



Automatização do Processo de Tratamento de Avisos de Pagamento usando UiPath

GONÇALO DOS SANTOS COELHO RIBEIRO SOARES

Junho de 2023

Automatização do Processo de Tratamento de Avisos de Pagamento usando *UiPath*

Gonçalo dos Santos Coelho Ribeiro Soares

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Software**

Orientador: António José Rocha de Oliveira

Supervisor: Roberto Silva

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 22 de junho de 2023

A handwritten signature in black ink, reading "Gonçalo Soares". The signature is written in a cursive, flowing style.

Dedicatória

Aos meus pais, avós, irmão, e colegas, por me obrigarem a ficar neste mestrado. A mim mesmo por não ter desistido.

Resumo

A automatização de processos manuais tornou-se cada vez mais importante no ambiente de dinâmico contexto empresarial atual. A presente dissertação centra-se na automatização de um processo manual específico na Inapa, com os objetivos de reduzir o tempo gasto no mesmo e reaproveitar os colaboradores para outras tarefas. O processo visado foi redesenhado e parcialmente automatizado usando UiPath, tendo esta automatização resultando numa abordagem significativamente mais eficiente em comparação com a execução manual tradicional. O objetivo deste estudo passou pela avaliação da eficiência da automatização em termos de tempo de execução do processo.

A metodologia de pesquisa envolveu a análise do processo manual existente e a identificação de áreas adequadas para automatização. Por meio de um redesenho cuidadoso e da utilização do UiPath, o processo manual foi significativamente automatizado. Após a implementação da solução automatizada, foram recolhidos dados sobre o tempo de execução para as abordagens manual e automatizada e avaliados dentro do âmbito operacional da empresa.

Os resultados do estudo demonstram uma melhoria notável na eficiência, com o processo automatizado a superar consistentemente a abordagem manual em cerca de 90%. A automatização não reduziu apenas o tempo necessário para o processo, mas também libertou os trabalhadores de tarefas repetitivas, permitindo o seu reaproveitamento para outras atividades de maior valor para a Inapa.

Esta pesquisa contribui para o campo da engenharia de software ao fornecer evidências empíricas dos benefícios da automatização de processos em termos de eficiência temporal. As descobertas destacam ainda o potencial da automatização para otimizar processos manuais e melhorar o desempenho organizacional geral. O estudo demonstra ainda a aplicação prática da ferramenta UiPath numa automatização bem-sucedida do processo em questão.

Palavras-chave: Automatização, *UIPATH*, Compreensão de Documentos, Processos Manuais.

Abstract

The automation of manual processes has become increasingly important in today's fast-paced business environment. This research focuses on the automation of a specific manual process within Inapa, aimed at reducing time consumption and freeing up workers for other tasks. The targeted process was redesigned and partially automated using UiPath tools, resulting in a significantly more efficient approach compared to the traditional manual execution. The objective of this study was to evaluate the effectiveness of automation in terms of time savings.

The research methodology involved analyzing the existing manual process and identifying areas suitable for automation. Through careful redesign and the utilization of UiPath tools, the manual process was automated to a significant extent. After the implementation of the automated solution, data was collected on time consumption for both the manual and automated approaches and evaluated within the company's operational environment.

The results of the study demonstrate a remarkable improvement in efficiency, with the automated process consistently outperforming the manual approach by over 90%. The automation not only reduced the time required for the process but also relieved workers from repetitive tasks, allowing them to focus on more value-added activities.

This research contributes to the field of software engineering by providing empirical evidence of the benefits of process automation in terms of time efficiency. The findings highlight the potential for automation to optimize manual processes and improve overall organizational performance. The study also sheds light on the practical application of UiPath tools in achieving successful process automation.

Keywords: Automation, UIPATH, Document Understanding, Manual Processes.

Agradecimentos

Agradeço à minha família por zelarem sempre pelos meus melhores interesses e me obrigarem a ficar neste mestrado.

Agradeço aos meus colegas Estugo Heves e Priguel Meto por terem embarcado nesta aventura apesar de todas as adversidades encontradas ao longo destes últimos 3 anos. Agradeço aos meus colegas Doze, Leandro e Lucas por terem um papel fundamental no meu bem-estar e saúde mental.

Agradeço em particular ao professor António José Rocha de Oliveira pela sua orientação, profissionalismo, paciência e disponibilidade ao longo deste projeto.

Agradeço à Xana por me inspirar, ensinar e aturar e nos bons e nos maus dias!

Agradeço a deus por me ter dado paciência, e agradeço a mim por conseguir usá-la.

Índice

1	Introdução	1
1.1	Apresentação da empresa	1
1.2	Descrição do Problema	1
1.3	Objetivos.....	2
1.4	Estrutura do Documento	3
2	Estado da Arte	5
2.1	Visão Geral.....	5
2.1.1	Análise Tecnológica Inapa	5
2.1.2	SAP ERP.....	5
2.2	Automatização de Processos	6
2.2.1	Automatização Robótica de Processos	6
2.2.2	Soluções Existentes	7
2.2.3	Análise Comparativa.....	9
2.2.4	Automatização de componentes	11
2.2.5	Soluções existentes	11
2.2.6	Análise Comparativa.....	14
3	Análise do Problema	19
3.1	Domínio do Problema	19
3.2	Análise.....	24
3.2.1	Detalhe Mecanismos de Visibilidade	24
3.2.2	Requisitos funcionais	25
3.2.3	Requisitos não funcionais	29
3.3	Análise de valor	29
3.3.1	Fuzzy Front End	30
3.3.2	New Concept Development.....	30
3.3.3	Valor	39
4	Desenho da Solução.....	41
4.1	Redesenho do Processo	41
4.2	Abordagem.....	41
4.2.1	Abordagem escolhida Processamento em 2 processos.....	42
4.2.2	Alternativa Tratamento dos Avisos de Pagamento à Medida	45
4.3	Solução de Métricas de <i>Process Summaries</i>	46
4.3.1	Vista Desenvolvimento	47
4.3.2	Vista Física.....	50
4.3.3	Alternativas de Design	51
5	Implementação da Solução	53

5.1	Automatização do Processo de Tratamento dos Avisos de Pagamento.	53
5.1.1	Inicialização.....	54
5.1.2	Recuperação de Transação	56
5.1.3	Processamento da Transação.....	60
5.1.4	Finalização	72
5.2	Solução Complementar de Métricas de <i>Process Summaries</i>	73
5.2.1	Dashboard <i>Backend</i>	76
5.2.2	Dashboard <i>Frontend</i>	81
5.2.3	Folder Monitor	84
6	Experimentação e Avaliação	89
6.1	Hipótese	89
6.2	Indicadores e fontes de informação	89
6.3	Ambiente de Controlo	90
	Cenários e casos de teste	91
6.4	Avaliação de Resultados.....	93
6.4.1	Avaliação das Amostras	95
6.4.2	Significância Estatística	96
7	Conclusões	97
7.1	Objetivos Atingidos	97
7.2	Limitações e Trabalho Futuro	98
7.3	Apreciação Final	98
8	Bibliografia.....	99

Lista de Figuras

Figura 1 - Diagrama BPMN Tratamento de Avisos de Pagamentos	19
Figura 2 -Diagrama BPMN do Subprocesso "Trata A.P"	19
Figura 3 - Diagrama BPMN Subprocesso Publicação do A.P	20
Figura 4 - Resultado da Transação VF03	21
Figura 5 - Resultado da Transação FBL5N	22
Figura 6 - Etapa 8: Validação das referências de Aviso de Pagamento presentes no SAP.....	23
Figura 7 - SD Consumo de um Process Summary.....	27
Figura 8 - SD Recolha de Métricas.....	27
Figura 9 - SD Atualização de Avisos de Pagamento.....	28
Figura 10 - Modelo NCD (Koen, et al., 2014).....	30
Figura 11 - Análise SWOT - Enquadramento estratégico da oportunidade	32
Figura 12 - Tendência de Mercado para Robot Process Automation	33
Figura 13 - Tendência Geográfica para Robot Process Automation	34
Figura 14 - Canvas Value Proposition (Osterwalder, et al., 2014).....	39
Figura 15 - Diagrama BPMN Preparo de Avisos de Pagamento (Alto Nível).....	42
Figura 16 - Diagrama BPMN Preparo de Avisos de Pagamento (Detalhe)	43
Figura 17 - Diagrama BPMN Publicação de Avisos de Pagamento (Alto Nível)	43
Figura 18 - Diagrama BPMN Publicação de Avisos de Pagamento (Detalhe).....	44
Figura 19 - Diagrama BPMN Alternativa Tratamento dos Avisos de Pagamento à Medida (Alto Nível)	45
Figura 20 - Diagrama BPMN Trata Aviso de Pagamento (Detalhe)	45
Figura 21 - Alternativa Tratamento dos Avisos de Pagamento à Medida - Publicação de Avisos de Pagamento (Detalhe)	46
Figura 22 - Diagrama de Componentes da Solução Complementar	48
Figura 23 - Diagrama de Componentes - Backend da Dashboard.....	49
Figura 24 - Diagrama de Componentes - Folder Monitor	50
Figura 25 - Diagrama de Implantação Solução Complementar.....	51
Figura 26 - Alternativa de Design Solução Dashboard	51
Figura 27 - Vista Geral UiPath Processo de Tratamento dos Avisos de Pagamento	54
Figura 28 - Inicialização	55
Figura 29 - Inicialização Aplicação SAP	55
Figura 30 - Fluxo Get Transactions (BPMN).....	56
Figura 31 - Diagrama <i>Execute Feba Transactions</i>	57
Figura 32 - UiPath Preparação Transação FEBA	58
Figura 33 - Execução Transação FEBA	59
Figura 34 - Processamento de Transação UiPath.....	61
Figura 35 - Decisão de Aplicação de Processo do Cliente	62
Figura 36 - Preparação para Processo de Cliente.....	63
Figura 37 - Processamento de Transação.....	64
Figura 38 - Subprocesso UiPath Get Account Code	64

Figura 39 - Instruções UiPath Invocação da Extração de Informação do PDF	66
Figura 40 - Subprocesso Detalhe Extração de Informação do PDF do Aviso de Pagamento	67
Figura 41 - <i>UiPath Obtain Open Invoices</i>	68
Figura 42 - Fluxo de Subprocesso <i>Calculate Differences</i>	69
Figura 43 - Subprocesso cálculo de descontos.....	70
Figura 44 - Subprocesso Aplicação de Cor ao <i>Excel</i> Aviso de Pagamento.....	72
Figura 45 - Subprocesso End	72
Figura 46 - <i>Dashboard General</i>	74
Figura 47 - <i>Dashboard Metrics</i>	75
Figura 48 - <i>Dashboard Files</i>	75
Figura 49 - <i>Dashboard Clients</i>	76
Figura 50 - Estrutura Dashboard Backend.....	77
Figura 51 - Execução de Stored Procedures via MySqlConnection.....	77
Figura 52 - Demonstração Uso ORM Entity Framework Core	78
Figura 53 - Centralização de Configurações de Injeção de Dependências.....	79
Figura 54 - Exemplo de utilização de Interfaces nos Controllers	80
Figura 55 - Exemplo de utilização de Interfaces nos Serviços	80
Figura 56 - Estrutura <i>Dashboard Frontend</i>	81
Figura 57 - Função Exemplo Transformação de Dados <i>Mapper</i>	82
Figura 58 - Invocação de uma Função do <i>Mapper</i>	83
Figura 59 - Exemplo de utilização de Imutabilidade	84
Figura 60 - Estrutura <i>Folder Monitor</i>	85
Figura 61 - Invocação do Serviço de Monitorização de Diretório	86
Figura 62 - Monitorização do Diretório Configurado	86
Figura 63 - Método de Leitura de Process Summary CSV	87
Figura 64 - Processamento de Process Summary	88
Figura 65 - Utilização de Testes Estatísticos.....	95

Lista de Tabelas

Tabela 1 – Comparação das Principais Ferramentas RPA	10
Tabela 2 - Comparação de Ferramentas de Extração de Informação não estruturada de ficheiros ...	15
Tabela 3 - Requisitos não funcionais FURPS+.....	29
Tabela 4 - Tabela Inicial TOPSIS.....	36
Tabela 5 - Matriz de Decisão Normalizada.....	36
Tabela 6 - Matriz de Decisão Normalizada com Pesos.....	37
Tabela 7 - Separação das soluções Ideal e Ideal Negativa	38
Tabela 8 - Resultados de Caso de Teste 1: Processo Único Pré-validado	94
Tabela 9 - Cálculo do índice de performance para os resultados dos testes realizados.....	94
Tabela 10 - Resultados de Teste de <i>Shapiro-Wilk</i>	96

Acrónimos e Símbolos

Lista de Acrónimos

ERP	<i>Enterprise Resource Planning</i>
RPA	<i>Robotic Process Automation</i>
BW	<i>Business Warehouse</i>
FI	<i>Financial</i>
MM	Material Management
SD	<i>Sales Distribution</i>
CO	<i>Controlling</i>
TOPSIS	<i>Technique for Order of Preference by Similarity to Ideal Solution</i>

1 Introdução

Num mundo tecnológico cada vez mais dinâmico, a produtividade dos trabalhadores e sua alocação a tarefas adequadas é fundamental para o crescimento e sustentabilidade do negócio no qual estão inseridos. Por outro lado, as estatísticas demonstram que a automatização tem sido um tópico de consideração séria por parte de muitas empresas, podendo eliminar ou reduzir a necessidade de intervenção manual por parte dos trabalhadores. A presente dissertação aborda justamente a exploração de um processo manual e a apresentação de uma solução para a automatização do mesmo. O projeto, decorrido na empresa Inapa, pressupõe o estudo de diferentes abordagens de automatização, desenho, implementação e testagem da solução delineada. Ao capítulo corrente cabe a responsabilidade de enquadrar o leitor, sendo realizada a apresentação da empresa, da temática do problema, dos principais objetivos esperados e um breve resumo sobre a estruturação do documento.

1.1 Apresentação da empresa

Fundada em 1965, a Inapa ocupa uma posição de referência no mercado europeu de distribuição de papel e detém ainda posições de destaque na distribuição de soluções de embalagem e comunicação visual. Atualmente a empresa conta com uma equipa de 1.709 colaboradores, e opera em 10 geografias distintas – Alemanha, França, Espanha, Portugal, Bélgica, Luxemburgo, Áustria, Holanda, Turquia e Angola.

Dentro do ramo de distribuição de papel e de soluções de embalagem, a Inapa oferece a possibilidade de realização de encomendas dos seus produtos mediante a sua disponibilidade (stock) em armazém. Esta gestão é efetuada numa plataforma reservada para acesso pelos colaboradores internos da Inapa, que a utilizam de modo a consultar e modificar os detalhes de encomendas de clientes, faturação e prazos de entrega das mesmas.

1.2 Descrição do Problema

Como apresentado na secção anterior, a Inapa é uma empresa focada na venda, distribuição, compra e revenda de papel a vários clientes e fornecedores. O processo de venda de papel a um cliente não é finalizado num único passo, mas apresenta antes várias fases:

- O sistema do cliente envia automaticamente para o sistema *EDI* da Inapa a encomenda por vários formatos (XML, PDF, etc.);
- É persistido o pedido de ordem nos centros de dados Inapa, e é gerado um artefacto que espelha a ordem desejada no sistema de ordens da Inapa;

- Os colaboradores internos Inapa designados para o efeito verificam a encomenda e decidem sobre a sua aceitação ou rejeição;
- A encomenda fica efetivamente confirmada e é transmitida para o ERP respetivo.

Aquando da realização da encomenda, além de ser gerada uma fatura correspondente ao valor total a liquidar ao cliente e esse valor final é ainda persistido na plataforma SAP. Quando o pagamento é efetuado, é gerado um aviso de pagamento – um comprovativo dos valores pagos, via ficheiro PDF.

Este aviso de pagamento enviado pelos clientes deve ser validado, os valores conferidos com os valores calculados na realização da encomenda, deve ser validado se foi corretamente aplicado algum dos descontos tabelados ao cliente, entre outras regras de negócio.

Após a validação, existe a necessidade de marcar no SAP a liquidação de uma certa encomenda como correta ou não. Atualmente, todo este processo de validação de um aviso de pagamento e confirmação do sucesso da transferência é realizado manualmente por colaboradores Inapa. Consequentemente, da realização manual do processo referido, destacam-se os seguintes problemas:

- É um processo moroso e repetitivo que impacta a produtividade do colaborador Inapa auferido à sua realização;
- É um processo suscetível a falhas, uma vez que toda a verificação dos valores das faturas e comparação com os valores presentes no aviso de pagamento é verificada pelo ser humano;
- É um processo que não tem uma periodicidade de execução bem definida.

1.3 Objetivos

Dado o âmbito manual no qual o processo de tratamento de avisos de pagamentos está inserido, e dado o facto do processo em questão ser complexo, composto por várias etapas que quando executadas manualmente são mais propensas a erro, pretende-se que a implementação da automatização do presente processo seja dividida em duas fases.

A primeira fase corresponde a recuperação dos ficheiros dos avisos de pagamento a tratar, as suas faturas em aberto e a validação das mesmas (compreendidas entre as **etapas 1 e 6**). A segunda fase corresponde a identificação das referências dos avisos de pagamento cuja validação das faturas tenha sido bem-sucedida, a sua validação, e o fecho dos processos validados (**etapas 7,8 e 9**).

Adicionalmente, a solução pretendida implica o desenvolvimento de uma integração dos serviços já existentes, substituindo/reduzindo o trabalho manual realizado. No entanto, um dos objetivos que deve ser contemplado na solução passa também pela criação de mecanismos que permitam que os colaboradores Inapa tenham visibilidade e controlo sobre os ficheiros tratados. Ou seja, mecanismos

de visibilidade e controlo que registem que ficheiros podem ser tratados e que ficheiros têm de ser revistos, que ficheiros foram tratados com sucesso ou com insucesso. Deve ser ainda possível visualizar o detalhe extraído e o resultado das validações para cada aviso de pagamento a tratar.

Resumindo os principais objetivos do projeto são:

1. Desenho de uma solução de automatização do processo de tratamento de aviso de pagamentos temporalmente mais eficiente que a abordagem manual;
2. Implementação da 1ª fase (**etapas 1-6**) da solução desenhada para automatização do processo, para 5 clientes pré-selecionados pela organização Inapa;
3. Desenvolvimento de mecanismos de visibilidade, transparência e controlo sobre os ficheiros validados e tratados (detalhados em 3.2.1).

1.4 Estrutura do Documento

Este documento é composto por cinco capítulos:

1. **Introdução:** subdividido em apresentação da empresa, descrição do problema, objetivos e estrutura do documento. Este capítulo é dedicado ao enquadramento do leitor no problema identificado;
2. **Estado de Arte:** responsável pela apresentação da pesquisa conduzida ao tema do documento. Este capítulo inclui ainda a análise tecnológica à organização, um estudo de soluções e tecnologias semelhantes ao problema identificado e a sua comparação e análise;
3. **Análise:** dedicado à análise mediante a necessidade do projeto. Esta engloba uma análise aprofundada sobre o problema, os objetivos concretos esperados da implementação da solução, apresentação sobre requisitos funcionais e não funcionais, e a exposição da análise de valor.
4. **Desenho da Solução:** onde é explicada a solução delineada, o *design* concebido para a mesma e as alternativas de *design* ponderadas;
5. **Experimentação e Avaliação:** reservado à apresentação dos testes a serem implementados de forma a avaliar a solução.

2 Estado da Arte

O presente capítulo trata de apresentar uma análise aos vários conceitos e tecnologias que possam ser exploradas no âmbito de uma possível solução para o problema proposto. Após a examinação dos diferentes conceitos e tecnologias, dentro de cada subgrupo serão expostas as análises comparativas realizadas.

2.1 Visão Geral

Atualmente, dos muitos processos levados a cabo pela Inapa, o processo de **tratamento dos avisos de pagamento**, que é responsável pela verificação e comparação dos movimentos enviados pelo cliente com os registos dos mesmos no sistema interno da Inapa, é realizado manualmente. Este processo implica inevitavelmente o tratamento de informação não estruturada. De um modo geral, informação não estruturada poder ser entendida como informação que não está em conformidade com nenhum modelo de dados e pode ser representada sob múltiplos formatos como texto, áudio, ficheiros, imagens, vídeos e outros (Seng, 2019). O facto de este tipo de informação não seguir um modelo concreto faz com o que o seu tratamento seja imprevisível e demorado. Uma vez que uma das partes mais relevantes do processo mencionado implica o tratamento de informação não estruturada, foi efetuado um levantamento das diferentes maneiras viáveis de se proceder à sua automatização. Este levantamento poderá ser consultado no subcapítulo 2.2.5.

No entanto, primeiramente será realizada uma breve análise tecnológica à organização Inapa, de modo a demonstrar que outras tecnologias poderão ser impactantes para a automatização do processo como um todo.

2.1.1 Análise Tecnológica Inapa

A Inapa é uma empresa tecnologicamente vasta, sendo utilizadas várias tecnologias consoante as necessidades e objetivos da parte de negócio correspondente. No entanto, como o processo relativo ao tratamento de aviso de pagamentos é maioritariamente manual, o número de tecnologias que merecem destaque é reduzido, frisando o uso da plataforma SAP (SAP SE, s.d.).

2.1.2 SAP ERP

SAP ERP é um gestor de recursos de empresa desenvolvido pela empresa SAP SE. Um sistema ERP (do inglês Enterprise Resource Planning) é um sistema modular feito para integrar as principais funções do núcleo negocial de uma empresa num único sistema homogéneo.

Dos 25 módulos oferecidos pelo SAP, divididos em módulos funcionais e módulos técnicos, a Inapa faz uso de um total de 5 módulos, nomeadamente:

1. **BW** (*Business Warehouse*);
2. **FI** (*Financial*);
3. **MM** (*Material Management*);
4. **SD** (*Sales Distribution*);
5. **CO** (*Controlling*).

O principal objetivo da Inapa ao utilizar o **SAP** seria possuir uma abordagem que permitisse unificar todas as suas geografias, em vez ter uma abordagem heterogénea por geografia. O processo de tratamento de avisos de pagamento tem como foco o módulo financeiro.

2.2 Automatização de Processos

Neste subcapítulo serão apresentadas diferentes formas e tecnologias exploradas que contribuam para a automatização de processos como um todo ou das suas componentes individuais.

2.2.1 Automatização Robótica de Processos

Automação robótica de processos ou **RPA** (do inglês **Robotic Process Automation**) refere-se a aplicações de automação de processos de tecnologias de informação baseadas em software de robô com capacidade de capturar e interpretar processos de organizações (Vajgel, et al., 2021).

Atualmente, esta automatização robótica de processos tem sido cada vez mais apreciada por empresas. O facto de uma licença software **RPA** poder custar entre 1/3 e 1/5 do custo de um trabalhador a tempo inteiro e do robô poder, discutivelmente, realizar tarefas estruturadas equivalentemente a um número entre 2 e 5 humanos, têm sido argumentos de peso de como **RPA** ambiciona revolucionar a redução de custos e aumentar a produção (Enríquez, et al., 2020).

O **RPA**, além de vários benefícios e capacidades, apresenta também vários desafios, como corroborado pelos documentos (Vajgel, et al., 2021) e (Enríquez, et al., 2020):

- Benefícios
 - Eficiência Operacional;
 - Qualidade de Serviço Realizada;
 - Gestão de Risco;
 - Facilidade de Configuração – Dispensa a necessidade de Engenheiros de Software especializados;
 - Plataforma robusta em termos de segurança e auditabilidade.
 - Disponibilidade praticamente ilimitada.
- Capacidades
 - Capacidades ao nível de um trabalhador;
 - Transparência e padronização na realização de tarefas;
 - Flexibilidade, escalabilidade e controlo.

- Desafios
 - Mecanismos de Avaliação de Infraestruturas;
 - Modelos de Avaliação de capacidades organizacionais;
 - Maximização de capacidades analíticas;
 - Técnicas de Gestão de Escalabilidade;
 - Técnicas de Seleção de Tarefas.

No entanto, nem todos os processos são adequados à utilização de RPA. De acordo com o referido em (Deloitte, s.d.), os processos mais adequados à implementação de uma solução RPA são aqueles que apresentam alguma ou várias das seguintes características:

- São processos repetitivos;
- São processos muito propensos à existência de erro;
- São processos com regras bem definidas.

Tendo em conta que o processo de Tratamento de Aviso de Pagamentos é um processo repetitivo, fortemente propenso a erros e com regras de tratamento bem definidas, conseguimos concluir que o mesmo seria adequado para automatização segundo **RPA**. A utilização **RPA** seria uma mais-valia para o processo em mãos uma vez que permitiria automatizá-lo, diminuindo a intervenção humana apenas aos casos estritamente necessários. Dada esta compatibilidade, foram estudadas as principais ferramentas RPA fundamentais para uma implementação automatizada do processo.

2.2.2 Soluções Existentes

No mercado de automatização segundo **RPA**, existem várias ferramentas distintas, cada qual com as suas vantagens e desvantagens. A análise seguinte pretende explorar as diferentes soluções existentes no mercado, bem como as principais diferenças entre as mesmas. Será ainda apresentada uma comparação detalhada entre as várias ferramentas exploradas.

2.2.2.1 UIPATH

UiPath é uma empresa de software, fundada em Bucareste, em 2005, por Daniel Dines. A empresa é sobretudo dedicada ao desenvolvimento de software de automatização robótica de processos (**RPA**).

O produto principal desenvolvido pela empresa denomina-se *UiPath Automation Platform* e foca-se na automatização repetitiva de tarefas digitais, que são normalmente executadas pelo ser humano. Esta plataforma combina um *IDE* (do inglês, *Integrated Development Environment*) visual à base de *low code*, o *UiPath Studio*, com agentes (robôs de *software*) que executam os processos de utilizador. Estes robôs disponibilizados tanto realizam tarefas de *front-office* como também de *back-office*, sendo versáteis para virtualmente qualquer necessidade digital. Posteriormente, os processos UiPath criados no *UiPath Studio* podem ser implantados, monitorizados e geridos remotamente através da ferramenta de controlo central chamada *UiPath Orchestrator*.

O facto de ter um IDE visual e uma interface *low code* permite diminuir a curva de aprendizagem de utilização, fazendo uso de um mecanismo de *drag and drop*, que pode ser utilizado por qualquer tipo de utilizadores, mesmo sem nenhuma experiência de programação. Adicionalmente, a plataforma contém de raiz mais de 300 componentes de automatização e outros tantos que podem ser adicionados através da utilização de pacotes disponibilizados por terceiros.

2.2.2.2 Blue Prism

O produto, com o mesmo nome que a empresa, *Blue Prism* foi construído sob a linguagem *Java* para automatizar virtualmente qualquer operação em qualquer plataforma. O *Blue Prism* vem munido de várias funcionalidades indispensáveis e úteis como recursos de segurança, encriptação de informação, distribuição de carga, entre outros. O produto oferece uma *interface* visual que permite criar, editar e testar processos de negócio e requer apenas um nível básico de experiência em programação. Apesar de não ter um **macrorecorder**, o produto disponibiliza robôs que realizam tarefas *back-office*

A ferramenta em questão é composta pelos componentes *Object Studio*, *Process Studio* e *Control Room*.

- **Object Studio** – É um *canvas* de *design* onde o robô disponibilizado pode ser programado e treinado para utilizar aplicações usadas pela empresa em questão. Do seu uso resultam blocos reutilizáveis correspondentes à execução de processos de negócio;
- **Process Studio** – Permite testar a lógica de negócio desenvolvida, através do controlo de *loops*, variáveis e chamadas aos blocos desenvolvidos no *Object Studio*;
- **Control Room** – Permite monitorizar e gerir o estado dos recursos interconectados no ambiente de produção.

2.2.2.3 Automation Anywhere

A empresa *Automation Anywhere* disponibiliza vários produtos, sendo o seu principal a ferramenta *Automation Anywhere*. Esta ferramenta tem como foco a automatização de operações do negócio ponta-a-ponta das empresas e é usada sobretudo para tarefas repetitivas com regras bem definidas. Esta ferramenta oferece ainda automação cognitiva para tarefas com informação semiestruturada e possui uma vertente analítica que disponibiliza um conjunto de métricas para analisar o negócio em tempo real. A ferramenta referida é composta por dois componentes:

- 1- **Enterprise Client** – Representa a interface primária para os subcomponentes *Bot Creator* e *Bot Runner*.
 - **Bot Creator** - Descrevendo um *bot* no presente contexto como uma tarefa independente programada para ser executada com pouca ou nenhuma intervenção humana (Automation Anywhere, 2020). O *Bot Creator* é o local de desenvolvimento de *bots* da ferramenta **Automation Anywhere** e conta com um vasto leque de funcionalidades, nomeadamente o editor de tarefas, o observador de eventos, o *bot player*, o login automático e a agenda local;
 - **Bot Runner** – É a máquina de software responsável pela execução dos *bots* criados com o *Bot Creator*.

2- **Control Room** – Permite monitorizar e gerir todas as tarefas automatizadas.

O produto *Automation Anywhere* foi construído baseado na arquitetura cliente-servidor e oferece robôs tanto para *front-office* como *back-office*.

2.2.3 Análise Comparativa

Após a análise individual às mais importantes ferramentas do mercado no âmbito de utilização de RPA, segue-se uma análise comparativa entre as mesmas, de forma a expor resumida e comparativamente as diferenças fundamentais entre elas. Esta análise comparativa basear-se-á em seis atributos com relevância para a distinção das ferramentas e para a organização Inapa.

Estes atributos são os seguintes:

- Tipo de Robots;
- Existência de Macro-Recorder;
- Interação com a plataforma;
- Versões Gratuitas;
- Suporte e Recursos de aprendizagem;
- Nível mínimo de experiência de programação.

Relativamente ao atributo de tipo de robots, é importante perceber a versatilidade de tarefas que os robots oferecidos por cada uma das plataformas consegue realizar, pois no âmbito do projeto podem ser necessárias tarefas impossíveis de realizar pelos robots oferecidos.

A importância da existência de um **macro-recorder** prende-se à capacidade de guardar certos eventos (o clicar no rato ou teclas premidas no teclado). Desta maneira, desde que o contexto seja o mesmo, certas tarefas podem ser automatizadas reproduzindo as instruções previamente gravadas.

Outra vertente importante é a vertente da formação e aprendizagem de futuros utilizadores dentro da organização. Para tal, são analisadas as características de interação com a aplicação, versões existentes, recursos de aprendizagem e nível mínimo de experiência em programação. Para tal será averiguado se as plataformas em causa oferecem uma maneira mais intuitiva de interação com as mesmas, preferencialmente mais visual do que descritiva e se as plataformas oferecem um tipo de versão gratuita.

Também significativo são as temáticas da facilidade de acesso a recursos de aprendizagem, uma vez que permitem acelerar a formação de utilizadores, e o nível mínimo de experiência de programação necessária, uma vez que dita a abrangência da aplicação a diferentes tipos de utilizadores.

A 1 apresenta a comparação das três plataformas identificadas relativamente aos critérios apresentados.

CrITÉRIOS/FERRAMENTAS	UiPath	Blue Prism	Automation Anywhere
Tipo de Robôs	<i>Front-office e Back-office</i>	<i>Back-office</i>	<i>Front-office e Back-office</i>
Macro-Recorder	Sim	Não	Sim
Interação com a plataforma	Visual com <i>Drag and Drop</i>	Visual com <i>Drag and Drop</i>	Visual com <i>Drag and Drop</i>
Versões oferecidas	Versão comunidade e versão teste 30 dias	Versão teste 30 dias	Versão comunidade para estudantes e desenvolvedores, e versão teste 30 dias geral
Suporte e Recursos de Aprendizagem	Contratos de Treino, Vídeos tutoriais, Fórum da comunidade.	Guia de Ajuda, Email, Contratos de treino	Treino e Certificações
Nível de Experiência recomendado em programação	Nenhum	Nenhum	Nenhum

Tabela 1 – Comparação das Principais Ferramentas RPA

Através da visualização e análise da 1, é possível verificar que de um modo geral, nos requisitos considerados, as aplicações apresentam apenas ligeiras diferenças entre si. No requisito de **tipo de robôs**, as aplicações de *UiPath* e *Automation Anywhere* oferecem tanto a possibilidade de realizarem tarefas *front-office* como *back-office*, a *Blue Prism* não é tão versátil, disponibilizando apenas robôs para *back-office*. No segundo requisito, a oferta de *macro-recorder*, o padrão mantém-se, uma vez que a *Blue Prism* é o único produto que não dispõe desta funcionalidade. Avaliando o terceiro requisito – Interação com a plataforma – consegue-se entender que existe um empate técnico, uma vez que ambas as aplicações possuem uma interação visual via *drag and drop*.

