hana. Consultado a 15 de julho de 2018, de *Purestorage*: https://www.purestorage.com/resources/type-a/road_to_sap_hana.html

Rashid, M., Hossain, L., & Patrick, J. (2002). The evolution of ERP systems: a historical perspective. Em M. Rashid, L. Hossain, & J. Patrick (Eds.), *Entreprise Resourse*

Planning Global Opportunities and Challenges (pp. 1-16). United States of America: Idea Group Pub.

Retailsolutions. (2018). Neptune Software – SAP Fiori Apps in only a few days! Consultado a 4 de julho de 2018, de *Retailsolutions*: https://www.retailsolutions.de/en/portfolio/solutions/neptune-software.html

Rouse, M. (2017). SAP HANA. Consultado a 4 de julho de 2018, de *SearchSAP*: https://searchsap.techtarget.com/definition/HANA-SAP-HANA

SAP. (2017). SAP Leonardo Digital Innovation System: Capabilities & Success Stories. Consultado a 24 de Agosto de 2018, de *SAP*: https://itelligencegroup.com/wp-content/usermedia/solution-brief-sap-leonardo-glo-en.pdf

SAP. (2018). Why SAP Fiori? Consultado a 8 de setembro de 2018, de *SAP*: https://www.sap.com/products/fiori.html

SAP. (s.d.a). SAP: A 46-year history of success. Consultado a 3 de agosto de 2018, de *SAP*: https://www.sap.com/corporate/en/company/history.html

SAP. (s.d.b). ABAP Development Community. Consultado a 19 de agosto de 2018, de *SAP*: https://www.sap.com/portugal/community/topic/abap.html

SAP. (s.d.c). Classes. Consultado a 13 de agosto de 2018, de *SAP* Documentation: https://help.sap.com/saphelp_nw70/helpdata/en/c3/225b5c54f411d194a60000e8353423/frameset.htm

Scruminc. (2018). Scrum Training & Certification. Consultado a 7 de setembro de 2018, de *Scruminc*: https://www.scruminc.com/

SelectHub. (2018). ERP Advantages and Disadvantages. Consultado a 3 de setembro de 2018, de *SelectHub Confidence in Software*: https://selecthub.com/enterprise-resource-planning/erp-advantages-and-disadvantages/

Sistemasgrainhill. (23 de junho de 2013). Implementar Notas SAP. Consultado a 3 de Agosto de 2018, de *Sistemasgrainhill*: https://sistemasgrainhill.wordpress.com/2013/06/24/implementar-notas-sap/

SoftINSA. (2018a). Sobre a Softinsa. Consultado a 15 de fevereiro de 2018, de *SofINSA*: http://www.softinsa.pt/section/softinsa

SoftINSA. (2018b). Centros de inovação tecnológica. Consultado a 15 de fevereiro de 2018, de *SofINSA*: http://www.sofinsa.pt/section/centros-de-inovacao

SoftINSA. (2018c). SAP Consulting. Consultado a 1 de setembro de 2018, de *SoftINSA*: http://www.softinsa.pt/section/sap

STechies. (s.d.). SAP HANA. Consultado a 10 de agosto de 2018, de *STechies*: https://www.stechies.com/hana/

Stellman, A., & Greene, J. (2005). *Applied software project management*. USA: O'Reilly Media.

Techopedia. (s.d.). SAP HANA. Consultado a 6 de setembro de 2018 de *Techopedia*: https://www.techopedia.com/definition/28540/sap-hana

Turban, E., McLean, E., & Wetherbe, J. (1999). *Information Technology for Management – Making Connections for Strategic Advantage*, 2ª edição, New York John Wiley.

Tutorialspoint. (2018a). SAP ABAP - Overview. Consultado a 19 de maio de 2018 de *Tutorialspoint*: https://www.tutorialspoint.com/sap_abap/sap_abap_overview.htm

Tutorialspoint. (2018b). SAP Fiori - Introduction. Consultado a 20 de maio de 2018 de *Tutorialspoint*: https://www.tutorialspoint.com/sap_fiori/sap_fiori_introduction.htm

Universo-Projeto. (19 de março de 2014). Qual é o melhor entre o Modelo Ágil e o Tradicional? Consultado a 25 de setembro de 2018, de *Universo-Projeto*: https://universoprojeto.wordpress.com/tag/waterfall/

Vinicius. (2016). Processo de teste ágil x tradicional. Consultado a 11 de agosto de 2018, de *Devmedia*: https://www.devmedia.com.br/processo-de-teste-agil-x-tradicional/36854

Wailgum, T., & Perkins, B. (12 de fevereiro de 2018). What is ERP? a guide to enterprise resourse planning systems. Consultado a 3 de Agosto de 2018, de *CIO*: https://www.cio.com/article/2439502/enterprise-resource-planning/enterprise-resource-planning-erp-definition-and-solutions.html

Wong, A. (14 de maio de 2018). The SAP Leonardo Digital Innovation System Explained! Consultado a 17 de agosto de 2018, de *Techarp*: https://www.techarp.com/enterprise/sap-leonardo-explained/