

Otimização da análise de OSS Notes em SAP

Ricardo José Cruz Bastos

*Relatório de Projeto apresentado à Escola Superior de Tecnologia e Gestão para
obtenção do Grau de Mestre em Informática*

Trabalho realizado sob a orientação de:

João Paulo Ribeiro Pereira

Bragança

Outubro de 2023

Resumo

O presente relatório retrata o projeto desenvolvido no âmbito da dissertação do mestrado em Informática.

O projeto desenvolvido teve como propósito desenvolver uma aplicação no ERP SAP capaz de analisar notas OSS de forma a permitir que, além de transformar um processo extremamente demorado num processo automatizado, não ocorram falhas no processo de identificação de pré-requisitos e ordem de implementação.

É finalidade deste projeto auxiliar não só os consultores SAP, como também qualquer programador ou utilizador do ERP a determinar uma estimativa acertada para a implementação de notas, diminuindo o tempo de análise que, quando feita incorretamente, pode gerar múltiplas inconsistências nos sistemas.

O plano de trabalho encontra-se estruturado em quatro fases. Numa fase inicial, foi realizado o estudo das tecnologias a utilizar permitindo a sua integração no projeto.

A segunda fase consistiu no desenvolvimento da arquitetura da aplicação, através da modelação e da criação das *mockups*. No que concerne à terceira fase, esta compreendeu o desenvolvimento da solução, iniciando-se na implementação do protótipo, passando pelo desenvolvimento da solução em SAP e finalizando com a criação de uma aplicação transaccional em Fiori.

Finalmente, a última fase consistiu nos testes e validação da solução através da comparação da análise de notas utilizando duas metodologias. A metodologia manual, utilizada no dia a dia por consultores, programadores e utilizadores finais e a metodologia automatizada desenvolvida.

Palavras-Chave: Oss Notes, SAP ERP, Fiori, SMTP, Desenvolvimento de aplicações.

Abstract

The present report portrays the project developed within the scope of the master thesis in Informatics.

The purpose of the project was to develop an application within the SAP ERP system capable of analyzing OSS Notes in a way that not only transforms an extremely time-consuming process into an automated one but also ensures no failures occur in the manual identification of prerequisites and implementation order.

The goal of this project is to assist not only SAP consultants but also any programmer or end user in accurately estimating the implementation of notes, reducing the analysis time, which, when done incorrectly, can lead to multiple inconsistencies in the systems.

The workplan was divided into four phases. In the initial phase, a study of the technologies to be used was conducted to enable their integration into the project. The second phase involved the development of the application's architecture through modeling and creating the mockups. As for the third phase, it encompassed the development of the solution, starting with the implementation of the prototype to download SAP Notes, moving on to the SAP application development and concluding with the creation of a transactional application in Fiori.

Finally, the last phase consisted of testing and validating the solution by comparing the analysis of OSS Notes using two different methodologies. The manual methodology, used on a daily basis by consultants, programmers and end users, and the automated methodology developed.

Keywords: Oss Notes, SAP ERP, Fiori, SMTP, Application Development.

Índice Geral

Resumo	iii
Abstract.....	v
Índice Geral	vii
Lista de Siglas/Abreviaturas	ix
Índice de Figuras	xi
Índice de Tabelas	xiii
Capítulo 1 Introdução	1
1.1. Enquadramento	2
1.2. Objetivos	2
1.3. Estrutura do documento	3
Capítulo 2 Problema /Contextualização	5
2.1. Contextualização.....	5
2.2. Estado de Arte.....	5
2.2.1. Transação SNOTE.....	6
2.2.2. SAP <i>Support Portal</i>	7
Capítulo 3 Revisão Bibliográfica	9
3.1. ERP	9
3.2. Empresa SAP	10
3.3. ERP SAP.....	11
Capítulo 4 Ferramentas/Tecnologias.....	15
4.1. SAP ABAP	15
4.2. SAP Fiori	15
4.2.1. Tipos de aplicações Fiori.....	16
4.3. Notas OSS.....	16
4.4. SAP SMTP.....	17
4.5. Figma	18
4.6. Visual Paradigm.....	18
Capítulo 5 Arquitetura e Modelação do Sistema	19
5.1. Requisitos Funcionais	19
5.1.1. Analisar notas SAP.....	19
5.1.1.1. Gerar Relatório da análise	20
5.1.1.2. Consultar listagem ordenada de notas	20
5.1.1.3. Consultar passos manuais.....	20
5.1.1.4. Gerir estimativa	20

5.1.1.5.	Enviar estimativa por e-mail	21
5.2.	Requisitos Não-Funcionais	21
5.2.1.	Requisitos Gerais	21
5.2.1.1.	Performance e Confiabilidade	21
5.2.1.2.	Utilidade	21
5.2.1.3.	Interface Gráfica	21
5.3.	Diagrama de Casos de Uso	22
5.1.	Diagrama de Atividades	23
5.2.	Mockups	24
Capítulo 6	Desenvolvimento	27
6.1.	Implementação do protótipo de <i>download</i> de Notas	28
6.2.	Solução no SAP GUI	33
6.2.1.	<i>Include</i> ZSNOTEN_RCB_DAT	34
6.2.2.	<i>Include</i> ZSNOTEN_RCB_SRC	34
6.2.3.	<i>Include</i> ZSNOTEN_RCB_CLA	35
6.2.3.1.	Método DOWNLOAD_OSS_NOTAS	36
6.2.3.2.	Método CHECK_OSS_NOTES	37
6.2.3.3.	Método CREATE_FINAL_TABLE	38
6.2.3.4.	Método CREATE_AND_DISPLAY_ALV	41
6.2.3.5.	Método USER_COMMAND	43
6.2.3.6.	Método GET_ESTIMATIVA_FINAL	44
6.2.3.7.	Método ENVIA_EMAIL	45
6.2.3.8.	Método APRESENTA_MSG	47
6.2.4.	<i>Include</i> ZSNOTEN_RCB_F01	48
6.3.	Aplicação Transacional em Fiori	48
6.3.1.	Criação do <i>Intent</i>	49
6.3.2.	Customização do <i>launchpad</i>	49
6.3.3.	Catálogos	50
Capítulo 7	Análise e Discussão de Resultados	61
7.1.	Introdução	61
7.2.	Metodologia e aspetos da avaliação	61
7.3.	Resultados obtidos	62
7.4.	Trabalho futuro	64
Capítulo 8	Conclusões	65
Bibliografia	67

Lista de Siglas/Abreviaturas

ABAP *Advanced Business Application Programming.*

ALV *ABAP List View.*

BAPI *Business Application Programming Interface.*

ERP *Enterprise resource planning.*

FUI *Fiori User Interfaces.*

GUI *Graphical User Interface.*

SAP *Systems Applications and Products.*

SMP *SAP Mobile Platform.*

SMTP *Simple Mail Transfer Protocol.*

SQL *Structured Query Language.*

UC *Unidade Curricular.*

XML *Extensible Markup Language.*

Índice de Figuras

Figura 1 - Download de notas na transação SNOTE.....	6
Figura 2 - SAP <i>Note Browser</i> (permite implementar e visualizar notas SAP).....	7
Figura 3 - SAP <i>Support Portal</i>	8
Figura 4 - Logotipo Empresa SAP [15].....	10
Figura 5 - Módulos SAP.....	11
Figura 6 - SAP <i>Easy Access</i>	12
Figura 7 - Serviço SAP SMTP	17
Figura 10 - Diagrama de Casos de Uso	22
Figura 9 - Diagrama de Atividades.....	24
Figura 11 - <i>Launchpad</i> de Fiori.....	25
Figura 12 - Ecrã de seleção da aplicação.....	25
Figura 13 - <i>Report</i> com a listagem ordenada de notas	26
Figura 14 - Metodologia utilizada	27
Figura 15 - Download Notas SAP	28
Figura 16 - Assistente de notas.....	29
Figura 17 - Classe CL_SCWN_NOTE_SAR_FILE_N	30
Figura 18 - SM59 - Conexões RFC	31
Figura 19 - Configurações técnicas da conexão RFC	31
Figura 20 - Configurações de Logon & Segurança	32
Figura 21 - Ecrã de seleção.....	35
Figura 22 - Ecrã de seleção - Alterar estatísticas.....	35
Figura 23 - Tipo de tabela TT_CWB_NOTE_DISPLAY	37
Figura 24 - Estrutura CWB_NOTE_DISPLAY	38
Figura 25 - Estrutura ZSNOTEN	41
Figura 26 - Exemplo listagem gerada.....	43
Figura 27 - Exemplo de e-mail enviado	46
Figura 28 - Exemplo de listagem em anexo no e-mail	47
Figura 29 - Transação ZSNOTEN.....	48
Figura 30 - Secções do painel de customização do launchpad.....	50
Figura 31 - Criação do Catálogo.....	51
Figura 32 - Criação do <i>Tile</i>	52
Figura 33 - Tipo do <i>Tile</i>	53
Figura 34 - Configuração do <i>Tile</i>	53
Figura 35 - Criação do <i>target mapping</i>	54
Figura 36 - Configuração do <i>target mapping</i>	54
Figura 37 - Criação do <i>tile</i> + <i>target mapping</i> por referência	55
Figura 38 - Criação do grupo.....	56
Figura 39 - Configuração do <i>role</i>	57
Figura 40 - <i>Tile</i> de acesso à aplicação transacional.....	58
Figura 41 - Ecrã de seleção na aplicação transacional em Fiori.....	58
Figura 42 - <i>Output</i> da execução teste	59

Índice de Tabelas

Tabela 1 - Códigos de transação.....	13
Tabela 2 - Estrutura da solução	34
Tabela 3 - Mapeamento entre objeto e tipo de objeto	40
Tabela 4 - Explicação da configuração do tile	53
Tabela 5 - Explicação da configuração do <i>target mapping</i>	55
Tabela 6 - Resultados obtidos na análise manual	63
Tabela 7 - Resultados obtidos através da utilização da solução	63

Capítulo 1 Introdução

No contexto atual de consultoria, nomeadamente no suporte a clientes que utilizem o ERP SAP, a principal metodologia de análise de comportamento *standard* anômalo é a procura de notas OSS com correções para os objetos em falha nos sistemas.

OSS Notes é definido como um serviço online da SAP que fornece atualizações sobre *patches* em diferentes módulos e informações atualizadas sobre notas SAP. As notas SAP são instruções de correção para bugs ou problemas encontrados nos sistemas [1].

É responsabilidade do consultor realizar uma análise para garantir que não há código personalizado que interrompa o fluxo do código *standard*. Se o código e/ou objetos que está(ão) a causar o(s) problema(s) for efetivamente *standard* é então realizada a pesquisa de notas OSS pelos sintomas que o sistema indica (mensagens de erro, *dumps*, requisitos legais...) [2].

Uma vez determinada a nota, ou o conjunto de notas, é necessário confirmar se as mesmas são válidas para a versão de *support package* do sistema dos clientes. Além disso, é requisito de implementação verificar todos os pré-requisitos e seus passos manuais (caso existam).

Embora esta seja a técnica utilizada pela maioria dos consultores, programadores ou utilizadores finais, é uma técnica bastante propícia a erros quando feita manualmente, gerando múltiplas inconsistências nos sistemas. Dito isto, foram exploradas ferramentas *standard* para tornar este processo automatizado e garantir, não só a ordem pela qual as notas devem ser implementadas, mas também a determinação de uma estimativa de implementação das notas mais adequada possível.

1.1. Enquadramento

Em consultoria SAP, sempre que um cliente identifica um comportamento anómalo no sistema que seja *standard*, são identificadas, na maioria dos casos, notas com instruções de correções aos sintomas detetados nos sistemas.

Tendo em conta todo o processo de implementação (pré-análise, validação de pré-requisitos e identificação de passos manuais), podemos definir que a análise de notas SAP é uma tarefa exaustiva, morosa e bastante propícia a erros.

Por esse motivo, automatizar este processo é uma vantagem não só para os consultores, como também para os clientes que, além de evitar falhas na análise manual de pré-requisitos, tornamos o processo mais célere e menos custoso.

1.2. Objetivos

O objetivo deste projeto teve como principal foco o desenvolvimento de uma solução que funcionasse como auxílio na análise de pré-requisitos e identificação de passos manuais durante o processo de implementação de notas SAP.

De forma detalhada, tencionou-se:

- Desenvolver uma aplicação no ERP SAP para determinar a ordem pela qual as notas devem ser instaladas;
- Identificar os pré-requisitos e passos manuais das notas, validando o seu estado de implementação;
- Determinar uma estimativa aproximada para a implementação;
- Enviar por e-mail a listagem ordenada e detalhada das notas, além de incluir o tempo esperado para a sua implementação.

1.3. Estrutura do documento

De forma a facilitar a leitura deste relatório e de demonstrar o trabalho realizado, este documento foi dividido em oito capítulos.

- **Capítulo 1 – Introdução**

No primeiro capítulo é apresentada uma introdução, bem como um enquadramento ao tema em questão, descrevendo os objetivos estabelecidos para a implementação do mesmo.

- **Capítulo 2 – Problema/Contextualização**

Neste capítulo é apresentada uma breve contextualização em que surge o problema enunciado bem como o estado de arte.

- **Capítulo 3 – Revisão Bibliográfica**

No terceiro capítulo são apresentados os conceitos teóricos abordados no âmbito deste projeto.

- **Capítulo 4 – Ferramentas/Tecnologias**

Neste capítulo são apresentadas todas as ferramentas e tecnologias que serviram de apoio ao desenvolvimento deste projeto.

- **Capítulo 5 – Arquitetura e Modelação do Sistema**

O quinto capítulo expõe a arquitetura e modelação desenvolvida. Nesse sentido, são apresentados os diagramas *Unified Modeling Language* (UML) e as *mockups*, tendo como finalidade ilustrar com mais clareza e detalhe o funcionamento da aplicação.

- **Capítulo 6 – Desenvolvimento/Implementação**

O penúltimo capítulo encontra-se dividido em três secções. Cada uma dessas secções descreve de forma pormenorizada a implementação da solução desenvolvida, desde o desenvolvimento do *report* que lista a ordem de implementação das notas à configuração do envio de e-mail e criação posterior do tile transacional.

- **Capítulo 7 - Análise e Discussão de Resultados**

No capítulo sete é apresentada uma reflexão de todo o trabalho desenvolvido, apresentando os resultados obtidos bem como as dificuldades ultrapassadas.

- **Capítulo 8 – Conclusão**

No último capítulo é apresentada a conclusão, fazendo uma breve *overview* do trabalho realizado discutindo os resultados obtidos.

Capítulo 2 Problema /Contextualização

2.1. Contextualização

Como já referenciado no Capítulo 1, sempre que são detetadas anomalias no comportamento *standard* dos sistemas SAP e essa(s) anomalia(s) não foram criadas por código personalizado, é realizada uma listagem de notas SAP, tendo por base os sintomas dos sistemas.

Através dessa listagem, é necessário identificar quais das notas são válidas para a versão do *support package* dos sistemas e identificar quais das notas são válidas para implementar. Além disso, é necessário realizar uma listagem dos pré-requisitos das notas a implementar, bem como identificar os passos manuais a realizar.

Dada a necessidade de automatizar o processo de análise de notas SAP, este projeto surge de forma a auxiliar os consultores, programadores e utilizadores finais a identificar todos os passos necessários para a implementação das notas, permitindo determinar uma estimativa adequada para a sua implementação.

2.2. Estado de Arte

De forma a implementar uma solução capaz de produzir uma listagem ordenada de notas, bem como a identificação dos seus pré-requisitos e passos manuais, realizaram-se várias pesquisas sobre o que são notas SAP, métodos de determinação de pré-requisitos, passos

manuais e foram também analisadas metodologias *standard* de hierarquias e dependências das notas.

Das várias pesquisas realizadas, foi tomado como exemplo a transação SNOTE (ferramenta assistente de notas SAP) e o *SAP Support Portal* (portal da SAP que permite a pesquisa por notas, artigos de base de conhecimento, conteúdo da comunidade como *threads* entre outros).

2.2.1. Transação SNOTE

A transação SNOTE no ERP SAP, permite o *download* de notas OSS diretamente do servidor de aplicações do *marketplace* da SAP [3].

Uma das vantagens da SNOTE é que, além de permitir o *download* das notas (Figura 1).

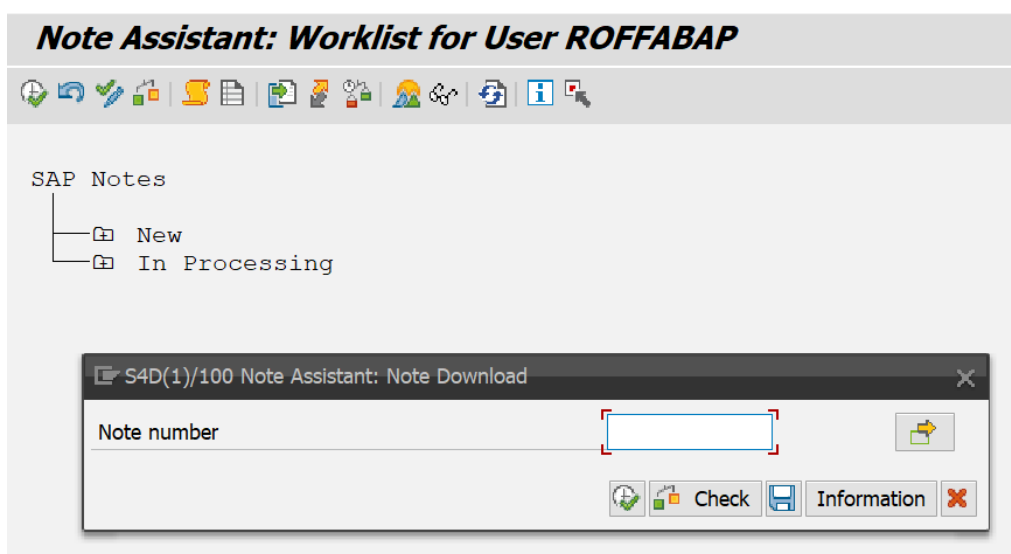


Figura 1 - Download de notas na transação SNOTE

É nesta transação que podemos proceder com a implementação e visualização do documento referente à nota utilizando o *SAP Note Browser* (Figura 2). Este documento contém a descrição, sintomas, estado de implementação e uma listagem dos componentes de suporte (*support packages* correspondentes).