Os restantes requisitos propostos serão analisados no contexto da vertente de formação e aprendizagem de futuros utilizadores, definida anteriormente no presente capítulo. No quarto requisito, de **versões oferecidas**, o *UiPath* destaca-se da concorrência ao oferecer tanto uma versão de teste por 30 dias, como uma versão comunidade limitada. Este ponto é considerado relevante uma vez que pode ser utilizado *gratuitamente* para formar novos utilizadores sem a necessidade de mais custos com licenças sem os utilizadores conseguirem tirar proveito das suas capacidades. Neste requisito, a *Automation Anywhere* oferece também uma versão teste por 30 dias e uma versão comunidade mais limitada que a do *UiPath*, e a *Blue Prism* oferece apenas uma versão teste gratuita por 30 dias.

No que toca ao requisito de suporte e recursos de aprendizagem, ambas as empresas oferecem contratos de treino, dando-se novamente destaque ao *UiPath* por expor um fórum comunidade. Este fórum promove a vertente de **facilidade de acesso a recursos de aprendizagem**, uma vez que permite a partilha de conhecimento e soluções, contribuindo positivamente para a formação de

utilizadores. Por fim, ainda na vertente de formação e aprendizagem de futuros utilizadores, no requisito **nível de experiência recomendado em programação**, considera-se que não há nenhuma aplicação a destacar uma vez que nenhuma das aplicações requer algum tipo de conhecimento e experiência de programação. Após a exposição desta análise, o **UiPath** foi considerado como tendo uma margem residual sobre as outras aplicações. No entanto, apesar da avaliação imparcial sobre as três tecnologias apresentadas, há que realçar que a organização tem também preferência pela utilização da tecnologia **UiPath**, uma vez que já concluiu processos internos recorrendo à mesma ferramenta.

2.2.4 Automatização de componentes

Nos subcapítulos precedentes ao presente foi apresentada uma introdução à abordagem de automatização robótica de processos e de como é que a mesma se poderia encaixar para uma possível resolução do problema apresentado.

No entanto, a abordagem **RPA** não foi a única abordagem explorada. Um dos principais e mais críticos pontos no processo de tratamento de aviso de pagamentos passa pela interpretação dos campos e valores presentes nos ficheiros de Avisos de Pagamento (e posterior validação dos mesmos), enviados pelos clientes da Inapa. Nesse sentido, foram também exploradas abordagens focadas na extração automática de dados e informação não estruturada de ficheiros. Estas abordagens diferem sobretudo da abordagem **RPA** por terem como cerne isolado a vertente de extração de informação de ficheiros, em vez de constituírem uma solução global para o problema descrito.

Apesar desta abordagem não permitir uma automatização total do processo, a sua utilização contribuiria significativamente para a otimização da atual extração desgastante de informação dos avisos de pagamento, realizada manualmente. Adicionalmente, esta poderia ser combinada com outras técnicas de modo a automatizar o processo como um todo.

De um modo geral, ambas as abordagens têm os seus próprios benefícios e desvantagens, e as mesmas podem ser consultadas em mais detalhe no capítulo 3.3.

2.2.5 Soluções existentes

No âmbito de extração de informação não estruturada de documentos existem atualmente no mercado várias ferramentas disponíveis, sendo que este tipo de produtos não é necessariamente novo. Dessas ferramentas disponíveis foi feita uma seleção baseada nas melhores soluções de automatização de faturas segundo (Tripathi, 2023) , e sobre essa seleção foram realizadas análises individuais às ferramentas e uma análise comparativa entre todas elas.

2.2.5.1 Rossum

A **Rossum** é uma organização que se descreve sendo uma solução moderna de processamento inteligente de documentos, com centenas de clientes espalhados pelo planeta. O produto oferecido

pela organização trata-se da **plataforma Rossum**, cujo objetivo passa pela diminuição do tempo gasto na realização de trabalho manual (Rossum, s.d.).

A plataforma oferecida pela empresa oferece funcionalidades que se enquadram em duas vertentes distintas (Rossum, s.d.):

1. **Gestão de documentos**
 - a. Suporte de documentos em vários formatos;
 - b. Filtro de documentos;
 - c. Organização e ordenação com um sistema de fila flexível.
2. **Extração de dados**
 - a. Extração e compreensão de objetos complexos (e.g tabelas e grelhas);
 - b. Permite criação de regras de negócio a aplicar aos *templates*;
 - c. Leitura de documentos que suporta mudanças na estrutura (*layout*) dos documentos.
 - d. Validações aplicadas ao tipo de dados, podendo ser corrigidas manualmente.

A plataforma **Rossum** é uma solução alimentada por uma inteligência artificial, baseada na nuvem e que assegura uma precisão de 98% na extração de campos. Além disso, em (Tripathi, 2023) e (Capterra, s.d.) temos uma visão geral de mais alguns prós e contras desta plataforma, nomeadamente:

Vantagens:

- Fornecimento de materiais de treino sob a forma de aulas ao vivo, *webinars* e documentação sobre a sua utilização;
- Integração com email e com **RPA (Robot Process Automation)**;
- Disponibilização de uma API para uma integração mais flexível;
- Proteção de dados na nuvem;
- Versão teste gratuita.

Desvantagens:

- Cobrança por segurança acrescida na comunicação;
- Poucas funcionalidades base apresentadas.

2.2.5.2 Tipalti

A **Tipalti** (Tipalti, s.d.) é uma solução **SaaS (Software as a Service)** baseada na nuvem que ajuda na automatização de contas de pagamento, ideal para pequenos e médios negócios. Esta solução agiliza os fluxos de trabalho de pagamento de fornecedores, permite a integração de fornecedores, a cobrança de faturas e gestão legal adjacente. Além de oferecer suporte e integração por parte de terceiros, a ferramenta permite ainda que os utilizadores criem relatórios de reconciliação de pagamentos que podem ser vinculados diretamente aos sistemas de gestão ERP (Tripathi, 2023).

Da análise desta solução foram encontradas as seguintes vantagens e desvantagens em (Tripathi, 2023) e (Capterra, s.d.):

Vantagens:

- Automatização de extração de dados de faturas;
- Automatização de fluxos de aprovação de faturas;
- Disponibilização de vários métodos de pagamento em vários tipos de moeda;
- Armazenamento de documentos e validações bancárias.

Desvantagens:

- A plataforma necessita que lhe sejam fornecidos os documentos a processar, não o fazendo automaticamente;
- Falta de detalhes em caso de erro de pagamento;
- Pouca integração com fornecedores como por exemplo bancos.

2.2.5.3 ABBYY FlexiCapture

A **ABBYY FlexiCapture** é uma solução robusta de captura e processamento de dados projetada para trabalhar com fluxos de documentos com virtualmente qualquer complexidade e *layout*.

Adicionalmente, a solução em causa é altamente escalável e permite a classificação automática de documentos, permitindo às organizações aumentarem a sua eficiência e reduzindo os custos associados ao processamento destes documentos de forma manual (Tripathi, 2023).

Da análise desta solução foram encontradas as seguintes vantagens e desvantagens em (Tripathi, 2023) e (Capterra, s.d.):

Vantagens:

- Verificação de dados em documentos;
- Conversão de documentos físicos para meio digital;
- Captura distribuída de informação;

Desvantagens:

- Uma interface mais técnica e complexa de se navegar;
- Necessita de atualizações massivas que requerem reinstalação do software;
- A entrega de resultados é lenta;
- Eficácia de alguns processos não é tão alta quanto o expectável.

2.2.5.4 Docsumo

A **Docsumo** é uma solução de processamento inteligente de documentos otimizada para extrair informação com uma precisão de 99%. Além desta solução trabalhar com virtualmente qualquer tipo de documentos independentemente do seu layout sem necessitar de configuração manual prévia, a mesma permite ainda treinar um modelo para *templates* mais específicos. A **Docsumo** disponibiliza forma de integrar com software de terceiros, alimentando-o com a extração de dados em formatos como *excel*, *json*, *txt* e *csv* (Tripathi, 2023).

Da análise desta solução foram encontradas as seguintes vantagens e desvantagens em (Tripathi, 2023):

Vantagens:

- Captura e transferência inteligente de informação;
- Detecção de fraude de documento em tempo real;
- Extração de dados em tabelas e validação dos dados consoante o seu tipo de dados;
- Proteção de dados na nuvem.

Desvantagens:

- A documentação da API está incompleta;
- Não suporta documentos escritos à mão.

2.2.6 Análise Comparativa

Após a análise individual sob as ferramentas apresentadas, segue-se uma análise comparativa entre as mesmas, de forma a expor resumida e comparativamente as diferenças fundamentais entre elas. Esta análise comparativa basear-se-á em quatro atributos com relevância para a distinção das ferramentas para a organização Inapa.

Estes atributos são os seguintes:

- Disponibilização de uma *API* para comunicação com aplicações;
- Validação de dados recolhidos
- Facilidade de uso;
- Documentação e recursos de aprendizagem.

A **disponibilização de uma API** para comunicação com aplicações de terceiros é praticamente um requisito obrigatório. Esta seria necessária de modo a permitir a alimentação de ficheiros e a respetiva recolha de dados.

Uma outra vertente crucial espelhada nos critérios escolhidos é da coerência dos dados recolhidos. O critério **validação de dados recolhidos** pretende justamente avaliar que ferramentas é que oferecem mecanismos de prevenção, validação e correção das extrações de informação não estruturada de documentos. Dentro deste critério encaixam-se mecanismos como validações de tipo de dados recolhidos e possibilidade de conferência e correção de dados por recursos humanos.

Por fim, os últimos dois critérios prendem-se com o uso da ferramenta em si, nomeadamente se a mesma é acessível e pouco complexa de se usar, e se as ferramentas apresentam materiais que permitam a sua fácil compreensão (e.g tutoriais, documentação, fórum-comunidade, entre outros).

Apesar de o preço também ser um critério interessante a considerar, na pesquisa efetivada não nos foi possível averiguar os preços do uso das ferramentas, uma vez que estes dependiam diretamente de consulta com as equipas de vendas e/ou do volume páginas a processar mensalmente, cuja estimativa não foi possível obter. Como não seria possível retirar conclusões sobre esse critério, o mesmo não foi considerado para a tabela comparativa correspondente.

A Tabela 2 - Comparação de Ferramentas de Extração de Informação não estruturada de ficheiros Tabela 2 exhibe justamente a comparação das plataformas identificadas relativamente aos critérios previamente referidos.

Crítérios/Ferramentas	Rossum	Tipalti	Docsumo	ABBY FlexiCapture
Disponibiliza API	Sim	Só para pagamentos	Sim	Sim
Suporta validação de dados recolhidos	• Validação de tipos de dados; • Validação manual por colaboradores.	• Validação de tipo de dados.	• Validação de tipos de dados; • Validação manual por colaboradores.	• Validação de tipos de dados; • Validação manual por colaboradores; • Validação condicional de regras.
Facilidade de uso	Fácil, interface intuitiva	Fácil, interface intuitiva	Fácil, interface intuitiva	Média, interface mais técnica.
Recursos de aprendizagem	• Documentação; • E-books; • Vídeos-demo; • Webinars sob demanda.	• Guias; • Demo ao cliente.	• Documentação incompleta; • Demo ao cliente.	• Vídeos Tutoriais; • Documentação da API.

Tabela 2 - Comparação de Ferramentas de Extração de Informação não estruturada de ficheiros

Após a análise dos dados da Tabela 2 e da correlação com a respetiva apresentação das ferramentas realizada no presente capítulo é possível entender que embora as ferramentas estudadas permitam de um modo geral a extração de informação não estruturada de documentos, as mesmas servem diferentes públicos-alvo. Enquanto a **ABBY FlexiCapture** é uma ferramenta de extração geral de informação de virtualmente qualquer documento, as ferramentas **Docsumo** e **Rossum** são mais especializadas para a leitura de faturas e ficheiros financeiros. Por sua vez, a solução **Tipalti** apresenta todo um arsenal de integração com a área financeira, permitindo não só a extração de informação de documentos, mas também a própria a submissão de despesas e até efetuar pagamentos.

Começando a análise pelo requisito fundamental de **disponibilização de uma API**, foi verificado que apesar de “tecnicamente” todos cumprirem com este critério, a **API** disponibilizada pela **Tipalti** se destina apenas a pagamentos. Como a realização de pagamentos se encontra excluída do âmbito do projeto proposto, considera-se que a **Tipalti** não apresenta uma API válida para a extração de dados. Qualquer extração terá de ser feita na plataforma disponibilizada. Isto constitui uma séria desvantagem face às outras soluções, uma vez que não permite uma automatização de fluxos sem um Input manual, e portanto, poderia não ser benéfico para a solução idealizada.

Em segundo lugar, no critério de **validação de dados recolhidos** é passível de observação que todas as ferramentas cumprem com este critério, apresentando a possibilidade de validação de dados. Apesar de todas as soluções apresentarem a possibilidade de validação manual dos dados por colaboradores, a **ABBY FlexiCapture** deve ser destacada pela oferta de validação de regras condicional. Esta validação permite que sejam definidas regras para a validação dos dados (e.g valor deve ser positivo e maior do que 10). Esta funcionalidade seria relevante para o projeto em questão, uma vez que o processo manual atual implica a realização de validações de natureza semelhante.

Face ao terceiro critério, **facilidade de uso**, conclui-se que a **ABBY FlexiCapture** tem um destaque negativo, uma vez que é a única ferramenta que tem como uma desvantagem uma interface mais técnica e complexa. Este terá um impacto negativo na acessibilidade para colaboradores Inapa que necessitem de realizar validações dos dados. Uma interface pouco intuitiva agrava a curva de aprendizagem e poderá contribuir para uma experiência negativa dos colaboradores e uma maior resistência à utilização de uma certa solução em detrimento de outras.

Por fim, examinando o quarto e último critério – **recursos de aprendizagem** – verifica-se uma certa distinção da aplicação **Rossum**. Esta solução apresenta um elevado número de materiais de estudo, incluindo não só documentação, mas também vídeos-tutorial e até *webinars* consoante a necessidade. Estes recursos auxiliares poderiam revelar-se extremamente úteis para permitir uma rápida integração da solução e um acelerado enquadramento por parte dos colaboradores que necessitam de verificações manuais.

Após a inspeção efetuada aos critérios e ferramentas demonstrados ao longo desta análise, foi possível chegar a uma solução. Tendo em conta o facto da ferramenta ser especializada para a leitura de faturas e documentos financeiros, de a mesma possuir uma **API**, de apresentar uma interface intuitiva e o maior número de recursos de aprendizagem, aliado ao facto de a mesma não

apresentar nenhum defeito crítico evidente, justifica-se assim a escolha da ferramenta **Rossum** como a ferramenta melhor indicada para o contexto do projeto proposto.

3 Análise do Problema

Este capítulo destina-se à apresentação de assuntos relacionados com a análise do problema. Inicialmente, começar-se-á por apresentar o domínio do problema, para uma melhor compreensão do mesmo. A este segue-se a apresentação de requisitos funcionais acompanhados de diagramas dos casos de uso mais relevantes a considerar, e finalmente, uma análise dos requisitos não funcionais do problema, segundo o modelo FURPS+.

3.1 Domínio do Problema

O problema em questão pode ser visto como uma aplicação do desafio global de automatização de tarefas de utilizador, direcionado ao âmbito reduzido do processo interno de tratamento de avisos de pagamento. Este processo é, de um modo geral, responsável pela verificação financeira de que as faturas das encomendas realizadas pelos clientes da Inapa se encontram efetivamente liquidadas. Atualmente é realizado manualmente por colaboradores da Inapa e apresenta extrema relevância, uma vez que pode impactar compras futuras dos clientes. De modo a fomentar o entendimento facilitado do processo, o mesmo é primeiramente apresentado pelas Figura 1, Figura 2 e Figura 3, sendo seguidamente descritas e detalhadas as etapas presentes nos respetivos diagramas.

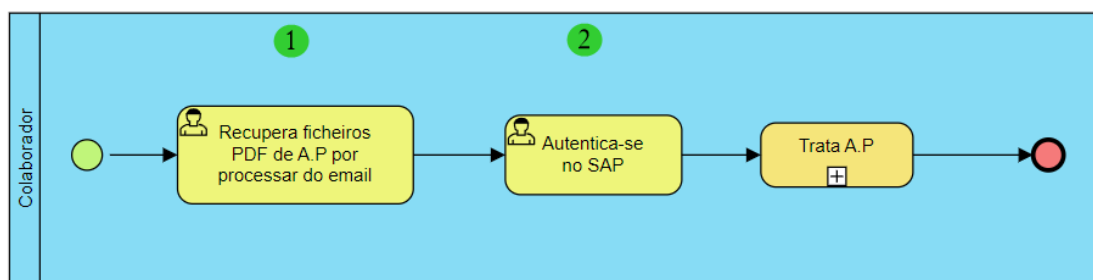


Figura 1 - Diagrama BPMN Tratamento de Avisos de Pagamentos

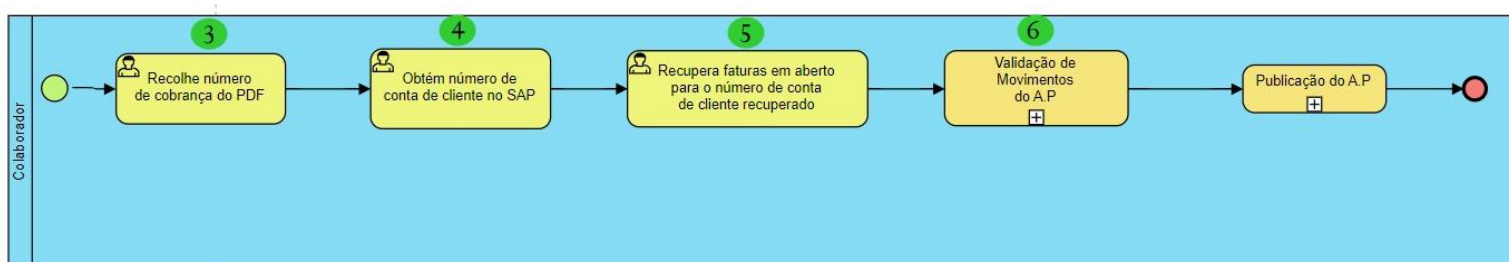


Figura 2 -Diagrama BPMN do Subprocesso "Trata A.P"

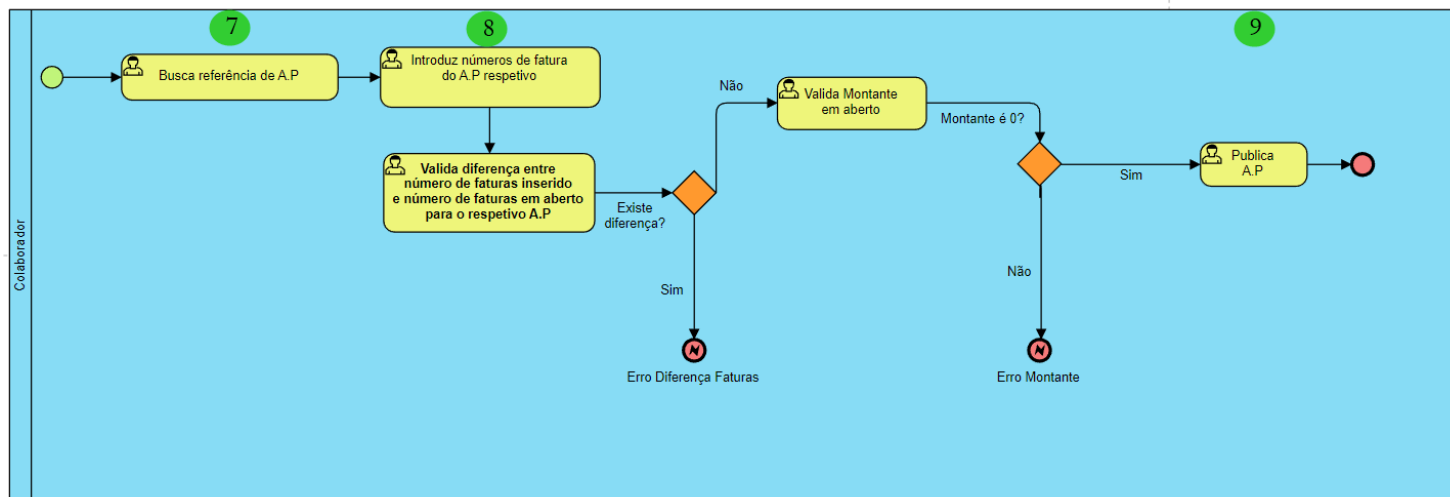


Figura 3 - Diagrama BPMN Subprocesso Publicação do A.P

Os diagramas do processo contêm ainda círculos numerados, sendo que a cada um corresponde a respectiva etapa abaixo ilustrada, sendo elas:

1. O colaborador verifica o email e recupera os ficheiros *PDF* correspondentes aos avisos de pagamento por verificar;
2. O colaborador autentica-se no *SAP* para a realização dos passos seguintes;
3. Para o *PDF* recuperado, o colaborador recolhe o número de cobrança que vêm no aviso de pagamento em questão;
4. Através desse número de cobrança, o colaborador vai fazer uso do sistema *SAP*, para obter o número da conta associada ao número de cobrança pesquisado;
5. Usando o *SAP*, o colaborador faz uso do número da conta recuperado no passo anterior, recuperando as faturas (*invoices*) em aberto associados ao número em questão.
6. O colaborador deve validar cada movimento do Ficheiro *PDF* de aviso de pagamento. Caso o aviso de pagamento tenha alguma discrepância, terá de ser revisto manualmente por pessoal autorizado;
7. No caso de o aviso de pagamento não apresentar qualquer discrepância, o colaborador deve através do sistema *SAP* encontrar a referência do pagamento em questão no *SAP*;
8. O colaborador deve realizar validações sobre a referência de aviso de pagamento recuperada do *SAP*. Caso a referência do aviso de pagamento no *SAP* tenha alguma discrepância, o mesmo necessitará de ser revisto manualmente por pessoal autorizado;
9. Caso a referência do aviso de pagamento no *SAP* não apresente qualquer discrepância, o mesmo deverá ser marcado como fechado, recorrendo a um botão de *POST* disponibilizado pela plataforma *SAP*.

Devido à necessidade de algumas das etapas previamente referidas do processo de Tratamento de Aviso de Pagamentos de serem detalhadas, as mesmas serão apresentadas de seguida.

Etapa 3 – Recuperação do número de cobrança nos Avisos de Pagamentos

Os ficheiros *PDF* correspondentes aos avisos de pagamento têm diferentes formatos, vindo de diferentes modelos, variando sobretudo com a geografia do cliente e com o próprio cliente em questão. No entanto, todos os avisos de pagamento têm alguns campos em comum (que podem ter designações diferentes). Dos campos em comum destacam-se o que pode ser traduzido português

como o número de cobrança. Este número identifica exclusivamente uma transação monetária associada.

Etapas 4 – Recuperação do número da conta associada ao número de cobrança

A plataforma SAP apresenta a possibilidade de visualizar documentos de cobrança o número de cobrança associado, através da realização da transação padrão VF03 (SAP, 2006). Um documento de cobrança apresenta um conjunto de informação relevante, como observável na Figura 4. No entanto, para o contexto do processo de tratamento de avisos de pagamento, o campo de maior relevância é o campo **Payer**, representante do código de conta associada à cobrança realizada.

Item	Material	Item Description	Billed Quantity	SU	Net Value	Curr...	Tax Armo
42100011387	42100011387	tecno colors 160g 210x297 R ..	40 VE		188,00	EUR	

Figura 4 - Resultado da Transação VF03

Etapa 5 – Recuperação das faturas em aberto associadas ao número de conta

A plataforma SAP apresenta através da transação padrão FBL5N a possibilidade de listar vários itens (no presente caso, faturas) associadas a um conjunto de fatores (Johns Hopkins Medical Institutions, 2006). Associando o número de conta previamente recuperado e o código da empresa interna Inapa, recupera-se um conjunto de faturas em aberto para os campos referidos, como observável na Figura 5.

St	Type	PM	Reference	Assignment	Doc. Date	Amount in Local Crpy	D/C	DD	Discount	Amount PayT	Net Due Date	Arrear	Dun. Text	Orig doc.	Last Dunned	Block	Va
VR			9001057023	9001057023	04.12.2020	154,05	S	2	0,00	C063	13.01.2021	217	1		09.08.2021		
VR			9001058263	9001058263	07.12.2020	79,34	S	2	0,00	C063	16.01.2021	214	1		09.08.2021		
VR			9001058275	9001058275		80,74	S	2	0,00	C063	16.01.2021	214	1		09.08.2021		
VR			9001059446	9001059446	08.12.2020	85,38	S	2	0,00	C063	17.01.2021	213	1		09.08.2021		
VR			9001059447	9001059447		77,12	S	2	0,00	C063	17.01.2021	213	1		09.08.2021		
VR			9001060616	9001060616	09.12.2020	883,92	S	2	0,00	C063	18.01.2021	212	1		09.08.2021		
VR			9001060617	9001060617		85,38	S	2	0,00	C063	18.01.2021	212	1		09.08.2021		
VR			9001060621	9001060621		77,26	S	2	0,00	C063	18.01.2021	212	1		09.08.2021		
VR			9001060623	9001060623		24,24	S	2	0,00	C063	18.01.2021	212	1		09.08.2021		
VR			9001060624	9001060624		77,49	S	2	0,00	C063	18.01.2021	212	1		09.08.2021		
VR			9001060625	9001060625		80,74	S	2	0,00	C063	18.01.2021	212	1		09.08.2021		
VR			9001060629	9001060629		87,93	S	2	0,00	C063	18.01.2021	212	1		09.08.2021		
VR			9001061931	9001061931	10.12.2020	85,38	S	2	0,00	C063	19.01.2021	211	1		09.08.2021		
VR			9001064534	9001064534	14.12.2020	765,60	S	2	0,00	C063	23.01.2021	207	1		09.08.2021		
VG			9070002080	9070002080	16.12.2020	154,23	H	2	0,00	C063	16.12.2020	245	1		09.08.2021		
VG			9070002086	9070002086		964,30	H	2	0,00	C063	16.12.2020	245	1		09.08.2021		
VG			9070002164	9070002164	17.12.2020	740,97	H	2	0,00	C063	17.12.2020	244	1		09.08.2021		
VR			9001069383	9001069383	18.12.2020	85,38	S	2	0,00	C063	27.01.2021	203	1		09.08.2021		
VR			9001069386	9001069386		83,52	S	2	0,00	C063	27.01.2021	203	1		09.08.2021		
VR			9001070429	9001070429	21.12.2020	176,78	S	2	0,00	C063	30.01.2021	200	1		09.08.2021		
VR			9001070444	9001070444		127,14	S	2	0,00	C063	30.01.2021	200	1		09.08.2021		
VR			9001072220	9001072220	22.12.2020	1.344,67	S	2	0,00	C063	31.01.2021	199	1		09.08.2021		
VR			9001072223	9001072223		896,45	S	2	0,00	C063	31.01.2021	199	1		09.08.2021		
VG			9070002339	9070002339		446,60	H	2	0,00	C063	22.12.2020	239	1		09.08.2021		
VG			9070002345	9070002345		883,92	H	2	0,00	C063	22.12.2020	239	1		09.08.2021		
VG			9070002346	9070002346		1.064,88	H	2	0,00	C063	22.12.2020	239	1		09.08.2021		
VG			9070002348	9070002348		909,44	H	2	0,00	C063	22.12.2020	239	1		09.08.2021		
VG			9070002351	9070002351		1.064,88	H	2	0,00	C063	22.12.2020	239	1		09.08.2021		
DG			2000000402	2000000402	31.12.2020	15.199,12	H	2	0,00		31.12.2020	230	1	Verkaufsbonus manuel	09.08.2021		
DG			MANUELLER BO.			60.125,96	H	2	0,00		31.12.2020	230	1	Verkaufsbonus manuel	09.08.2021		
DG						17.178,86	H	2	0,00		31.12.2020	230	1	Verkaufsbonus manuel	09.08.2021		

Figura 5 - Resultado da Transação FBL5N

Etapa 6 – Validação dos movimentos do ficheiro PDF e das faturas em aberto associadas ao número de conta

O processo de validação é composto por 3 passos, correspondentes a 3 validações individuais que se têm de verificar para cada movimento do ficheiro PDF.

1. O valor final do movimento no detalhe do ficheiro PDF deve igualar o valor da fatura correspondente em SAP, pelo que a diferença entre ambos os valores deve ser 0;
2. A diferença entre o valor a pagar e o valor esperado deve ser 0. Apesar de cada cliente apresentar campos diferentes nos movimentos dos ficheiros dos Avisos de Pagamento, o valor correto deve ser calculado recorrendo a uma fórmula Montante – (Desconto + Montante bloqueado + Abatimentos);

3. Validar o desconto ao movimento. A aplicação de um desconto deve ser validada no ecrã SAP das faturas em aberto. O desconto a ser aplicado depende de um conjunto de critérios, como o próprio cliente, o valor da fatura, tempo decorrido desde a realização da encomenda até ao seu pagamento, entre outros critérios.

Etapas 8 – Validação das referências de Aviso de Pagamento presentes no SAP

Tendo sido validados todas as faturas referentes aos movimentos apresentados no ficheiro PDF de Aviso de Pagamento, deve ser validada a referência do mesmo na Plataforma SAP. Tal é feito recorrendo novamente à Transação FEBA. Este segundo processo de validação também é realizado seguindo um conjunto de padrão de passos:

1. Introdução dos números de fatura recuperados do ficheiro PDF de Aviso de Pagamento no campo “**Notes to Payee**”;
2. Deve ser validado que o número de faturas em aberto previamente recuperadas na **Etapas 5 – Recuperação das faturas em aberto associadas ao número de conta** é o mesmo que o número de faturas introduzidas;
3. Por fim, após a introdução do número das faturas, deve ser verificado que o campo **Amount** referente ao Aviso de Pagamento se encontra com o valor 0, e que o campo **Open** se encontra com a cor verde, indicando que está de facto em aberto.

As validações referidas podem ser consultadas na Figura 6.

Figura 6 - Etapa 8: Validação das referências de Aviso de Pagamento presentes no SAP

3.2 Análise

O presente subcapítulo apresenta a identificação das necessidades e a análise elaborada tendo em conta o contexto do problema referido em 3.1 e o contexto dos objetivos mencionados em 1.3 . A análise realizada será composta pelo detalhe dos mecanismos de visibilidade e pela apresentação dos requisitos funcionais e não funcionais imperativos para solução.

3.2.1 Detalhe Mecanismos de Visibilidade

Com a finalidade de responder ao terceiro objetivo proposto - **Desenvolvimento de mecanismos de visibilidade, transparência e controlo sobre os ficheiros validados e tratados** – foram contemplados os seguintes recursos:

1. Criação do ficheiro *Excel* auxiliar **Process Summary** – destinado ao resumo da execução levada a cabo pelo robô, incluindo informação sobre todos os avisos de pagamento não processados automaticamente recolhidos por cada execução (cliente, montante, data) e o estado do mesmo, de modo que este possa ser atualizado, caso necessário, nas diferentes etapas;
2. Artefacto *Excel* **Payment Advice Details** – destinado ao resumo de detalhes sobre um aviso de pagamento específico. Contém os dados relevantes extraídos do ficheiro *PDF* para o aviso de pagamento correspondente, bem como dados necessários extraídos do **SAP** e dados correspondentes às validações a serem efetuadas;
3. Solução tecnológica de consumo intermédio de ficheiros *Process Summary* – destinada à fácil e rápida apresentação de métricas e detalhes sobre os vários ficheiros *Process Summary*. Esta solução foi pensada com o objetivo de auxiliar os colaboradores internos a escolherem os ficheiros *Process Summary* mais urgentes. A solução permite averiguar quais os *Process Summary* que se encontram em falha, quais os clientes mais afetados, quais os ficheiros com maior montante preso, entre outras métricas relevantes ao fim destinado.

Foi ainda planeada a existência de várias pastas para onde cada tipo de ficheiro deve ser enviado de modo a promover uma fácil perceção do que o que se encontra tratado e em erros. Os diretórios idealizados foram os seguintes:

1. Ficheiro **Process Summary**: Por cada ficheiro ter um diretório com a data do dia atual sob o formato *ddmmyyyy*
 - a. Caso o **Process Summary** ainda não tenha sido processado, deverá encontrar-se na pasta **ToPost**;
 - b. Caso contrário, quando o mesmo for processado deverá ser movido para pasta **Processed**.
2. Ficheiro **Payment Advice Details**: Cada um dos clientes pré-definidos para serem englobados na solução terão uma pasta destinada com o seu nome;
 - a. Caso o **Payment Advice Details** se encontre por validar, o mesmo deverá estar na pasta com o nome do respetivo do cliente;

- b. Caso o **Payment Advice Details** tenha sido validado e não apresenta quaisquer erros, o mesmo deverá ser movido para a subpasta **GoodToGo** do cliente respetivo;
- c. Caso o **Payment Advice Details** tenha sido validado e apresente algum erro, o mesmo deverá ser movido para a subpasta **ToBeChecked** do cliente respetivo.

3.2.2 Requisitos funcionais

Este subcapítulo trata da apresentação dos requisitos funcionais tanto da solução de automatização do processo de tratamento de aviso de pagamentos, como da solução complementar de consumo de ficheiros *Process Summary*. Enquanto que os primeiros serão expostos de uma forma mais suave devido à natureza do processo ao qual dizem respeito, os segundos serão detalhados e, nos casos mais relevantes, acompanhados de diagramas que auxiliem a sua compreensão.

3.2.2.1 Automatização de Tratamento dos Avisos de Pagamento

REQ 01 - Autenticação no SAP

Descrição: O robô deverá conseguir aceder à plataforma SAP residente na máquina local de um utilizador, introduzir as credenciais para si configuradas, e clicar no botão de login existente no ecrã. Como resultado, a plataforma SAP apresentará a página principal associada.

REQ 02 - Recuperação das referências de Avisos de Pagamento

Descrição: O robô deverá conseguir aceder ao ecrã SAP da transação *FEBA*, preencher os campos existentes com valores predefinidos e extrair a informação resultante para um ficheiro *excel Process Summary*. Adicionalmente, deve nomear o ficheiro *excel* de acordo com regras definidas e proceder à gravação dos dados após a sua inserção.

Pré-condições:

1. O robô deve ter acesso aos diretórios necessários;
2. O robô deve ter a capacidade de extrair dados de PDF.

REQ 03 - Recuperação dos ficheiros PDF correspondentes aos avisos de pagamento não processados em SAP

Descrição: O robô deverá para cada referência de um aviso de pagamento presentes no *Process Summary*, recuperar os valores dos campos que permitam identificar o *PDF* correspondente. De seguida, deve aceder ao diretório INAPA para o respetivo cliente (**Business Partner**), recuperar o *PDF* em questão e extrair a informação pré-estipulada sobre o *PDF*. Posteriormente, o robô deve ainda ser capaz de criar exportar a informação recuperada para um novo ficheiro *Excel* correspondente ao aviso de pagamento em questão, cujo o nome depreende da seguinte fórmula: **[Account code]_[date of payment advice]_[amount of payment advice]_[current_ddmmyy_hhmmss]**.

Pré-condições:

1. O robô deve ter acesso aos diretórios necessários.
2. O robô deve ter a capacidade de extrair dados de *PDF*.

REQ 04 – Recuperação da conta de cliente

Descrição: O robô deverá, através dos dados extraídos do ficheiro *PDF*, fazer uso ecrã de **SAP** correspondente, preencher os campos pedidos, e recuperar a conta de cliente associada ao número do aviso de pagamento que se encontra a tratar.

REQ 05 – Obtenção das faturas em aberto para uma determinada conta de cliente

Descrição: O robô deverá, para um ficheiro *Excel* gerado correspondente a um dado aviso de pagamento, fazer uso da plataforma **SAP** (ecrã de transação FBL5N) e recuperar as faturas em aberto associadas à conta em questão. De seguida, deve extraí-las para uma segunda folha no mesmo ficheiro *Excel* e persistir as alterações realizadas.

REQ 06 – Validação de movimentos do *PDF* de Aviso de Pagamento

Descrição: O robô deverá proceder à validação individual dos movimentos recuperados do ficheiro *PDF* correspondente ao aviso de pagamento em tratamento. Devem ser realizadas validações (Etapa 6 – Validação dos movimentos do ficheiro *PDF* e das faturas em aberto associadas ao número de conta) face ao desconto a aplicar, desconto efetivamente aplicado e diferenças entre o valor monetário da fatura em SAP e do movimentos comunicado (previamente recuperados no REQ 03 - Recuperação dos ficheiros *PDF* correspondentes aos avisos de pagamento não processados em SAP). Consoante os resultados das validações, o robô deverá ainda marcar o registo acabado de validar no ficheiro *Excel* correspondente ao aviso de pagamento com uma das seguintes cores: verde, azul ou vermelho.

REQ 07 – Arquivo do ficheiro *Excel* referente a um aviso de pagamento e sucessiva atualização no ficheiro *Process Summary*

Descrição: O robô deverá mover o ficheiro *Excel* correspondente ao aviso de pagamento em tratamento conforme o sucesso das validações efetuadas (para o diretório **GoodToGo** no caso dos movimentos do aviso de pagamento estarem de acordo com o esperado e, caso contrário, para o diretório **ToBeChecked**). O robô deverá ainda atualizar os campos de **Status** e **PA Filename** na linha correspondente ao aviso de pagamento em tratamento, no ficheiro **Process Summary**.

3.2.2.2 Solução Dashboard

REQ 01 – Consumo de um ficheiro *Process Summary*

Descrição: A solução deve ser capaz de detetar a chegada de um aviso de pagamento a um diretório configurado, de o consumir e armazenar.

Da análise da Figura 7 considera-se o seguinte fluxo de eventos:

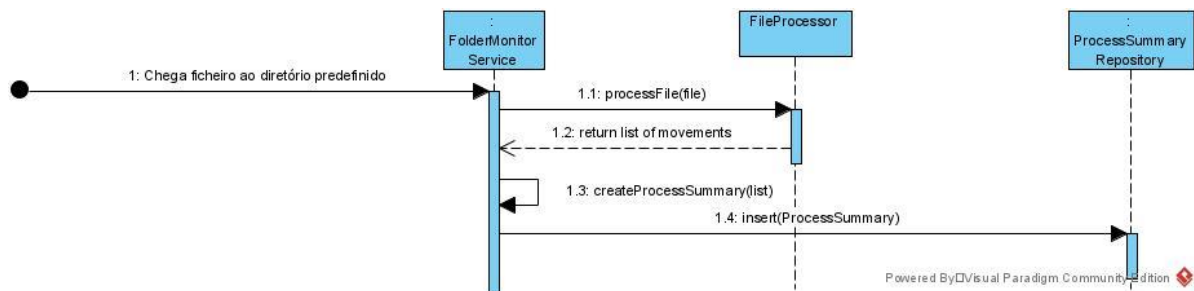


Figura 7 - SD Consumo de um Process Summary

1. É transferido um ficheiro *Process Summary* para o diretório configurado, sendo prontamente detetado pelo *Folder Monitor*;
2. O ficheiro é lido e traduzido para objetos de domínio;
3. Os objetos gerados são persistidos na base de dados Inapa.

REQ 02 – Recolha de Métricas

Descrição: A solução deve ser capaz de produzir métricas relevantes para que os colaboradores Inapa consigam saber quais os principais processos em erro.

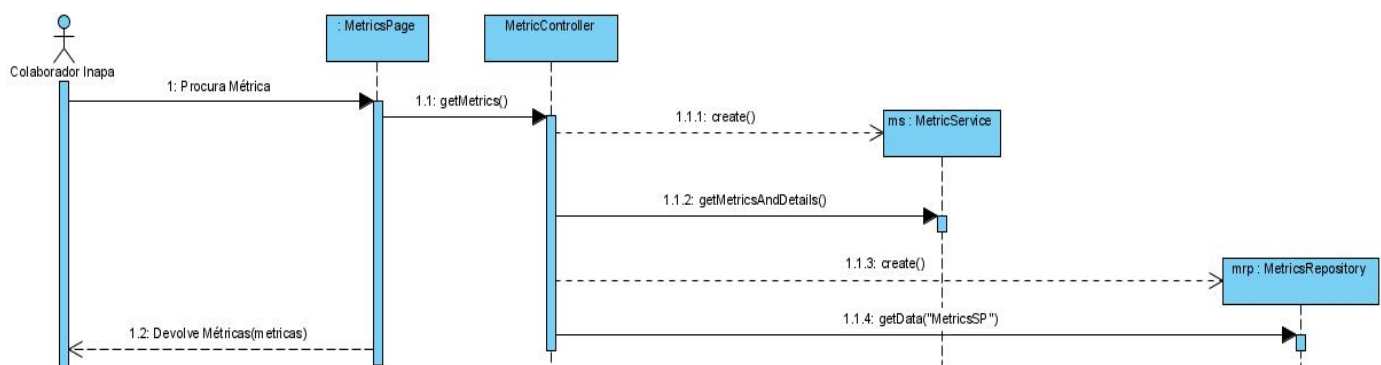


Figura 8 - SD Recolha de Métricas

Da análise Figura 8 considera-se o seguinte fluxo de eventos:

1. O colaborador Inapa acede à página das métricas;
2. A aplicação recolhe e devolve as métricas requisitadas.

REQ 03 – Atualização de Avisos de Pagamento

Descrição: A solução deve ser capaz de permitir que um colaborador marque como corrigido uma ou várias referências de avisos de pagamento.

Pré-condições:

O colaborador acedeu à página **Files** e recuperou os detalhes dos avisos de pagamento.

Fluxo:

Da análise Figura 9 considera-se o seguinte fluxo de eventos:

1. O colaborador selecciona marca uma referência de aviso de pagamento como corrigida;
2. A aplicação marca a referência de aviso de pagamento como corrigida. Caso não existam mais referências de avisos de pagamento do mesmo *Process Summary* em erro, o próprio *Process Summary* é atualizado de modo a ter um estado de validado.

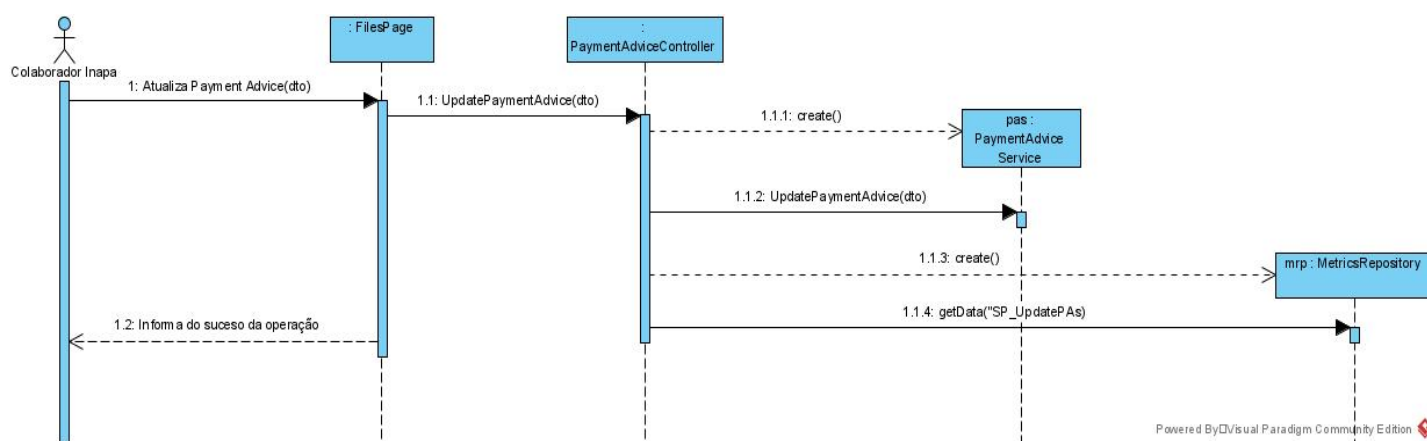


Figura 9 - SD Atualização de Avisos de Pagamento

3.2.3 Requisitos não funcionais

Esta subsecção é dedicada à exposição dos requisitos não funcionais definidos para a solução.

Requisitos não funcionais são um conjunto de especificações que descrevem as capacidades e restrições do sistema considerado (altexsoft, 2022). Na Tabela 3 encontram-se os principais requisitos não funcionais identificados, levantados recorrendo ao modelo FURPS+.

FURPS+	Descrição do Requisito
Suportabilidade (configurabilidade)	O diretório ao qual a solução irá buscar os avisos de pagamento deverá ser configurável
+ (implementação)	A solução deverá proceder à leitura de ficheiros <i>PDF</i> realizando uma extração textual do mesmo. A solução deve ser desenvolvida na linguagem <i>Visual Basic</i> .
+ (<i>design</i>)	Os processos de validação e publicação devem ser independentes.
+ (<i>performance</i>)	A solução deve ser pelo menos 90% mais rápida que a sua contraparte manual

Tabela 3 - Requisitos não funcionais FURPS+

3.3 Análise de valor

O presente subcapítulo destina-se à apresentação da análise de valor no contexto do projeto abordado.

A análise de valor é um método sistemático baseado em processos lógicos que tem como objetivo melhorar o valor de um projeto. A aplicação contínua desta metodologia ao longo da vida útil do projeto permite não só manter um controlo significativamente maior dos riscos financeiros, como também permite uma melhoria na otimização dos recursos disponíveis. Além destas vantagens, a análise de valor visa ainda a agregação de valor ao projeto e a minimização dos riscos associados ao mesmo.

De modo a averiguar não só a criação de valor, mas também para garantir a minimização de consequências relacionadas com a adoção de um produto, deve ser contemplado o processo de invocação.

Este processo de inovação, segundo Peter Koen, pode ser dividido em três partes (Koen, et al., 2014):

- Fuzzy Front End
- New Product Development
- Comercialização

3.3.1 Fuzzy Front End

O **Fuzzy Front End** é uma etapa que pode ser encarada como a raiz da inovação, pretendendo a definição e exploração de conceitos de inovação (Eschberger-Friedl, 2018), e de onde surge um primeiro conceito de produto. As atividades englobadas nesta etapa são normalmente descritas como caóticas, imprevisíveis e desordenadas. (Koen, 2007)

Devido à elevada discrepância entre diferentes empresas e organizações no que toca às definições e termos comuns dos elementos presentes no *Fuzzy Front End*, Koen criou o modelo NCD (**New Concept Development**) (Koen, et al., 2014).

3.3.2 New Concept Development

Este novo modelo, também ele proposto por **Koen**, com o intuito de definir uma linguagem comum para os passos do processo inovação, compreende fundamentalmente três conceitos:

1. O motor – Simboliza a liderança da organização, a estratégia de negócio e a cultura. O motor dá origem a 5 elementos controláveis;
2. Os cinco elementos - Sendo eles a identificação da oportunidade, a sua análise, a génese da Ideia, a seleção da ideia e por fim, a definição do conceito (Figura 10);
3. Outros fatores - Elementos externos incontroláveis, como a concorrência, fatores sociais, económicos, legais, entre outros.

Apesar da representação visual do modelo em questão ser difundida sob a forma de um círculo, dando a entender que o processo em questão não é linear, é proposto que se inicie no elemento **identificação da oportunidade**, e termine na **definição do conceito**. Posteriormente, deve ser seguido um de dois processos: O desenvolvimento de um novo produto, **NPD (New Product Development)**, ou o processo de gestão tecnológica, **TSG (Technology Stage Gate)**.

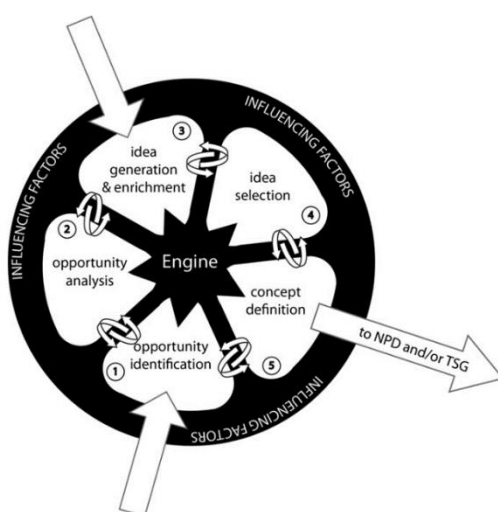


Figura 10 - Modelo NCD (Koen, et al., 2014)

3.3.2.1 Identificação da Oportunidade

Esta é a fase que é mais aceita como sendo a primeira fase do NCD. Deve estar alinhada com objetivos da organização e do negócio, de onde podem surgir as novas oportunidades.

Devido aos valores de excelência e camaradagem que definem a cultura da Inapa, a identificação de oportunidades tanto pode provir de fatores externos, internos ou de uma mistura de ambos. No caso do problema abordado neste documento (1.2), a oportunidade de automatizar o processo de tratamento de avisos de pagamento foi motivada pelo fator interno de alívio de trabalho repetitivo dos colaboradores Inapa que o realizam e pelo fator externo da execução de um processo mais eficiente e barata.

3.3.2.2 Análise da Oportunidade

A análise da oportunidade trata-se de um processo formal e iterativo que permite retirar ilações sobre o valor da oportunidade aferida. Dependendo da importância dada à redução de incerteza sobre a atratividade da oportunidade, podem ser realizadas atividades tais como grupos de foco ou estudos de mercado. Apesar de tudo, a análise de oportunidade tipicamente é composta pelos seguintes passos:

- 1- **Enquadramento estratégico:** Determina como é que a oportunidade em questão se enquadra nos quadrantes das forças, fraquezas e ameaças para a empresa.
- 2- **Avaliação do segmento de mercado:** Análise do mercado e dos seus fatores económicos, culturais, tecnológicos e demográficos.
- 3- **Análise da concorrência:** Determina quem são os maiores concorrentes identificados na segmentação do mercado.
- 4- **Avaliação do cliente:** Determina quais as necessidades dos clientes que não estão a ser eficientemente tratadas atualmente.

Enquadramento Estratégico

De modo a determinar como é que a oportunidade em questão se situa nos quadrantes referidos, foi realizada uma análise SWOT. Esta análise SWOT (*Strengths Weaknesses Opportunities Threats*), que passa por uma ferramenta de planeamento estratégico permite avaliar as principais forças, fraquezas, oportunidades e ameaças. Na realização desta análise consideram-se as forças e fraquezas como tendo uma origem interna, e as oportunidades e ameaças como tendo uma origem externa.

Para a realização desta análise SWOT, deve ser entendido como identificar e distinguir entre o que deve ou não ser considerado para cada quadrante.

- 1- **Forças:** Recursos ou capacidades internas da organização que podem ser usadas para atingir os objetivos propostos.
- 2- **Fraquezas:** Limitações ou defeitos internos da organização que dificultam/impedem que a mesma atinga os objetivos propostos.

- 3- **Oportunidades:** Situações externas favoráveis ao ambiente da organização que permitem que a mesma retire vantagens na perseguição dos objetivos propostos.
- 4- **Ameaças:** Situações externas desfavoráveis ao ambiente da organização, potencialmente nefastas à estratégia da mesma e com impacto da prossecução dos objetivos.

Da aplicação desta análise à nossa oportunidade obtivemos o seguinte resultado, representado na Figura 11.

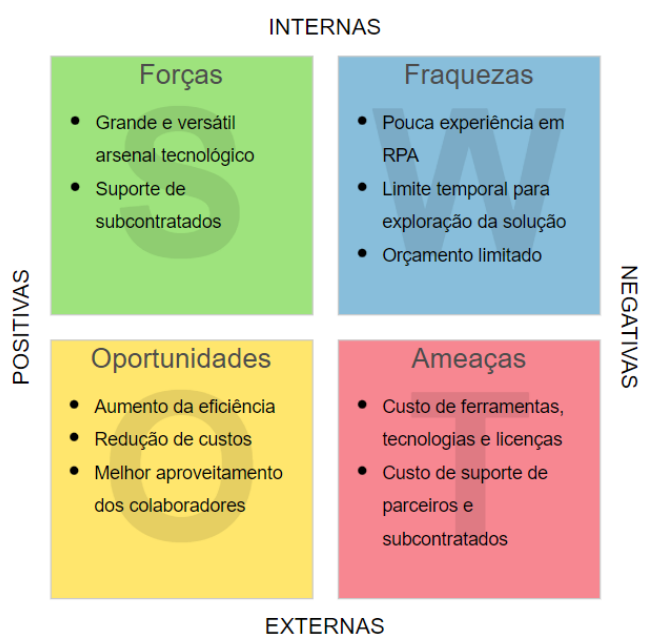


Figura 11 - Análise SWOT - Enquadramento estratégico da oportunidade

Averiguando a análise realizada podemos concluir que as principais forças percecionadas passam pela adaptabilidade e experiência a lidar com um vasto arsenal tecnológico, e pela possibilidade de suporte por parte de subcontratados para esclarecimento de dúvidas e até caso necessário suporte à implementação da solução. Quanto às fraquezas identificadas, sobressai-se a limitada experiência com o ramo tecnológico (**RPA**) subjacente à oportunidade, e os limites temporais e orçamentais para o desenvolvimento da solução. O limite orçamental é sobretudo um entrave uma vez que o desenvolvimento de uma solução implica várias fases, desde o seu estudo, desenho, implementação até à testagem da mesma. Em termos de ameaças externas, temos os preços cuja organização não tem controlo, como o custo de licenças e ferramentas necessárias à implementação da solução, bem como o custo de parceiros requisitados para apoio com questões tecnológicas.

Avaliação de Segmento de Mercado

Considera-se que o segmento de mercado no qual esta oportunidade se insere é **Robot Process Automation**. Tendo isto em conta, foi realizada uma pesquisa com o objetivo de conferir a sua popularidade, bem como os seus fatores económicos, culturais, tecnológicos e demográficos. Da análise da Figura 12 podemos verificar que apesar de uma certa diminuição recente em 2022 e 2023, desde maio de 2015 o interesse por este ramo tem sido renovado a sua popularidade aumentado culminando num pico antes no ano de 2020. Em termos de fatores culturais e demográficos, à exceção de uma grande parte dos países no continente africano e alguns no continente asiático, existe uma distribuição praticamente global quanto à procura por **Robot Process Automation**, como mostra a Figura 13.

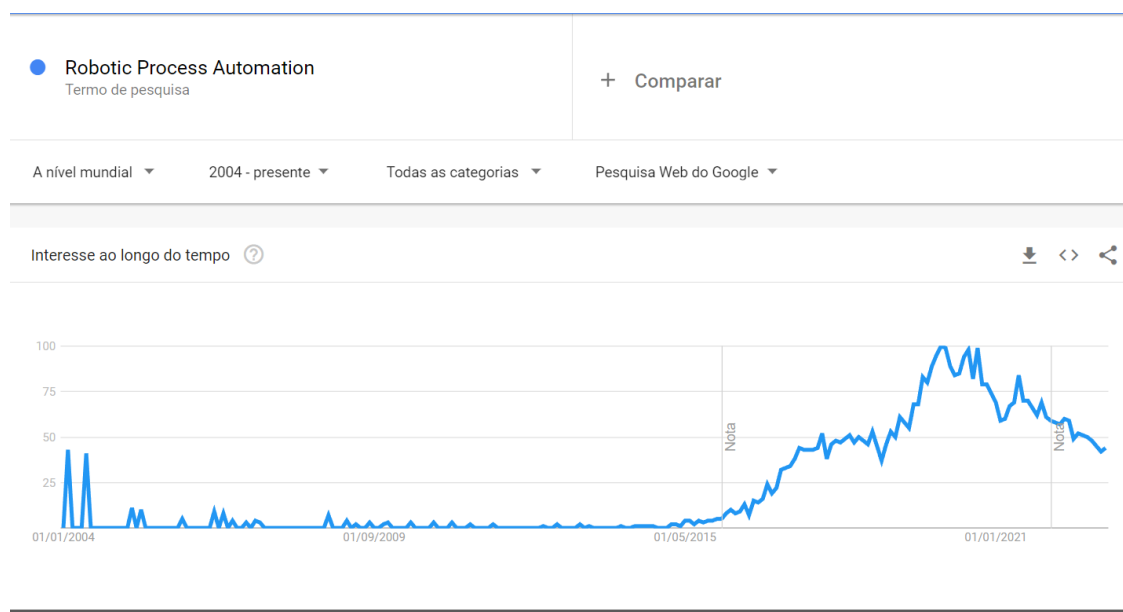


Figura 12 - Tendência de Mercado para Robot Process Automation

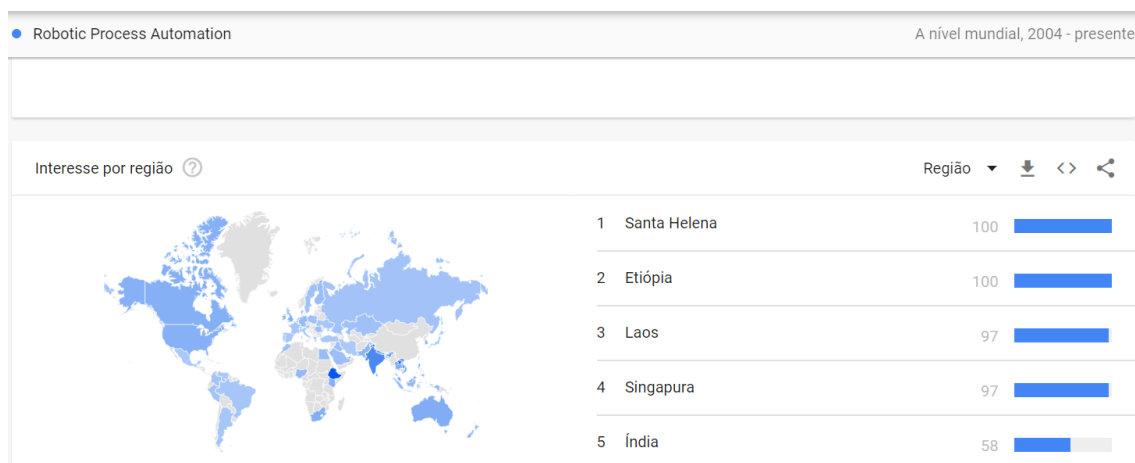


Figura 13 - Tendência Geográfica para Robot Process Automation

Análise da concorrência

Derivado do facto da oportunidade ter sido pensada apenas com o âmbito de uso interno, não havendo intenção de externalizar este uso, não se revelou oportuno a realização de um estudo e análise da concorrência.

Avaliação do cliente

Uma vez que o principal cliente e usufruidor desta oportunidade passa pela própria empresa em si, podemos afirmar que atualmente, nenhum dos produtos existentes e em uso pela empresa satisfaz as necessidades elaboradas na secção correspondente à **Descrição do Problema**.

3.3.2.3 Génese e enriquecimento da Ideia

A génese da ideia é um processo evolutivo que compreende as fases de produção, desenvolvimento e maturação da ideia. Neste processo as ideias podem ser construídas, combinadas, eliminadas, estudadas, recusadas e desenvolvidas.

Tal como referido em (Koen, et al., 2002) os brainstormings são uma das técnicas mais eficazes para a geração de ideias. Como tal, foi seguida essa mesma abordagem, tendo sido realizadas várias sessões de brainstorming com uma equipa interna da Inapa ao longo de 2 semanas, de onde resultaram 2 ideias, cada qual com o seu leque de vantagens e desvantagens.

Como resultado das sessões de brainstorming surgiram as seguintes soluções:

- 1- Implementação de raiz, de um sistema feito à medida, que tratasse da leitura e extração de informação de ficheiros PDF, comunicação com e extração de informação da plataforma SAP, comparação das informações extraídas, e manipulação da plataforma SAP;
- 2- Implementação de uma solução recorrendo a tecnologias de **Robotic Process Automation** que tratará de gerir a leitura, extração e comparação de informação tanto de ficheiros PDF como da plataforma SAP. A solução deve ser ainda capaz de manipular a plataforma SAP.

3.3.2.4 Seleção da Ideia

Após a génese de ideias ter resultado em mais do que uma ideia, passa-se à classificação e análise de cada uma das soluções. Para tal, existe uma multitude de técnicas de decisão (MCDA – Multi-Criteria Decision Analysis).

- AHP – Analytic Hierachy Process;
- SMART- Simple Multi-Attribute Rating Technique;
- TOPSIS – Technique Order of Preference by Similarty to Ideal Solution.

Para a tomada de decisão fomos auxiliadas pelo emprego da técnica MCDA TOPSIS, uma vez que temos de tomar uma decisão entre duas possíveis escolhas, e segundo (Widianta & Rizaldi, 2018) a técnica TOPSIS apresenta uma elevada exatidão na obtenção de resultados face a outras técnicas semelhantes.

TOPSIS

Este método caracteriza-se pela apresentação de duas alternativas artificiais.

- **Alternativa ideal:** a que tiver melhor classificação para todos os atributos considerados;
- **Alternativa ideal negativa:** a que tiver pior classificação para todos os atributos considerados.

Posteriormente o objetivo passa pela avaliação das ideias em questão e pela sua comparação com as alternativas ideal e ideal negativa. O método TOPSIS trata de selecionar a solução mais perto da solução ideal e mais longe da solução ideal negativa, e é composto pelo seguinte conjunto de passos:

- 1- Construção de uma matriz normalizada de decisão (usando as soluções e atributos considerados);
- 2- Construção de uma matriz normalizada de decisão com atribuição de pesos;
- 3- Determinar as soluções alternativas ideal e ideal negativa;
- 4- Calcular as medidas de separação para cada alternativa;
- 5- Calcular a proximidade relativa em relação à solução ideal.

Para aplicarmos esta técnica, foram considerados os seguintes atributos:

- Esforço de implementação;
- Recursos Tecnológicos;
- Custo.

Tendo as ideias a verificar (capítulo Gênese e enriquecimento da Ideia), e os atributos, passaremos à apresentação de uma matriz não normalizada, com os valores e pesos já atribuídos para os atributos de cada solução.

Ideia/Atributo	Esforço de implementação	Recursos tecnológicos	Custo
Ideia desenvolvimento RPA	6	4	9
Ideia software à medida	9	8	7
Peso	0.3	0.3	0.4

Tabela 4 - Tabela Inicial TOPSIS

Uma vez que este método envolve comparações há que realçar as conotações dos atributos escolhidos. Entenda-se que quanto maior o valor de cada atributo, mais negativa a solução (e.g maior custo é menos favorável que um menor custo). Quanto à distribuição dos pesos, o mesmo foi distribuído praticamente uniformemente, uma vez que os três atributos em questão são decisivos para o sucesso de qualquer uma das ideias. No entanto, o custo acaba por se destacar por ser justamente um ponto mais crítico, estando até identificado na análise *SWOT* como uma das principais ameaças.

Matriz de Decisão Normalizada

Para a normalização da matriz de decisão, teremos de nos guiar pelas fórmulas matemáticas adequadas.

Primeiramente devemos aplicar a seguinte fórmula $(\sum x_{ij}^2)^{1/2}$. Esta fórmula representa a raiz quadrada do somatório das células, e deve ser aplicada para cada coluna. De seguida devemos dividir cada célula pelo valor obtido pela fórmula para a coluna respetiva. Da aplicação dos passos referidos, obtemos a **matriz de decisão normalizada**, presente na Tabela 5.

Ideia/Atributo	Esforço de implementação	Recursos tecnológicos	Custo
Ideia desenvolvimento RPA	0.55	0.45	0.79
Ideia software à medida	0.83	0.89	0.61

Tabela 5 - Matriz de Decisão Normalizada

Matriz de Decisão Normalizada com Pesos

O fator peso, previamente definido na Tabela 4 - Tabela Inicial TOPSIS deve ser aplicado, multiplicando o seu valor pelo de cada célula da tabela (à exceção das células de cabeçalho). Da sua aplicação, resulta a Tabela 6.

Ideia/Atributo	Esforço de implementação	Recursos tecnológicos	Custo
Ideia desenvolvimento RPA	0.15	0.14	0.32
Ideia software à medida	0.25	0.27	0.24

Tabela 6 - Matriz de Decisão Normalizada com Pesos

Determinação da Alternativa Ideal

A determinação da solução ideal faz-se utilizando a célula ideal de cada atributo (coluna). Tal deve ser feito após a normalização da matriz e aplicação do fator peso. Após ser feito esse levantamento, determinou-se que a solução ideal, representada por A^* apresenta os seguintes valores.

$$A^* = \{0.15, 0.14, 0.24\}.$$

Determinação da Alternativa Ideal Negativa

Tal como a determinação da solução ideal, a determinação da solução alternativa ideal negativa faz-se utilizando a célula com o pior resultado ideal de cada atributo (coluna). Após ser feito esse levantamento, determinou-se que a solução ideal negativa, representada por A' apresenta os seguintes valores

$$A' = \{0.25, 0.27, 0.32\}$$

Cálculo da Separação da Alternativa Ideal

O cálculo da separação deve ser realizado para cada solução proposta face à alternativa ideal. Para cada linha deve ser aplicada a seguinte fórmula $S_i^* = [\sum (v_j^* - v_{ij})^2]^{1/2}$. A fórmula referida traduz-se na raiz quadrada do somatório do quadrado das diferenças entre o valor de cada célula de uma linha, e o valor correspondente na solução ideal.

$$\text{Separação Ideia RPA} = \sqrt{(0.15 - 0.15)^2 + (0.14 - 0.14)^2 + (0.32 - 0.24)^2}$$

$$\text{Separação Ideia Tradicional} = \sqrt{(0.25 - 0.15)^2 + (0.27 - 0.14)^2 + (0.24 - 0.24)^2}$$

O valor obtido após a aplicação da fórmula pode ser encontrado na Tabela 7 - Separação das soluções Ideal e Ideal Negativa.