Note Assistant: Note Browser

SAP Note Number to

Application Component to

Processing Status

Implementation State

Processors

Selection

Software Component

Release

Sort Order

Number

Application Component

Figura 2 - SAP Note Browser (permite implementar e visualizar notas SAP)

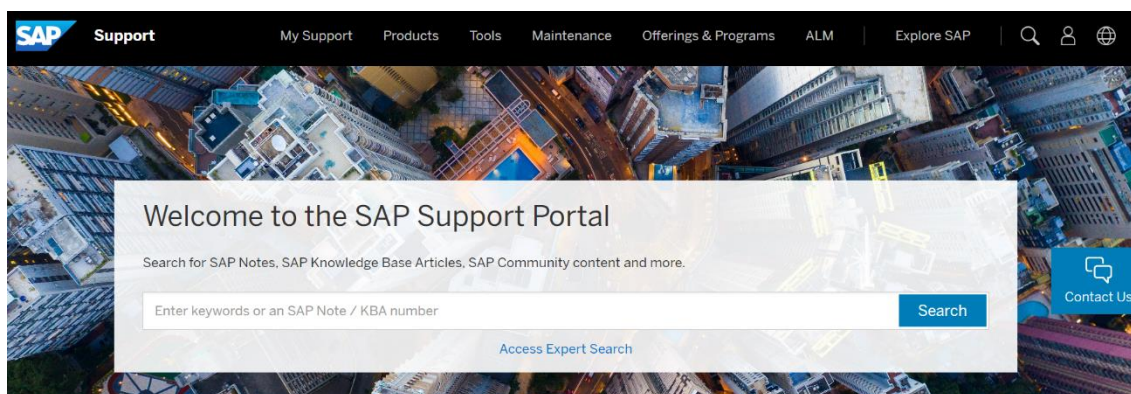
2.2.2. SAP Support Portal

O SAP Support Portal é uma plataforma projetada para fornecer ferramentas de suporte e serviços [4].

Para aceder a este portal, é necessário um *s-user* e *password*. É maioritariamente utilizado para:

- Obter suporte de especialistas em SAP, através da criação de incidentes;
- Procurar notas e artigos de conhecimento;
- Adicionar e remover utilizadores;
- Gerir todos os sistemas e instalações;
- Registrar chaves de desenvolvedor e objetos (SSCR), chaves de licença e instalação entre outras.

As chaves de desenvolvedor permitem atribuir permissões a utilizadores para poderem editar objetos e as chaves de objetos autorizam a edição de objetos *standard*.



Use the **quick links below** for direct access to top tasks in **SAP for Me**, your **central access point** for all support related tasks.

Figura 3 - SAP Support Portal

Capítulo 3 Revisão Bibliográfica

3.1. ERP

ERP, “*Enterprise Resource Planning*” (em português, Planeamento de Recursos Empresariais), surge com o objetivo de reformular a organização interna das empresas. Estima-se que o ERP teve a sua origem na empresa alemã SAP quando lançou o R/2, uma versão pioneira do posterior software ERP.

Um sistema ERP consiste num conjunto de módulos de software integrados e uma base de dados central que permite às organizações gerir os seus recursos de forma eficaz e eficiente, através da automação de processos empresariais, partilha de dados e acessos a informações atualizadas em tempo real [5].

A implementação dos ERP nas mais variadas empresas tem vantagens comuns, nomeadamente, a redução de tempo e custos associados a diversos processos de trabalho, aumento da eficiência e eficácia globais, maior facilidade em tomar decisões estratégicas, operacionais, organizacionais e financeiras [6, 7, 8]. Além disso, [9] aponta também a melhoria do desempenho comercial e do posicionamento das organizações, otimização do capital de trabalho e a melhoria do atendimento ao cliente como consequências positivas dos ERP nas empresas. Apesar de todas estas vantagens, um ERP tem alguns fatores menos positivos como: custo elevado, tempo de implementação, personalização (necessária para responder à necessidade do cliente, mas que pode aumentar a possibilidade de erros e dificultar atualizações futuras), migração de dados (transferir dados para um novo sistema ERP pode ser complexo e a execução incorreta pode levar a perda de dados, p.e.) e resistência dos funcionários (que necessitarão de se adaptar a uma nova forma de proceder).

Estas vantagens têm feito com que as empresas, à medida que evoluem – procurando melhores desempenhos comerciais e estratégias competitivas duradouras, optem pela adoção de sistemas ERP em detrimento da melhoria dos sistemas de formação internos tradicionais que possuem [10].

Ainda relativamente aos sistemas de informação internos tradicionais, a sua implementação é muito diferente da implementação de um ERP. A adoção bem-sucedida de um ERP requer planeamento e uma gestão que abranja tanto o segmento tecnológico como organizacional [11].

3.2. Empresa SAP

A SAP SE foi fundada na Alemanha por cinco ex-funcionários da IBM em 1972. Esta é uma multinacional de *software* reconhecida pelo fornecimento de soluções tecnológicas para empresas em todo o mundo. Esta empresa encontra-se associada à comercialização de softwares de gestão empresarial (ERP) e também de diversos produtos e serviços relacionados com sistemas de informação.

A SAP é uma das maiores empresas do mundo da sua vertente, sendo reconhecida por fornecer produtos/soluções que permitem otimizar processos operacionais, gerir recursos humanos, contabilidade, logística, *supply-chain*, entre outros.

Além da versatilidade de áreas funcionais possíveis de trabalhar nos ERP da SAP, destaca-se também a versatilidade da ótica de integração com outras tecnologias - em maio de 2023 a SAP anunciou investimentos para investigação e desenvolvimento de novas tecnologias em parceria com a Microsoft, Google Cloud e IBM [12, 13, 14].

A SAP encontra-se em Portugal desde 1993, altura em que foi inaugurado o escritório de Lisboa [15].

Figura 4 - Logotipo Empresa SAP [15]



3.3. ERP SAP

De acordo com a(s) área(s) de especialização que existem numa empresa, existe um módulo associado no ERP. Atualmente, a SAP providencia os seguintes módulos: [16]:

- CO – *Controlling* (Contabilidade)
- FI – *Finance* (Finanças)
- HR – *Human Resources* (Recursos Humanos) ou *HCM* – *Human Capital Management* (Gestão de Capital Humano)
- MM – *Material Management* (Gestão de Materiais)
- PM – *Plant Maintenance* (Manutenção do Edifício)
- PP – *Production Planning* (Planeamento de Produção)
- QM – *Quality Management* (Gestão da Qualidade)
- SD – *Sales & Distribution* (Vendas e Distribuição)
- WM – *Warehouse Management* (Gestão de Armazém)
- TR – *Transport Request* (Solicitação de Transporte)
- WF – *Workflow*
- IS – *Industry Solutions* (Soluções Industriais)
- BW – *Business Warehousing* (Armazém Empresarial)
- RE – *Real State* (Imobiliário)

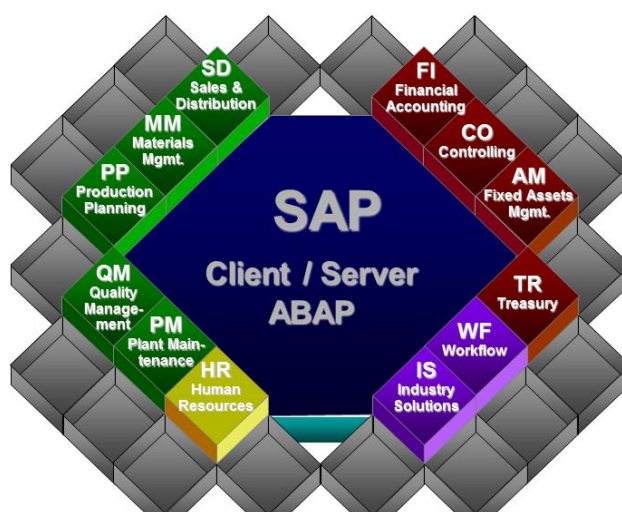


Figura 5 - Módulos SAP

Os utilizadores acedem ao ERP através do SAP GUI. SAP GUI é a aplicação para o cliente que permite ao utilizador aceder aos diferentes sistemas SAP. Após autenticado, os utilizadores são redireccionados para o ecrã principal – *SAP Easy Access*.

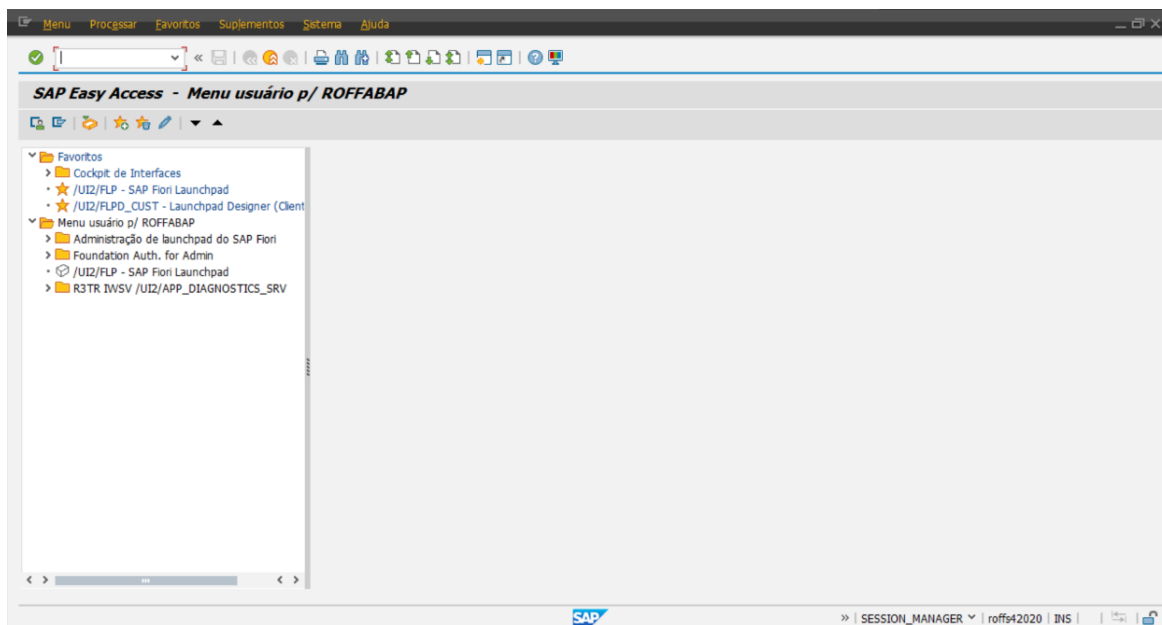


Figura 6 - *SAP Easy Access*

Cada um dos nós apresentados contém várias ramificações (sub-nós) que, por sua vez, possuem outros sub-nós até chegar à unidade fundamental: transações. Cada transação é uma operação específica (ou uma sequência de operações) que o utilizador pode executar em prol de realizar uma tarefa específica.

Estas operações são ações como criar uma fatura, processar uma ordem de compra, verificar informações de stock, entre outras. Embora seja possível aceder a todas as transações através do SAP Easy Access, cada transição tem um código exclusivo. Ao pesquisar diretamente por este código na barra de pesquisa, acede-se diretamente à transação pretendida.

Na Tabela 1, são apresentadas algumas transações utilizadas no âmbito do desenvolvimento deste projeto.

Tabela 1 - Códigos de transação

Código de Transação	Descrição
SE11	Permite o acesso ao dicionário de dados ABAP
SE16N	Permite a exibição geral de tabelas
SE37	Permite a gestão de módulos de função
SE38	Permite a gestão de programas
SM59	Permite a gestão de conexões
SNOTE	Assistente de notas OSS
SOST	Permite visualizar e administrar todas as mensagens enviadas
SPRO	Permite gerir objetos por módulo
PFCG	Permite a criação de <i>roles</i>
/UI2/FLP	Permite o acesso ao <i>launchpad</i> de Fiori
/UI2/FLPD_CONF	Permite o acesso ao <i>launchpad</i> de customização de Fiori
/UI2/SEM OBJ_SA	Permite a criação de objetos semânticos

Quando falamos da organização interna do SAP, é fundamental conhecer o conceito de Dados Mestre que envolve todo o tipo de informação necessário para gerir uma empresa, nomeadamente os dados dos clientes, fornecedores, contas, funcionários, matérias, entre outros. O acesso a estas informações específicas requer a utilização do Mandante.

O Mandante é uma unidade organizacional no sistema que representa o grupo corporativo e, ao mesmo tempo, representa o nível mais alto da hierarquia no sistema SAP. Deste modo, o Mandante tem como objetivo configurar, desenvolver e testar. Dentro de um Mandante é possível criar um número ilimitado de empresas, cada uma com as suas tabelas e configurações. Deste modo, evitam-se duplicações de dados e informações, tornando o Mandante um espaço de dados exclusivo e único.

Capítulo 4 Ferramentas/Tecnologias

4.1. SAP ABAP

ABAP, “*Advanced Business Application Programming*” (em português, Programação Avançada de Aplicações Empresariais), é uma linguagem de programação desenvolvida pela SAP para a implementação de aplicações empresariais no ambiente SAP. O Open SQL está incorporado na linguagem de programação ABAP para ler e modificar o conteúdo da base de dados [17].

ABAP List Viewer, mais conhecido por ALV, é um dos elementos mais utilizados para criar *reports* e possui três formas diferentes de visualização, são elas ALV de lista simples, ALV *tree* e ALV sequencial hierárquica.

No âmbito deste projeto, foi implementada uma ALV de lista simples, nomeadamente, o módulo função REUSE_ALV_GRID_DISPLAY_LVC para apresentar listagens ordenadas das notas a implementar.

4.2. SAP Fiori

O SAP Fiori é uma linguagem de design desenvolvida pela SAP para ser utilizada pelos seus clientes, parceiros e pela própria SAP em aplicações empresariais [18].

As aplicações que utilizam esta linguagem são frequentemente designadas por aplicações Fiori ou interfaces Fiori User Interfaces (FUIs).

Os designs do SAP Fiori podem ser implementados utilizando qualquer tecnologia, embora a SAP forneça bibliotecas de UI compatíveis com Fiori na sua

biblioteca SAPUI5 JavaScript, bem como o kit de desenvolvimento de software (SDK) - SAP Cloud Platform para iOS Android [19].

4.2.1. Tipos de aplicações Fiori

Existem três tipos diferentes de aplicações Fiori, são eles [20]:

- Aplicações transacionais

Estas aplicações são reconhecíveis a partir das transações ABAP clássicas encontradas no SAP GUI. A aplicação criada no âmbito deste projeto foi implementada através da criação de um tile transacional. Um tile é um container que representa uma aplicação na página inicial do SAP Fiori Launchpad.

- Aplicações analíticas

Estas aplicações possuem recursos analíticos integrados. O objetivo destas aplicações é fornecer *insights* de negócios utilizando algoritmos complexos.

- Aplicações de fichas técnicas

Estas aplicações utilizam as capacidades de pesquisa empresarial da base de dados SAP HANA para fornecer resultados de pesquisa em contextos específicos. As aplicações de fichas técnicas, apesar de serem aplicações por direito próprio, são normalmente navegadas a partir de outra aplicação - uma aplicação transacional ou uma aplicação analítica.

4.3. Notas OSS

OSS NOTES é um portal de serviços SAP que fornece informações atualizadas sobre notas SAP. Fornecem frequentemente correções de erros, *patches*, novos desenvolvimentos ou melhorias de programas e outras atualizações *standard*.

Além disso, as notas SAP indicam uma lista de instruções de correções para objetos SAP, podendo essas instruções ser ou não, manuais [21]. De forma a garantir que uma nota está presente num sistema, podemos clicar no botão “SAP Note Browser” na transação SNOTE, como demonstrado na Figura 2.

As notas SAP podem ter sete diferentes estados:

- Pode ser implementada;
- Não é possível implementar;
- Completamente implementada;
- Implementada de modo incompleto;
- Obsoleta;
- Versão obsoleta implementada;
- Estado indefinido de implementação.

4.4. SAP SMTP

O NetWeaver ABAP vem com tudo o que é necessário para o envio de e-mails. Tudo o que é necessário é um servidor SMTP externo que atue como um servidor entre o SAP e o recetor e uma rota para os e-mails recebidos [22].

Como a entrada DNS MX de um domínio aponta normalmente para um servidor SMTP, é necessário configurar uma regra de encaminhamento entre o servidor SMTP e o servidor SMTP SAP, como demonstrado na Figura 7.



Figura 7 - Serviço SAP SMTP

4.5. Figma

Figma é uma ferramenta de design direcionada principalmente para pessoas que necessitem de criar designs de interfaces. Segundo Danny Maiorca, Figma é a melhor escolha para equipas de design UX/UI [23].

Esta foi a ferramenta escolhida para a criação das *mockups* do SAP Fiori.

4.6. Visual Paradigm

É uma ferramenta que permite criar diagramas que definem o funcionamento e a estrutura da aplicação, garantido assim, que numa equipa de desenvolvimento todos os programadores possam trabalhar de forma unida e concisa [24].

Suporta diversos tipos de modelação: UML, BPMN, ArchiMate, DFD, ERD, SoaML, SysML, CMMN [25].

Capítulo 5 Arquitetura e Modelação do Sistema

Neste capítulo, é apresentada toda a arquitetura e modelação desenvolvida no processo de implementação do projeto.