Cálculo da Separação das Alternativa Ideal Negativa

Tal como no cálculo da separação da alternativa ideal, para a alternativa ideal negativa deve ser aplicada a mesma fórmula, substituindo, no entanto, para cada célula, o valor a subtrair pelo valor correspondente da alternativa ideal negativa.

$$\text{Separação Ideia RPA} = \sqrt{(0.15 - 0.25)^2 + (0.14 - 0.27)^2 + (0.32 - 0.32)^2}$$

$$\text{Separação Ideia Tradicional} = \sqrt{(0.25 - 0.25)^2 + (0.27 - 0.27)^2 + (0.24 - 0.32)^2}$$

O valor obtido pode ser encontrado na Tabela 7 - Separação das soluções Ideal e Ideal Negativa.

Ideia/Atributo	Separação da Alternativa Ideal	Separação da Alternativa Ideal Negativa
Ideia desenvolvimento RPA	0.08	0.16
Ideia software à medida	0.16	0.08

Tabela 7 - Separação das soluções Ideal e Ideal Negativa

Cálculo da Proximidade Relativa em relação à Solução Alternativa Ideal

O passo final da técnica TOPSIS consiste em, para cada uma das Ideias, aplicar a fórmula $C_i^* = S_i' / (S_i^* + S_i')$. Distinga-se nesta fórmula o símbolo S_i' representante do cálculo da separação face à alternativa ideal negativa. Por outro lado, o símbolo S_i^* representa o cálculo da separação face à alternativa ideal.

Aplicando a fórmula enunciada para as ideias estudadas obtemos os seguintes resultados~

$$\text{Ideia RPA} = 0.16 / (0.16 + 0.08) = 0.67$$

$$\text{Ideia Tradicional} = 0.08 / (0.16 + 0.08) = 0.33$$

Tendo em conta os resultados apresentados a ideia RPA foi preferida em relação à ideia de desenvolvimento de software tradicional.

3.3.3 Valor

O conceito de valor é um conceito subjetivo, podendo ter variadas interpretações dependendo do contexto no qual se insere. Apesar de geralmente o valor de um produto ser também diferente para produtor (organização) e consumidor (cliente), no presente caso, uma vez que o produto se destina a uso interno, será analisada apenas na vertente de consumidor (cliente).

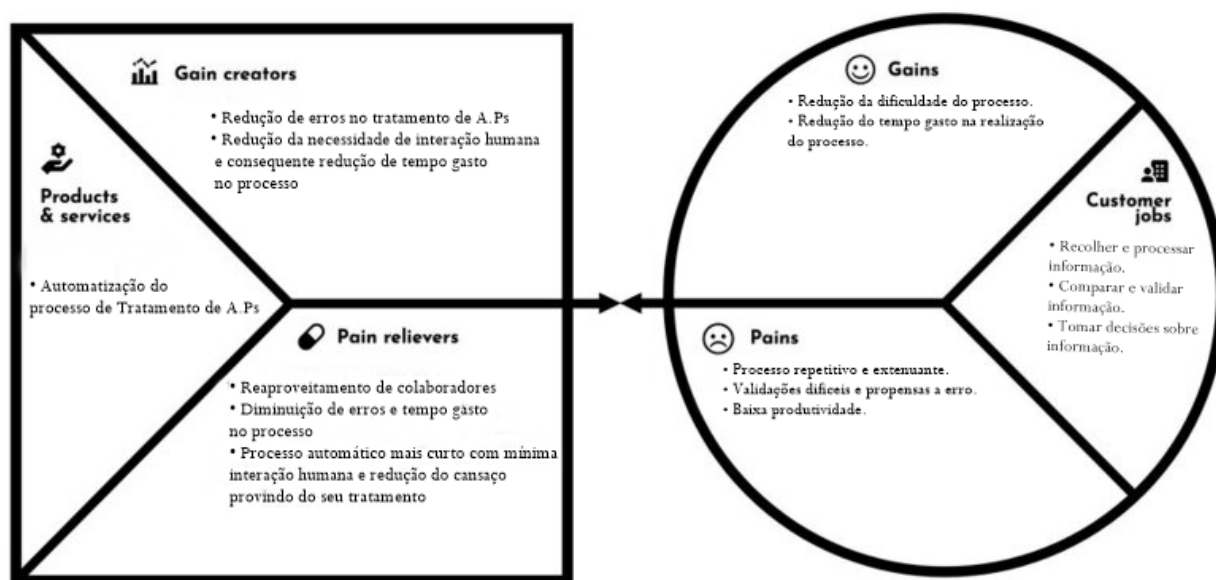


Figura 14 - Canvas Value Proposition (Osterwalder, et al., 2014)

O valor para o cliente pode ser definido como qualquer percepção pessoal de vantagem decorrente da associação do cliente com a oferta de uma organização e pode ocorrer sob a forma de redução de sacrifícios/dores do cliente, da presença de benefícios ou de uma combinação dos dois (Woodall, 2003). Esta **valor percebido** pelo cliente pode ainda ser influenciado por outros fatores como a experiência de compra, a imagem da marca e a própria identidade do produto, além dos benefícios percebidos pelo mesmo (Jiang & Zhang, 2018). De maneira a expressar a relação entre os principais benefícios da concretização da ideia, com as necessidades da organização como cliente, foi elaborada uma **proposta de valor**. Segundo (Kambil & Ari, 1996) uma **proposta de valor** define como é que as funcionalidades de um produto ou serviço se enquadram para colmatar as necessidades dos clientes a ele destinados. A Figura 14 representa a proposta de valor efetuada sobre o modelo proposto por Alex Osterwalder (Osterwalder, et al., 2014).

A proposta de valor identificada na Figura 14 representa a automatização de componentes do processo de Tratamento de avisos de pagamento. Do lado direito, sob forma circular, situa-se a

perspetiva do cliente. Nesta perspetiva é possível observar as tarefas que o cliente tem em mãos (**customer jobs**), que no presente caso correspondem às etapas do processo de tratamento de avisos de pagamentos, as “dores” (**pains**) que atrapalham o cliente na realização dessas tarefas, e os principais ganhos que gostariam de ver refletidos na realização das tarefas. Do lado esquerdo, sob a forma quadrada, observa-se o **mapa de valor**, constituído pelos produtos e serviços oferecidos (**products & services**), quais os principais benefícios (**gain creators**) que a sua utilização traz, e que dores são combatidas (**pain relievers**) pelo seu uso. Apesar de ser composto por duas partes distintas, este modelo deve ser interpretado como um todo, onde os benefícios gerados e as dores combatidas do modelo de valor devem estar associados às dores e aos ganhos expectáveis da perspetiva do cliente.

4 Desenho da Solução

Este capítulo destina-se à apresentação do desenho da solução e suas alternativas.

4.1 Redesenho do Processo

Derivado da solução a desenvolver passar pela automatização de processo de tratamento de avisos de pagamento, o principal desafio de design passou por redesenhar a solução manual. Este desenho não passa apenas pela transição de um processo manual para um processo automático, mas também da sua maturação devido às necessidades da solução. O desenho da solução descrito no presente capítulo já inclui a criação dos tipos de ficheiros *Excel Process Summary* e *Payment Advice Details* e o processamento de vários ficheiros. De modo a auxiliar a compreensão serão expostos como artefactos de design diagramas de processo em notação BPMN. Adicionalmente, tratando-se de um processo com uma certa complexidade, serão apresentados diagramas correspondentes aos subprocessos que forem sendo referidos de modo a aprofundar o detalhe incidido sobre as diferentes partes do processo.

Por fim, há que realçar que foram contempladas **alternativas de design** para o problema em causa, pelo que as mesmas estarão expostas no presente capítulo nas subsecções que se seguem.

4.2 Abordagem

Apesar do processo manual do **tratamento de avisos de pagamento** ser um processo bem definido, não existe uma solução única para redesenhar a abordagem automática. Foram identificadas duas alternativas de desenho válidas para o solucionar, cada qual com os seus prós e contras. As alternativas identificadas foram as seguintes:

1. **Processamento em 2 processos: Preparo dos Avisos de Pagamento e Publicação dos Avisos de Pagamento.** Esta abordagem implica não só a divisão do processo, mas também permite uma implementação faseada da solução pensada. Além desta vantagem, esta solução permite um certo grau de independência entre os processos, podendo dar o benefício de execuções paralelas. No entanto, o processo como todo tem a desvantagem duplicar passos de busca de informação (pesquisa e abertura de ficheiros *Excel*), levando a um impacto no tempo demorado.
2. **Tratamento dos Avisos de Pagamento à Medida:** Esta solução implica que o processo corra como um todo. O processo tem como foco um aviso de pagamento de cada vez, fazendo o

seu tratamento e a sua publicação e apenas depois passando para o aviso de pagamento seguinte. Esta abordagem tem como vantagem teórica um melhor tempo de execução face à abordagem de divisão de processo. Tem como principal desvantagem obrigar a implementação do processo como um todo, e de apenas disponibilizar o ficheiro **Process Summary**, com um resumo sobre os casos corretos e incorretos no final da publicação do último aviso de pagamento. Desta maneira, pode impactar os casos que precisem de ser revistos manualmente, que vão ficar retidos até à finalização do processo.

Tendo em conta que o problema proposto foi pensado para ser dividido em duas fases distintas como referido e explicado nos objetivos (1.3), a hipótese escolhida para a abordagem foi a 1ª - **Processamento em 2 processos**.

4.2.1 Abordagem escolhida Processamento em 2 processos

Como referido, esta abordagem a divisão do processo de tratamento de avisos de pagamento em dois subprocessos. Esta divisão prima pela manutenibilidade pois permite que cada subprocesso fique responsável pela sua cadeia de responsabilidades. Outra vantagem da divisão em subprocessos é que uma alteração num subprocesso não envolve necessariamente alterações no outro. E o mesmo se verifica no comprometimento da solução como um todo. Um erro num subprocesso não implica uma falha necessária no funcionamento do sistema. De seguida serão apresentados os diagramas referentes aos subprocessos desta abordagem.

4.2.1.1 Preparo dos Avisos de Pagamento

Este subprocesso do processo principal é composto pela recolha dos avisos de pagamento por processar, pela validação dos seus movimentos e atualização em conformidade dos ficheiros de **Process Summary** e **Payment Advice Details**.

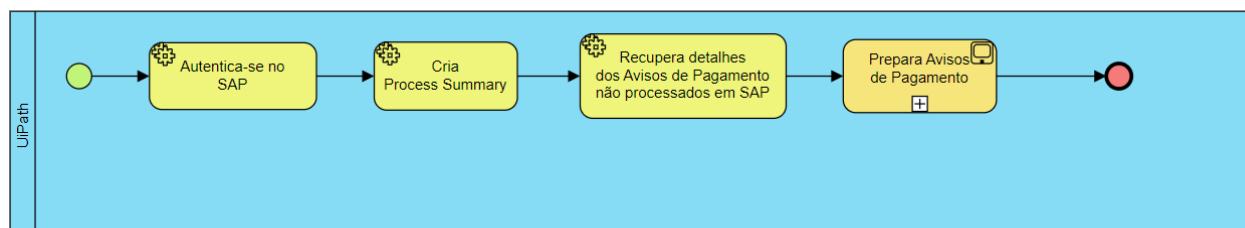


Figura 15 - Diagrama BPMN Preparo de Avisos de Pagamento (Alto Nível)

Através da visualização das Figura 15 e Figura 16 é possível ter tanto uma percepção geral como uma mais detalhada sobre a constituição deste subprocesso. Todas as atividades (*tasks*) representadas são atividades de sistema, realizadas por software. É ainda relevante realçar que o subprocesso **Prepara Avisos de Pagamento** é realizado para todas as referências de pagamento que estiverem no ficheiro **Process Summary**.

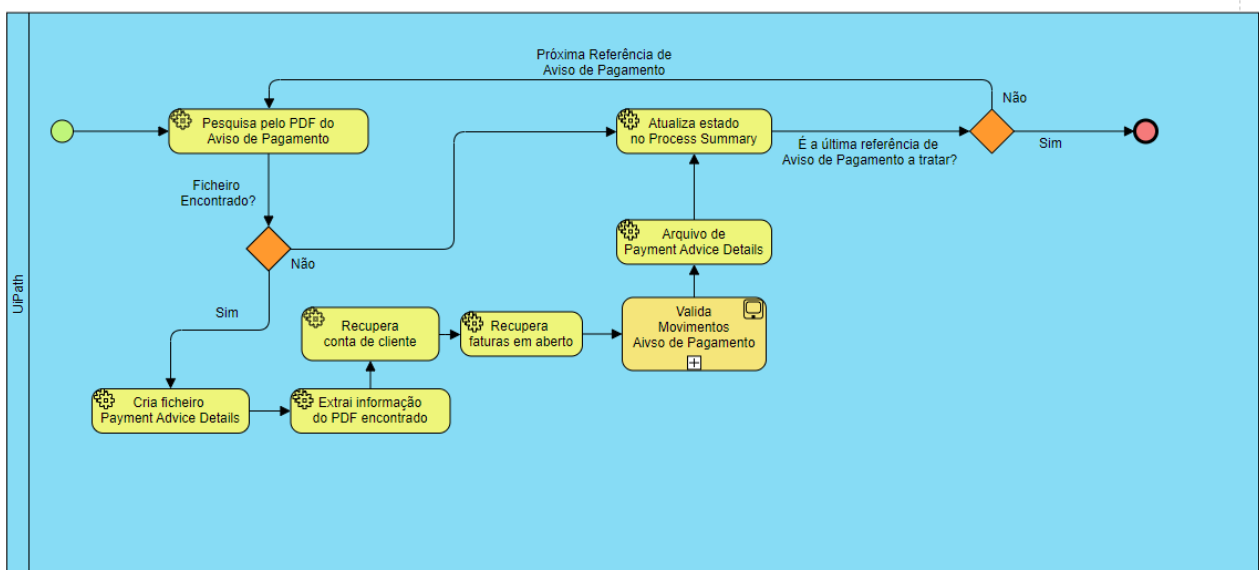


Figura 16 - Diagrama BPMN Preparo de Avisos de Pagamento (Detalhe)

4.2.1.2 Publicação dos Avisos de Pagamento

O segundo subprocesso – **Publicação dos Avisos de Pagamento** – compreende as etapas de validação em **SAP** (validação de aviso de pagamento em aberto, validação de montante igual a zero, validação de número de faturas inseridas), a etapa de publicação do aviso de pagamento e a atualização do ficheiro **Process Summary** para refletir o novo estado do aviso de pagamento.

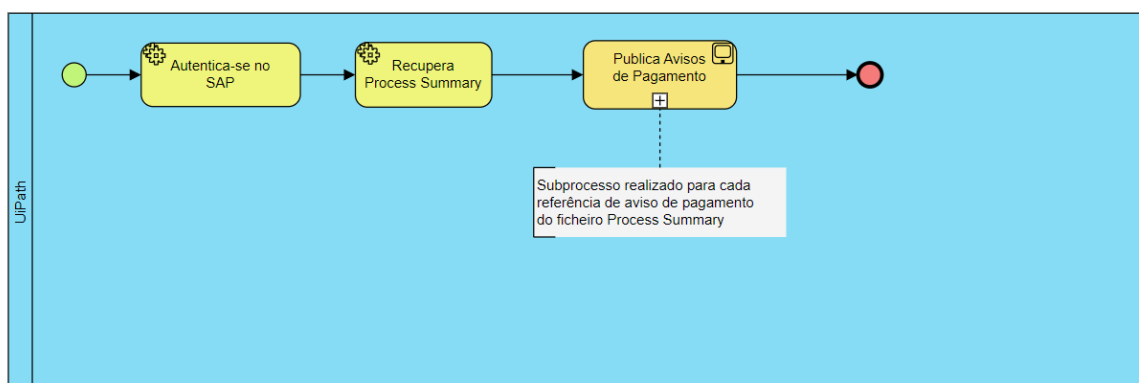


Figura 17 - Diagrama BPMN Publicação de Avisos de Pagamento (Alto Nível)

Como é possível observar na Figura 17, a realização deste subprocesso implica uma nova autenticação no SAP e a recuperação dos ficheiros **Process Summary** que estiverem por processar. De seguida, é executado o subprocesso **Publica Avisos de Pagamento**. Este subprocesso irá validar se a referência do aviso de pagamento está no estado **Confirmed**.

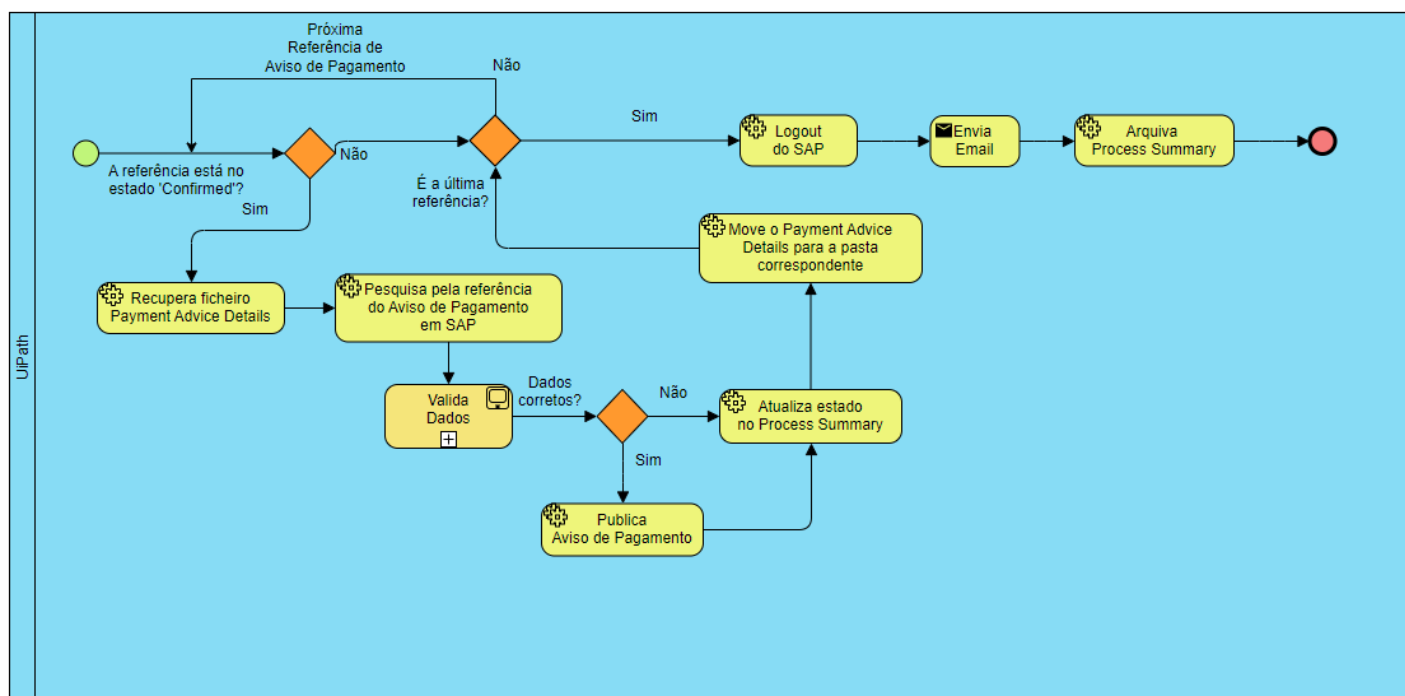


Figura 18 - Diagrama BPMN Publicação de Avisos de Pagamento (Detalhe)

Em caso afirmativo, recupera-se o ficheiro **Payment Advice Details** correspondente, recupera-se os números de fatura no último ficheiro e valida-se contra os números de faturas em **SAP** para esse aviso de pagamento. Caso não seja detetada nenhuma discrepância nas validações o aviso de pagamento deve ser publicado. Posteriormente, atualiza-se o estado no ficheiro **Process Summary** consoante tenha sido encontrado um erro ou não, e move-se o **Payment Advice Details** para a pasta correspondente. Por fim, depois de validadas todas as referências de pagamento no **Process Summary**, efetua-se o **logout** do **SAP**, envia-se um email para a equipa da Inapa responsável pelo tratamento manual a informar que o ficheiro **Process Summary** foi tratado, e arquiva-se o mesmo na pasta **Processed**.

4.2.2 Alternativa Tratamento dos Avisos de Pagamento à Medida

Como indicado em 4.2, nesta alternativa o processo tem como foco um aviso de pagamento de cada vez, fazendo o seu tratamento e a sua publicação e apenas depois passando para o aviso de pagamento seguinte.

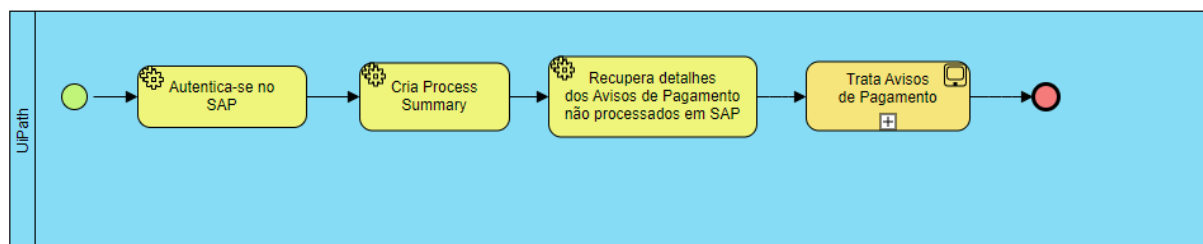


Figura 19 - Diagrama BPMN Alternativa Tratamento dos Avisos de Pagamento à Medida (Alto Nível)

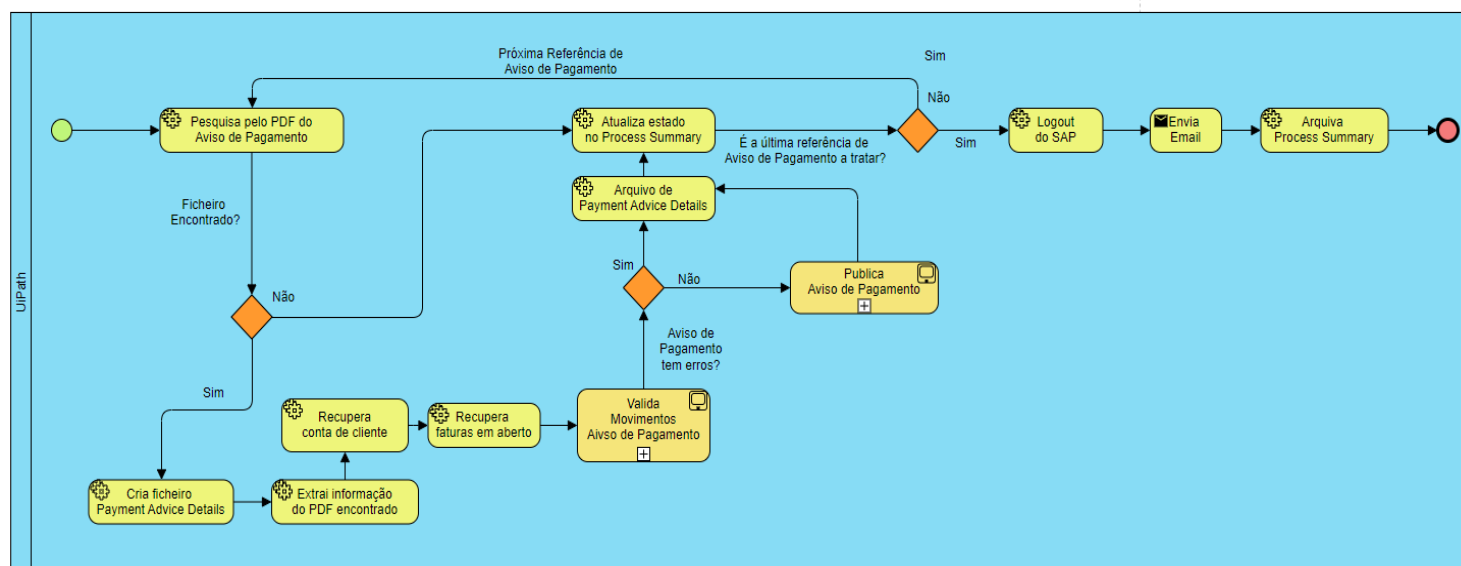


Figura 20 - Diagrama BPMN Trata Aviso de Pagamento (Detalhe)

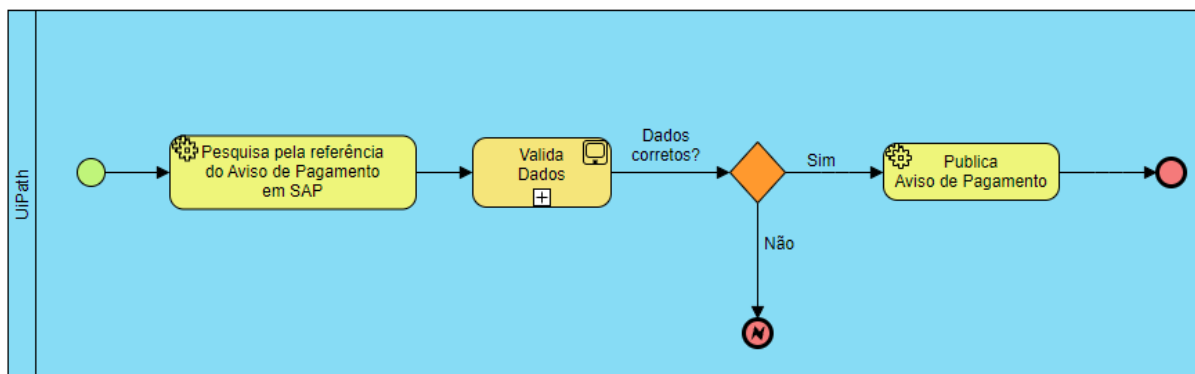


Figura 21 - Alternativa Tratamento dos Avisos de Pagamento à Medida - Publicação de Avisos de Pagamento (Detalhe)

Esta alternativa expressa também um *tradeoff* no que toca a complexidade dos subprocessos quando contrastada com a solução escolhida. Comparando a Figura 16 (da solução escolhida) e Figura 20 (da presente alternativa), é possível verificar que a última apresenta uma maior complexidade por conter, integrado no processo, a chamada do subprocesso de publicação do aviso de pagamento e a atividade responsável pelo arquivo do ficheiro **Payment Advice Details**. No entanto, a complexidade acrescida deste subprocesso permite reduzir em muito a complexidade do outro subprocesso “publicação do aviso de pagamento”, como observável comparando a Figura 18 (da solução atual) com a Figura 21 (da presente alternativa).

4.3 Solução de Métricas de *Process Summaries*

De modo a munir os colaboradores Inapa de ferramentas que lhes permitam rapidamente analisar o estado dos vários ficheiros *Process Summary* que possam apresentar erros, foi desenhada uma solução complementar que integra já parte do ecossistema da Inapa. A solução em questão é fundamentalmente uma solução de *reporting* que, aquando da chegada de ficheiro *Process Summary* a um diretório predefinido que está a ser monitorizado, o lê e insere os dados na base de dados interna da Inapa. Por outro lado, os colaboradores internos fazem uso da solução ao consultar o componente de **Dashboard** onde são apresentadas métricas relevantes sobre os ficheiros processados.

A solução encontra-se explorada em maior detalhe nas Figura 22, Figura 23, Figura 24 e Figura 25 que procuram ilustrar respetivamente as vistas de desenvolvimento (Figura 22, Figura 23 e Figura 24) e física (Figura 25) dos seus diferentes constituintes.

Nesta etapa foram também contempladas **alternativas de design**, expostas no subcapítulo correspondente (4.3.3).

4.3.1 Vista Desenvolvimento

Esta vista, segundo o modelo arquitetural 4+1, descreve o sistema do ponto de vista do programador, sendo comum a sua representação ser feita através de Diagramas UML de Componentes (Jayawardene, 2021). Como tal, encontram-se apresentados três diagramas de componentes. O primeiro é referente à composição geral do sistema, representado pela Figura 22, e os dois restantes dizem respeito aos componentes de ***Dashboard backend*** e ***Folder Monitor***.

4.3.1.1 Diagrama de Componentes da Solução Complementar

O diagrama de componentes exposto na Figura 22 retrata o sistema desenvolvido como um todo, possibilitando a vista dos seus componentes individuais e das interações entre os mesmos. Analisando o artefacto em questão, entende-se que a solução é composta por três componentes distintos:

- Uma aplicação ***Dashboard***, composta pelos componentes de *frontend* e *backend*, que comunicam entre si através da API disponibilizada pelo componente *backend*. Este último componente consome ainda uma API disponibilizada por uma base de dados externa ao sistema da Inapa;
- Uma aplicação independente ***Folder Monitor***, dedicada ao processamento dos diferentes ficheiros *Process Summary*, que também comunica através de uma API com a base de dados da Inapa.

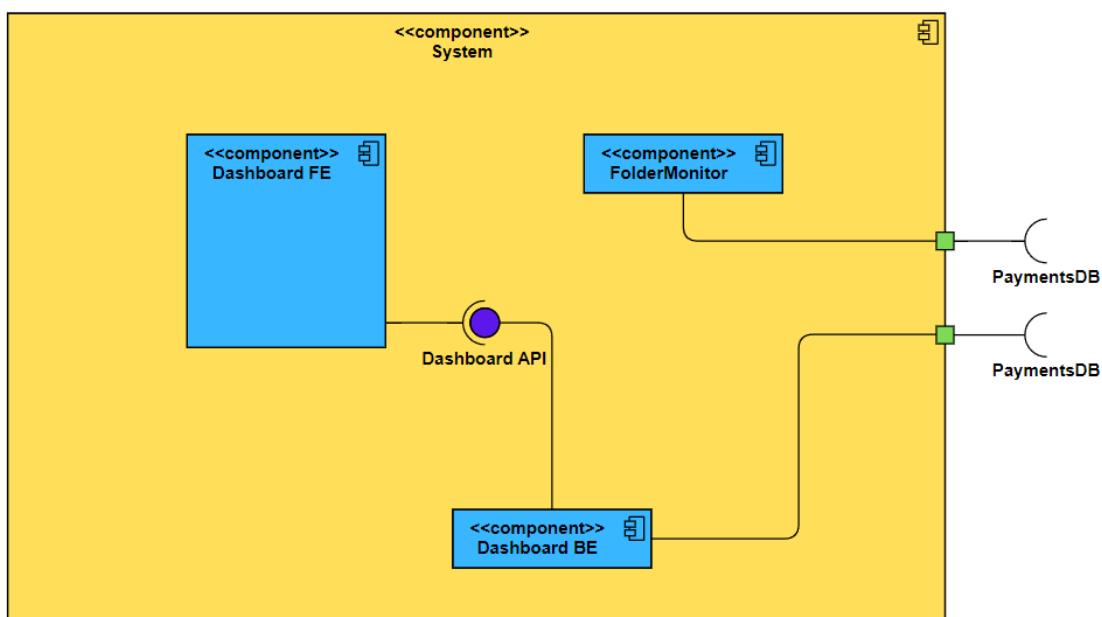


Figura 22 - Diagrama de Componentes da Solução Complementar

4.3.1.2 Diagrama de Componentes *Dashboard (backend)*

O diagrama de componentes exposto na Figura 23 detalha o componente *Dashboard backend* da solução. A arquitetura do componente em questão foi inspirada no padrão **Layered Architecture**, caracterizado por ser um método tradicional de design de software que organiza os vários componentes em camadas horizontais, onde cada uma desempenha o seu papel (Walpita, 2019). No presente caso, todas as camadas utilizadas podem ser consideradas **camadas fechadas** uma vez que um pedido passa de uma camada para a imediatamente a seguir, seguindo um percurso previsível. O desenho desta solução envolve a existência de 3 fulcrais camadas:

- **Controllers:** Responsável por ser o ponto de contacto da aplicação ao disponibilizar uma API REST para receção dos pedidos dos clientes;
- **Services:** Responsável pela comunicação entre a camada de **Controllers** e **Repositories**. Esta camada é também responsável por elementos da lógica de negócio e quaisquer validações que sejam necessárias;
- **Repositories:** Responsável pela comunicação com a base de dados da Inapa de modo a persistir e recolher dados necessários.

Da análise da figura supramencionada ainda é possível distinguir que existe referências a objetos de domínio (representados por Domain) e objetos DTO (Data Transfer Objects), dedicados à comunicação entre as camadas.

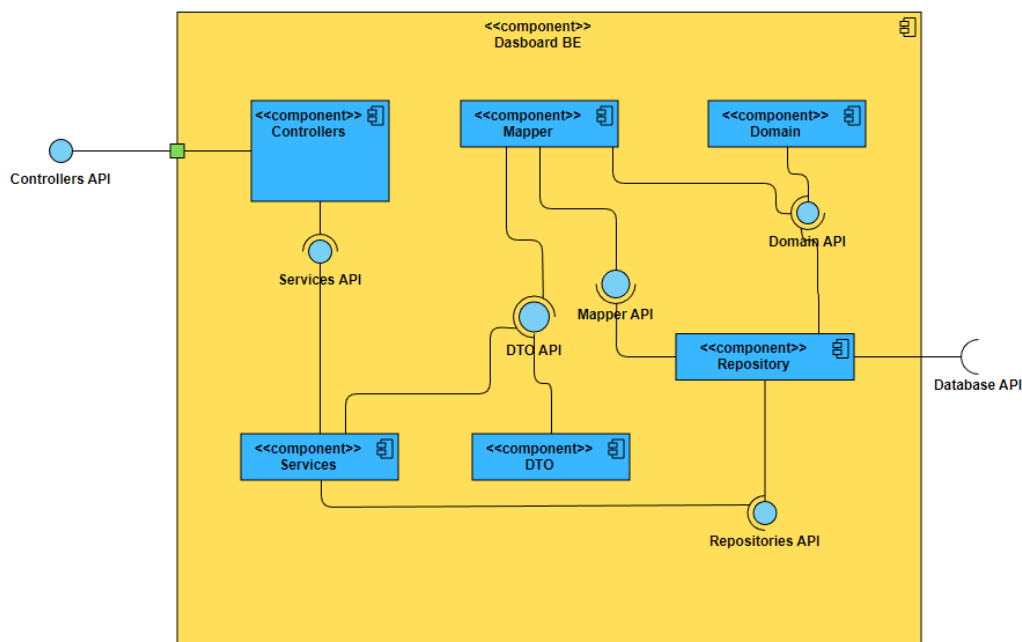


Figura 23 - Diagrama de Componentes - Backend da Dashboard

4.3.1.3 Diagrama de Componentes *Folder Monitor*

O diagrama de componentes da Figura 24 apresenta componente *Folder Monitor* da solução. Este componente foi pensado com o objetivo de realizar um controlo intermédio sobre os ficheiros *Process Summary*, averiguando para cada movimento neles presentes, se o mesmo apresenta algum tipo de erros ou se se encontra preparado para a publicação. A arquitetura do componente em questão é semelhante à previamente explicada em 4.3.1.2, com a ligeira de diferença de não ter a camada de **Controllers**. Esta decisão justifica-se uma vez que a aplicação em questão, tratando-se de uma aplicação consola, não disponibiliza qualquer ponto de contacto para interação com outras aplicações.

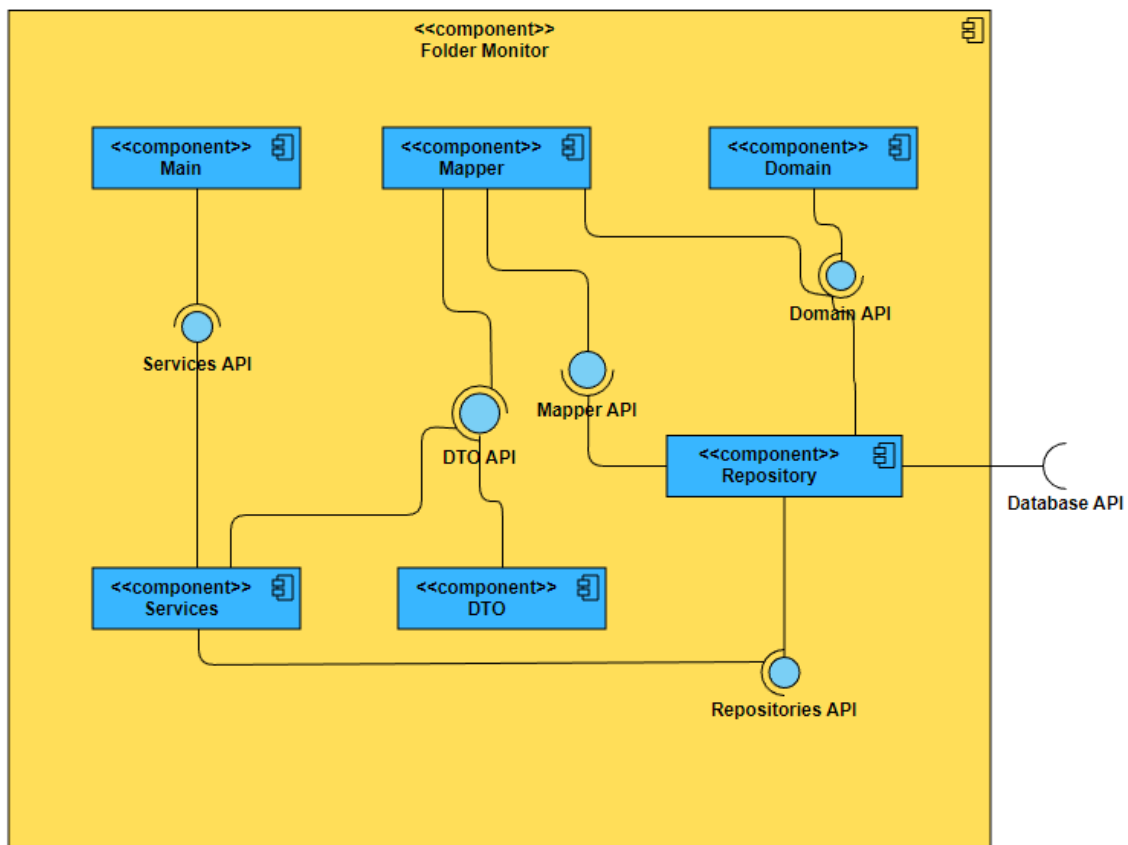


Figura 24 - Diagrama de Componentes - Folder Monitor

4.3.2 Vista Física

Esta vista, segundo o modelo arquitetural 4+1, descreve o sistema do ponto de vista do engenheiro de sistemas, pretendendo representar a topologia dos componentes de sistema, bem como as conexões entre eles (Jayawardene, 2021). Da interpretação do diagrama de implantação (Figura 25) usado para o efeito, depreende-se que ambos os artefactos correspondentes ao *Dashboard* vão ser implantados num servidor **Apache**, numa máquina **AWS**, onde também está alojado o servidor interno de base de dados da **Inapa**. Por outro lado, o componente **Folder Monitor** será implantado numa máquina interna **Inapa**, a mesma onde a solução UiPath está alojada. Desta maneira, é possível o componente **Folder Monitor** monitorizar o diretório para onde irão os ficheiros **Process Summary** depois de validados pelo processo do UiPath, sendo depois redireccionados para o diretório mais adequado correspondente. Por fim, há que realçar que é usado um aparelho computacional pelos diferentes colaboradores internos de modo a aceder ao componente *Dashboard FE*.

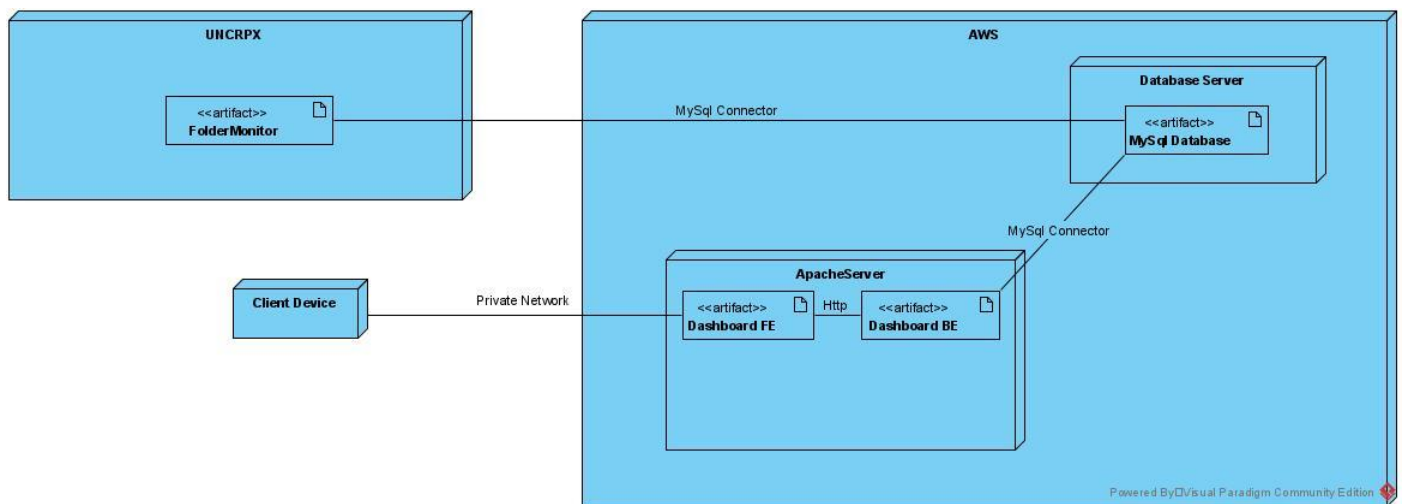


Figura 25 - Diagrama de Implantação Solução Complementar

4.3.3 Alternativas de Design

Foi considerada como alternativa de design à solução desenvolvida a fusão dos componentes de **Dashboard backend** e **Folder Monitor** e o uso de apenas uma interface de comunicação com a base de dados como observável na Figura 26.

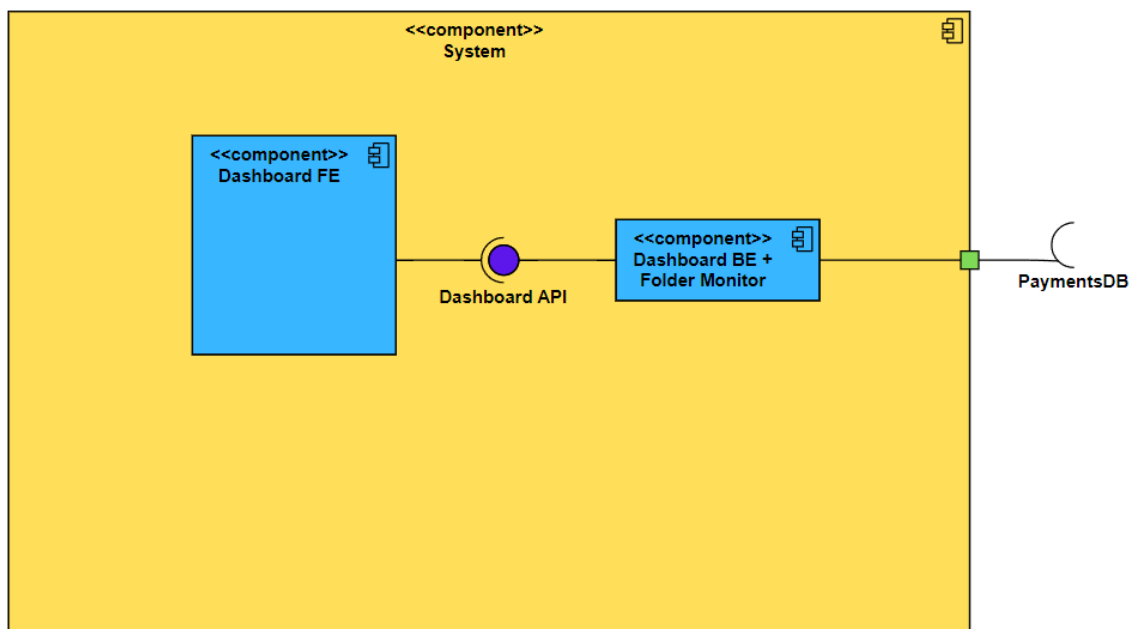


Figura 26 - Alternativa de Design Solução Dashboard

No entanto esta solução alternativa foi preterida em detrimento da escolhida por se considerar que havia uma grande mistura de responsabilidades, que fariam com que o componente resultante da fusão do **Dashboard** com o **Folder Monitor** apresentasse tanto comportamentos de uma WEB API como de uma aplicação consola. Além do mais, seria obrigatório que esse componente estivesse alojado na máquina *Inapa* onde o processo UiPath seria executado.

5 Implementação da Solução

A análise e o desenho de um sistema culminam na sua implementação. Este capítulo é dedicado à implementação e explicação da solução exposta nos capítulos precedentes. Serão apresentados os artefactos de código e justificadas as decisões mais importantes tomadas.

5.1 Automatização do Processo de Tratamento dos Avisos de Pagamento.

Derivado da solução a desenvolver passar pela automatização de processo de tratamento de avisos de pagamento, o principal desafio de design passou por redesenhar a solução manual. Este desenho não passa apenas pela transição de um processo manual para um processo automático, mas também da sua maturação devido às necessidades da solução. O desenho da solução descrito no presente capítulo já inclui a criação dos tipos de ficheiros *Excel Process Summary* e *Payment Advice Details* e o processamento de vários ficheiros. De modo a auxiliar a compreensão serão expostos como artefactos de design diagramas de processo em notação BPMN. Adicionalmente, tratando-se de um processo com uma certa complexidade, serão apresentados diagramas correspondentes aos subprocessos que forem sendo referidos de modo a aprofundar o detalhe incidido sobre as diferentes partes do processo.

Analisando a Figura 27, vemos que a solução desenhada é composta por 4 secções relevantes, dedicadas a fases e responsabilidades diferentes do processo:

1. Inicialização
2. Recuperação da Transação
3. Processamento da Transação
4. Finalização

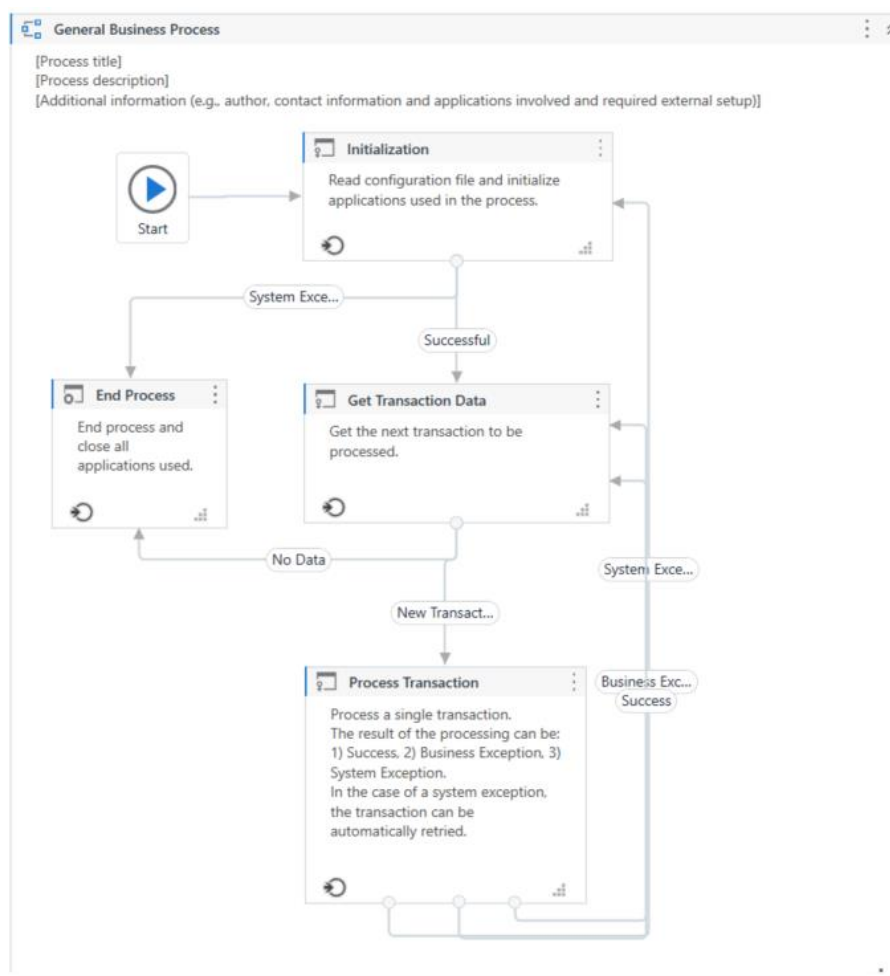


Figura 27 - Vista Geral UiPath Processo de Tratamento dos Avisos de Pagamento

5.1.1 Inicialização

A inicialização é caracterizada pela responsabilidade de garantir o começo do processo num estado coerente. Para tal, como observável na Figura 28 são terminadas as aplicações que estiverem a ser executadas e que venham ser usadas pelo *UiPath* para completar o processo. Apesar da única aplicação usada pelo *UiPath* na realização deste processo ser o **SAP**, como no futuro poderão ser adicionadas mais aplicações necessárias (por exemplo a aplicação do **Outlook mail**) as atividades responsáveis pelo término e iniciação das aplicações foram distribuídas para diferentes fluxos. A Figura 29, que transparece o detalhe do fluxo “InitAllApplications” da Figura 28 mostra que é feito uso de uma das grandes vantagens do *UiPath* - a integração com outras aplicações – para invocar uma atividade pré-existente de login na aplicação SAP, à qual apenas de fornecer o caminho de alojamento da aplicação na máquina e o nome da conexão **SAP** usada. Posteriormente são recuperadas as credenciais configuradas e usadas para efetivar o login.

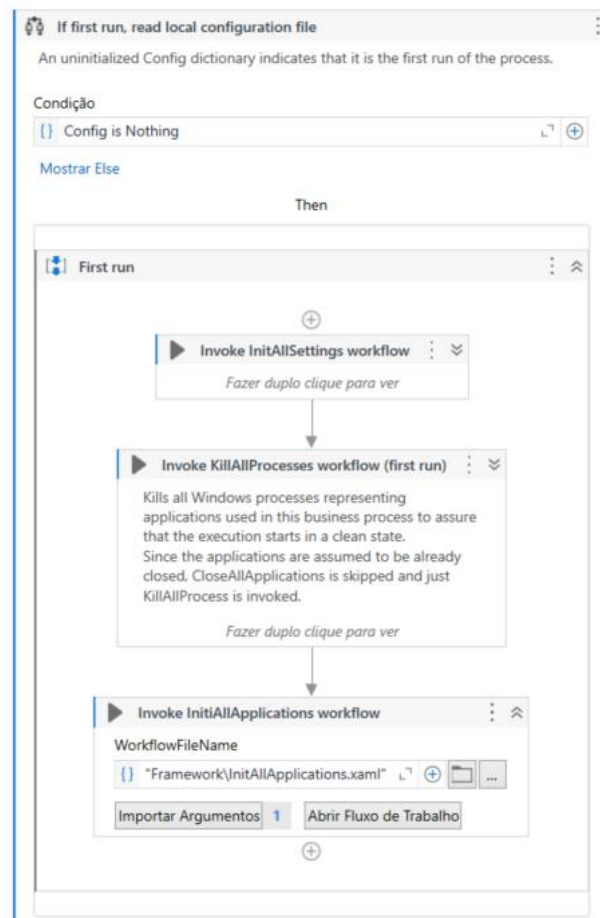


Figura 28 - Inicialização

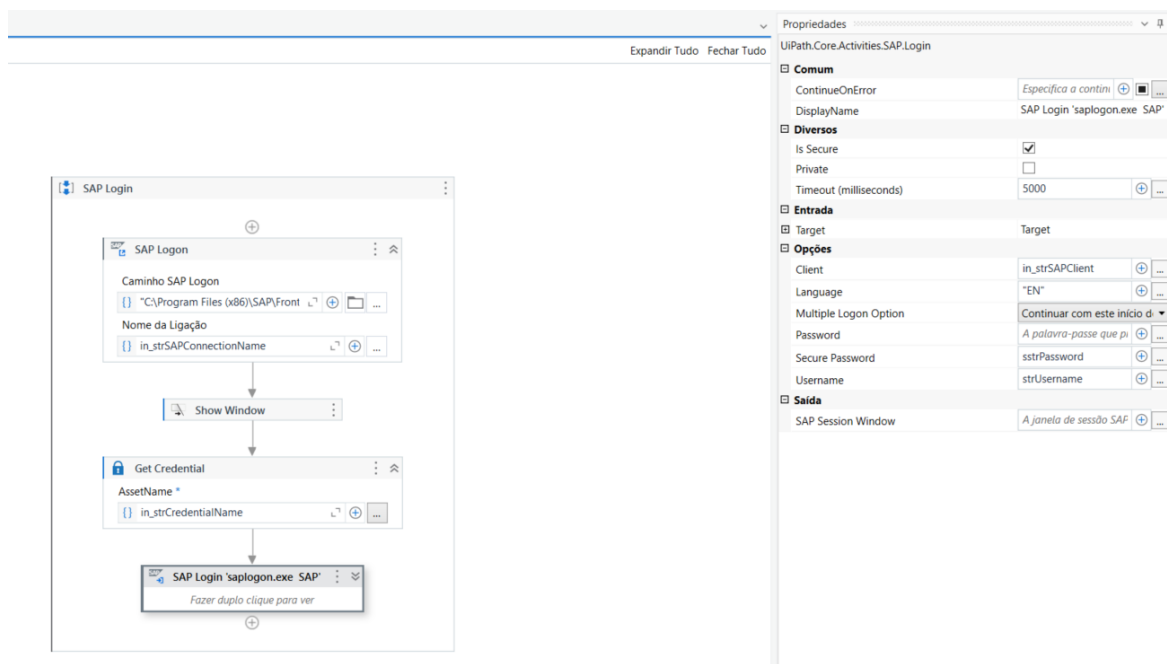


Figura 29 - Inicialização Aplicação SAP

5.1.2 Recuperação de Transação

Esta subprocesso é responsável pela extração de informação com as referências e detalhes dos avisos de pagamento a tratar do SAP e criação do respetivo ficheiro **Process Summary**. Fica ainda com a responsabilidade de determinar a próxima transação que deverá ser tratada. De maneira a extrair a informação necessária do SAP, o UiPath faz uso da transação **FEBA** existente, executando-o com dois conjuntos de parâmetros predefinidos. De modo a ser perceptível os fluxos gerais deste subprocesso, além de figuras representativas do código serão também exibidos diagramas que auxiliem a sua compreensão. A Figura 30 transmite um entendimento geral sobre o fluxo primário do processo **Get Transactions**, sendo ele:

1. Verificação de que estamos a tratar a primeira transação. Esta verificação é relevante, uma vez que se não for a primeira transação, é escusado de proceder a uma nova leitura do *Process Summary*, uma vez que a informação extraída se encontra numa tabela, em memória no UiPath;
2. No caso de ser a primeira vez, é validado se já existe um ficheiro **Process Summary** por processar. Em caso negativo, procede-se à extração de informação do SAP através da transação **FEBA** e à criação do respetivo **Process Summary**. Este ficheiro vai ser posteriormente recuperado e carregado para uma tabela em memória;
3. Posteriormente, é validado se a tabela tem alguma transação e, em caso afirmativo, se ainda faltam serem processadas. Caso a resposta seja negativa a qualquer uma destas duas questões, o valor do elemento transação deve ser marcado como vazio, servindo depois como condição de paragem da aplicação. No entanto, no caso de pelo menos uma transação ainda não ter sido processada, a mesma é recuperada, sendo depois processada em outro fluxo (5.1.3).

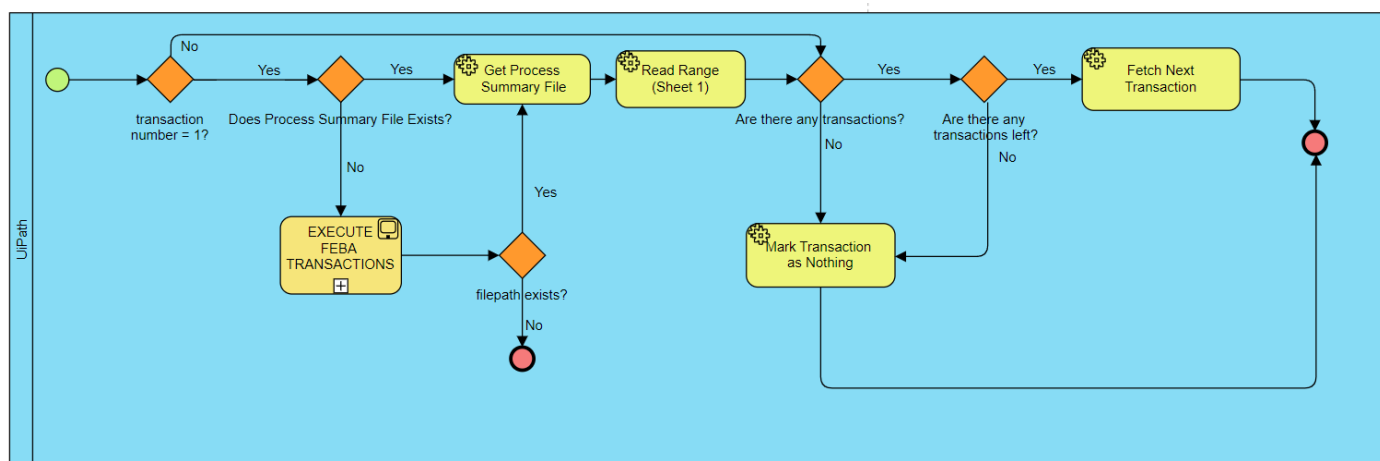


Figura 30 - Fluxo Get Transactions (BPMN)

O subprocesso “**Execute FEBA Transactions**” da Figura 30, encontra-se detalhado na Figura 31. Este processo está responsável pela invocação da atividade “**Executes FEBA Transaction**” com um conjunto de valores predefinidos, validação dos resultados, e caso existam, criação do ficheiro *Process Summary* (no diretório padrão configurado - *RPA\ProcessSummary\ddmmmyyyy\ProcessSummary*). Posteriormente, é realizada a mesma atividade com um segundo conjunto de parâmetros, onde os resultados são adicionados aos já presentes no *Process Summary* (no caso de existirem). Caso não haja resultados em nenhuma das invocações da atividade, o processo regista esse acontecimento e termina.

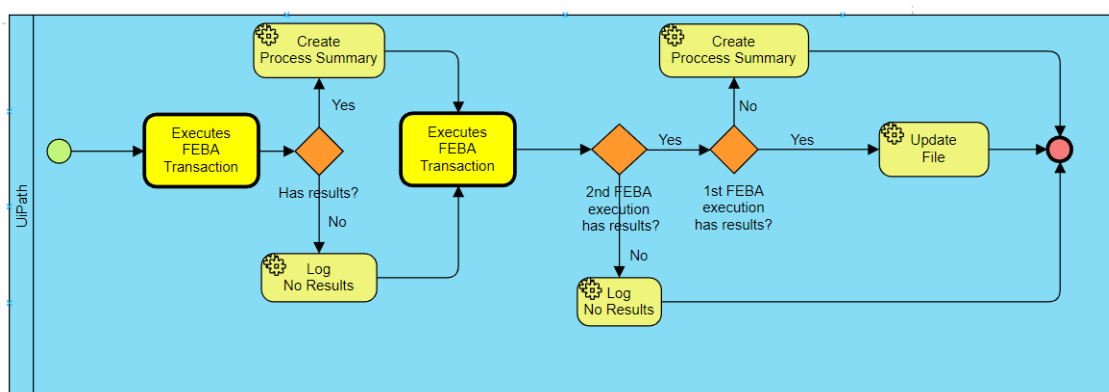


Figura 31 - Diagrama *Execute Feba Transactions*

Quanto à atividade “**Executes FEBA Transaction**”, invocado duas vezes no diagrama da Figura 31, encontra-se detalhado nas Figura 32 e Figura 33 representativas do conjunto de instruções UiPath correspondentes à sua implementação. O seguinte fluxo de baixo nível descreve de forma sucinta as etapas seguidas na implementação:

1. O robô UiPath deve, através do menu apresentado pelo SAP, selecionar a caixa de texto de pesquisa, localizada no canto superior esquerdo, escrever na caixa de texto o nome da transação a realizar (*FEBA_BANK_STATMT*) e simular um clique na tecla de *ENTER*.
2. Após a devolução do menu correspondente para o ecrã por parte da plataforma SAP, o robô deve preencher os campos **Company Code**, **House Bank** e **AccountID** com valores previamente estipulados, bem como clicar no botão de verificação dos valores introduzidos, e por fim pelo botão responsável pela pesquisa.
3. Após a plataforma SAP apresentar a lista total de avisos de pagamento não processados automaticamente, o robô deve selecionar a opção no SAP de layout para o robô, filtrando de seguida os resultados para conterem apenas como **Business Partner** os clientes que foram considerados no âmbito do projeto.
4. De seguida, o robô deve extrair os dados filtrados apresentados pelo SAP, fazendo uso da atividade **UiPath Extract Structured Data** (UiPath, s.d.) para uma tabela em memória no **UiPath**. A essa tabela é acrescentada ainda a coluna **Status** com o valor por defeito de “**Open**”.

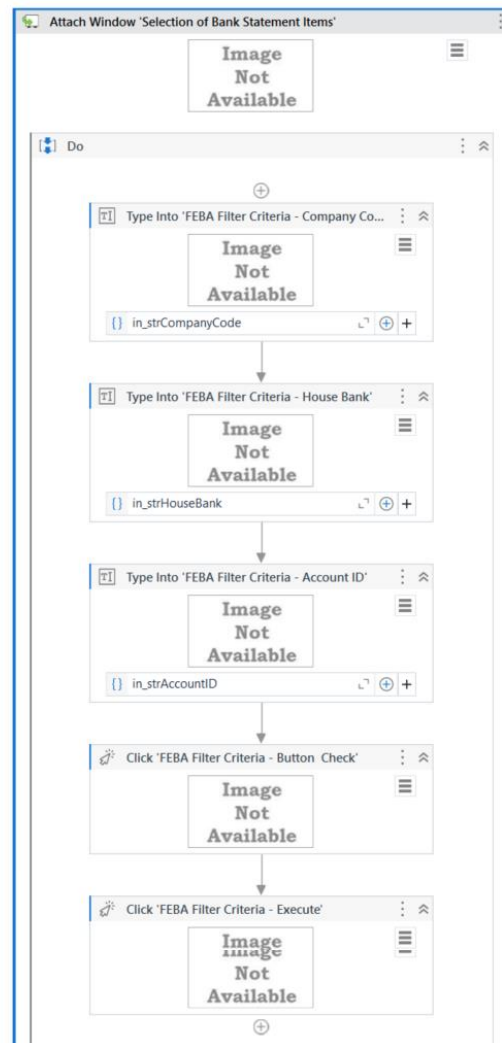


Figura 32 - UiPath Preparação Transação FEBA



Figura 33 - Execução Transação FEBA

5.1.3 Processamento da Transação

Após a definição da transação a tratar, é o presente fluxo que está incumbido de a processar. De um modo mais generalizado, a perspectiva *macro* da Figura 34 demonstra que os passos primordiais desta fase passam pela validação do estado da transação, pela aplicação do processo de cliente respetivo ao valor do campo **business partner** extraído previamente, e por fim, pela atualização do estado para o respetivo resultante do processamento. Apesar de cada cliente ter um modelo predefinido consoante os ficheiros PDF de aviso de pagamento, a estrutura dos subprocessos entre os diferentes clientes é muito semelhante.

As Figura 35 e Figura 36 apresentam a generalidade das validações até à aplicação do processo principal para o cliente em questão. A Figura 35, mostra que esse componente é somente responsável por decidir se poderemos chamar os restantes subprocessos do cliente. Esta decisão é feita baseando-se na sua habilidade de encontrar o ficheiro PDF correspondente à transação em tratamento, cujo nome é composto pelo cliente, data e montante. Em caso de o ficheiro não ser encontrado, é preenchida uma variável com o valor adequado (**PDF Not Found**), servindo depois para atualizar o valor da coluna **status** da presente transação no *Process Summary*. Por sua vez, a Figura 36 trata das validações para verificar que o ficheiro PDF em questão corresponde ao modelo montado para o presente cliente, validando a existência do nome indicativo do cliente e de informações relativas ao pagamento.