Nesse sentido, são demonstrados os requisitos funcionais e não funcionais, o diagrama de casos de uso, o diagrama de atividades e as *mockups*, tendo como objetivo ilustrar com mais clareza e detalhe o funcionamento da aplicação, bem como as atividades que possam ser realizadas pelo utilizar desde o momento em que indica as notas a serem analisadas até ao momento em que gera a listagem e calcula a estimativa.

5.1. Requisitos Funcionais

5.1.1. Analisar notas SAP

- **Ator:** Qualquer utilizador da aplicação;
- **Descrição:** O sistema deve permitir aos utilizadores analisar notas SAP, devolvendo uma listagem ordenada de notas a implementar, identificando os passos manuais e calculando uma estimativa prevista para a implementação (a estimativa deve ser editável). O sistema deve permitir aos utilizadores enviar a estimativa por e-mail.

5.1.1.1. Gerar Relatório da análise

- **Ator:** Qualquer utilizador da aplicação;
- **Descrição:** O sistema deve permitir aos utilizadores gerar um relatório com uma listagem das notas a implementar, identificando os passos manuais e calculando uma estimativa prevista para a implementação (a estimativa deve ser editável).

5.1.1.2. Consultar listagem ordenada de notas

- **Ator:** Qualquer utilizador da aplicação;
- **Descrição:** O sistema deve permitir aos utilizadores consultar a listagem ordenada de notas a implementar, de forma que, no momento de implementação, não sejam geradas inconsistências nos sistemas.

5.1.1.3. Consultar passos manuais

- **Ator:** Qualquer utilizador da aplicação;
- **Descrição:** O sistema deve permitir aos utilizadores consultar os passos manuais das notas (caso existam), de forma a permitir visualizar os passos que são necessários de realizar antes e/ou depois de implementar as notas.

5.1.1.4. Gerir estimativa

- **Ator:** Qualquer utilizador da aplicação;
- **Descrição:** O sistema deve permitir aos utilizadores visualizar e ajustar a estimativa para implementar as notas de forma que a estimativa seja o mais pertinente possível, uma vez que a duração de implementação de uma nota pode variar de sistema para sistema.

5.1.1.5. Enviar estimativa por e-mail

- **Ator:** Qualquer utilizador da aplicação;
- **Descrição:** O sistema deve permitir aos utilizadores enviar a estimativa das notas a implementar. No e-mail enviado deve ser anexada a listagem gerada no relatório de forma que o destinatário entenda o tempo previsto de implementação.

5.2. Requisitos Não-Funcionais

5.2.1. Requisitos Gerais

5.2.1.1. Performance e Confiabilidade

- **Descrição:** O *download* das notas deve ser realizado apenas se necessário, de forma a evitar *workload* desnecessário. Tanto o *download* como o envio da listagem deve ser realizado de forma segura, utilizando metodologias *standard*.

5.2.1.2. Utilidade

- **Descrição:** A aplicação deve ser fácil de utilizar e deve garantir a redução do tempo de análise das notas, que é uma tarefa morosa e propicia a falhas.

5.2.1.3. Interface Gráfica

- **Descrição:** O sistema deve fornecer uma interface graficamente simples, mas agradável e fácil de entender.

5.3. Diagrama de Casos de Uso

O diagrama de Casos de Uso evidencia as funcionalidades que o utilizador da aplicação pode desempenhar. Uma vez que o foco do desenvolvimento consistiu na implementação de funcionalidades da solução, apenas temos presente a entidade Utilizador.

Um utilizador que tenha acesso à aplicação tem acesso a todas as funcionalidades da aplicação, desde analisar notas SAP a calcular uma estimativa de implementação.

No que diz respeito à análise de notas, o utilizador é obrigado a efetuar o *download* das notas ao executar o programa (caso ainda não estejam no sistema). Se ocorrer algum problema com o *download*, será possível consultar uma listagem de erros. Além dos pontos referidos, o utilizador será capaz de ajustar os parâmetros de auxílio ao cálculo da estimativa e, após a execução do programa, poderá consultar os passos manuais das notas, caso existam.

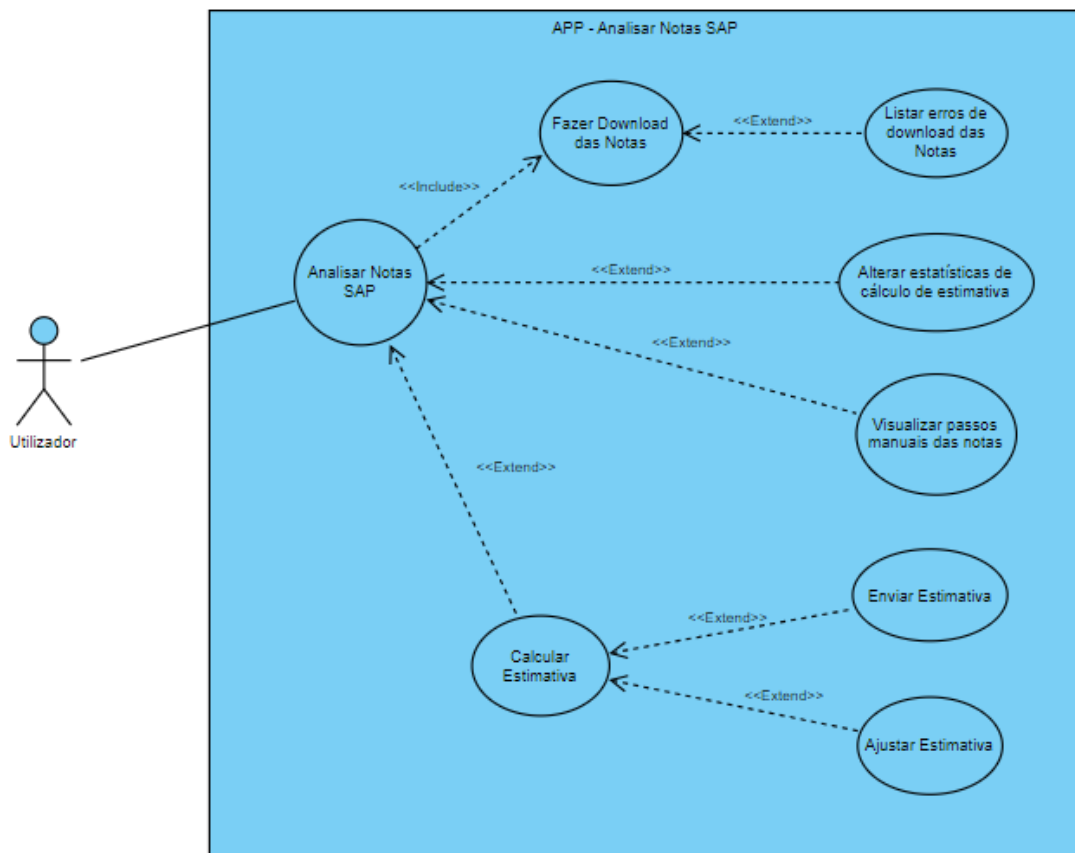


Figura 8 - Diagrama de Casos de Uso

Relativamente ao cálculo da estimativa, o utilizador pode ajustar manualmente o tempo esperado de implementação de cada nota, uma vez que o valor definido inicialmente para o cálculo não é fixo, até porque, por vezes existem notas que, apesar de afetarem vários objetos, contêm modificações com impactos mínimos e levam a que a estimativa possa ser menor. Por outrora, podem existir vários passos manuais levem ao aumento do tempo estimado de implementação.

5.4. Diagrama de Atividades

O fluxo do diagrama de atividades, demonstra todas as tarefas que um interveniente pode desempenhar.

Na Figura 9, conseguimos ver que a primeira etapa desempenhada por um utilizador é o preenchimento do ecrã de seleção. Como parâmetros obrigatórios, o utilizar deve indicar as notas que deseja analisar, o *e-mail* do destinatário da estimativa e definir os valores que o programa deve ter em conta para calcular a estimativa, dependendo do tipo de objeto afetado.

Após preencher o ecrã de seleção, o utilizador pode executar o programa, para analisar a notas. Por sua vez, o programa valida os elementos do ecrã de seleção e existem três opções. Em caso de parâmetros inválidos ou em falta, é devolvida uma mensagem de erro ao utilizador, que apenas tem duas opções, preencher corretamente o ecrã de seleção ou terminar a sua execução. Caso os dados sejam válidos, o programa efetua o *download* (se necessário), cria uma listagem de notas ordenadas, identifica os seus passos manuais (caso existam), verifica o estado de implementação e calcula o tempo esperado tendo por base os parâmetros do ecrã de seleção.

Finalmente, é devolvida a listagem ao utilizador sob a forma de um *report*. O utilizador pode terminar a execução, enviar a estimativa caso o valor esteja de acordo com a sua análise ou ajustar se achar que o tempo esperado não está de acordo.

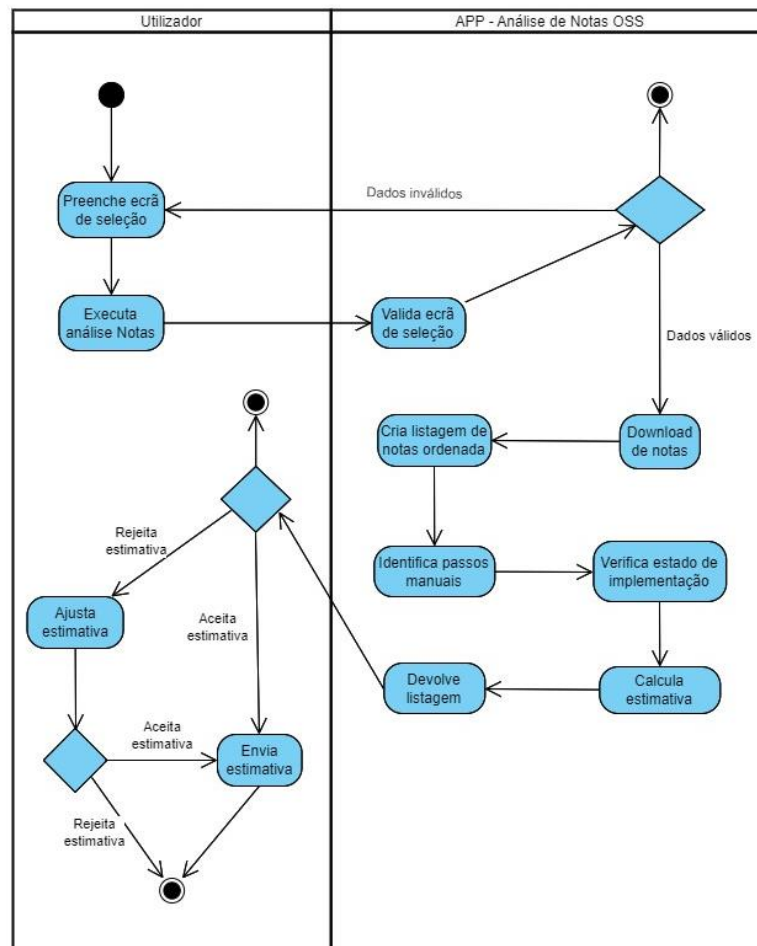


Figura 9 - Diagrama de Atividades

5.5. Mockups

Na Figura 10 é visível o *launchpad* de Fiori, a app criada foi desenhada para enquadrar um grupo dedicado à análise de notas OSS, denominado “Análise de Notas OSS”. Este grupo é apenas visível por utilizadores com *roles* específicos.

Esses *roles* serão criados posteriormente na transação PFCG, identificando os grupos e catálogos que devem ser assignados aos utilizadores correspondentes.

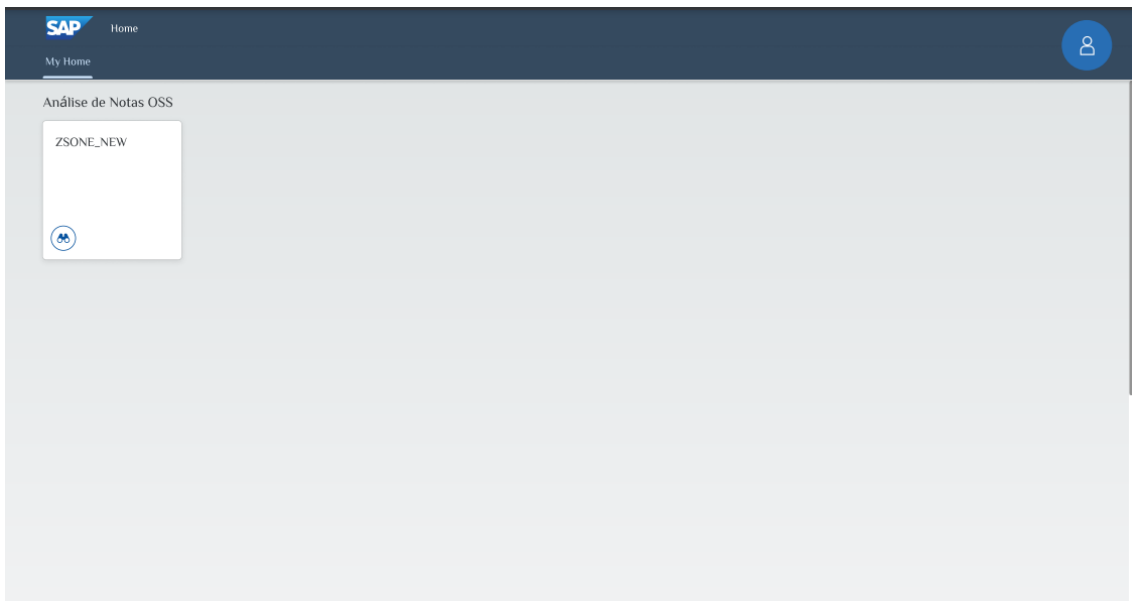


Figura 10 - Launchpad de Fiori

Após abertura do *tile*, é apresentado ao utilizador o ecrã de seleção da solução desenvolvida (Figura 11). É neste momento em que são escolhidas as notas que devem ser analisadas pelo programa e devem ser definidos os endereços de e-mail dos destinatários a enviar a estimativa.

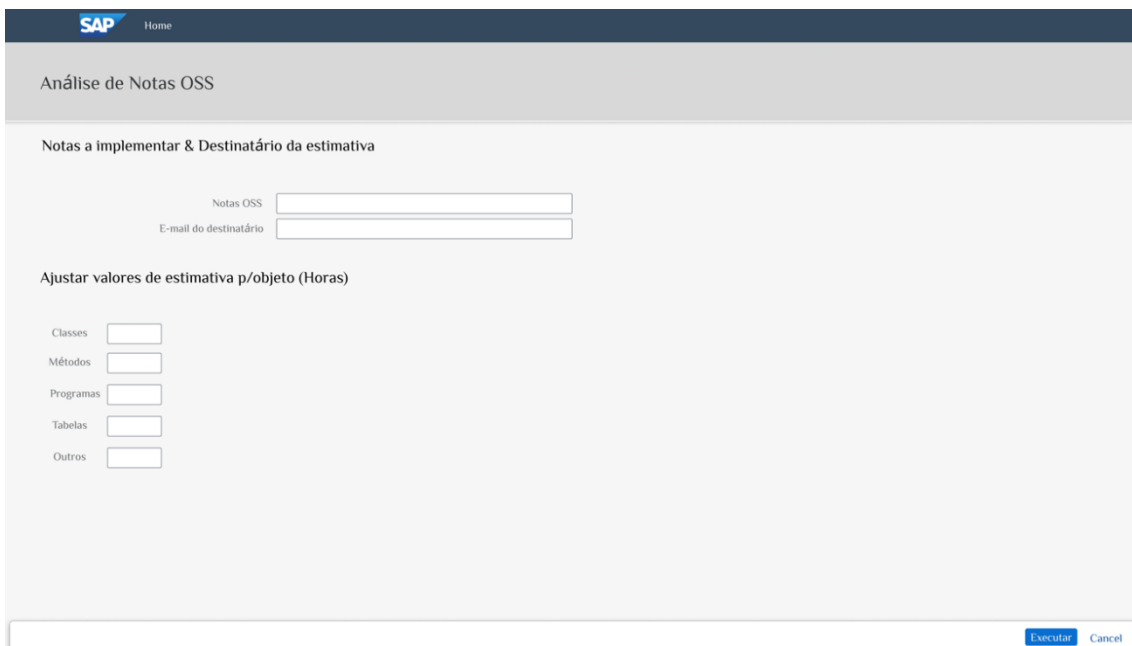
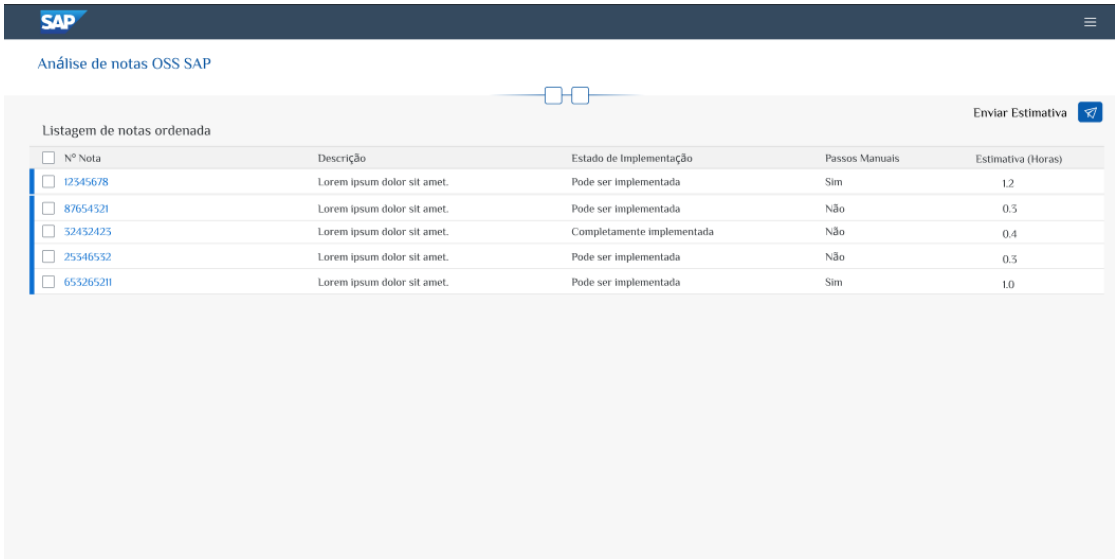


Figura 11 - Ecrã de seleção da aplicação

Além disso, neste mesmo ecrã é possível ajustar os parâmetros de cálculo da estimativa, indicando o tempo em horas, para cada tipo de objeto afetado pelas notas, seja ele uma classe, método, programa, tabela ou outro. Estes parâmetros são úteis uma vez que há objetos que, quando modificados incorretamente por uma nota, têm tendência a gerar múltiplas inconsistências e fazem com que o processo de implementação seja bastante moroso.