A Figura 37 detalha o fluxo do processamento de uma transação para um dado cliente, sendo seguidos os seguintes passos:

1. Extração do montante do PDF do aviso de pagamento;
2. Comparação com o montante em sistema (recuperado do SAP);
3. Extração de outras informações do PDF (número do aviso de pagamento, número do movimento, data do pagamento);
4. Recuperação do código de conta, através da realização da transação VF03 em SAP;
5. Extração dos movimentos presentes no PDF de aviso de pagamento e criação de um ficheiro *Excel* representativo do aviso de pagamento;
6. Recuperação das faturas em aberto em sistema através da realização da transação FBL5N em SAP;
7. Comparação dos movimentos e cálculo de diferenças entre os montantes em sistema e montantes do PDF, diferença entre o valor pago e o valor calculado correspondente ao que é suposto ser efetivamente pago e cálculo do desconto aplicado;
8. Cálculo do desconto que deveria ser aplicado com base na data da fatura e data do pagamento;
9. Aplicação de cor às células do ficheiro *Excel* criado de aviso de pagamentos consoante os valores calculados das diferenças.

De seguida, serão explorados cada um dos principais subprocessos mencionadas Figura 37.

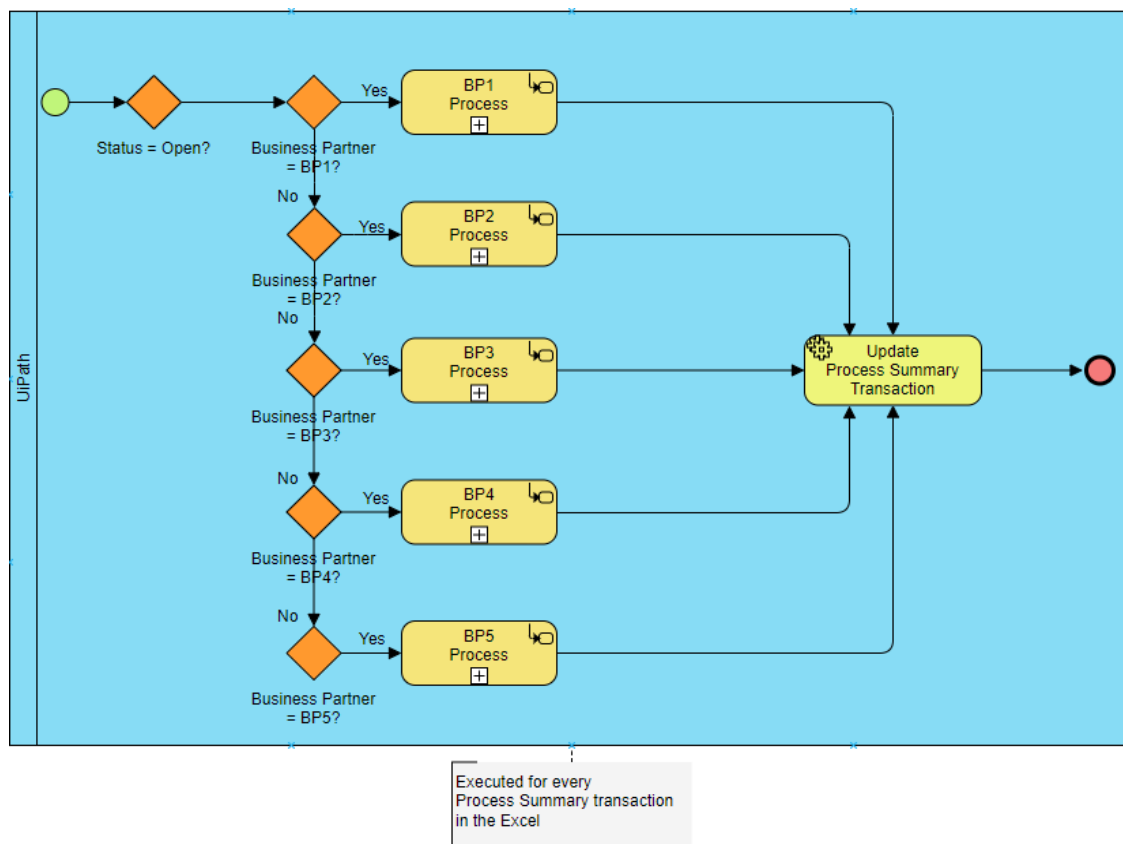


Figura 34 - Processamento de Transação UiPath

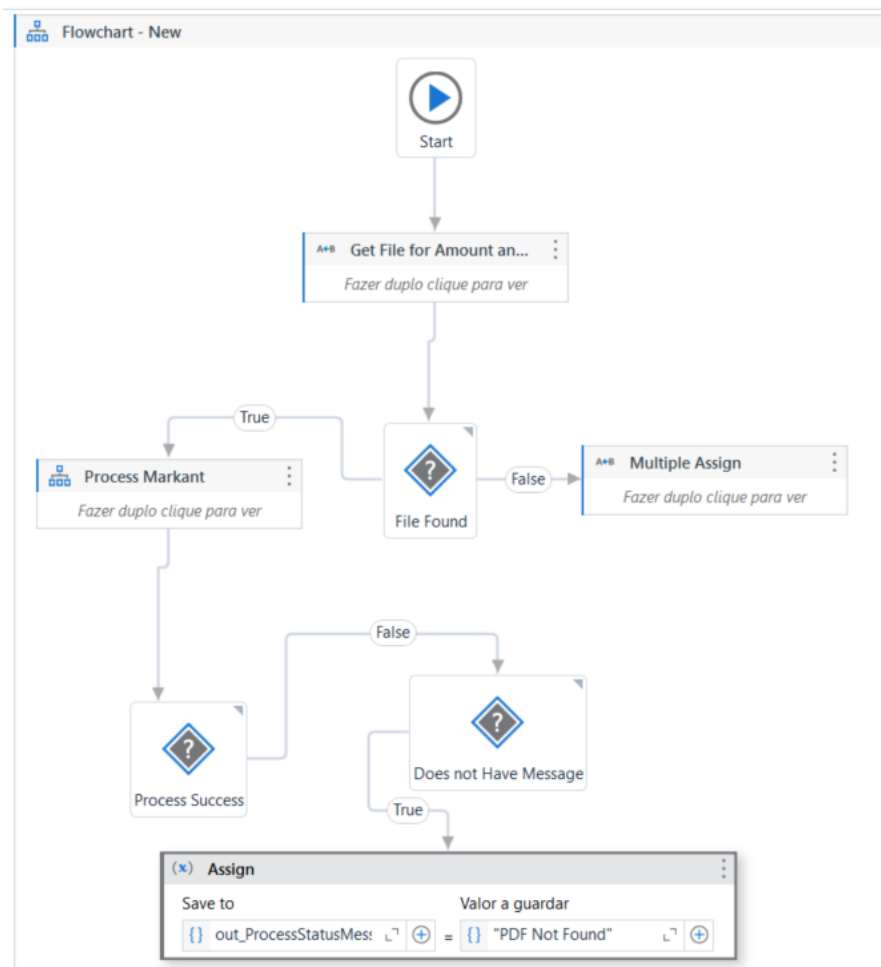


Figura 35 - Decisão de Aplicação de Processo do Cliente

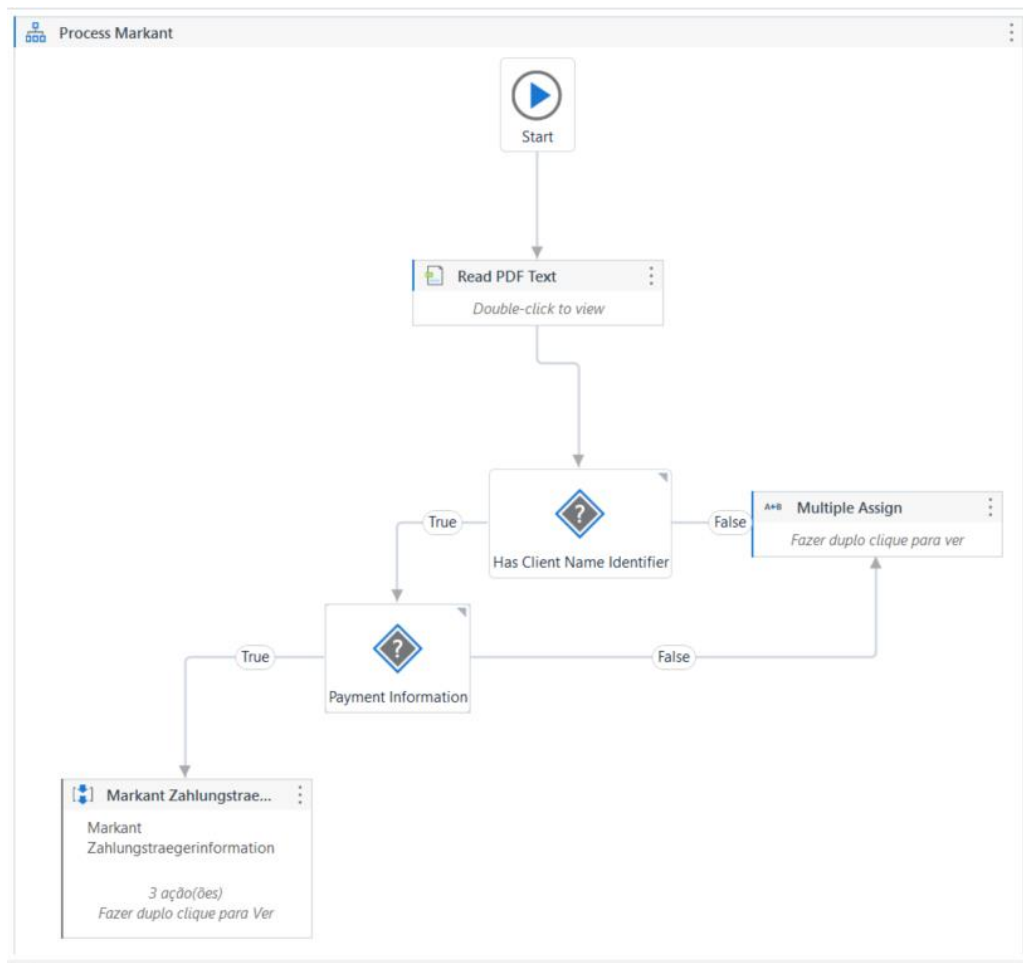


Figura 36 - Preparação para Processo de Cliente

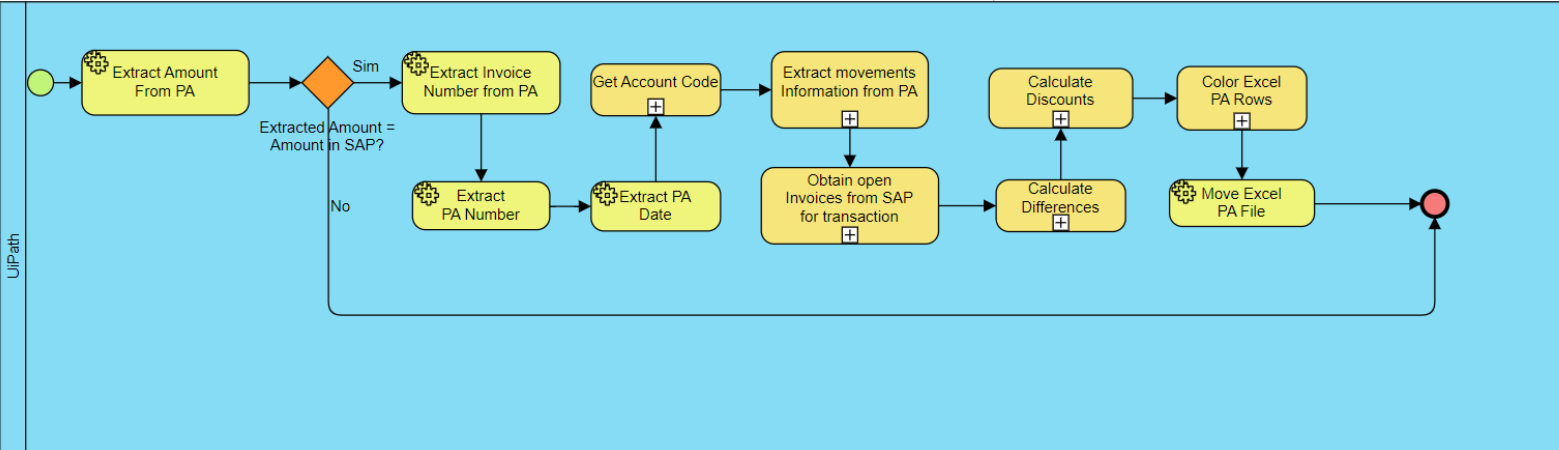


Figura 37 - Processamento de Transação

5.1.3.1 Subprocesso Get Account Code

Através da visualização da Figura 38 verifica-se que a recuperação do código de conta é um processo simples, no qual, após um acesso ao *SAP*, fazendo uso da transação VF03 e fornecendo o número da fatura é possível extrair o código de conta apresentado.

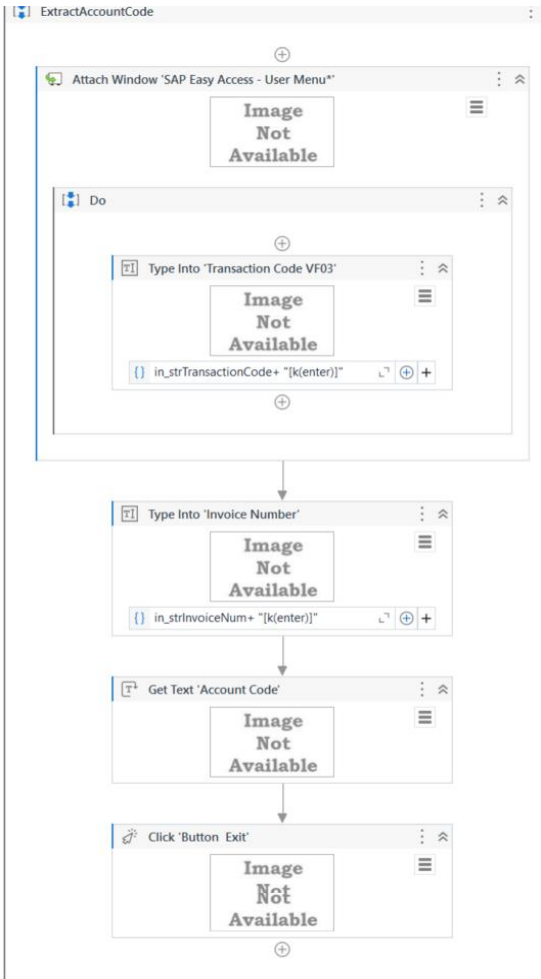


Figura 38 - Subprocesso UiPath Get Account Code

5.1.3.2 Subprocesso *Extract Data*

Este subprocesso, também ele composto por subprocessos, ilustra a abordagem tomada para recolher as informações necessárias do PDF do aviso de pagamento para um determinado cliente. A informação a extrair foca-se nos detalhes dos movimentos, e em conjunto com a informação previamente recolhida apresentam-se os campos:

- Payment Advice Number (previamente recolhido);
- Date of Payment Advice (previamente recolhido);
- Account Code (previamente recolhido);
- Document number;
- Payment Date (previamente recolhido);
- Blocked Amount;
- Foreign Number;
- Amount;
- Cash Discount;
- Payment Amount;
- Currency;
- Rebate 1;
- Rebate 2;
- Rebate 3;
- Invoice Date (recolhido posteriormente).

Começando pela análise da Figura 39, representativa do fluxo geral, verifica-se que o fluxo do processo é simples, sendo composto por:

1. A criação de uma tabela em memória no *UiPath*, que tem como colunas os nomes dos campos supramencionados;
2. A extração dos diferentes movimentos presentes no PDF recorrendo à leitura textual do PDF e à aplicação de expressões regulares;
3. A extração de informação dos movimentos recuperados;
4. A criação do ficheiro *Excel* do aviso de pagamento, tendo preenchido a primeira folha (**Sheet 1**) com os valores atualizados.

Por sua vez, complementando a Figura 40, que exhibe o grosso do fluxo do ponto 3, anotam-se as seguintes informações:

- Os movimentos no ficheiro PDF dos avisos de pagamento podem não conter todos os campos. Como tal, quando existe a omissão de exatamente um campo, considera-se que o campo em falta é o terceiro abatimento (**rebate 3**);

- Quando existe a omissão de mais do que um campo, consideram-se que todos os campos relacionados com descontos e abatimentos são ignorados, ficando a possuir o valor zero;
- Os momentos de extração de dados são feitos recorrendo a atividades *Matches* (**UiPath.Core.Activities.Matches**), fornecendo expressões regulares. A atribuição às respetivas variáveis é feita com base nas posições dos valores extraídos.

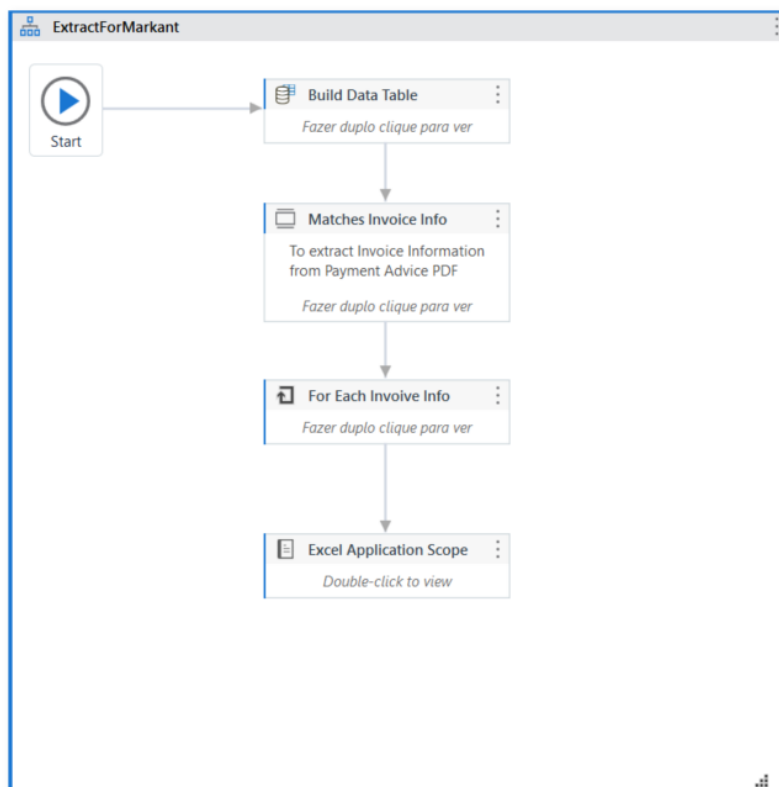


Figura 39 - Instruções UiPath Invocação da Extração de Informação do PDF

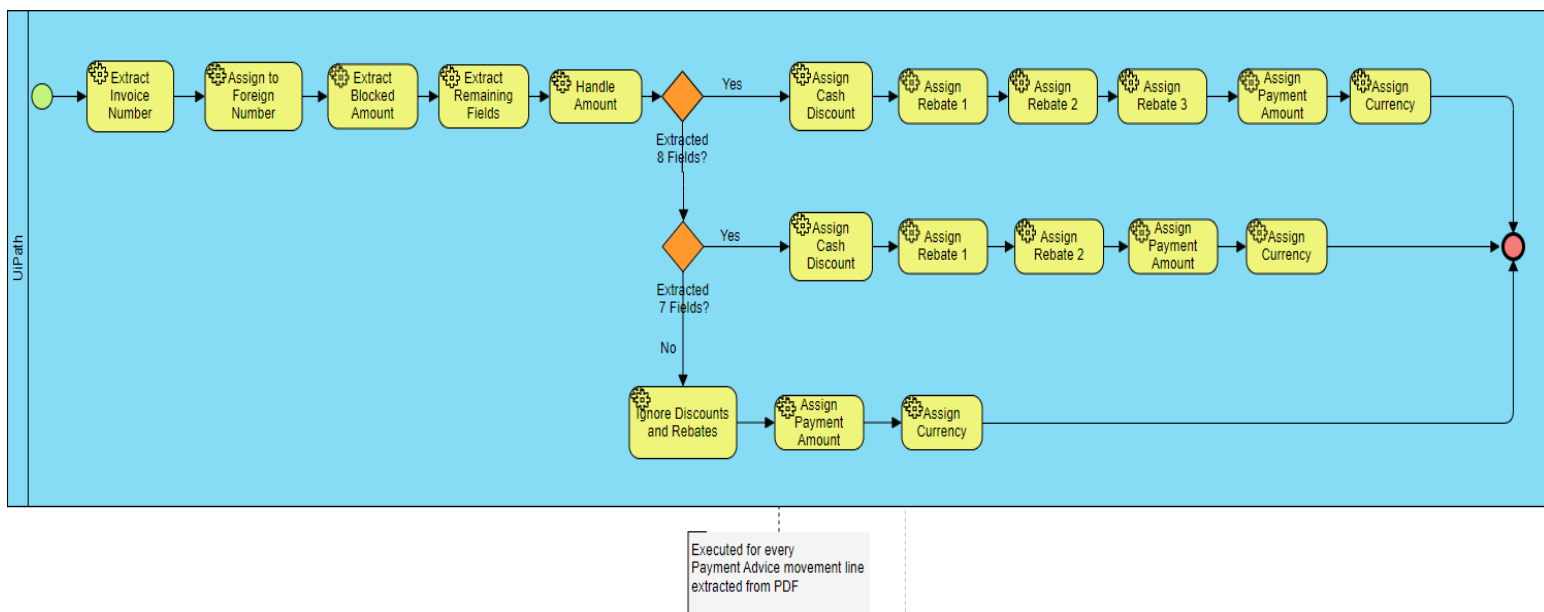


Figura 40 - Subprocesso Detalhe Extração de Informação do PDF do Aviso de Pagamento

5.1.3.3 Subprocesso Obtain Open Invoices from SAP

A Figura 41 mostra que o para a obtenção das faturas em aberto no *SAP* é necessária a realização da transação existente **FBL5N**. De modo a realizar esta transação é necessário o preenchimento dos campos de **Company Code** e **Customer Account** previamente recuperados, e de modo que os dados sejam captados pelo robô, passar o **layout** esperado (no presente caso, o layout robô). Posteriormente, após verificações quanto à presença de resultados, os mesmos são extraídos recorrendo a uma atividade de **Extract Data** (UiPath, s.d.). Os dados extraídos consistem nos seguintes campos:

- Reference;*
- Document Date;*
- Amount in local Currency;*
- Discount Amount;*
- Payment terms;*
- Disc.percent 1;*
- Days 1;*
- Disc.percent 2;*
- Days 2 ;*
- Net Days.*

Aos campos **Amount in local Currency**, **Disc.percent 1** e **Disc.percent 2** são ainda aplicadas transformações dos valores recuperados (nomeadamente eliminação de pontos finais do valor original, quando aplicável, e a substituição das vírgulas por pontos). Por fim, os valores tratados são inseridos na segunda folha (**Sheet2**) do ficheiro *Excel* de aviso de pagamento.

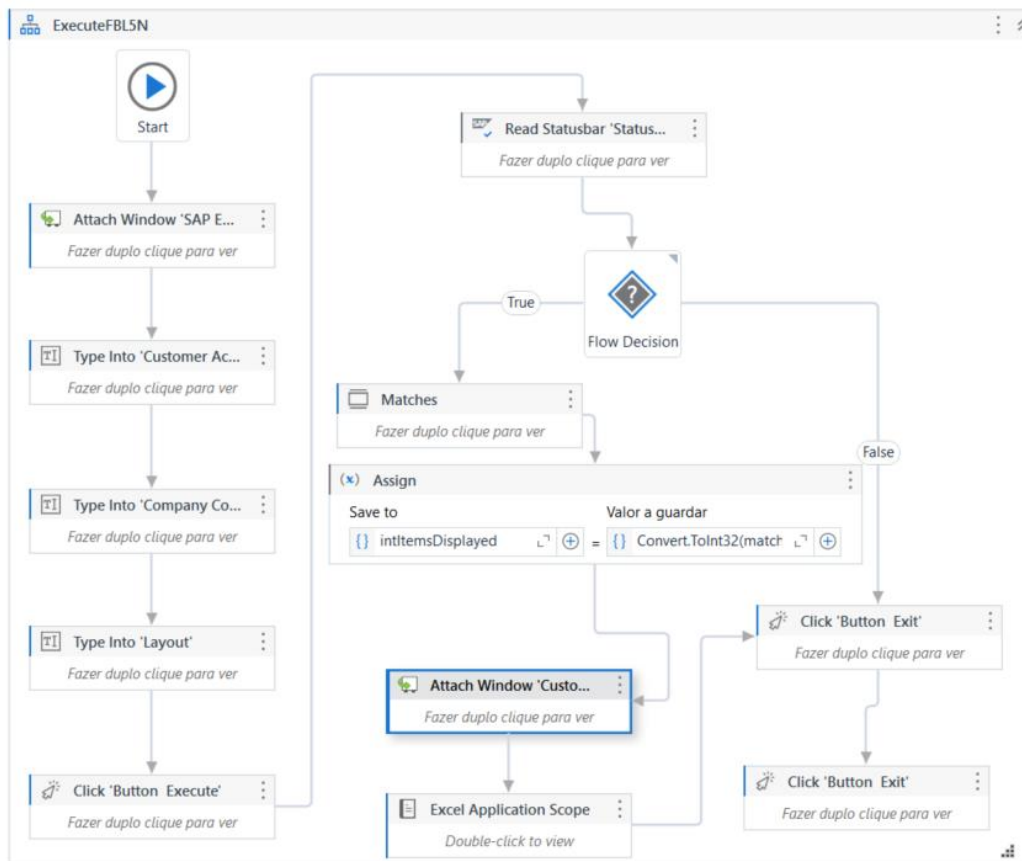


Figura 41 - UiPath Obtain Open Invoices

5.1.3.4 Subprocesso Calculate Differences

O subprocesso da Figura 42 foi pensado com o intuito de calcular se existem diferenças no aviso de pagamento que possam ser impeditivas da sua liquidação no sistema interno SAP da Inapa. Para tal, ele é responsável pelo cálculo de duas diferenças, que são calculadas para cada movimento extraído do PDF:

1. Diferença entre o montante em sistema e o montante extraído do PDF;
2. Diferença entre o montante de pagamento extraído do PDF e o valor que deveria ter sido efetivamente pago (calculado pela fórmula *montante – (desconto aplicado + montante bloqueado + abatimentos)*).

Este subprocesso é composto pelas seguintes etapas:

1. Adição das colunas **Difference**, **Difference to the Amount in the System** e **Discount Applied** à tabela que contém os dados da **primeira folha** do Excel do aviso de pagamento, criado em 5.1.3.2;
2. Para cada movimento previamente extraído do PDF, que esteja na tabela mencionada no ponto superior, extrair o valor do montante em SAP correspondente. Esse valor encontra-se na segunda folha do mesmo ficheiro, preenchida no subprocesso 0, e pode ser recuperado através da pesquisa do valor do campo **Foreign Number** da primeira folha, na coluna **Reference** da segunda folha. Tal é feito recorrendo à atividade UiPath **LookupDataTable**, que permite recuperar o valor de uma célula realizando uma pesquisa condicional;

3. Posteriormente, procede-se ao cálculo da **diferença para o montante em sistema**, subtraindo ao montante do movimento no PDF (coluna **Amount**) o montante em sistema SAP recuperado no ponto anterior;
4. Cálculo do valor que efetivamente deveria ter sido pago, recorrendo à fórmula mencionada em [2];
5. Cálculo da diferença entre o montante pago e o valor que deveria ter sido efetivamente pago;
6. Cálculo da percentagem de desconto aplicada face ao montante total a pagar;
7. Atualização da primeira folha do *Excel* do aviso de pagamento com os valores calculados.

Há, no entanto, que referir que este processo não é um processo genérico, uma vez que cada cliente tem a sua própria fórmula de cálculo do **valor que deveria ter sido efetivamente pago**.

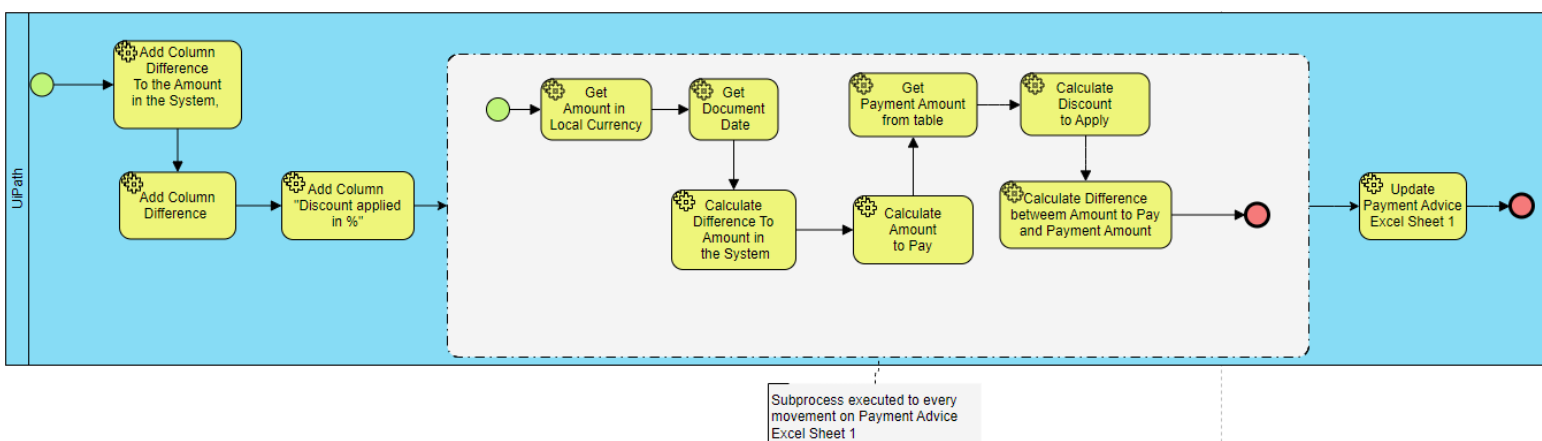


Figura 42 - Fluxo de Subprocesso *Calculate Differences*

5.1.3.5 Subprocesso *Calculate Discounts*

O subprocesso em questão tem como objetivo o cálculo do desconto que deveria ter sido efetivamente aplicado. Para isso são tidos em conta os termos de pagamento, a percentagem de desconto 1 (em caso de pagamento acima de **days1** dias) e a percentagem de desconto 2 (em caso de pagamento acima de **days2** dias). A percentagem de desconto 1 (**Disc percent 1**) representa uma maior percentagem de desconto (que a percentagem de desconto 2), mas para ser aplicada, a data de pagamento tem de ser relativamente curta (uma vez que o valor **days1** é reduzido). Por outro lado, apesar da percentagem de desconto 2 apresentar um valor de desconto mais reduzido, a margem para a sua aplicação é maior, ou seja, há um maior prazo até ao qual a data de pagamento poderá ser realizada.

Analisando a Figura 43 depreende-se o seguinte fluxo:

1. À tabela atualizada dos movimentos dos avisos de pagamento, acrescenta-se a coluna **Disc. To Apply**, que, como o nome indica, representa o desconto a aplicar;
2. É realizada uma verificação quanto à data da fatura (**invoice date**). Caso a mesma seja inválida, não será aplicado nenhum desconto;
3. Caso contrário, são extraídos os valores dos termos de pagamentos (**Payment Terms**), e dos descontos a aplicar (**Disc percent1** e **Disc percent2**) caso sejam verificadas certas condições;
4. Começam a ser verificadas as regras de negócio necessárias para o cálculo do desconto que deveria ter sido efetivamente aplicado, começando primeiramente por validar os termos de pagamento. Caso o valor do campo não esteja preenchido ou contenha **NA**, o desconto a aplicar deverá ser 0;
5. Em caso de os termos de pagamento estarem validados, deve-se validar o valor da percentagem de desconto 1. Se esse valor for 0, como ele tem precedência face ao valor da percentagem de desconto 2, então o desconto a aplicar deve ser 0;
6. Caso contrário, é validado se a data da fatura (**invoice date**) + **days1** é superior ou igual à data de pagamento. Ou seja, se o pagamento foi efetuado dentro do intervalo necessário para aplicação da maior percentagem de desconto (**Disc percent 1**). Caso afirmativo, como esta percentagem tem precedência, ela é escolhida e não se realizam mais cálculos para a presente linha;
7. Por outro lado, caso o pagamento tenha sido feito fora deste intervalo, dever-se-ão validar as mesmas condições para a percentagem de desconto 2 (**Disc percent2**). Caso se verifique que o pagamento foi feito no intervalo do tempo entre a data da fatura e a data da fatura + **days2**, então o valor do desconto a aplicar é o valor do campo **Disc percent2**, caso contrário, deverá ser considerado que o cliente não tem direito a nenhum tipo de desconto para o movimento efetuado;
8. Por fim, tendo sido calculado o desconto a aplicar por cada movimento, a primeira folha (**Sheet1**) do ficheiro Excel do aviso de pagamento em questão deve de ser atualizada com os novos valores calculados.

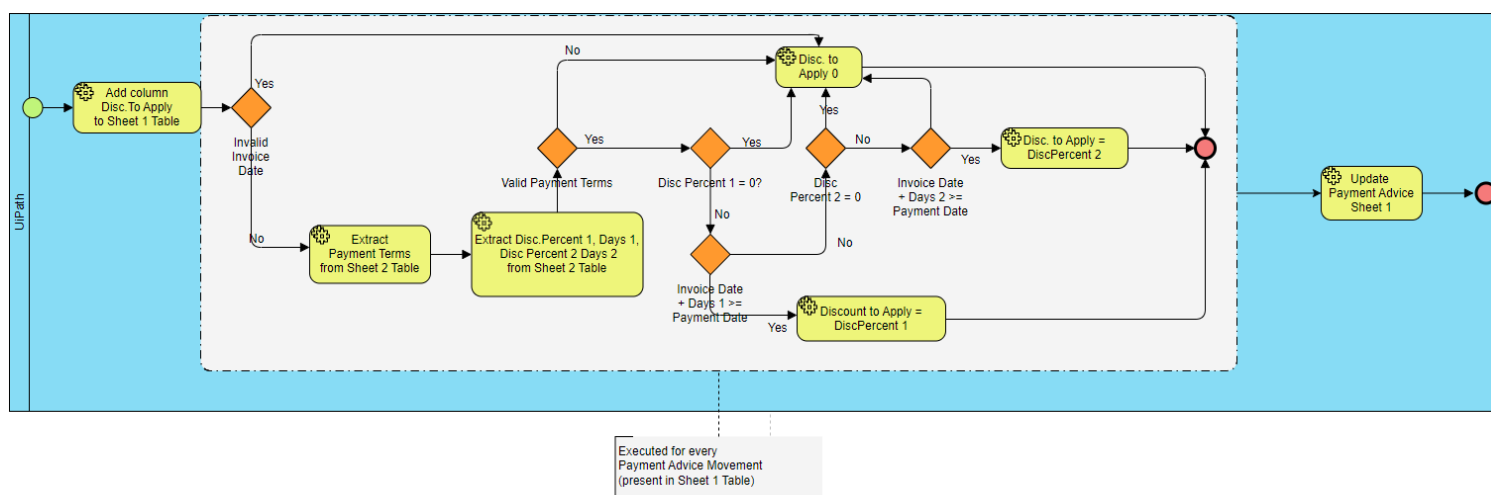


Figura 43 - Subprocesso cálculo de descontos

5.1.3.6 Subprocesso Color Excel PA Rows

O objetivo do presente subprocesso passa pela sinalização visual a colaboradores da Inapa auferidos do estado dos movimentos de um certo aviso de pagamento. Para tal, foi definido como critério, para cada uma das diferenças verificadas no Subprocesso *Calculate Differences* a aplicação de uma cor diferente, sendo aplicadas as seguintes cores nas seguintes situações:

- **Vermelho:** Quando a diferença entre o montante pago e o montante calculado é diferente de zero;
- **Amarelo:** Quando a diferença entre o montante em sistema e o montante pago é diferente de zero;
- **Azul:** Quando a diferença entre o desconto a aplicar e o desconto aplicado é diferente de zero;
- **Verde:** Quando não se verifique qualquer diferença das previamente enunciadas.

Ao consultar a Figura 44, é possível verificar que estas validações e respetiva aplicação da cor são realizadas para cada um dos movimentos presente na primeira folha (**Sheet 1**) do *Excel* do aviso de pagamento a ser tratado. Na mesma figura, estão destacadas ainda atividades de aplicação de cor após certas condições serem cumpridas. Essas atividades são compostas pelas instruções *UiPath ExcelSetColorRange*, responsável por colorir um certo conjunto de células de uma determinada cor, e pela instrução de atribuição de variável, usada no presente caso para incrementar um contador de células da respetiva cor mudada.

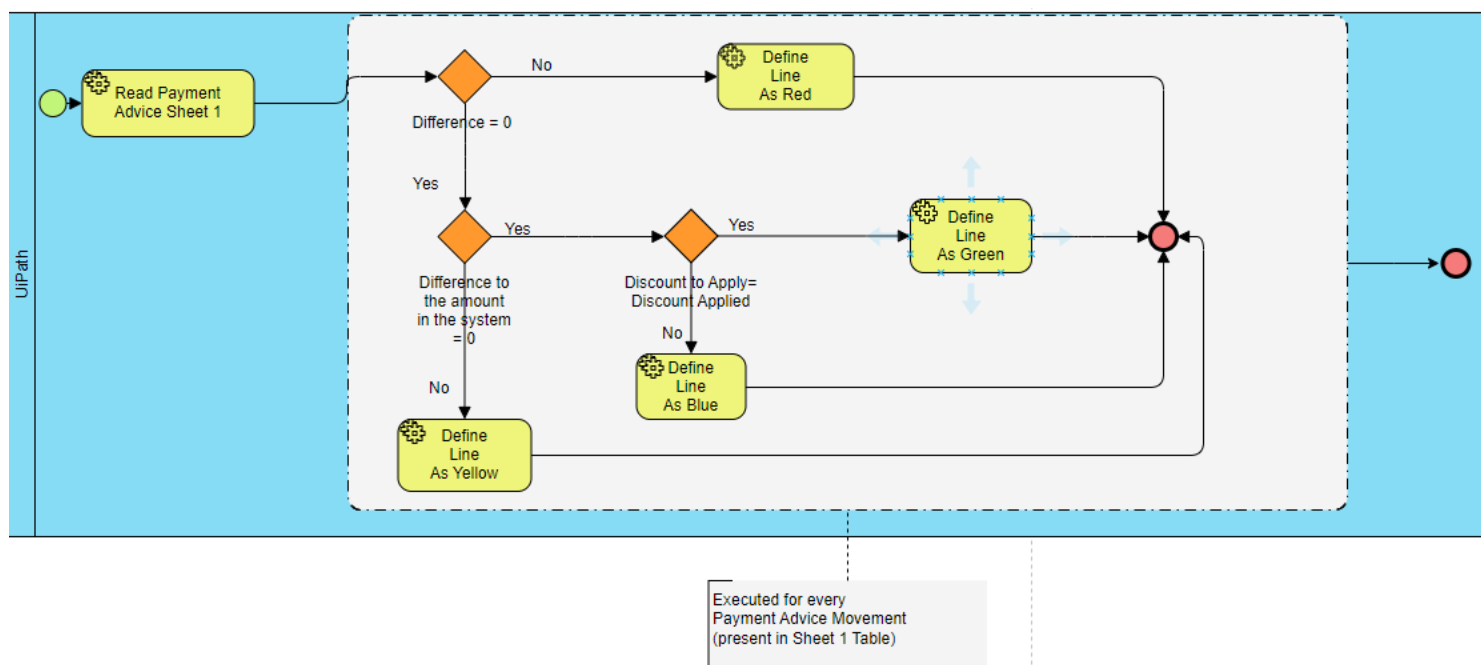


Figura 44 - Subprocesso Aplicação de Cor ao *Excel* Aviso de Pagamento

5.1.4 Finalização

Este subprocesso é caracterizado por garantir que a aplicação acaba num estado coerente, para tal finalizando a aplicação SAP. Como visto na Figura 45, cabe-lhe ainda a responsabilidade de, após todos os movimentos serem validados, mover o ficheiro *Process Summary* para a respetiva pasta de sucesso ou falha, consoante o processamento para todas as transações.

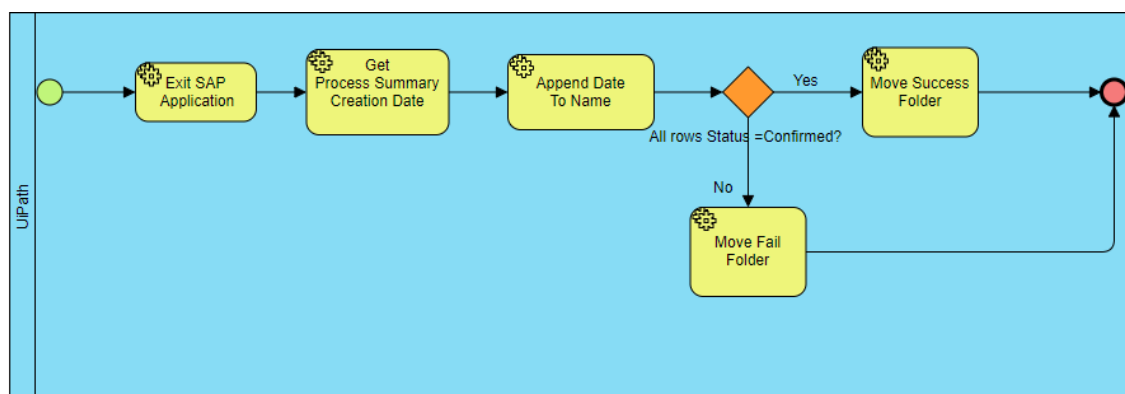


Figura 45 - Subprocesso End

5.2 Solução Complementar de Métricas de *Process Summaries*

Este subcapítulo é dedicado à apresentação dos artefactos e decisões relevantes tomadas no que toca à parte da solução descrita em 4.3, composta pelos componentes **Folder Watcher** e **Dashboard**.

A solução em causa, como referido em 3.2.1, foi pensada para auxiliar os colaboradores internos a facilmente analisarem ficheiros *Process Summary* e que avisos de pagamento possam ter tido algum erro. Para tal, a solução montada teve consideração pelas métricas que poderiam facilitar a função dos colaboradores. Estas métricas estão disponíveis em duas zonas distintas:

1. **Página General** – Na página principal são apresentados resumos de informação relevante (Figura 46), como o total de montante preso nos *Process Summaries*, o total de ficheiros que não foram processados com sucesso, e quais o mais recente e o mais antigo processo que não tiveram sucesso. Além dessas métricas é ainda fornecida uma visão sobre a quantidade de ficheiros *Process Summary* que foram processados com erro e com sucesso e ainda quais os resultados para o presente dia, no caso de o processo já ter sido executado.
2. **Página Metrics** – Nesta página *web* (representada pela Figura 47) encontram-se métricas e os seus detalhes, de modo fácil de pesquisa. A informação desta página destina-se a fins puramente visuais, mas o colaborador pode interagir mudando os períodos entre os quais quer visualizar cada uma das métricas. A informação encontra-se orientada aos ficheiros *Process Summary* e aos clientes neles contidos. As métricas escolhidas foram:
 - a. Ficheiros *Process Summary* com maior montante bloqueado;
 - b. Ficheiros *Process Summary* com mais movimentos falhados;
 - c. Clientes com maior montante bloqueado;
 - d. Erro mais frequente de ocorrer por cada cliente (e sua periodicidade);
 - e. Clientes com mais movimentos falhados;
 - f. Clientes com mais processos falhados.

O conjunto de métricas apresentados permite aos colaboradores facilmente perceber quais são os ficheiros em erro e quais é que devem de ser tratados primeiro. Depois de tratados os ficheiros, de modo a poder notificar o serviço que um determinado *Process Summary* ou um dos seus movimentos foi corrigido, o colaborador Inapa pode fazer uso da página **Files** (Figura 48). Esta página apresenta todos os *Process Summary* com pelo menos um movimento falhado e oferece a possibilidade de marcar o movimento como corrigido, atualizando assim a informação referente ao mesmo. Uma vez que o *Process Summary* tenha todos os movimentos como corrigidos, ele deixa de ser contabilizado para as métricas, garantindo a consistência da informação.

Por fim, a aplicação apresenta ainda a página **Clients** (Figura 49), cujo objetivo é mostrar para cada cliente em específico quais os *Process Summaries* que contêm avisos de pagamento em falha, e os detalhes dos mesmos. Esta perspetiva orientada ao cliente é fulcral para uma rápida análise dos principais impactos nos clientes e urgência na resolução de assuntos relacionados com os mesmos.

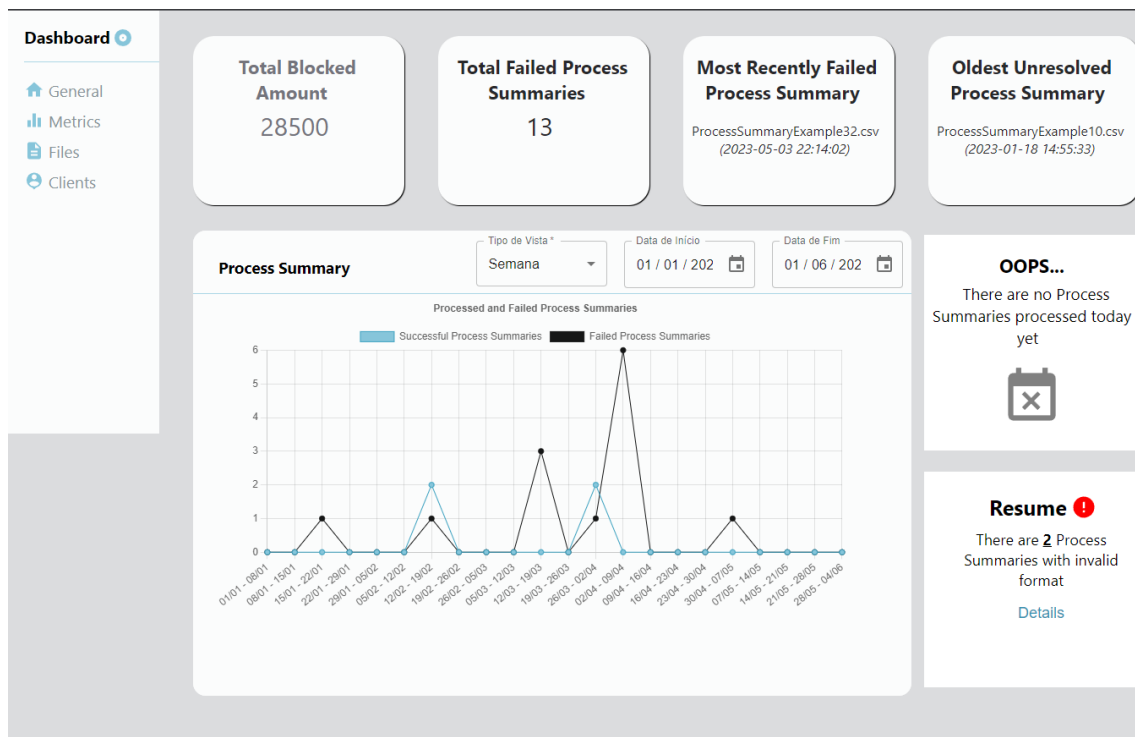
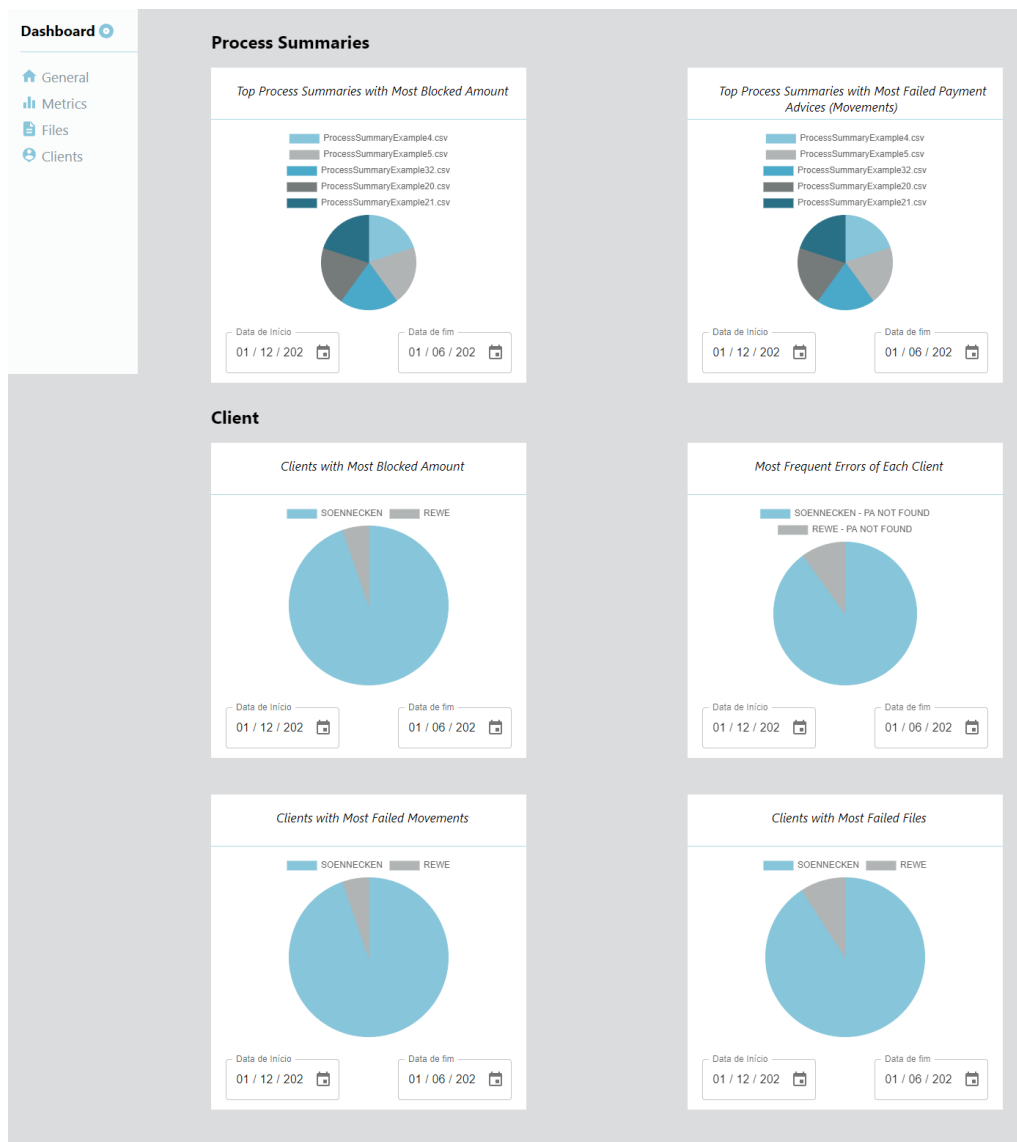


Figura 46 - Dashboard General



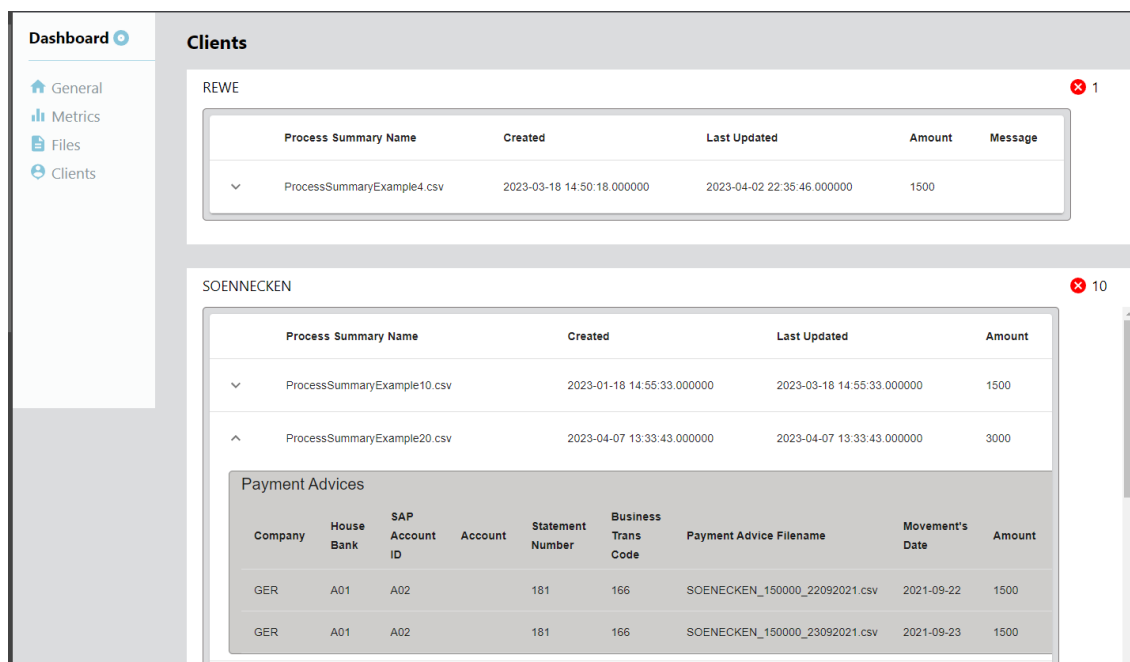


Figura 49 - *Dashboard Clients*

De seguida as aplicações serão detalhadas a um nível mais técnico de modo a elaborar como é que a solução foi construída.

5.2.1 Dashboard *Backend*

No desenvolver deste componente procurou-se aplicarem-se as boas práticas de engenharia de software, nomeadamente na aplicação dos padrões GRASP e SOLID. De modo a organizar o código e a preparar uma divisão adequada das responsabilidades, o mesmo foi dividido por packages (Figura 50) onde cada qual representa um subconjunto de elementos de natureza semelhante.

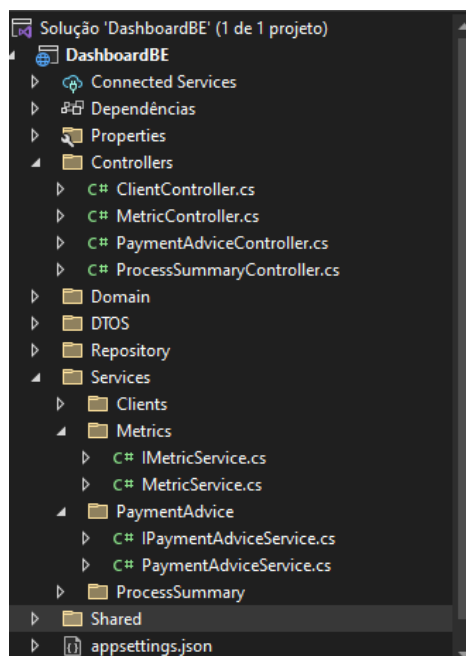


Figura 50 - Estrutura Dashboard Backend

5.2.1.1 Uso de dois mecanismos de acesso à camada de dados

A solução em causa é sobretudo uma solução de *reporting* e por isso a mesma é fortemente dedicada à apresentação de dados estatísticas sobre o domínio da aplicação. No entanto estas estatísticas por si só não fazem parte do domínio. Portanto, de modo a tirar o máximo partido tanto da obtenção do domínio como da obtenção de métricas, foram utilizadas duas abordagens de recolha de dados da base de dados diferentes: **Execução de Stored Procedures via**

MySQLConnection: Esta abordagem (Figura 51) foi escolhida para contornar as limitações de realizar *queries* complexas à base de dados via *ORM*, focadas na manipulação e apresentação de dados que não constituem objetos de domínio. Esta execução foi baseada num único método genérico, que aceita um infinito número de parâmetros e o nome do *Stored Procedure* a executar. O desenvolvimento deste método evidencia boas práticas de software ao reduzir o código necessário para a implementação da comunicação com a base de dados. Adicionalmente ao abstrair a responsabilidade de comunicar com a base de dados diretamente no serviço, é ainda respeitado o princípio *Single Responsibility Principle* do padrão **SOLID** por garantir que cada classe só tem uma razão para mudar.

ORM Entity Framework Core: Esta abordagem foi escolhida devido a ter como vantagens a eficaz e coerente recuperação de objetos de domínio (*Process Summaries* e *PaymentAdvices*), presente na Figura 52.

```

2 referências
public async Task<JObject> GetMetricsResume()
{
    try
    {
        var metricResume = await _context.SpJsonToJObject("SP_GetMetricsResumeFinal");
        return metricResume;
    }
    catch (Exception ex)
    {
        var error = new JObject();
        error.Add("Status", "NOK");
        return error;
    }
}

```

Figura 51 - Execução de Stored Procedures via MySqlConnection

```

27 public async Task<List<ProcessSummaryDTO>> GetBetweenDates(DateTime beforeDate, DateTime afterDate)
28 {
29     try
30     {
31         var errorProcessSummariesBetweenDatesList = await _context.ProcessSummary
32             .Include(p => p.PaymentAdvices.Where(pa => !pa.Status.Equals("Confirmed")))
33             .Where(psm => psm.Created <= afterDate && psm.Created >= beforeDate)
34             .ToListAsync();
35         if (errorProcessSummariesBetweenDatesList != null && errorProcessSummariesBetweenDatesList.Count() > 0)
36         {
37             var listDTO = errorProcessSummariesBetweenDatesList.Select(ps => ProcessSummaryDTO.ConvertToDTO(ps)).ToListAsync();
38             return listDTO;
39         }
40     }
41     catch (Exception ex)
42     {
43         Console.WriteLine(ex.ToString());
44     }
45     return null;
46 }
47
48
49

```

Figura 52 - Demonstração Uso ORM Entity Framework Core

5.2.1.2 Injeção de dependências utilizando interfaces

A utilização de injeção de dependências proporciona vários benefícios (Kralj, s.d.) , como a centralização de configuração de código (Figura 53), aumento manutenibilidade do código e diminuição do acoplamento da solução.

Ao utilizar interfaces, promovemos o baixo acoplamento e alta da coesão da solução, que são pilares dos padrões **GRASP**. Esta utilização (Figura 54 e Figura 55) contribui para o baixo acoplamento da solução uma vez que as classes invocadoras não ficam dependentes da implementação concreta da classe, apenas da interface. Portanto, caso haja alguma mudança na implementação da classe concreta, a mesma não é refletida na classe invocadora. Ao termos uma injeção de dependências baseada na utilização de interfaces, respeitamos também os princípios PID (**Princípio de Inversão de Dependências**) e PSI (Princípio de Segregação de Interfaces) do SOLID. Enquanto o primeiro princípio (PID) invoca que devemos sempre depender de abstrações e nunca de implementações concretas, o PSI cita que é preferível termos várias interfaces mais específicas do que apenas uma interface genérica (Paixão, 2019). Isto é refletido no código ao termos várias interfaces de serviços diferentes, em vez de apenas uma genérica.

```

9     var builder = WebApplication.CreateBuilder(args);
10
11     // Add services to the container.
12
13     builder.Services.AddControllers().AddNewtonsoftJson();
14     // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
15     builder.Services.AddEndpointsApiExplorer();
16
17
18     builder.Services.AddScoped<DataAccess>();
19     builder.Services.AddScoped<IProcessSummaryService, ProcessSummaryService>();
20     builder.Services.AddScoped<IPaymentAdviceService, PaymentAdviceService>();
21     builder.Services.AddScoped<IMetricService, MetricService>();
22     builder.Services.AddScoped<IClientService, ClientService>();
23     builder.Services.AddScoped<IProcessSummaryRepository, ProcessSummaryRepository>();

```

Figura 53 - Centralização de Configurações de Injeção de Dependências

```

13
14 [ApiController]
15 [Route("[controller]")]
16 public class ClientController : ControllerBase
17 {
18
19     private readonly IClieintsService _clientService;
20
21     0 referências
22     public ClientController(IClieintsService clientService)
23     {
24         _clientService = clientService;
25     }
26
27     [HttpGet]
28
29     0 referências
30     public async Task<IActionResult> GetClientsProcessSummary()
31     {
32         var clientsDetails = await _clientService.GetClientDetails();
33         return Ok(clientsDetails);
34     }
35 }

```

Figura 54 - Exemplo de utilização de Interfaces nos Controllers

```

7 namespace DashboardBE.Services.ProcessSummary
8 {
9     2 referências
10     public class ProcessSummaryService : IProcessSummaryService
11     {
12
13         private readonly IProcessSummaryRepository _repository;
14
15         0 referências
16         public ProcessSummaryService(IProcessSummaryRepository repository)
17         {
18             _repository = repository;
19         }
20
21         2 referências
22         public async Task<List<ProcessSummaryDTO>> GetProcessSummaries(JObject dates)
23         {
24             var verifiedDates = validateDates(dates);
25             var beforeDate = verifiedDates.Item1;
26             var afterDate = verifiedDates.Item2;
27             return await _repository.GetBetweenDates(beforeDate, afterDate);
28         }
29     }
30 }

```

Figura 55 - Exemplo de utilização de Interfaces nos Serviços

5.2.2 Dashboard *Frontend*

O desenvolvimento deste componente seguiu de um modo geral as mesmas boas práticas de engenharia de software que se encontram expostas em 5.2.1, sendo um código focado na manutenibilidade da solução, apresentado sobretudo características de baixo acoplamento e alta coesão.

A organização do código também seguiu uma distribuição por *packages*, de modo a distinguir código usado para diferentes objetivos (observável na Figura 56 as pastas principais de ***components***, ***pages***, ***services*** e ***utils***). Enquanto a pasta de serviços contém o código responsável pela realização de chamadas *HTTP* à contraparte do *backend*, a pasta ***utils*** apresenta funções auxiliares utilizadas em diversos componentes. A pasta ***components*** evidencia outra boa prática na engenharia de software, pois nela encontram-se blocos de código genéricos até certo ponto, feitos propositadamente para serem reutilizados em diversas páginas. Por fim, a pasta ***pages*** alberga as páginas, que fazem uso dos serviços e dos componentes para servir o colaborador Inapa com o *frontend* da solução.