Resta referir que esse valor não é considerado fixo uma vez que o tempo estimado por cada nota é ajustável antes de enviar a estimativa final, até porque, por vezes existem notas que, apesar de afetarem vários objetos, contêm modificações com impactos mínimos.



Análise de notas OSS SAP

Listagem de notas ordenada

Enviar Estimativa

<input type="checkbox"/> N° Nota	Descrição	Estado de Implementação	Passos Manuais	Estimativa (Horas)
<input type="checkbox"/> 12545678	Lorem ipsum dolor sit amet.	Pode ser implementada	Sim	1.2
<input type="checkbox"/> 87654321	Lorem ipsum dolor sit amet.	Pode ser implementada	Não	0.5
<input type="checkbox"/> 52452425	Lorem ipsum dolor sit amet.	Completamente implementada	Não	0.4
<input type="checkbox"/> 25346552	Lorem ipsum dolor sit amet.	Pode ser implementada	Não	0.5
<input type="checkbox"/> 65326521	Lorem ipsum dolor sit amet.	Pode ser implementada	Sim	1.0

Figura 12 - Report com a listagem ordenada de notas

Após a execução do programa, será apresentado um *report* com a listagem ordenada das notas que o utilizador deseja implementar (Figura 12). Além disso, é apresentado o estado de implementação da nota, a indicação da existência de passos manuais (bem como o documento contendo os mesmos – caso existam) e uma estimativa do tempo esperado para a implementação das notas (ajustável).

Finalmente, resta referir que o botão enviar “Enviar Estimativa”, enviará a listagem em formato *excel* para os destinatários definidos no ecrã de seleção, indicando o somatório das de todas as notas a implementar.

Capítulo 6 Desenvolvimento

Este projeto foi desenvolvido sobre uma metodologia *Scrumban* (*Scrum* + *Kanban*). Esta, por sua vez, consistiu em sprints mensais, onde foram definidas tarefas por secções de forma a agilizar o processo de implementação e comunicação [26].

Deste modo, foi iniciado um estudo aprofundado sobre a forma como iríamos abordar o projeto, desde a descrição do serviço, requerimentos funcionais e aspetos a implementar.

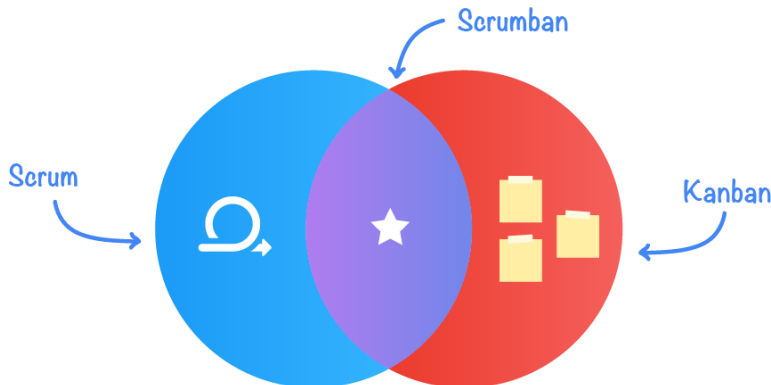


Figura 13 - Metodologia utilizada

Como já referido no Capítulo 1, este projeto subdividiu-se em três principais fases:

- **Primeira fase:** Consistiu na implementação de um protótipo focado no desenvolvimento de uma aplicação em SAP, que permitiu fazer o *download*, determinar pré-requisitos, passos manuais e ordenar notas pela sua ordem de implementação.

- **Segunda fase:** Desenvolvimento da solução no ERP SAP, com o objetivo de calcular a estimativa de implementação das notas e enviar a mesma via e-mail;
- **Terceira fase:** Resume-se na conceção de uma aplicação transaccional em Fiori através da solução desenvolvida no SAP GUI, cujo objetivo é providenciar compatibilidade com dispositivos móveis. Além disso, uma aplicação em Fiori providencia menos custos aos clientes para gerir a aplicação em múltiplas plataformas.

6.1. Implementação do protótipo de *download* de Notas

De forma a definir a melhor estratégia para o *download* de notas, foi analisada a metodologia *standard* utilizada na transação SNOTE. Antes de se poder listar uma nota, na SNOTE é necessário fazer o seu download, seguindo os passos indicados na Figura 14.

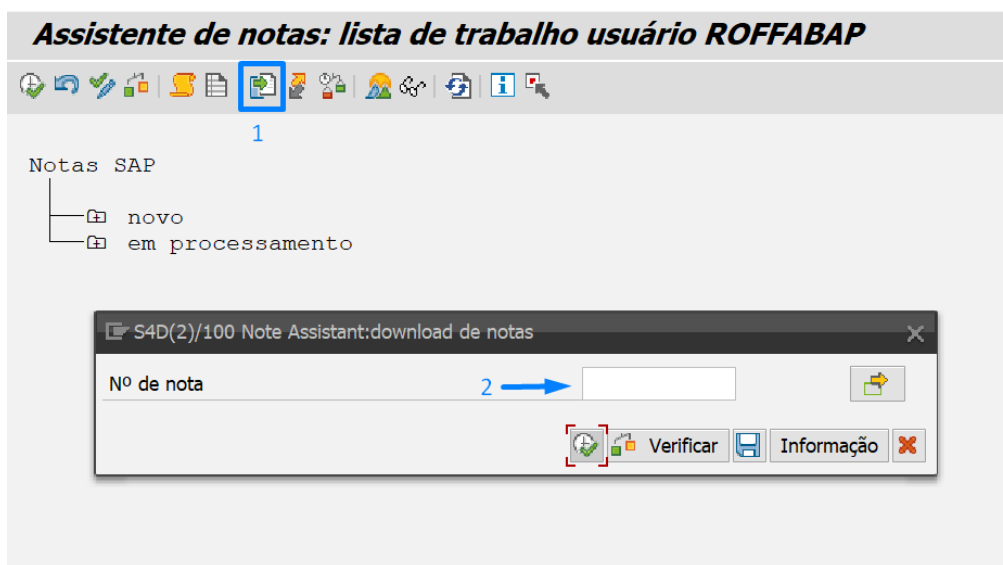


Figura 14 - Download Notas SAP

Além disso, listar as notas para a sua implementação é uma etapa realizada num ecrã diferente, o que não é prático para o utilizar final, como demonstrado na Figura 15.

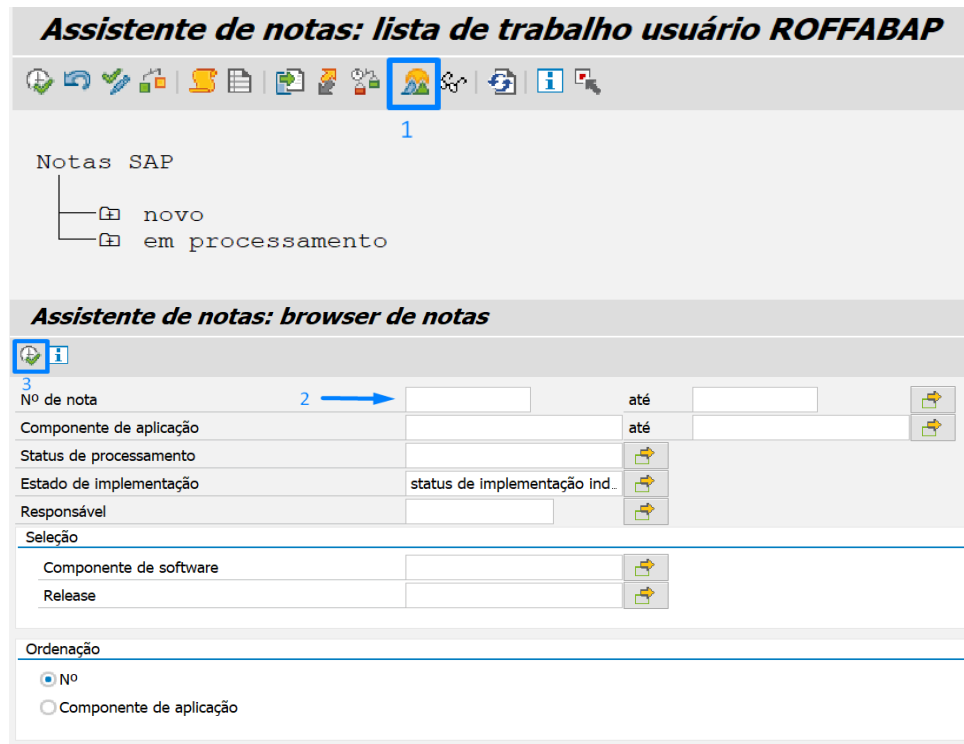


Figura 15 - Assistente de notas

Módulo Função SCWN_NOTE_DOWNLOAD

A transação SNOTE, utiliza o módulo função SCWN_NOTE_DOWNLOAD para efetuar o *download/update* das notas, mas antes, é utilizado o módulo função SCWB_NOTE_ENQUEUE, que permite bloquear a(s) nota(s) e garantir que ninguém a está a utilizar/implementar simultaneamente.

É também utilizada a classe CL_SCWN_NOTE_SAR_FILE_N, a sua estrutura está definida na Figura 16 e a sua utilização será explicada mais à frente.

Esta classe é importante uma vez que providencia métodos para o *download* e *upload* de notas.

Os métodos de *upload* são apenas utilizados quando as conexões RFC não estão válidas e os utilizadores procedem com a inserção manual das notas. Este *upload* é realizado através do ficheiro .SAR obtido no portal da SAP.

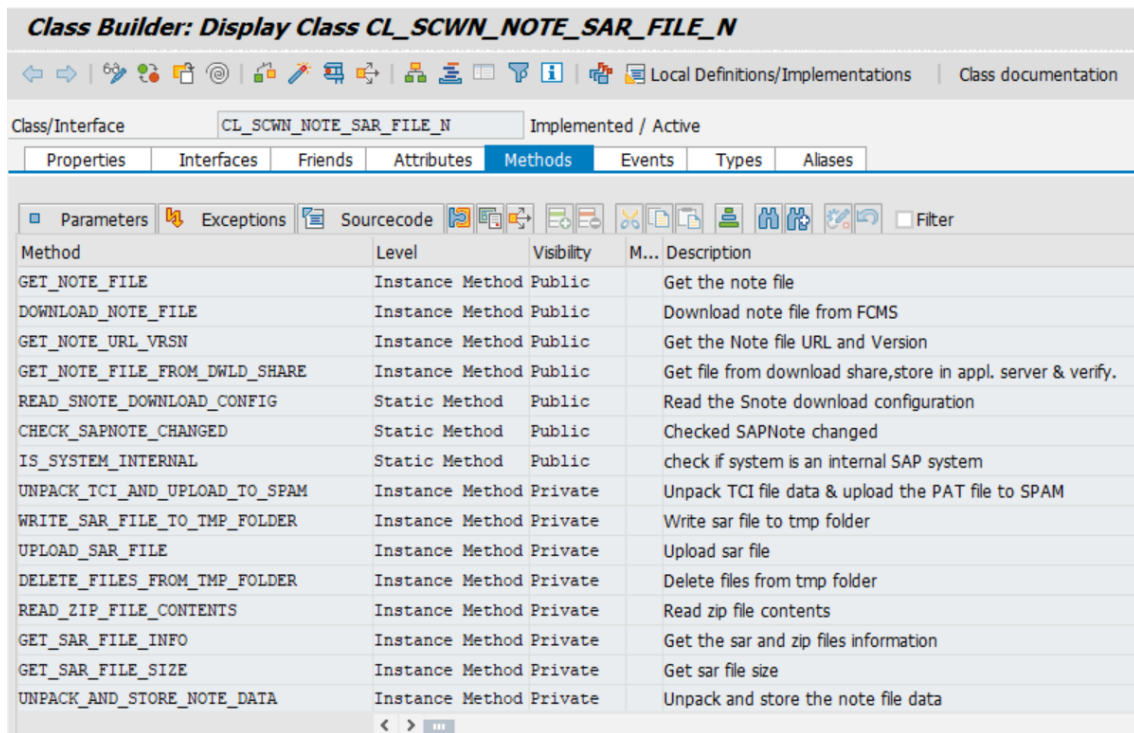


Figura 16 - Classe CL_SCWN_NOTE_SAR_FILE_N

Para o download ser feito via HTTPS/RFC, é necessário fornecer uma estrutura do tipo CWBNTKEYLG, esta estrutura contém três campos:

- NUMM – NUMC10 – N° da nota – Obrigatório
- VERSNO – NUMC4 – N° da versão da nota – Opcional
- LANGU – CHAR01 – Código de Idioma – Obrigatório

Para obter o procedimento de *download* de notas, é consultada a tabela CWB_DWNLD_PROC. Esta tabela contém o nome das conexões criadas que permitem a comunicação com um servidor externo.

A título de exemplo, uma das conexões é denominada SAP-SUPPORT_NOTE_DOWNLOAD e a sua configuração pode ser consultada na transação SM59, como demonstrado na Figura 17.



Figura 17 - SM59 - Conexões RFC

A configuração desta conexão consiste na definição do *host* e porta destino, dos dados de autenticação e da definição dos certificados, como é visível na Figura 18 e Figura 19.

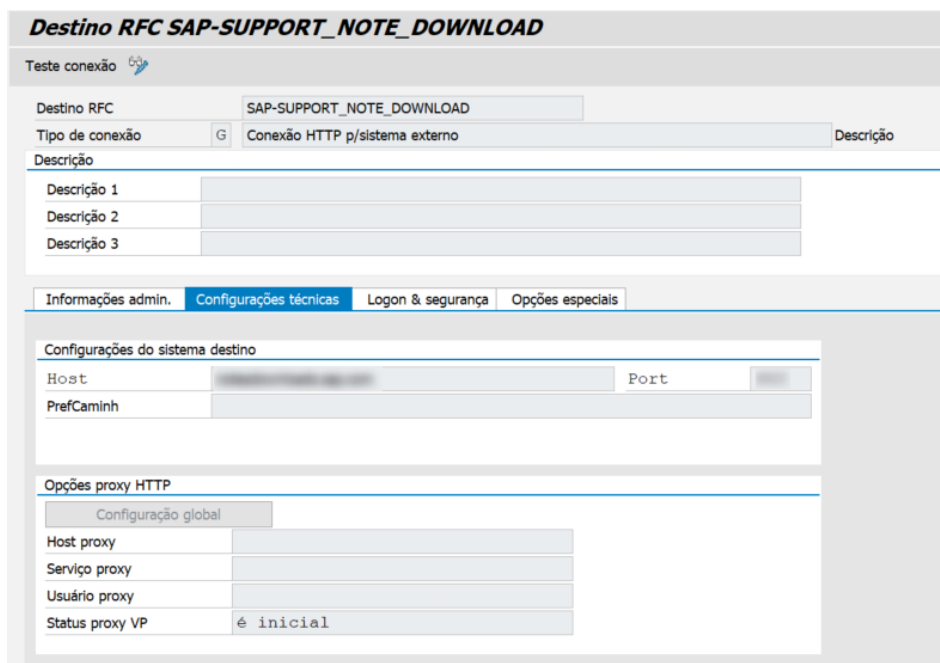


Figura 18 - Configurações técnicas da conexão RFC

Figura 19 - Configurações de Logon & Segurança

Com as conexões configuradas, é obtido o URL da nota através do método GET_NOTE_URL_VRSN.

Método GET_NOTE_URLVRSN

Este método é utilizado para obter as informações das notas (nome, descrição, versão, prefixo do url, entre outros) do SMP (SAP Mobile Platform). Essas informações são obtidas através de uma transformação XML com o número da nota e os dados de segurança da conexão, que são utilizados quando é enviado o *request*.

Neste momento, é aberta uma conexão HTTP para o serviço com o prefixo do *url* da nota e a sua chamada devolve os dados da nota em formato XSTRING.

Finalmente, e de forma resumida, a resposta é convertida num binário e é guardada no sistema.

Módulo Função SCWB_DOWNLOAD_OUTDATED_NOTES

Após uma vasta pesquisa, foi identificado que, em sistemas mais recentes é possível fazer o *download* das notas que não estejam atualizadas, através da utilização do módulo função SCWB_DOWNLOAD_OUTDATED_NOTES.

Resumidamente, este módulo função verifica se existe uma versão mais recente disponível e só faz *download* das notas se necessário. Esta validação é feita comparando a versão da nota no sistema local com a versão mais recente da nota. Para obter a versão local, é utilizado um *perform* denominado GET_NEWEST_LOCAL_VERSION.

Após obter a versão local, a mesma é comparada com a versão mais recente da nota, através da utilização do método CHECK_SAPNOTE_CHANGED da classe CL_SCWN_NOTE_SAR_FILE_N.

Este foi o módulo função de eleição para o *download* das notas uma vez que foi notável a eficiência do programa com a sua utilização.

Em notas com bastantes pré-requisitos, foram detetados ganhos significativos de *performance* na execução do protótipo. Quando a(s) nota(s) já estava(m) no sistema, se fosse utilizado o módulo função SCWN_NOTE_DOWNLOAD, a nota era descarregada de qualquer das formas, ao contrário do que acontecia com a utilização do módulo função SCWB_DOWNLOAD_OUTDATED_NOTES.

Em suma, o protótipo desenvolvido foi adaptado de forma que o *download* fosse realizado no mesmo momento em que o programa é executado.

Com este módulo de função, é aplicada uma metodologia que valida se as notas já se encontram no sistema e/ou estão atualizadas, evitando um *workload* desnecessário.

6.2. Solução no SAP GUI

A solução desenvolvida no SAP GUI foi estruturada da seguinte forma:

Tabela 2 - Estrutura da solução

<i>Include</i>	Descrição
ZSNOTEN_RCB_DAT	Declarações globais
ZSNOTEN_RCB_SCR	Ecrã de seleção
ZSNOTEN_RCB_CLA	Classe local
ZSNOTEN_RCB_F01	Rotinas auxiliares

6.2.1. *Include* ZSNOTEN_RCB_DAT

O *include* ZSNOTEN_RCB_DAT foi utilizado para declarar tipificações, tabelas, estruturas, variáveis e constantes globais à solução desenvolvida.

6.2.2. *Include* ZSNOTEN_RCB_SRC

Neste *include* foi criado o ecrã de seleção da aplicação. Ao contrário do que foi definido nas *mockups* apresentadas na secção 5.5, foi criado um ecrã de seleção dinâmico, de forma que não apareça toda a informação ao utilizador final.

Para obter o resultado pretendido, o ecrã de seleção foi dividido em três blocos. O primeiro contém um *radiobutton* que, tal como o nome indica, permite a escolha de uma opção.

Neste caso o *radiobutton* criado no primeiro bloco foi utilizado para permitir ao utilizador escolher entre a opção de determinar a estimativa e a opção de alterar estatísticas. Desta forma, foi aplicado um *loop* ao ecrã que permite esconder e apresentar os restantes blocos.

Por defeito, está ativado o botão de determinar a estimativa, mas, caso o utilizador escolha o botão de alterar estatísticas, o bloco com os elementos necessários para determinar a estimativa é escondido e aparece o bloco que tem os elementos que permitem alterar as estatísticas de cálculo de estimativas. Desta forma, ao executar a aplicação, o ecrã de seleção é o demonstrado na Figura 20.

Análise OSS Notes

Qual opção?

Determinar Estimativa
 Alterar Estatísticas

Notas a Implementar & Informações para envio de Estimativa

OSS Notes

E-mail do Destinatário até

Figura 20 - Ecrã de seleção

Caso o utilizador escolha alterar as estatísticas, o ecrã fica definido como demonstrado na Figura 21.

Análise OSS Notes

Qual opção?

Determinar Estimativa
 Alterar Estatísticas

Ajustar valores de estimativa (Horas)

Classes	0, 10
Métodos	0, 10
Programas	0, 10
Tabelas	0, 10
Outros	0, 10
Passos manuais	0, 50

Figura 21 - Ecrã de seleção - Alterar estatísticas

6.2.3. *Include* ZSNOTEN_RCB_CLA

Este *include* foi utilizado para declarar classes locais a utilizar pelo programa. Neste cenário em específico foi declara uma classe denominada **main** com os seguintes métodos:

- DOWNLOAD_OSS_NOTES – Faz o *download* das notas indicadas no ecrã de seleção (apenas se necessário);
- CHECK_OSS_NOTES – Identifica pré-requisitos das notas;
- CREATE_FINAL_TABLE – Cria a tabela final;

- CREATE_AND_DISPLAY_ALV – Configuração da ALV e apresentação do *report*;
- USER_COMAND – Identifica as ações do utilizador no *report*;
- TOP_OF_PAGE – Mapeamento do cabeçalho da ALV;
- PF_STATUS_SET – Definição do ecrã GUI que apresentará o *report*;
- TRATA_FIELDCATALOG – Mapeamento do catálogo de dados da ALV;
- TRATA_LAYOUT – Mapeamento de estilos da ALV;
- GET_MANUAL_INST_PDF – Obtém os documentos em formato .pdf com os passos manuais;
- ENVIA_EMAIL – Envia e-mail para os destinatários definidos no ecrã de seleção com a estimativa para a implementação e a listagem de notas ordenada em formato .xlsx;
- GET_ESTIMATIVA_FINAL – Obtém o valor final da estimativa;
- APRESENTA_MSGS – Apresenta as mensagens de erro.

O construtor da classe começa por definir alguns valores iniciais do ecrã de seleção. São eles os valores iniciais de estatísticas que representam 0.1h (6 min) por objeto afetado e 0.5h (30 min) caso a nota tenha passos manuais.

Além disso, é utilizada a **BAPI_USER_GET_DETAIL** para obter o endereço de e-mail do utilizador que está a executar o programa e pré-preencher o campo do destinatário da estimativa. Caso o utilizador não tenha o seu endereço de e-mail parametrizado no sistema, o campo do destinatário é pré-preenchido com um e-mail genérico.

6.2.3.1. Método DOWNLOAD_OSS_NOTAS

Este método simplesmente mapeia uma tabela com as notas inseridas pelo utilizador no ecrã de seleção e executa o módulo função

SCWB_DOWNLOAD_OUTDATE_NOTES que, tal como discutido na secção 6.1, foi o módulo função escolhido para o *download* de notas.

Além disso, foram implementadas validações para devolver mensagens ao utilizador relativamente a erros de *download* de notas, sejam eles por erro na conexão RFC ou pela inexistência das notas inseridas.

6.2.3.2. Método CHECK_OSS_NOTES

Após o *download* das notas, o número de notas válidas é armazenado numa tabela interna. Neste método, a tabela é percorrida de forma a obter dados específicos de cada nota, identificando os pré-requisitos e definindo-os consoante a sua hierarquia de implementação.

O módulo função SCWB_CINST_PRECONDITION_DATA foi utilizado para determinar os pré-requisitos de cada nota inserida, uma vez que este módulo função devolve uma tabela do tipo TT_CWB_NOTE_DISPLAY.

Este tipo de tabela tem a categoria de linha CWB_NOTE_DISPLAY, como demonstrado na Figura 22.

The screenshot shows the SAP Dictionary configuration for the table type **TT_CWB_NOTE_DISPLAY**. The title is "Dictionary: exibir tipo de tabela". The table type is set to **TT_CWB_NOTE_DISPLAY** and is marked as **ativo**. The description is "Note display for SWI Report". The configuration is shown under the "Características" tab, with the "Ctg.linha" sub-tab selected. The "Categoria da linha" is set to **CWB NOTE DISPLAY**. The "Tipo incorporado" section shows "Ctg.Dados" with "Nº de posições" set to 0 and "Casas decimais" set to 0. The "Tipo de referência" section shows "Nome tipo refer." as an empty field. The "Referência a tipo instalado" section shows "Tp.dados" as an empty field, "Compr" set to 0, and "Casas decimais" set to 0.

Figura 22 - Tipo de tabela TT_CWB_NOTE_DISPLAY

A categoria de linha nada mais é do que uma estrutura que, quando devolvida sob forma de tabela interna ao utilizar, através do módulo função SCWB_CINST_PRECONDITION_DATA, contém informações relevantes a cada nota,

como o seu número, versão, estado de implementação, último log da nota, nível de dependência, entre outros.

Dictionary: exibir estrutura

Representação de hierarquia

Estrut.: CWB_NOTE_DISPLAY ativo

Descrição breve: Note display for SWI Report

Características Componentes Entrs.possíveis/verificação Campos moeda/quantidade

Componente	Tp.atribuição tp.	Tipo componente	Categoria dados	Compr	Casas d...	SistCoords.	Descrição breve
NUMM	1 Tipo	▼ CWBNTNUMM	NUMC	10	0		0 Nº de nota
VERSNO	1 Tipo	▼ CWBCIVERS	NUMC	4	0		0 Nº de versão de uma instrução de correção
PRSTATUS	1 Tipo	▼	CHAR	256	0		0 Note Status
ALEID	1 Tipo	▼ CWBCIALEID	NUMC	10	0		0 Nº de uma instrução de correção
PARENT_ALEID	1 Tipo	▼ CWBCIALEID	NUMC	10	0		0 Nº de uma instrução de correção
MOD_DATE	1 Tipo	▼ CUS_DATE	DATS	8	0		0 Ampliação de cliente/modificação: data da última modificação
LOG	1 Tipo	▼	CHAR	256	0		0 Latest Note log
TRKORR	1 Tipo	▼ TRKORR	CHAR	20	0		0 Ordem/tarefa
CI_LEVEL	1 Tipo	▼	INT4	10	0		0 Dependent Level
INST_MODE	1 Tipo	▼	CHAR	10	0		0 Installation Mode for Corrections to Note

Figura 23 - Estrutura CWB_NOTE_DISPLAY

6.2.3.3. Método CREATE_FINAL_TABLE

Neste método é percorrida cada entrada da tabela devolvida pelo método CHECK_OSS_NOTES. Cada entrada dessa tabela contém todas as notas que são pré-requisitos das notas indicadas no ecrã de seleção, por nota.

As notas com PARENT_ALEID igual a '0000000000', são consideradas as notas principais e devem ser as últimas a serem implementadas. Foi determinado que as notas que tiverem o PARENT_ALEID igual ao ALEID de outras, significa que têm de ser implementadas primeiro.

Esta estratégia não se revelou muito útil, uma vez que várias notas pode ter o PARENT_ALEID igual ao ALEID de outras, ficava um bocado complexo decidir qual das notas deveria ser implementada primeiro. Foi então que, após análise de vários resultados, foi identificado que a nota principal (última a ser implementada), contém o CI_LEVEL (nível de dependência) igual a um.

Além disso, foi verificado que o nível de dependência funcionava de ordem inversa, ou seja, as primeiras notas a implementar eram as de nível de dependência superior, com a exceção das notas com nível de dependência zero, que deviam de ser implementadas antes das notas com nível de dependência um.

A estratégia definida passou por obter o maior número possível de dependência por nota, obtendo o registo com maior CI_LEVEL e assim, num *inner loop*, são lidas as notas por nível de dependência, de forma decrescente. Esta estratégia revelou-se ser a mais prática e permitiu criar uma listagem ordenada.

Obtenção da descrição

A descrição das notas foi obtida através da seleção à tabela CWBNTSTXT com o número da nota e a língua do sistema, adicionando uma validação para que, caso não exista na língua do sistema, selecione em inglês (uma vez que é recomendado pela SAP que se efetuem os *downloads* das notas em inglês).

Validação do estado de implementação

O estado de implementação de cada nota foi obtido através de associação direta, uma vez que já tínhamos essa informação ao executar o módulo função SCWB_CINST_PRECONDITION_DATA.

Identificação de passos manuais

Para verificar se uma nota tem passos manuais, foi utilizado o módulo função SCWB_NOTE_IMPL_STATUS que, devolve uma listagem com as instruções de implementação e objetos afetados pela nota. Foi então validado se essas instruções continham atividades manuais e foi ativada uma *flag* para identificar que existem passos manuais para a implementação da nota.

Uma das questões mais desafiantes foi a obtenção do binário do PDF com as instruções manuais, devido a problemas de *performance* da solução. Após analisar o comportamento *standard* ao visualizar notas na transação SNOTE, foi identificada a chamada do método *create* da classe CL_CWB_MAN_ACT_DISPLAY, que devolve uma instância com várias informações irrelevantes para a solução a desenvolver.

No entanto, um dos atributos relevantes era a tabela AT_MTXT_DISPLAY, que nada mais é do que uma tabela com as instruções de correção das notas e que contém um atributo denominado PDF_TABLE. Essa tabela é do tipo TBL1024, que nada mais é do que um RAW1024.

Neste momento, o problema seria devolver esse atributo, uma vez que era privado e não fazia sentido continuar a execução do código, apenas tornaria a solução menos eficiente. A solução encontrada foi modificar a classe *standard* CL_CWB_MAN_ACT_DISPLAY, para adicionar um método que devolva o atributo privado.

Finalmente, foram armazenados os passos manuais numa tabela auxiliar de forma a permitir apresentar o documento no momento de apresentação do *report* final.

Cálculo da estimativa

Para calcular a estimativa, foi aproveitada a tabela devolvida pelo módulo de função SCWB_NOTE_IMPL_STATUS, uma vez que devolvia os objetos afetados por cada instrução no formato de uma TADIR_KEY.

Uma TADIR_KEY nada mais é do que uma estrutura de três elementos. Destes três elementos, dois deles são relevantes para a identificação dos objetos afetados, são eles o TROBJECT (tipo de objeto) e TROBJ_NAME (nome do objeto). Desta forma, conseguimos contabilizar o número de objetos afetados e o seu tipo, facilitando a exclusão de objetos afetados em várias instruções na mesma nota, permitindo que a estimativa a determinar fosse o mais acertada possível, tendo em conta os parâmetros definidos pelo utilizador.

Com uma listagem única e não duplicada de objetos afetados, foi determinado o seu tipo de objeto e definido o valor adequado tendo por base o que o utilizador definiu. Os tipos de objeto que se podem definir encontram-se mapeados na Tabela 3.

Tabela 3 - Mapeamento entre objeto e tipo de objeto

Objeto	Tipo de objeto
Classe	CLAS
<i>Enhancement</i>	ENHO
Programa	PROG
Tabela	TABL

Além disso, se nenhum dos objetos afetados for do tipo mencionado na primeira tabela, o valor definido pelo utilizador em “outros” é atribuído a essa instrução. Caso a nota tenha passos manuais, é também atribuído o valor definido pelo utilizador no parâmetro “passos manuais” que, por defeito, é superior aos restantes.

6.2.3.4. Método CREATE_AND_DISPLAY_ALV

A implementação do método CREATE_AND_DISPLAY_ALV resume-se na configuração das definições técnicas da listagem ALV.

Neste método são utilizados os métodos TRATA_FIELDCATALOG (para configurar o catálogo de dados), TRATA_LAYOUT (ajustar alguns estilos do *report*) e utilizado o módulo função REUSE_ALV_GRID_DISPLAY_LVC (para apresentar a listagem ALV sob a forma de *report*).

TRATA_FIELDCATALOG

Neste método, é utilizado módulo função LVC_FIELDCATALOG_MERGE para converter uma estrutura (previamente criada na transação SE11 - Figura 24) numa tabela interna do tipo LVC_T_FCAT, que é o tipo do catálogo de dados necessário para fornecer ao módulo função REUSE_ALV_GRID_DISPLAY_LVC.

Dictionary: exibir estrutura

Representação de hierarquia Estrutura append...

Estrut. ZSNONEN_ALV ativo

Descrição breve Estrutura de apoio para apresentar a ALV no programa ZSNOTEN

Características Componentes Entrs.possíveis/verificação Campos moeda/quantidade

Tipo de compone 1 / 9

Componente	Tipo componente	Categoria dados	Compr	Casas d...	SistCoords.	Descrição breve
SNOTE	CWBNTNUMM	NUMC	10	0		0 Nº de nota
DESCR	CWBNTSTEXT	CHAR	80	0		0 Texto breve de uma nota
STATS	CHAR256	CHAR	256	0		0 Elemento de dados para callbacks WWI
PNOTE	CWBNTNUMM	NUMC	10	0		0 Nº de nota
DNLVL	INT4	INT4	10	0		0 Nº inteiro com sinal 4 bytes
MANUAL	XFELD	CHAR	1	0		0 Campo de seleção
INSTN		INT4	10	0		0 Nº objetos afetados
ICON	ICON_D	CHAR	4	0		0 Ícone em campos de texto (representação substituição, alias)
EST_TOT	PO1P DEC6_1	DEC	8	1		0

Figura 24 - Estrutura ZSNOTEN

Além disso, neste método são alterados vários textos de descrições a apresentar na listagem. O campo EST_TOT é também definido como editável e indicado para fazer

o somatório. Além disso, os campos DNLVL e MANUAL são definidos como campos técnicos (de forma a não serem apresentados no *report*). O campo ICON, foi utilizado para apresentar o PDF com os passos manuais, para isso foi definido como um *hotspot* de forma que seja ativada uma ação de utilizador quando clicado na célula.

Método TRATA_LAYOUT

Este método foi utilizado para definir os estilos da ALV. Simplesmente foi ativado o modo zebra, através da definição do campo técnico para apresentar linhas alternadas. Também foi ativado o campo técnico COL_OPT que faz uma otimização do tamanho das colunas para se ajustar ao conteúdo da listagem.

Módulo Função REUSE_ALV_GRID_DISPLAY_LVC

O módulo função REUSE_ALV_GRID_DISPLAY_LVC foi o escolhido para apresentar a listagem criada ao longo do programa.

Para apresentar a listagem, foram utilizados os seguintes parâmetros de importação:

- I_CALLBACK_PROGRAM – Nome do programa;
- I_CALLBACK_PF_STATUS_SET – Nome da rotina que define o ecrã;
- I_CALLBACK_USER_COMMAND – Nome da rotina que é chamada sempre que o utilizador final faz uma ação no *report*;
- I_CALLBACK_TOP_OF_PAGE – Nome da rotina que define o cabeçalho;
- I_STRUCTURE_NAME – Nome da estrutura definida para a listagem a apresentar;
- I_GRID_TITLE – Nome do título da listagem;
- IS_LAYOUT_LVC – Estrutura com o *layout* definido no método TRATA_LAYOUT;
- IT_FIELDCAT_LVC – Tabela com o catálogo de dados definido no método TRATA_FIELDCATALOG;
- T_OUTTAB – Tabela com o conteúdo a listar.

A utilização deste módulo função permitiu a apresentação da listagem, como demonstrado na Figura 25.

Análise OSS Notes

Enviar Estimativa ✖ Mensagens de Erro

Análise de notas OSS SAP

Data: 30.10.2023
Hora: 02:44:41
Relatório gerado por: ROFFABAP

Análise de notas OSS SAP

Nota	Descrição	Status	Nota Princ	Nº Obj.	Af	Passos Man	Estimativa
3301109	Information Retrieval Framework (IRF) Archive Data Retrieval for ILM Object MM_M	Pode ser implementado	3301109	4			0,9
3255358	RPFIEU_SAF: Interface note for SAF-T PT (61.5_PT)	Pode ser implementado	3318955	3		✖	0,3
3282308	RPFIEU_SAF: SAF-T Framework (94_F)	Pode ser implementado	3318955	10		✖	1,0
3272861	RPFIEU_SAF: SAF-T Framework (93_F)	Pode ser implementado	3318955	2		✖	0,2
3274180	RPFIEU_SAF: Interface note for SAF-T Framework (93_F)	Pode ser implementado	3318955	1		✖	0,1
3264751	RPFIEU_SAF: SAF-T Framework (92_F)	Pode ser implementado	3318955	11		✖	1,1
3231464	RPFIEU_SAF: SAF-T Framework (90_F)	Pode ser implementado	3318955	4		✖	0,4
3159260	RPFIEU_SAF: SAF-T Framework (86_F)	Pode ser implementado	3318955	3		✖	0,3
3125404	RPFIEU_SAF: SAF-T Framework (83_F)	Pode ser implementado	3318955	10		✖	1,0
3145009	RPFIEU_SAF: Interface note for SAF-T Framework (84_F)	Pode ser implementado	3318955	4		✖	0,4
3114226	RPFIEU_SAF: SAF-T Framework (82_F)	Pode ser implementado	3318955	21		✖	2,6
3103805	RPFIEU_SAF: SAF-T Framework (81_F)	Pode ser implementado	3318955	17		✖	1,7
3093957	RPFIEU_SAF: SAF-T Framework (80_F)	Pode ser implementado	3318955	6		✖	0,6
3283506	RPFIEU_SAF: SAF-T PT (64_PT)	Pode ser implementado	3318955	5		✖	0,5
3272871	RPFIEU_SAF: SAF-T PT (63_PT)	Pode ser implementado	3318955	4		✖	0,4
3294362	RPFIEU_SAF: SAF-T Framework (95_F)	Pode ser implementado	3318955	19		✖	1,9

Figura 25 - Exemplo listagem gerada

6.2.3.5. Método USER_COMMAND

Este método é utilizado para validar as ações do utilizador após a geração do *report*. No contexto deste desenvolvimento, este método validou as ações do utilizador para dois cenários, validação de *hotspots*, validação dos botões do ecrã.

Validação de *hotspots*

O primeiro cenário consistiu na validação de *hotspots* que, tal como referido na secção 6.2.3.3, validam quando o utilizador clica nas células que contém os passos manuais das notas.

Resumidamente, se não existirem passos manuais é enviada uma mensagem ao utilizador final a referir que não existem instruções manuais para a implementação dessa nota.

Caso existam passos manuais, é obtida a tabela com o binário do ficheiro e chamada a tela 0100 como *popup*, de forma a apresentar o documento PDF com as instruções manuais.

Antes de apresentar o *output* ao utilizador, a tabela em formato RAW1024 é convertida para RAW255 através da utilização do método CONVERT_1024_TO_255 da classe CL_RMPS_GENERAL_FUNCTIONS.

Esta conversão não foi aleatória, uma vez que o controlador que apresentar o documento precisa de uma tabela com os dados em formato RAW255.

Para apresentar o *output* do controlador, foi criado um container, através da utilização da de uma referência à classe CL_GUI_CUSTOM_CONTAINER. Com o container instanciado, foi também criada uma instância da classe CL_GUI_HTML_VIEWER, para criar um HTML *control proxy* de forma a gerar um *url* do documento (utilizando o método LOAD_DATA da classe CL_GUI_HTML_VIEWER).

Finalmente, o documento é apresentado utilizando o método SHOW_URL da classe CL_GUI_CUSTOM_CONTAINER utilizando o *url* obtido no método LOAD_DATA.

Validação dos botões do ecrã

De forma resumida, foi necessário adicionar dois botões ao ecrã para realizar o envio de e-mails e apresentar as mensagens de erro. Então, as validações implementadas foram para validar se o utilizador final clicou no botão de envio de estimativa ou de apresentar mensagens de erro.

Caso o utilizador clique no botão de enviar estimativa, são chamados os métodos GET_ESTIMATIVA_FINAL e ENVIA_EMAIL, caso o utilizador clique no botão mensagens de erro, é chamado o método APRESENTA_MSGS.

6.2.3.6. Método GET_ESTIMATIVA_FINAL

Este método nada mais é do que a chamada de uma instrução REDUCE à coluna EST_TOT da tabela final.

Resumidamente, a utilização de uma instrução REDUCE cria um resultado de um tipo de dados específico utilizando uma ou mais expressões de iteração [27].

Nos dias de hoje, a instrução REDUCE já não é muito utilizada, mas neste cenário, a sua utilização para fazer o somatório da coluna EST_TOT evitou a criação de um *loop* desnecessário.

6.2.3.7. Método ENVIA_EMAIL

O envio de e-mails foi realizado através da utilização das classes *standard* CL_BCS, CL_CAM_ADDRESS_BCS e CL_DOCUMENT_BCS.

Para iniciar o pedido de comunicação SAP, foi utilizado o método CREATE_PERSISTENT da classe CL_BCS, providenciando uma referência local da própria classe.

Com a classe instanciada, procedemos com a utilização do método SET_SEND_IMMEDIATELY para definir o processo de comunicação como de saída automática.

Outro dos métodos utilizados foi o método SET_MESSAGE_SUBJECT, definindo o assunto do pedido de comunicação SAP.

Para definir o e-mail do remetente, foi utilizado o método CREATE_INTERNET_ADDRESS da classe CL_CAM_ADDRESS_BCS, que devolve uma referência ao endereço criado. Com esta referência, é utilizado o método SET_SENDER da classe CL_BCS para definir o remetente do e-mail.

No que diz respeito aos destinatários da estimativa, o processo é identico ao de definição do remetente. É utilizado o método CREATE_INTERNET_ADDRESS da classe CL_CAM_ADDRESS_BCS para os endereços definidos no ecrã de seleção e é utilizado o método ADD_RECIPIENT para definir todos os destinatários.

O passo seguinte consistiu na criação de um documento com a mensagem a enviar no e-mail. A criação do documento nada mais foi do que a utilização do método CREATE_DOCUMENT da classe CL_DOCUMENT_BCS indicando três parâmetros. São eles o tipo do documento (RAW), o assunto e o conteúdo do e-mail.

A parte final consistiu na criação do anexo em formato .xlsx. Para isso, foi criada uma listagem nova mais simples, utilizando a classe CL_SALV_TABLE.

Foi escolhida esta listagem uma vez que a mesma permite-nos converter o *output* em xml, devolvendo uma variável em formato XSTRING.

Essa variável foi bastante útil uma vez que, com a utilização do módulo função SCMS_XSTRING_TO_BINARY, permitiu converter o *output* em binário.

Com este binário, já foi possível criar o anexo através da utilização do método ADD_ATTACHMENT da classe CL_DOCUMENT_BCS.

Finalmente, para colocar o documento com a mensagem na caixa de saída, foi utilizado o método SEND da classe CL_BCS.

Na Figura 26, é apresentado um exemplo de um e-mail enviado com a estimativa total prevista para a implementação das notas.

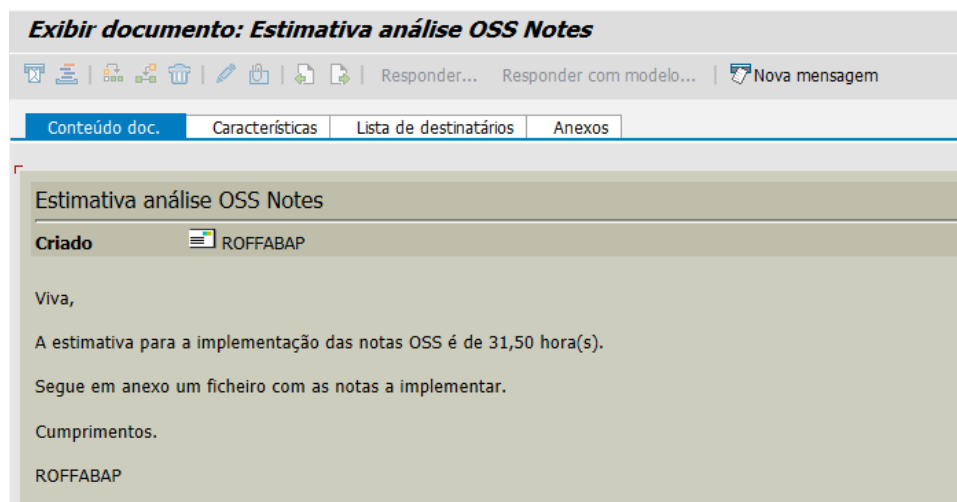


Figura 26 - Exemplo de e-mail enviado

Além disso, na Figura 27 é apresentado um exemplo de um anexo enviado por e-mail. Este anexo contém a listagem ordenada das notas, indicando a sua descrição, estado implementação, número da nota principal, a existência de passos manuais, o número de objetos afetados e o tempo esperado de implementação por nota.

	A	B	C	D	E	F	G	H
	Nº de nota	Descrição	Status	Nº de nota	Nota Principal	Passos Manuais ?	Nº Objetos Afetados	Estimativa Total
1	3301109	Information Retrieval Framework (IRF) Archive Data Retrieval for ILM Object MM_M	Pode ser implementado	3301109	0	X	4	0,9
2	3255358	RPFIEU_SAF-T: Interface note for SAF-T PT (61_5_PT)	Pode ser implementado	3318955	4		3	0,3
3	3282308	RPFIEU_SAF-T: SAF-T Framework (84_F)	Pode ser implementado	3318955	4		10	1,0
4	3272861	RPFIEU_SAF-T: SAF-T Framework (83_F)	Pode ser implementado	3318955	4		2	0,2
5	3274180	RPFIEU_SAF-T: Interface note for SAF-T Framework (93_F)	Pode ser implementado	3318955	4		1	0,1
6	3264751	RPFIEU_SAF-T: SAF-T Framework (92_F)	Pode ser implementado	3318955	4		11	1,1
7	3231464	RPFIEU_SAF-T: SAF-T Framework (90_F)	Pode ser implementado	3318955	4		4	0,4
8	3159260	RPFIEU_SAF-T: SAF-T Framework (86_F)	Pode ser implementado	3318955	4		3	0,3
9	3125404	RPFIEU_SAF-T: SAF-T Framework (83_F)	Pode ser implementado	3318955	4		10	1,0
10	3145009	RPFIEU_SAF-T: Interface note for SAF-T Framework (84_F)	Pode ser implementado	3318955	4		4	0,4
11	3114226	RPFIEU_SAF-T: SAF-T Framework (82_F)	Pode ser implementado	3318955	4	X	21	2,6
12	3103805	RPFIEU_SAF-T: SAF-T Framework (81_F)	Pode ser implementado	3318955	4		17	1,7
13	3093957	RPFIEU_SAF-T: SAF-T Framework (80_F)	Pode ser implementado	3318955	4		6	0,6
14	3283506	RPFIEU_SAF-T: SAF-T PT (64_PT)	Pode ser implementado	3318955	3		5	0,5
15	3272871	RPFIEU_SAF-T: SAF-T PT (63_PT)	Pode ser implementado	3318955	3		4	0,4
16	3294362	RPFIEU_SAF-T: SAF-T Framework (95_F)	Pode ser implementado	3318955	3		19	1,9
17	3248684	RPFIEU_SAF-T: SAF-T Framework (91_F)	Pode ser implementado	3318955	3		3	0,3
18	3218997	RPFIEU_SAF-T: SAF-T Framework (89_F)	Pode ser implementado	3318955	3		7	0,7

Figura 27 - Exemplo de listagem em anexo no e-mail

6.2.3.8. Método APRESENTA_MSG

Este método é utilizado quando o utilizador clica no botão de apresentar mensagens de erro definido no ecrã de *output*.

O que este botão faz é apresentar mensagens de erro coletadas ao longo da execução do programa que não afetem a sua correta execução.

Por exemplo, caso o utilizador indique várias notas para fazer *download* e uma delas não exista é adicionada uma mensagem relativa ao erro.

Outra das mensagens é caso a nota indicada não seja possível de implementar ou seja apenas informativa. Essas notas não irão aparecer na listagem de notas a implementar e podem gerar confusão ao utilizador final.

Assim, caso o utilizador pretenda, pode entender o motivo de certas notas não aparecerem no *output* final clicando no botão de mensagens de erro.

6.2.4. *Include* ZSNOTEN_RCB_F01

O *include* ZSNOTEN_RCB_F01 foi utilizado para criar rotinas auxiliares à solução desenvolvida.

Neste caso, uma vez que a ALV utilizada (REUSE_ALV_GRID_DISPLAY_LVC) apenas permite a execução de rotinas e não métodos de classe, foram criadas as rotinas USER_COMMAND, TOP_OF_PAGE e PF_STATUS_SET apenas e só para chamarem os métodos correspondentes, como explicado na secção 6.2.3.4.

6.3. Aplicação Transacional em Fiori

Após a implementação da aplicação no SAP GUI, foi desenvolvida uma aplicação transacional em Fiori.

The screenshot shows the 'Display Report Transaction' configuration screen in SAP Fiori. The title is 'Display Report Transaction'. Below the title is a navigation bar with icons for back, forward, search, and other functions. The main content area is divided into several sections:

- Transaction code:** ZSNOTEN
- Package:** [Redacted]
- Transaction text:** Análise de OSS Notes
- Start Options:**
 - Program:** [Redacted]
 - Selection screen:** 1000
 - Start with variant:** [Redacted]
 - Authorization Object:** [Redacted] [Values]
- Classification:**
 - Transaction classification:**
 - Professional User Transaction
 - Easy Web Transaction
 - Pervasive enabled
 - Service:** [Redacted]
- GUI support:**
 - SAP GUI for HTML
 - SAP GUI for Java
 - SAP GUI for Windows

Figura 28 - Transação ZSNOTEN

Para a aplicação ser reconhecível no *launchpad*, foi necessário criar uma transação para a solução previamente desenvolvida na *tcode* SE93 – a sua configuração pode ser consultada na Figura 28.

O desenvolvimento da aplicação transacional em Fiori foi dividido em três etapas:

- Etapa 1 – Criação dos *Intents*;
- Etapa 2 – Customização do *launchpad*;
- Etapa 3 - Criação e configuração de *Roles*;

Os *tiles* transacionais necessitam de um *target mapping*, desta forma é necessário criar o seu *intente-based-navigation*. O *intent* é definido por um objeto semântico e uma ação: <Objeto semântico>-<Ação>. Um objeto semântico pode ser definido como um uma entidade de negócio como um cliente, pedido de venda ou um produto [28].

6.3.1. Criação do *Intent*

De forma a criar o *intent*, foi necessário aceder à transação /N/UI2/SEMOBJ_SA para definir o objeto semântico.

No *landscape* atual, já se encontrava definido o objeto semântico ABAPSystem no sistema, que foi reutilizado, mas, caso fosse necessário criar um objeto semântico, a sua definição consiste na adição de registos com o preenchimento dos campos objeto semântica, nome do objeto semântico, componente aplicacional e descrição do objeto semântico.

6.3.2. Customização do *launchpad*

A customização do *launchpad* é repartida em duas secções, catálogos e grupos, como demonstrado na Figura 29.

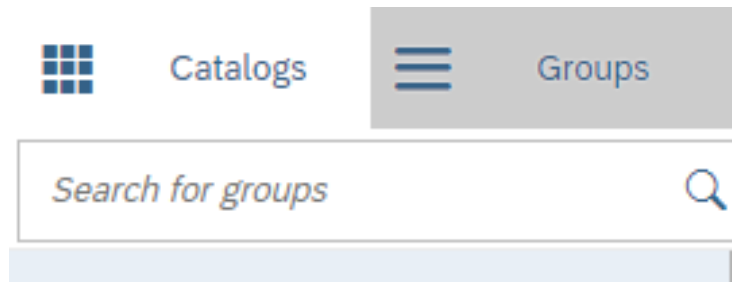


Figura 29 - Secções do painel de customização do launchpad

Para a criação dos grupos/catálogos é necessário aceder à customização do *launchpad*. É possível aceder das seguintes formas:

- Através da transação /UI2/FLPD_CONF;
- Através da transação SPRO em *SAP Solution Manager Implementation Guide* → *SAP Customizing Implementation*;

6.3.3. Catálogos

Um catálogo é um conjunto de aplicações (*tiles*) que se podem disponibilizar para funções específicas. Dependendo da sua função, os utilizadores podem navegar entre catálogos e escolher as aplicações que desejam exibir na página inicial do *launchpad* de Fiori [29].

Existem dois tipos de catálogos, os catálogos técnicos (*technical catalogs*) e os catálogos de negócio (*business catalogs*).

Um **catálogo técnico** atua como a *source of truth* para as definições da aplicação (*Tiles e Target Mapping*). Estes catálogos não devem ser adicionados a um *role*, a menos que seja para fazer uma validação rápida.

Os **catálogos de negócio** garantem o acesso do utilizador à aplicação. Estes catálogos são adicionados a *roles* (criados na transação PFCG), que permitem disponibilizar a um utilizador ou a um grupo de utilizadores específicos.

Resumindo, sempre que for necessário criar um catálogo que deve ser acessado através de *N roles*, o mais indicado é criar os *tiles* num catálogo técnico, para depois reutilizar esse catálogo em diversos catálogos de negócio.

Criação do Catálogo

No contexto do que foi referido anteriormente, foram criados dois catálogos, um catálogo técnico e um catálogo de negócio.

No catálogo técnico foram definidos os *tiles* e o catálogo de negócio foi utilizado para atribuir os *roles* para permitir o acesso à aplicação.

Após aceder ao *launchpad* de customização, basta clicar no mais que se situa na lista de seleção de catálogos, como apresentado na Figura 30.

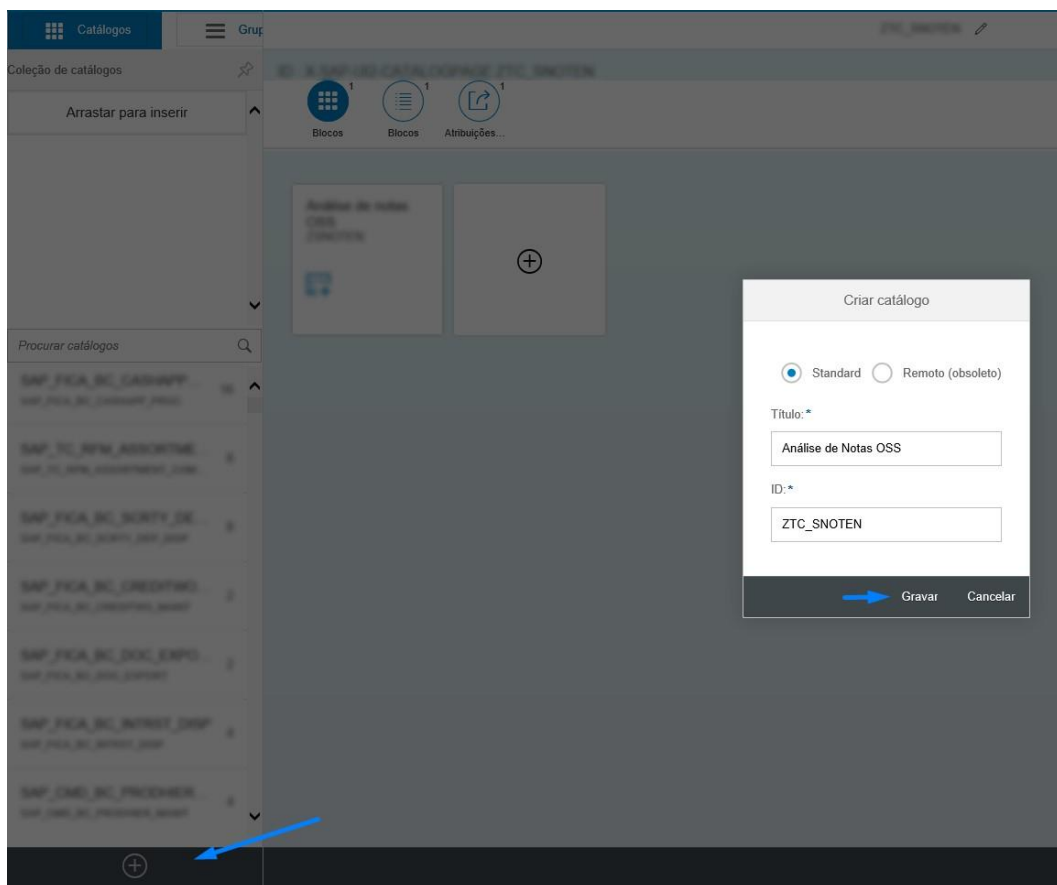


Figura 30 - Criação do Catálogo

Ambos os catálogos foram criados seguindo a seguinte nomenclatura:

Catálogo técnico – ZTC_<Transação em SAP>

Catálogo de negócio – ZBC_<Transação em SAP>

Criação do *Tile*

O *tile* tanto pode ser criado para um catálogo técnico como para um catálogo de negócio, o que define onde será criado é a finalidade do mesmo.

Tendo em conta a reutilização da criação do *tile* e do seu *target mapping*, foi apenas criado no catálogo técnico e depois, criado por referência no catálogo de negócio, herdando assim a configuração do *tile* + *target mapping*.

O *tile* foi criado clicando no bloco com o símbolo mais, como apresentado na Figura 31.



Figura 31 - Criação do *Tile*

Após indicarmos a intenção de criar um *tile*, deparamo-nos com a escolha do tipo do *tile*. Para criar um *tile* transacional, escolhemos o tipo “App Launcher – Estático”, como demonstrado na Figura 32.

Figura 32 - Tipo do *Tile*

O passo seguinte consiste na configuração do *tile*, a sua configuração está presente na Figura 33 e a explicação dos campos preenchidos na Tabela 4.

Figura 33 - Configuração do *Tile*

Tabela 4 - Explicação da configuração do tile

Título	Título exibido no <i>tile</i>
Subtítulo	Subtítulo exibido por baixo do título
Icon	Ícone do tile (podemos encontrar em SAP Icon Explorer)
Utilizar navegação de objetos semântica	Indicamos que pretendemos utilizar navegação por objeto semântico uma vez que vamos utilizar <i>target mapping</i>
Objeto semântico	Objeto semântico criado previamente na /N/UI2/SEMOBJ_SA
Ação	É preferível deixar algo alusivo à aplicação (no caso foi o nome da transação criada em SAP)
Parâmetro	sap-ui-tech-hint=gui , apenas um indicativo da tecnologia que será usada pela aplicação que está associada ao tile

Criação do *target mapping*

Após a criação do *tile*, foi criado o *target mapping*. O *target mapping* nada mais é do que *tiles* auxiliares utilizados para configurar a navegação baseada em *intents*.

O processo de criação está explícito na Figura 34 e a sua configuração está presente na Figura 35.

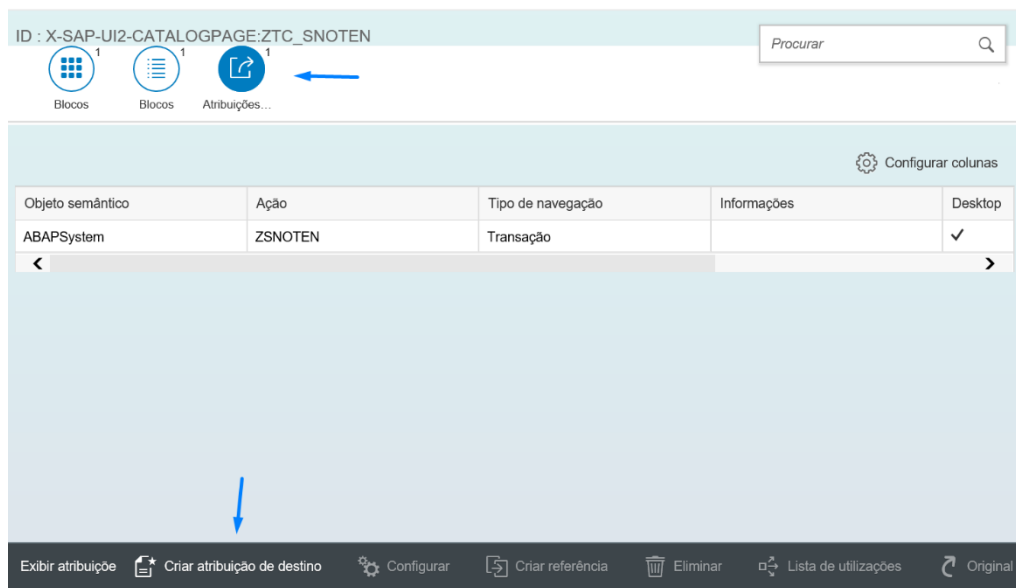


Figura 34 - Criação do *target mapping*



Figura 35 - Configuração do *target mapping*

Na Tabela 5, é apresentada a explicação da configuração do *target mapping*.

Tabela 5 - Explicação da configuração do *target mapping*

Objeto semântico	Objeto semântico indicado na criação do <i>tile</i>
Ação	Ação indicada na criação do <i>tile</i>
Tipo de aplicação	Transacional
Título	Título indicado na criação do <i>tile</i>
Transação	ZSNOTEN – Transação criada para a aplicação desenvolvida
Alias do sistema	Alias do sistema a ser utilizado

Criação da referência do *tile* + *target mapping*

Com ambos os catálogos criados, foi criada a referência do *tile* + *target mapping* original (do catálogo técnico) para o catálogo de negócio, desta forma reutilizamos o *tile* criado no catálogo original.

Para criar a referência para o catálogo de negócio, foi selecionado o *target mapping* criado anteriormente e clicado no botão de criar referência, selecionando o catálogo de negócio como demonstrado na Figura 36.

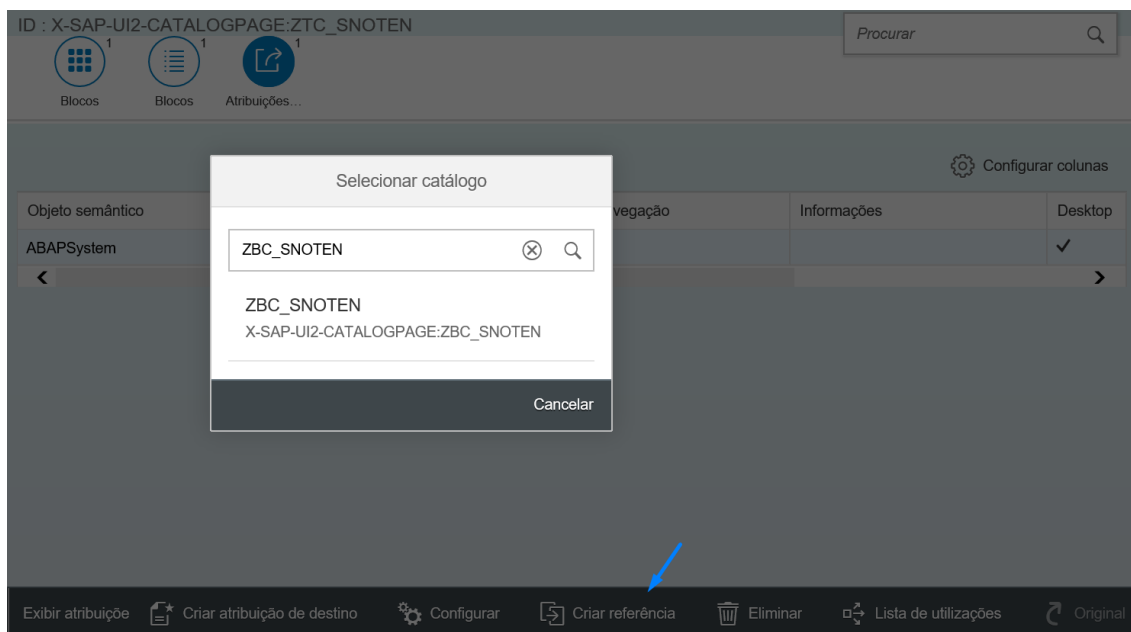


Figura 36 - Criação do *tile* + *target mapping* por referência

Criação do Grupo

Um grupo é um subconjunto de aplicações de um ou mais catálogos. Os *tiles* que são exibidos na página inicial do utilizador depende dos grupos que lhe são atribuídos. O utilizador também pode personalizar a sua página inicial, adicionando ou removendo aplicações para grupos pré-entregues ou autodefinidos.

Para criar o grupo, bastou clicar no mais que se situa na lista de seleção de grupo:

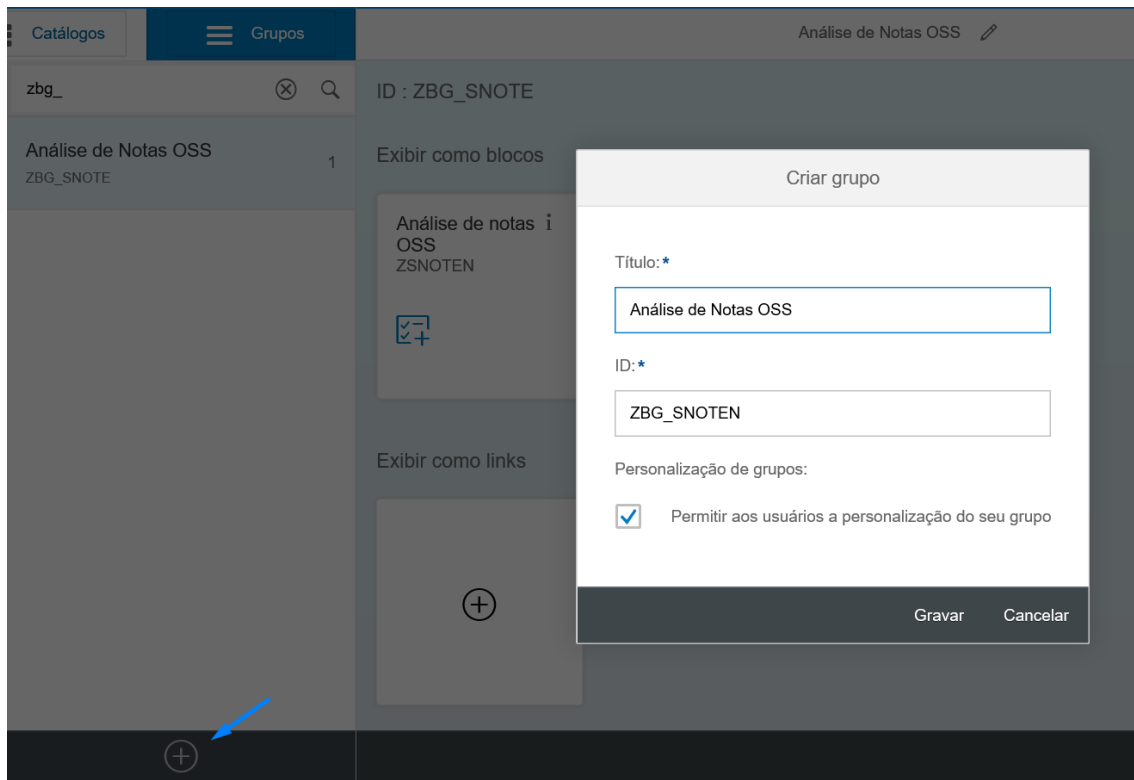


Figura 37 - Criação do grupo

O grupo foi criado seguindo a seguinte nomenclatura: ZBG_<Transação em SAP>. O passo final consistiu em adicionar os tiles criados, procurando pelo nome do catálogo de negócio.

Criação dos Roles

A criação dos *roles* tem de ser feita na máquina de *frontend*, no sistema SAP GUI, através da transação PFCG. A nomenclatura utilizada foi a seguinte: ZBR_<Transação em SAP>.

Para incluirmos o catálogo e o grupo, foi acedida à aba *menu* e selecionar as seguintes opções:

- SAP Fiori Launchpad => Catálogo de *launchpad*
- SAP Fiori Launchpad => Grupo de *launchpad*

Uma vez que é pretendido que os utilizadores tenham acesso às aplicações do catálogo e ao grupo, foi necessário adicionar ambos ao *role* (Figura 38).

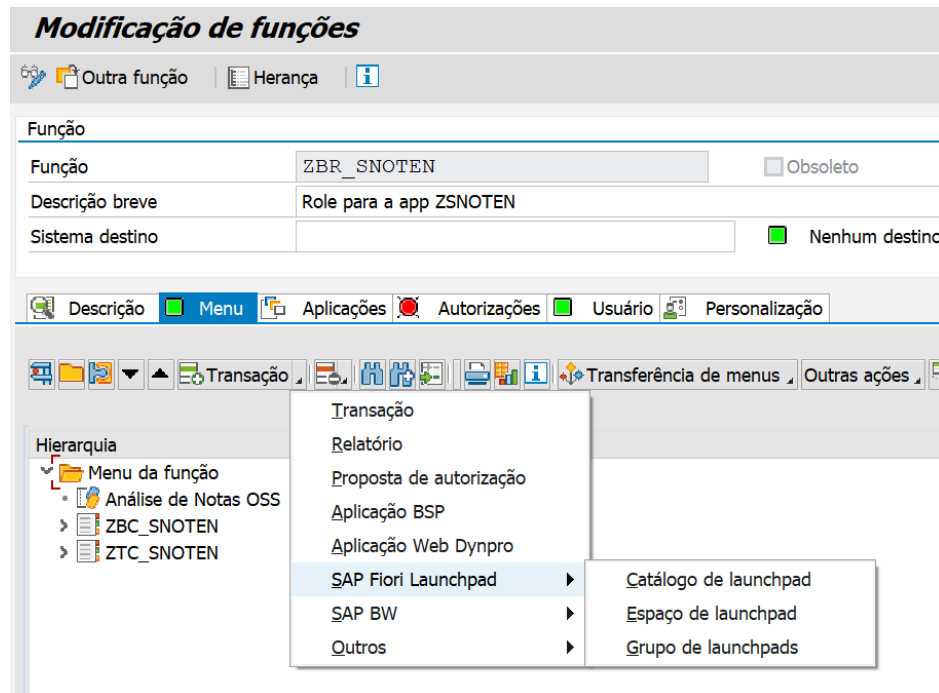


Figura 38 - Configuração do *role*

Se apenas fornecêssemos acesso ao catálogo, o utilizador teria acesso ao *tile* e poderia incluir a aplicação num grupo personalizável.

No final, foram indicados os utilizadores que terão acesso à *role*, através do mapeamento dos mesmos na aba *user*. Desta forma, todos os utilizadores com acesso a este *role*, ficaram com acesso à aplicação no *launchpad* de Fiori, como demonstrado na Figura 39.

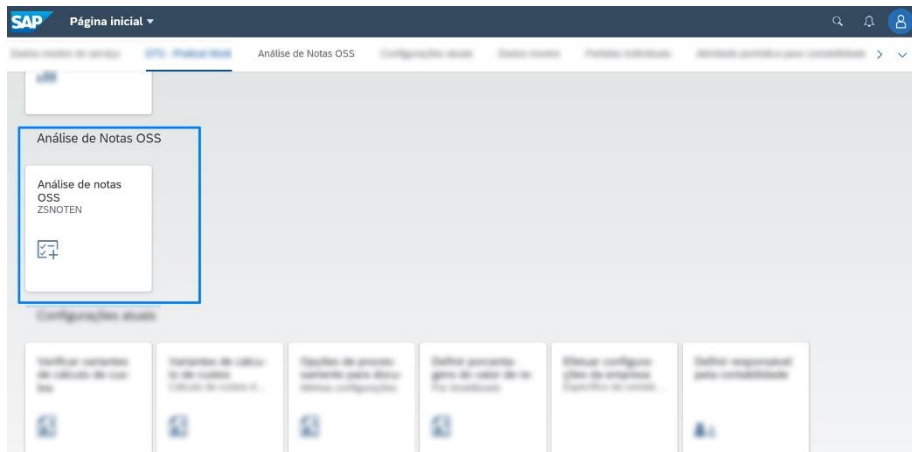


Figura 39 - Tile de acesso à aplicação transaccional

Na Figura 40, é apresentado o ecrã de seleção da aplicação transaccional que funciona como tal como explicado na secção 6.2.

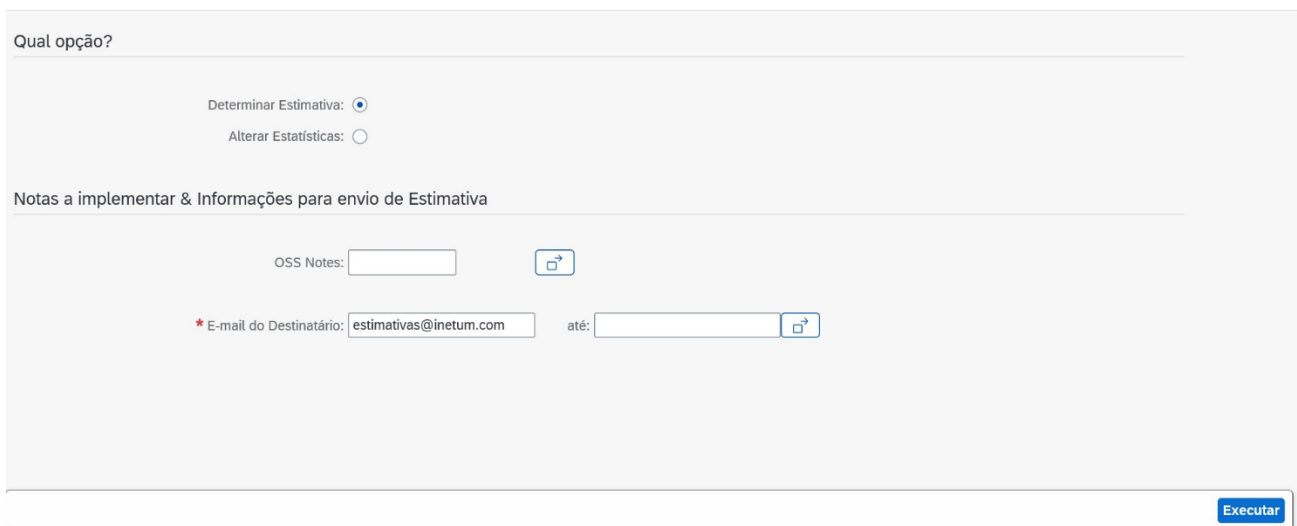


Figura 40 - Ecrã de seleção na aplicação transaccional em Fiori

Foi realizada uma execução teste para apresentar o *report* gerado. O seu *output* está apresentado na Figura 41.

Renovar [ícones] Enviar Estimativa Mensagens de Erro Mais

finalizar

Análise de notas OSS SAP

Data: 27.10.2023
 Hora: 17:32:31
 Relatório gerado por: ROFFABAP

Análise de notas OSS SAP

<input type="checkbox"/>	Nº de nota	Descrição	Status	Nota Princ	Nº Obj. Af	Passos Man	Estimativa
<input checked="" type="checkbox"/>	3301109	Information Retrieval Framework (IRF) Archive Data Retrieval	Pode ser implementado	3301109	4	[ícone]	0,9
<input type="checkbox"/>	3255358	RPFIEU_SAF: Interface note for SAF-T PT (61.5_PT)	Pode ser implementado	3318955	3	[ícone]	0,3
<input type="checkbox"/>	3282308	RPFIEU_SAF: SAF-T Framework (94_F)	Pode ser implementado	3318955	10	[ícone]	1,0
<input type="checkbox"/>	3272861	RPFIEU_SAF: SAF-T Framework (93_F)	Pode ser implementado	3318955	2	[ícone]	0,2
<input type="checkbox"/>	3274180	RPFIEU_SAF: Interface note for SAF-T Framework (93_F)	Pode ser implementado	3318955	1	[ícone]	0,1
<input type="checkbox"/>	3264751	RPFIEU_SAF: SAF-T Framework (92_F)	Pode ser implementado	3318955	11	[ícone]	1,1
<input type="checkbox"/>	3231464	RPFIEU_SAF: SAF-T Framework (90_F)	Pode ser implementado	3318955	4	[ícone]	0,4
<input type="checkbox"/>	3159260	RPFIEU_SAF: SAF-T Framework (86_F)	Pode ser implementado	3318955	3	[ícone]	0,3

[ícone] Não existiu nenhum problema ao fazer download de notas. Gravar Cancelar

Figura 41 - Output da execução teste

Capítulo 7 Análise e Discussão de Resultados

7.1. Introdução

Neste capítulo, são demonstrados os testes realizados à solução desenvolvida, bem como uma sugestão ao trabalho futuro para perfeição o projeto implementado. Dito isto, o capítulo ficou dividido em três secções.

Na primeira, são expostos os aspetos avaliados e as respetivas metodologias utilizadas para garantir a utilidade da solução. De seguida, são apresentados os resultados obtidos e finalmente, são analisados os resultados e apresentadas propostas de trabalho futuro.

É de ressaltar que, todas as funcionalidades previstas foram implementadas com sucesso e a aplicação já se encontra a ser utilizada por alguns consultores que têm providenciado feedback contínuo da sua utilização, o que permitiu projetar o trabalho futuro indicado na secção 7.4.

7.2. Metodologia e aspetos da avaliação

De modo a garantir uma avaliação correta e precisa, a solução desenvolvida foi testada com várias notas. No entanto, foram escolhidas três notas a título de exemplo para fazer uma comparação a nível temporal e funcional.

Para a avaliação realizada, foi tido em conta o tempo de duração da análise das notas, o número de notas a implementar, as notas que foram identificadas com ou sem passos manuais e a estimativa definida.

Os testes realizados, tiveram por base dois principais pontos de vista:

- O tempo de análise da implementação das notas;
- A veracidade do número de notas a implementar.

De modo a verificar que o *report* gerado pela solução era correto, foram comparados os resultados com os resultados de uma análise manual. Desta forma, além de garantirmos que o resultado era o correto, também podemos comparar o tempo entre as análises.

7.3. Resultados obtidos

Os resultados obtidos foram recolhidos através da execução da solução num sistema SAP ERP e através da análise das notas a implementar utilizando a transação SNOTE e o SAP *Support Portal*.

Na metodologia automatizada, foi contabilizado no tempo de análise a duração do *download* das notas, o tempo de análise do número de pré-requisitos que precisavam de ser implementados e validação das notas com passos manuais juntando um valor estimado.

Na análise manual, tiveram de ser analisadas mais de 300 notas que eram consideradas pré-requisitos (algumas repetidas por terem pré-requisitos comuns), tendo sido necessário fazer o *download* de cada nota e validar os passos manuais a implementar.

Na tabela Tabela 6 são apresentados os resultados obtidos durante a análise manual e na tabela Tabela 7, são apresentados os resultados obtidos utilizando a solução automatizada.

Tabela 6 - Resultados obtidos na análise manual

Nota	Tempo de análise (horas)	Nº pré-requisitos (implementáveis)	Nº notas com passos manuais	Estimativa (horas)
3318955	8	48	1	24
3314729	3	25	1	11
3301109	0.1	0	1	1.5

Tabela 7 - Resultados obtidos através da utilização da solução

Nota	Tempo de análise (horas)	Nº pré-requisitos (implementáveis)	Nº notas com passos manuais	Estimativa (horas)
3318955	0.01(6) (~1min)	48	1	30.6
3155420	0.005 (~20s)	25	1	14.6
3301109	0.0008(3) (~3s)	0	1	0.9

O cálculo da estimativa através da solução notou-se ligeiramente diferente ao definido quando analisado manualmente.

O motivo da diferença advém dos parâmetros definidos no ecrã de seleção, que foram 10 minutos por cada objeto afetado e 30 minutos por notas com passos manuais.

Apesar do valor determinado, a estimativa pode e deve ser ajustada após a execução da análise das notas. Até porque nem todas as notas devem seguir o padrão definido nos parâmetros de implementação.

7.4. Trabalho futuro

Como trabalho futuro, proponho a implementação de *features* que tornem a solução desenvolvida mais completa e agradável ao utilizador final, nomeadamente:

1. Implementar um duplo clique nas notas e visualizar os objetos que serão afetados durante a implementação das mesmas (o programa já contém uma tabela interna com essa informação);
2. Ajustar o estilo do ficheiro *excel* devolvido ao utilizador, uma vez que a listagem enviada é simples;
3. Permitir a implementação das notas na própria transação, evitando que o utilizador final tenha de executar a transação SNOTE;
4. Adotar diferentes estratégias para o cálculo das estimativas (por exemplo fornecer uma *tab* ao utilizador que sugere tempo de implementação para notas às quais o programa já analisou);
5. Anexar no envio do e-mail, os documentos PDF das etapas manuais.

Capítulo 8 Conclusões

A título de conclusão, a solução desenvolvida teve como principal objetivo, não só auxiliar consultores SAP a analisar as notas OSS que deve implementar como também ajudar qualquer programador que se depare com alguma anomalia *standard* e necessite também de analisar/implementar notas.

Ao longo do desenvolvimento, surgiram problemas relacionados com a eficiência da solução uma vez que, quando indicado ao programa para analisar múltiplas notas, a performance ia-se degradando cada vez mais. Com a utilização do módulo função SCWB_DOWNLOAD_OUTDATED_NOTES, que só fazia o *download* das notas que ainda não estavam no sistema e/ou estavam desatualizadas, foi notado um melhoramento significativo na *performance*. Além disso, foram adotadas validações para não repetir a análise de notas que já tinham sido consideradas e também se notaram melhorias significativas.

Surgiram também problemas com a recuperação dos binários com o PDF que continha os passos manuais, no entanto, com um ajuste à classe, foi implementado um método *get* para devolver o atributo privado.

Os objetivos propostos para este projeto consistiram no desenvolvimento da arquitetura da aplicação, através da modelação e da criação das *mockups*, a implementação de uma solução no SAP GUI e no desenvolvimento de uma aplicação transacional em Fiori para permitir a utilização da aplicação nos *launchpads*, seja ela via *web* ou *mobile*.

Desta forma, considero que todos os objetivos propostos foram cumpridos. Resta salientar que, como trabalho proposto, sugiro a implementação de algumas *features* de forma a tornar a solução mais completa, como a implementação de um duplo clique nas

notas de forma a permitir ao utilizador final visualizar os objetos afetados, ajustar o estilo do ficheiro *excel* devolvido em anexo no e-mail da estimativa, adicionar uma coluna extra com um *hotspot*, que permita a implementação das notas na própria aplicação, evitando que o utilizador final tenha de executar a transação SNOTE, adotar diferentes estratégias para o cálculo das estimativas (por exemplo fornecer uma *tab* ao utilizador que sugere tempo de implementação para notas às quais o programa já analisou). Outra das sugestões passaria por incluir nos anexos do e-mail da estimativa, os documentos PDF com as etapas manuais.

Bibliografia

- [1] M. Rouse, “OSS Notes,” Abril 2013. [Online]. Available: <https://www.techopedia.com/definition/28734/oss-notes-sap>. [Acedido em Outubro 2023].
- [2] K. N. Nampoothiry, “SAP OSS Notes: An Overview,” Setembro 2013. [Online]. Available: <https://blogs.sap.com/2013/09/18/sap-oss-notes-an-overview/>. [Acedido em Setembro 2023].
- [3] P. Alat, “SAP Community,” Maio 2014. [Online]. Available: <https://blogs.sap.com/2014/05/15/tcode-snote-step-by-step-guide/>. [Acedido em Outubro 2023].
- [4] S. Holitschke, “How to Use SAP's Support Portal and Applications to Get the Most Out of Your SAP Products,” Setembro 2023. [Online]. Available: <https://www.linkedin.com/pulse/how-use-saps-support-portal-applications-get-most-out-holitschke/>. [Acedido em Setembro 2023].
- [5] K. C. Laudon e J. P. Laudon, *Management Information Systems: Managing the Digital Firm*, New York: Prentice Hall, 2015.
- [6] S. Subramoniam, M. Tounsi e K. V. Krishnankutty, “The role of BPR in the implementation of ERP systems,” *Business Process Management Journal* 15.5, pp. 653-668, 2009.
- [7] R. Gabryelczyk e N. Roztockki, “Effects of BPM on ERP Adoption in the Public Sector,” *23^o Americas Conference on Information Systems*, 2017.
- [8] S. C. Santos, C. Santana e J. M. C. Elhimas, “Critical success factors for ERP implementation in sector public: An analysis based on literature and a real case.,” *26^o European Conference on Information Systems*, 2018.
- [9] P. C. Solutions, “2018 Report,” 2018. [Online]. Available: <https://cdn2.hubspot.net/hubfs/2184246/2018%20ERP%20Report.pdf>. [Acedido em Setembro 2023].
- [10] F. Mahmood, A. Z. Khan e R. H. Bokhari, “ERP issues and challenges: a research synthesis,” *Kybernetes* 49.3, pp. 629-659, 2020.
- [11] V. Kumar, B. Maheshwari e U. Kumar, “ERP systems implementation: Best practices in Canadian government organizations.,” *Government Information Quarterly* 19.2, pp. 147-172, 2002.
- [12] SAP e G. Cloud, “SAP and Google Cloud Expand Partnership to Build the Future of Open Data and AI for Enterprises,” Maio 2023. [Online]. Available: <https://news.sap.com/2023/05/sap-google-cloud-expanded-partnership-open-data-ai-enterprises/>. [Acedido em Setembro 2023].

- [13] SAP e IBM, “SAP to Embed IBM Watson Artificial Intelligence into SAP Solutions,” Maio 2023. [Online]. Available: <https://news.sap.com/2023/05/ibm-watson-artificial-intelligence-in-sap-solutions/>. [Acedido em Setembro 2023].
- [14] SAP e Microsoft, “SAP and Microsoft Collaborate on Joint Generative AI Offerings to Help Customers Address the Talent Gap,” Maio 2023. [Online]. Available: <https://news.sap.com/2023/05/sap-microsoft-joint-generative-ai-offerings-talent-gap/>. [Acedido em Setembro 2023].
- [15] SAP, “SAP Portal,” [Online]. Available: <https://support.sap.com/en/index.html>. [Acedido em Setembro 2023].
- [16] D. P. Cidre, “Desenvolvimento de soluções SAP,” 2019.
- [17] F. K. Febrianti, M. Saputra e W. Puspitasari, “COGS Report Customization Design for Profitability Analysis with ABAP List Viewer: Case Study of a Telecommunication Enterprise,” *1st International Conference on Information System & Information Technology (ICISIT)*, pp. 13-18, 2022.
- [18] TechTarget, “What is SAP Fiori?,” Julho 2018. [Online]. Available: <https://www.techtarget.com/searchsap/definition/SAP-Fiori>. [Acedido em Setembro 2023].
- [19] J. M. Mazzullo, *Working with SAP: Navigating and Reporting with the SAP S/4 HANA and FIORI GUIs*, 2023.
- [20] Eursap, “SAP Tips: What are the 3 different types of Fiori apps which are available?,” Agosto 2021. [Online]. Available: <https://eursap.eu/2021/08/17/sap-tip-the-three-different-types-of-fiori-apps/>. [Acedido em Setembro 2023].
- [21] S. Livingston, “What is OSS Notes? SAP SNOTE Tutorial,” Outubro 2023. [Online]. Available: <https://www.guru99.com/how-to-trace-if-an-oss-note-is-deployed-in-your-landscape.html>. [Acedido em Outubro 2023].
- [22] Blog SAP, “Configuring SAP SMTP service,” Setembro 2013. [Online]. Available: <https://blogs.sap.com/2013/09/17/configuring-sap-smtp-service/>. [Acedido em Setembro 2023].
- [23] D. Maiorca, “What Is Figma and What Is It Used For?,” Janeiro 2023. [Online]. Available: <https://www.makeuseof.com/what-is-figma-used-for/>. [Acedido em Outubro 2023].
- [24] Visual Paradigm, “Visual Paradigm Product Overview,” [Online]. Available: https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html. [Acedido em Outubro 2023].
- [25] Visual Paradigm, “Explore all the powerful features Visual Paradigm has to offer,” [Online]. Available: <https://www.visual-paradigm.com/features/>. [Acedido em Outubro 2023].

- [26] M. Sotille, “O que é Scrumban?,” 19 11 2022. [Online]. Available: <https://blog.pmtech.com.br/scrumban/>. [Acedido em 2 10 2023].
- [27] D. Santos, “Replace the LOOP for REDUCE operator,” Maio 2017. [Online]. Available: <https://blogs.sap.com/2017/05/25/replace-the-loop-for-reduce-operator/>. [Acedido em Outubro 2023].
- [28] Help SAP, “Defining Semantic Objects for Navigation,” [Online]. Available: https://help.sap.com/docs/ABAP_PLATFORM_NEW/a7b390faab1140c087b8926571e942b7/609c84bbacb04fd8a17bdb5da742815f.html?version=202009.000. [Acedido em Setembro 2023].
- [29] Help SAP, “Setup of Catalogs, Groups, and Roles in the SAP Fiori Launchpad,” [Online]. Available: https://help.sap.com/doc/fiori_bs2013/1.0%202016-01/en-US/6b/e47c525ae17154e10000000a44176d/content.htm?no_cache=true. [Acedido em Outubro 2023].