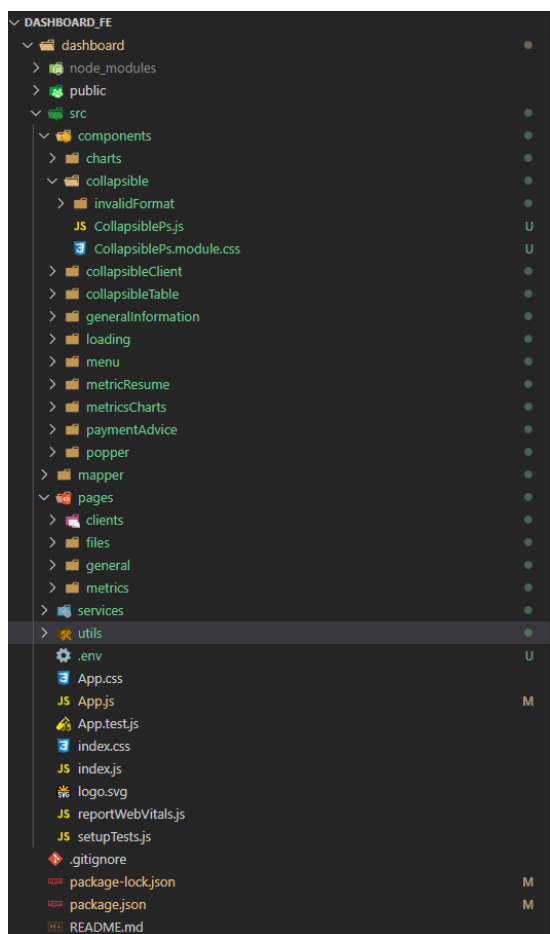


Figura 56 - Estrutura *Dashboard Frontend*

5.2.2.1 Utilização de um *Mapper* para abstração da responsabilidade

Quando são necessários dados do servidor é realizada uma chamada ao serviço correspondente. No entanto, a resposta vinda do serviço pode não se encontrar diretamente adaptada à sua imediata exibição ao

colaborador, sendo necessário proceder ao tratamento dos dados. Para tal foi pensada a existência e utilização de *Mappers* – conjuntos de funções dedicadas à conversão de dados para estarem adequados à sua exibição para uma determinada vista. O uso destas funções dedicadas tem como grande benefício a abstração de responsabilidades para uma função particular, e a consequente diminuição do acoplamento entre o invocador e os dados. Adicionalmente, existe também um incremento na manutenibilidade e legibilidade do código produzido. Uma possível alternativa seria a adaptação dos dados para a vista em que fossem necessários no momento da sua chamada. No entanto esta abordagem perde por não permitir reutilização do mapeamento (caso o mesmo seja necessário em mais algum lado), e por aumentar o acoplamento entre a classe invocadora e os dados. A Figura 57 representa uma das funções dos *Mappers* e a Figura 58 demonstra a aplicação da mesma num componente.

```
6 export function MetricToPieChartData(props) {
7
8   if (props === null || props === undefined || props.length === 0) {
9     return [];
10  }
11
12  let i = 0;
13  const colorArray = [];
14  const size = props != undefined ? props.data.length : 3;
15  while (i < size) {
16    let color;
17    if (i % 2 == 0 && i < 2 * blues.length) {
18      color = blues[(i / 2)];
19    }
20    else if (i % 2 == 1 && grayBlack.length > Math.floor(i / 2))
21      color = grayBlack[Math.floor(i / 2)];
22    else {
23      color = "#" + Math.floor(Math.random() * 16777215).toString(16).padStart(6, '0').toUpperCase();
24    }
25
26    if (!colorArray.includes(color)) {
27      colorArray.push(color);
28      i++;
29    }
30  }
31
32  const data = {
33    labels: props.labels,
34    datasets: [
35      {
36        label: props.legend,
37        data: props.data,
38        backgroundColor: colorArray,
39        borderColor: colorArray,
40        borderWidth: 1,
41      },
42    ],
43  };
44  return data;
45 };
```

Figura 57 - Função Exemplo Transformação de Dados *Mapper*

```

useEffect(() => {
  const servDataFunc = async () => {
    console.log("🔥 ~ file: ProcessSummaryLineChart.js:51 ~ servDataFunc ~ beforeDate:", beforeDate);
    console.log("🔥 ~ file: ProcessSummaryLineChart.js:51 ~ servDataFunc ~ afterDate:", afterDate);

    if ((afterDate !== null && afterDate !== undefined) || (beforeDate !== null && beforeDate !== undefined)) {
      if (props.func) {
        const metrics = await props.func(beforeDate, afterDate);
        if (metrics) {
          setTitle(metrics.title);
          const transformedData = MetricToPieChartData(metrics);
          let isArray = Array.isArray(transformedData)
          if ((isArray && transformedData.length > 0)) {
            setData(transformedData);
            setIsLoading(false);
          }
          else if (!isArray && transformedData !== null && transformedData !== undefined) {
            setData(transformedData);
            setIsLoading(false);
          }
        }
      }
    }
  };

  servDataFunc();
}, [beforeDate, afterDate, props.func]);

```

Figura 58 - Invocação de uma Função do *Mapper*

5.2.2.2 Aplicação imutável sobre os dados

Estando a utilizar o paradigma funcional permitido pela *framework React*, um dos seus princípios passa pelo favorecimento da manipulação de estruturas imutáveis face às mutáveis, uma vez que tal ajuda a manter o estado da aplicação coerente. Esse princípio indica que quando existe a necessidade de modificação dados, deve ser realizada uma cópia do objeto, e feitas as manipulações sobre a cópia do objeto, sendo depois o objeto original substituído pela sua cópia. Este princípio foi utilizado no desenvolvimento desta componente por ser considerado justamente uma boa prática de engenharia de software no que toca ao paradigma funcional. Como exemplo da sua utilização, pode-se consultar a Figura 59 Nesta figura, na invocação da função **validateCheck()** existe a necessidade de modificar o estado da aplicação (**correctStates**). No entanto, em vez desta variável ser modificada diretamente, é feita previamente uma cópia (variável **newState**), sendo depois esta variável cópia modificada, culminando na substituição do estado original (**setCorrectedStates**).

```

useEffect(() => {
  validateUpdate();
}, [correctStates])

const openCollapsible = () => {
  setOpen(!open);
}

const validateCheck = (event, id) => {
  const newState = [...correctStates];
  const currIndex = newState.findIndex(paRef => paRef.id === id);
  if (currIndex === -1) {
    newState.push({ id: id, checked: event.target.checked });
  } else {
    newState[currIndex].checked = event.target.checked;
  }
  setCorrectedStates(newState);
}

```

Figura 59 - Exemplo de utilização de Imutabilidade

5.2.3 Folder Monitor

A arquitetura deste componente da solução é bastante semelhante à apresentada do componente **Dashboard Backend** 5.2.1, e como tal o desenvolvimento deste componente seguiu de um modo geral as mesmas boas práticas de engenharia de software que se encontram expostas em 5.2.1 , sendo focada na manutenibilidade da solução, apresentado sobretudo características de baixo acoplamento e alta coesão. No entanto a diferença de natureza entre os projetos (sendo o **Dashboard Backend** uma *Web API* e o **Folder Monitor** uma aplicação de consola) levou à existência de certas nuances.

Apesar de se tratar do desenvolvimento de uma pequena aplicação, o código foi distribuído por *packages* (Figura 60) para facilitar a legibilidade e preparar uma divisão adequada das responsabilidades.

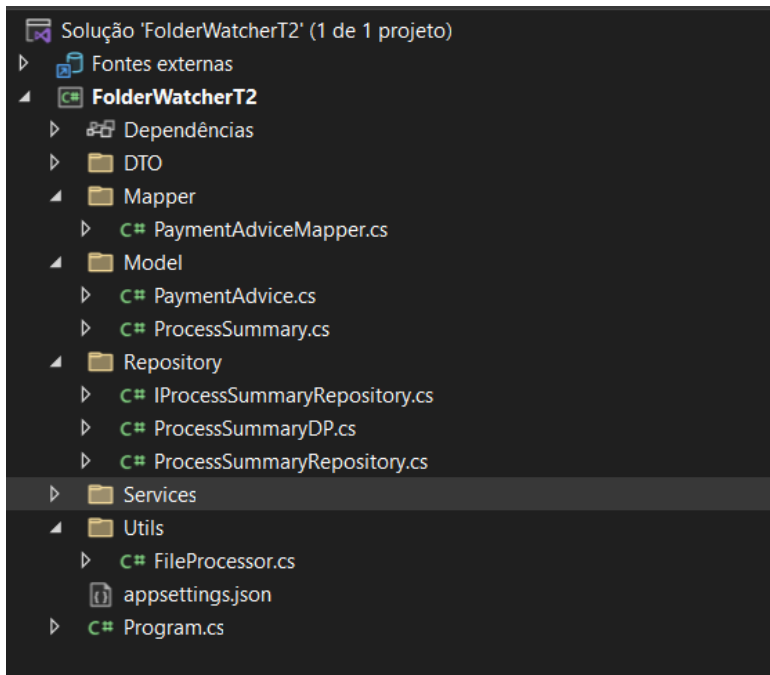


Figura 60 - Estrutura *Folder Monitor*

5.2.3.1 Monitorização do diretório

A monitorização do diretório pretendido foi feita recorrendo a um serviço. Este serviço está configurado para ser executado automaticamente pelo corpo principal da aplicação (linha 40 da Figura 61). O serviço em questão está continuamente e automaticamente a monitorizar o diretório definido no respetivo ficheiro de configurações, e sempre que existe um novo registo de edição de um ficheiro com a extensão CSV, o mesmo é tratado pelo método respetivo. Para tal, recorreu-se à classe nativa de *.NET FileSystemWatcher* (Figura 62). O tratamento deste tipo de ficheiros implica a sua conversão para registos de dados, sendo tal feito recorrendo a um método de uma classe utilitária dedicada (***FileProcessor***).

```

17 0 referências
18 public static void Main(string[] args)
19 {
20     var builder = new ConfigurationBuilder()
21     .SetBasePath(Directory.GetCurrentDirectory())
22     .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true);
23     var config = builder.Build();
24
25     string connectionString = config.GetValue<string>("ConnectionString");
26
27     var host = CreateHostBuilder(args, config, connectionString).Build();
28     host.Run();
29 }
30
31 1 referência
32 public static IHostBuilder CreateHostBuilder(string[] args, IConfiguration configuration, string connectionString) =>
33 {
34     Host.CreateDefaultBuilder(args)
35     .ConfigureAppConfiguration((hostContext, configBuilder) =>
36     {
37         configBuilder.Sources.Clear();
38         configBuilder.AddConfiguration(configuration);
39     })
40     .ConfigureServices((hostContext, services) =>
41     {
42         services.AddHostedService<FolderMonitorService>();
43         services.AddDbContext<ProcessSummaryDbContext>(options => options.UseMySQL(connectionString));
44         services.AddScoped<IProcessSummaryRepository, ProcessSummaryRepository>();
45     });
46 }

```

Figura 61 - Invocação do Serviço de Monitorização de Diretório

```

public async Task StartAsync(CancellationSource cancellationSource)
{
    string folderPath = _folderToMonitor;
    FileSystemWatcher watcher = new FileSystemWatcher(folderPath);
    watcher.Path = folderPath;
    watcher.NotifyFilter = NotifyFilters.LastWrite;
    watcher.Filter = "*.csv";
    watcher.Changed += new FileSystemEventHandler(OnFileChanged);
    watcher.EnableRaisingEvents = true;
    _cancellationSource = CancellationSource.CreateLinkedCancellationTokenSource(cancellationSource);
    await MonitorFolderAsync(_cancellationSource.Token);
    StopAsync(_cancellationSource.Token);
}

```

Figura 62 - Monitorização do Diretório Configurado

5.2.3.2 Leitura e Processamento do Ficheiro Process Summary

Sempre que um ficheiro **CSV Process Summary** é colado no diretório observado, ele é consumido pela aplicação **Folder Monitor** e processado. Fazendo uso da classe utilitária **FileProcessor** e recorrendo ao pacote de terceiros **CSV Helper**, começa-se por verificar a disponibilidade do ficheiro. Como um ficheiro **Process Summary** pode ter um tamanho variável, e como a classe utilizada **FileSystemWatcher** despoleta a ação de leitura mal o ficheiro começa a ser colado no diretório, para que a leitura do **Process Summary** não comece antes de o mesmo terminado, foi usada a instrução **FileMode.Open** (linha 39 da Figura 63). Esta instrução informa que a leitura do ficheiro será feita apenas se o ficheiro não estiver a ser escrito por outra aplicação. Posteriormente, é feita a conversão dos dados automática para um objeto DTO e o mesmo é retornado.

```
29 while (unavailable)
30 {
31     try
32     {
33         var config = new CsvConfiguration(CultureInfo.InvariantCulture)
34         {
35             Delimiter = ";",
36             PrepareHeaderForMatch = (args) => Regex.Replace(args.Header, @"\s", string.Empty),
37         };
38
39         using var stream = new FileStream(e.FullPath, FileMode.Open);
40         using (var reader = new StreamReader(stream))
41         using (var csv = new CsvReader(reader, config))
42         {
43             var records = csv.GetRecords<PaymentAdviceDTO>();
44             lista = records.ToList();
45         }
46         unavailable = false;
47     }
48     catch (System.IO.IOException ex)
49     {
50         //check here why it failed and ask user to retry if the file is in use.
51     }
52     catch (Exception ex)
53     {
54         unavailable = true;
55         return null;
56     }
57 }
58
59 return lista;
60
61
62 }
```

Figura 63 - Método de Leitura de Process Summary CSV

Depois de o ficheiro ser lido com sucesso, é verificado se não houve nenhum erro de leitura das linhas extraídas (linha 106 da Figura 64), e posteriormente, é confirmado que o valor da propriedade **status** da linha referente ao aviso de pagamento se encontra com o valor “confirmado”. Bastando apenas uma linha de aviso de pagamento ter um **status** diferente de “confirmado” para que o estado do **Process Summary** (variável **IsGood**, linha 95 Figura 64) obrigue à intervenção humana, o mesmo é alterado conforme o necessário. Por fim, dependente do estado **Process Summary**, o ficheiro é movido para a pasta correspondente - pasta de sucesso se o mesmo pode ser imediatamente processado, pasta de falha caso o mesmo necessite de ser revisto manualmente (linhas 120 e 121 da Figura 64).

```

92     fileNameQuantity.TryGetValue(e.Name, out isPresent);
93     if (isPresent == 1)
94     {
95         bool isGood = false;
96         fileNameQuantity.Remove(e.Name);
97         List<PaymentAdviceDTO> extractedPaymentAdvices = FileProcessor.ProcessFile(e);
98         var ps = new ProcessSummary
99         {
100             Name = e.Name,
101             Created = System.DateTime.Now,
102             LastUpdated = System.DateTime.Now,
103             IsGood = isGood,
104         };
105     };
106     if (extractedPaymentAdvices == null)
107     {
108         ps.Message = "An error has occurred. Process Summary has invalid format";
109         await repository.Insert(ps);
110     }
111     else
112     {
113         List<PaymentAdvice> paymentAdvices = extractedPaymentAdvices.Select(epa => PaymentAdviceMapper.toPaymentAdvice(epa)).ToList();
114         var confirmed = configuration.GetValue<string>("RightStatus");
115         isGood = paymentAdvices.AsQueryable().Count(pa => !pa.Status.Equals(confirmed)) == 0;
116         ps.IsGood = isGood;
117         ps.paymentAdvices = paymentAdvices;
118         await repository.Insert(ps);
119     }
120     string folderToMove = isGood ? configuration.GetValue<string>("SuccessFolderToMove") : configuration.GetValue<string>("FailedFolderToMove");
121     folderToMove = Path.Combine(folderToMove, e.Name);
122     File.Move(e.FullPath, folderToMove, true);

```

Figura 64 - Processamento de Process Summary

6 Experimentação e Avaliação

No capítulo da experimentação e avaliação serão apresentados, explicados e discutidos os resultados obtidos e de que maneira os mesmos se posicionam face à hipótese proposta.

6.1 Hipótese

A hipótese de uma investigação deve procurar exprimir os objetivos propostos para o projeto de uma forma quantificável, verificável e testável. A sua definição é um dos passos essenciais na planificação da pesquisa científica quantitativa (Lavrakas, 2008).

Como enunciado no capítulo 1.3 um dos principais objetivos deste projeto passa pelo desenvolvimento de uma solução que automatize parcial ou completamente o processo de tratamento de avisos de pagamentos para um conjunto de clientes escolhidos. Adicionalmente e não menos importante, a solução desenvolvida deve demorar menos tempo a efetuar o tratamento de um aviso de pagamento do que o demorado por um humano para tratar o mesmo ficheiro.

Considerando os objetivos previamente enunciados, formula-se a hipótese científica: “**É possível automatizar o processo de tratamento de avisos de pagamento de forma que o mesmo seja mais eficiente que a sua versão manual?**”.

6.2 Indicadores e fontes de informação

De maneira a testar a hipótese formulada é necessário decidir como o fazer. Uma vez que a automatização foi feita recorrendo a **RPA**, foram exploradas possíveis formas de averiguar a desempenho desta abordagem.

Como possível de confirmar em (CiGen, 2020) e (Teodorczuk, 2021) existem várias técnicas para averiguar a desempenho de uma abordagem **RPA**, entre as quais foram destacadas:

- **VTG (Value of Time Gains):** Diferença de custos de um processo levado a cabo por trabalhadores humanos e o custo do processo levado a cabo pelos robôs **RPA**. Dentro dos custos de trabalhadores inclui-se vencimento, imposto, custo de escritório e outros. Em termos de custos de robôs consideram-se os custos das licenças, custo de desenvolvimento e custo de manutenção (Teodorczuk, 2021).
- **Cost Per Error:** Custo médio causado por um erro. Nesta métrica incluem-se a soma de valor perdido devido a tempo de inatividade e custo das horas necessárias para corrigir o erro.
- **Gained Productivity:** Exprime a relação em termos de **FTEs (Full Time Equivalent)** ganhos pela automatização do processo.

- **Improved Accuracy:** Métrica que avalia a quantidade de trabalho necessária para compensar erros cometidos antes e depois da implementação da abordagem **RPA**.
- **Workforce Impact:** O Impacto na força de trabalho pode ser medido sob um vasto leque de métricas que considerem ganhos face ao trabalho manual. Exemplos dessas métricas são número de horas de trabalho gastas por ano, número de trabalhadores realocados, entre outras.
- **Cycle Time:** Medida que tem em conta a velocidade de um processo, ou seja, a quantidade de tempo necessária para completar um processo. O tempo de ciclo depende do volume total de trabalho, pelo que se deve ter em conta estimativas temporais para os volumes de trabalho considerados.

Apesar de todas as técnicas de avaliação de **RPA** apresentadas serem válidas, devido ao contexto do projeto e da impossibilidade de obter e expor dados necessários para alguns dos cálculos ilustrados nos pontos supramencionados (e.g. vencimentos, custos), algumas das técnicas de avaliação de desempenho de abordagens RPA foram excluídas. Ademais, a hipótese de investigação elaborada pressupõe a medição de eficiência da solução automatizada e a sua comparação com a versão manual. Tome-se por eficiência a capacidade de executar uma tarefa da melhor maneira possível, com o menor desperdício de tempo, esforço e recursos necessários. Uma vez que existe uma troca parcial de recursos humanos por recursos informáticos, no contexto do projeto em mãos a eficiência foi traduzida para considerar apenas o menor tempo gasto a completar o processo de tratamento de avisos de pagamentos. Sendo assim, a métrica mais adequada para a medição de eficiência temporal passa pelo **Cycle Time**, que mede justamente a velocidade de um processo para um dado volume de trabalho.

A metodologia de avaliação seguida foi a avaliação por indicadores, considerando-se o indicador de performance temporal da solução. Este indicador pretende avaliar o ganho percentual entre a aplicação da abordagem automática face à abordagem manual, e é dado pela seguinte fórmula de cálculo:

$$IPerformance = \left(1 - \frac{Tempo\ Solução\ Automática}{Tempo\ Abordagem\ Manual}\right) * 100$$

6.3 Ambiente de Controlo

Tendo a hipótese de investigação sido formulada e um critério para avaliação da mesma sido definido, o presente subcapítulo pretende detalhar o tipo de ambiente controlado pensado para testar a hipótese.

Um **ambiente controlado** é uma área com parâmetros previamente especificados regulados (Mine Arc Systems, 2019). Embora esta definição se aplique preferencialmente a espaços físicos como laboratórios, o conceito pode ser transposto para um contexto virtual de software. Aos parâmetros controlados podemos também dar o nome de variáveis de controlo. **Variáveis de controlo** é tudo

aquilo que possa ser restringindo ou limitado, refletindo um valor constante, por poder influenciar o resultado de um estudo (Bhandari, 2022).

De maneira a definir as variáveis de controlo e as variáveis teste, deve-se formalizar os casos e cenários de teste pensados.

Cenários e casos de teste

Uma vez que a hipótese de investigação passa pela validação e comparação da eficiência da abordagem **RPA** face à abordagem manual, deverá fazer-se uso de um caso de teste de maneira a efetivar uma testagem da hipótese de investigação. Um caso de teste é um cenário utilizado para averiguar um certo objetivo dado um conjunto de ações e condições do sistema (VISURE, s.d.).

A construção de um bom caso de teste deve seguir certas recomendações, como as descritas em (VISURE, s.d.). Algumas dessas recomendações implicam a **descrição do caso, informações de teste**, e quando verificável **pré-condições**. Estas pré-condições e outras restrições encontradas serão apresentadas no caso de teste sob o nome de **Condições e Restrições**.

Devido à natureza da solução planeada e desenvolvida, durante a elaboração de casos de teste surgiram algumas **nuances**:

- 1- A solução foi pensada para realizar tanto o tratamento de vários avisos de pagamento como de apenas um, no entanto estima-se que se tire maior proveito quando o volume de avisos de pagamento é maior;
- 2- A solução não contempla a realização em paralelo de múltiplos avisos de pagamento, mas sim uma execução sequencial dos ficheiros obtidos;
- 3- Caso exista alguma discrepância no ficheiro e a mesma seja contemplada pela abordagem desenvolvida, o ficheiro deverá ser marcado de acordo com o tipo de discrepância detetada. No entanto, não compete ao robô fazer o tratamento adequado da discrepância. Esse tratamento terá de ser sempre efetuado manualmente, uma vez que não foi contemplado para esta versão da automatização do processo;
- 4- O processo executado corresponde à implementação da primeira fase da solução, ou seja, até à etapa 6 (inclusive) do processo definido em 3.1.

6.3.1.1 Caso de Teste 1: Processo Único Pré-validado

Descrição: Um colaborador Interno deverá cronometrar-se a realizar continuamente o processo de tratamento de um aviso de pagamento de um dos clientes abrangidos no âmbito do projeto, no ambiente de qualidade. O aviso de pagamento em questão deverá ter sido previamente tratado no ambiente de produção, garantido que o mesmo não apresenta discrepâncias. Adicionalmente, deverá ser anotado o tempo decorrido entre a hora de início e a hora de fim de realização do processo.

Por sua vez, o processo automatizado de **RPA** deverá ser executado **para o mesmo aviso de pagamento**, no ambiente de testes, cronometrando o tempo demorado desde o início até ao fim do processo.

Por fim, o colaborador fica responsável por verificar se o processo levado a cabo pelo robô da abordagem **RPA** produz os resultados esperados.

Condições e Restrições: De modo a constituir um caso de teste válido deverão ser tidas em conta as seguintes condições:

- 1- O colaborador não deverá intercalar esta atividade com outras derivadas das suas funções;
- 2- O colaborador deverá medir rigorosamente tempo gasto na realização da tarefa, assumindo qualquer pausa como uma pausa no cronómetro;
- 3- O colaborador não deverá realizar o processo de tratamento de um aviso de pagamento sobre um ficheiro que ele próprio tenha tratado anteriormente;
- 4- Os ambientes de qualidade e testes devem espelhar o ambiente de produção nos quesitos necessários para os clientes abrangidos pela automatização;
- 5- O aviso de pagamento em questão deverá pertencer a um dos clientes abrangidos no âmbito do projeto e deverá ter um mínimo de quatro páginas e um máximo de dez páginas.
- 6- Antes do colaborador iniciar o processo a execução automática do processo por parte do robô, deverá ser validado que não existem mais ficheiros no diretório configurado.

Pós-condições: Devem ser recolhidos os tempos demorados pelo colaborador Interno e pelo Robô, bem como um identificador (nome) e o número de páginas do ficheiro do Aviso de Pagamento em questão.

Variáveis de controlo:

- Cliente do ficheiro de aviso de pagamento
- Número de páginas do ficheiro *PDF*
- Número de ficheiros por processar no diretório do robô

6.3.1.2 Caso de Teste 2: Processo único com erros

Este caso de teste foi pensado com o intuito de averiguar as diferenças temporais entre o colaborador humano e o robô na deteção de erros num Aviso de Pagamento.

Descrição: Um colaborador Interno deverá cronometrar-se a realizar continuamente o processo de tratamento de um Aviso de Pagamento de um dos clientes abrangidos no âmbito do projeto, no ambiente de qualidade. O aviso de pagamento em questão deverá ter sido previamente tratado no ambiente de produção, garantido que o mesmo apresenta alguma discrepância entre os dados nele contidos. Adicionalmente, deverá ser anotado o tempo decorrido entre a hora de início e a hora de fim de realização do processo.

Por sua vez, o processo automatizado de **RPA** deverá ser executado **para o mesmo Aviso de Pagamento**, no ambiente de testes, cronometrando o tempo demorado desde o início até ao fim do processo.

Por fim, o colaborador fica responsável por verificar se o processo levado a cabo pelo robô da abordagem **RPA** produz os resultados esperados e de apontar o tempo gasto pelo mesmo.

Condições e Restrições: De modo a constituir um caso de teste válido deverão ser tidas em conta as seguintes condições:

- 1- De maneira a eliminar qualquer parcialidade, o colaborador não deve ser informado à priori que o Aviso de Pagamento em questão apresenta discrepâncias entre os seus valores, de modo a simular um caso real, onde também não seria avisado;
- 2- O colaborador não deverá intercalar esta atividade com outras derivadas das suas funções;
- 3- O colaborador deverá medir rigorosamente tempo gasto na realização da tarefa, assumindo qualquer pausa como uma pausa no cronómetro;
- 4- O colaborador não deverá realizar o processo de tratamento de avisos de pagamento sobre ficheiros que ele próprio tenha tratado anteriormente;
- 5- Os ambientes de qualidade e testes devem espelhar o ambiente de produção nos quesitos necessários para os clientes abrangidos pela automatização;
- 6- O aviso de pagamento em questão deverá pertencer a um dos clientes abrangidos no âmbito do projeto e deverá ter um mínimo de quatro páginas e um máximo de dez páginas.
- 7- Antes do colaborador iniciar o processo a execução automática do processo por parte do robô, deverá ser validado que não existem mais ficheiros no diretório configurado.

Pós-condições: Deve ser recolhido o tempo total demorado pelo colaborador Interno e pelo Robô na realização do tratamento do Aviso de Pagamento. Adicionalmente, devem ser recolhidos um identificador (nome) e o número total de páginas do ficheiro do Aviso de Pagamento em questão.

Variáveis de controlo:

- Cliente do ficheiro de aviso de pagamento;
- Número de páginas do ficheiro PDF;
- Número de ficheiros por processar no diretório do robô.

6.4 Avaliação de Resultados

Tendo explícitos a hipótese, o ambiente de controlo e os cenários de teste a realizar, o presente subcapítulo pretende expor e detalhar os resultados obtidos e conclusões plausíveis sobre os mesmos.

Foram efetuadas 8 repetições do **Caso de Teste 1: Processo Único Pré-validado** no espaço de 5 dias, resultados presentes nas Tabela 8. Aos resultados obtidos, foi aplicada a fórmula descrita em 6.2 de

modo a calcular o índice de performance da abordagem automática face à abordagem manual. Estes cálculos e outras métricas de análise relevantes encontram-se presentes na Tabela 9.

Identificador do Aviso de Pagamento	Número de Páginas	Tempo abordagem Manual (segundos)	Tempo abordagem RPA (segundos)
Cliente1_AP1	6	5370	266
Cliente1_AP2	4	3528	238
Cliente2_AP1	7	6922	269
Cliente2_AP2	7	6754	277
Cliente2_AP3	6	5048	269
Cliente3_AP1	5	4163	246
Cliente4_AP1	9	8922	323
Cliente5_AP1	6	5508	256

Tabela 8 - Resultados de Caso de Teste 1: Processo Único Pré-validado

Identificador do Aviso de Pagamento	Número de Páginas	Tempo abordagem Manual (segundos)	Tempo abordagem RPA (segundos)	Índice de Performance	Tempo /Página Ab. Manual (segundos)	Tempo /Página Ab. Automática (segundos)
Cliente1_AP1	6	5370	266	95.05%	895.00	44.33
Cliente1_AP2	4	3528	238	93.25%	882.00	59.50
Cliente2_AP1	7	6922	269	96.11%	988.86	38.43
Cliente2_AP2	7	6754	277	95.90%	964.86	39.57
Cliente2_AP3	6	5048	269	94.67%	841.33	44.83
Cliente3_AP1	5	4163	246	94.09%	832.60	49.20
Cliente4_AP1	9	8922	323	96.38%	991.33	35.89
Cliente5_AP1	6	5508	256	95.35%	918.00	42.67

Tabela 9 - Cálculo do índice de performance para os resultados dos testes realizados

De maneira a podermos retirar conclusões sobre os resultados, teremos de escolher um método que nos permita, com um grau de confiança confortável rejeitar ou não a hipótese nula. Esta rejeição é feita com base no valor apresentado no *p-value*, que é a probabilidade de obter resultados pelo menos tão extremos quanto os obtidos no teste de hipóteses correspondente, assumindo a hipótese nula como correta (Beers, 2022). Um menor *p-value* permite concluir que existem evidências mais favoráveis que apoiam a hipótese alternativa. Normalmente a distinção é feita para valores de *p-value* inferiores a 0.05, uma vez que este valor é o considerado o limite para a significância estatística (Beers, 2022).

Apesar de existirem vários testes estatísticos, conhecendo as nossas amostras nos quesitos de normalidade da distribuição, dependência e número de grupos e através da Figura 65 conseguimos determinar o teste paramétrico mais apropriado ao problema em questão.

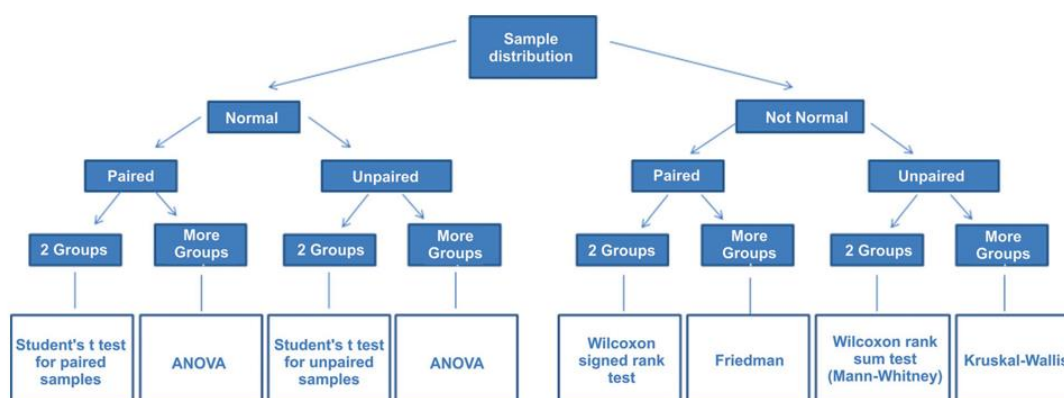


Figura 65 - Utilização de Testes Estatísticos

6.4.1 Avaliação das Amostras

Quanto ao quesito de **número de grupos** poderemos concluir que apenas estamos a avaliar dois grupos, um correspondente à abordagem manual e um correspondente à abordagem **RPA**.

Relativamente ao quesito de dependência entre as amostras, segundo (Minitab, s.d.), os conceitos de amostras dependentes e independentes são os seguintes:

- **Amostras dependentes:** Amostras dependentes provêm de medidas emparelhadas para um grupo de itens.
- **Amostras independentes:** Amostras independentes provêm de medidas para 2 grupos de itens diferentes, sem relação entre eles.

Como tal, uma vez que as medições efetuadas pressupunham a extração dos tempos nas diferentes abordagens para o mesmo conjunto de ficheiro, consideram-se as **amostras como dependentes**.

Quanto à normalidade da distribuição, a mesma foi testada através do teste de normalidade *Shapiro-Wilk* aplicado aos conjuntos de dados extraídos, estando o seu resultado e interpretação presentes na Tabela 10.

Abordagem	Resultado	Interpretação
Abordagem Manual	$W = 0.95874$, $p\text{-value} = 0.798$	Dado que o $p\text{-value}$ não é inferior a 0.05, não podemos rejeitar a hipótese nula, assumindo então a normalidade dos dados
Abordagem Automática	$W = 0.87399$, $p\text{-value} = 0.1648$	Dado que o $p\text{-value}$ não é inferior a 0.05, não podemos rejeitar a hipótese nula, assumindo então a normalidade dos dados

Tabela 10 - Resultados de Teste de *Shapiro-Wilk*

Tendo em conta a normalidade das distribuições, e a dependência entre os dois grupos das amostras, e fazendo uso da Figura 65, o teste escolhido foi o **Teste de *T Student* para amostras emparelhadas**.

6.4.2 Significância Estatística

O teste de *T Student* para amostras emparelhadas pretende avaliar se existe alguma significância estatística entre as médias de dois grupos. Ao ter em conta a natureza emparelhada dos dados, este teste tem uma maior precisão ao reduzir o impacto de diferenças individuais no todo.

Na realização deste teste, a hipótese nula significaria que não existe diferença significativa entre as medidas dos 2 grupos (a diferença média entre as medidas dos 2 grupos é 0). Aplicando este teste aos dados recolhidos, obtemos como valor $t = 9.2146$ e do $p\text{-value} = 3.659\text{e-}05$. Uma vez que o $p\text{-value}$ é menor que 0.05, é possível rejeitar a hipótese nula e afirmar que existe sim uma diferença significativa entre o tempo levado a cabo pela abordagem manual e pela abordagem automática na realização do processo de tratamento dos avisos de pagamento.

7 Conclusões

O presente capítulo dedica-se à apresentação de um resumo sobre o projeto desenvolvido, sendo explorados os objetivos atingidos, as principais limitações subjacentes ao projeto e uma apreciação final sobre o mesmo.

7.1 Objetivos Atingidos

Da análise do exposto no corpo da presente dissertação, foram elaborados (1.3) e abordados os vários objetivos subjacentes ao projeto. Resumidamente:

- Para o objetivo “Desenho de uma solução de automatização do processo de tratamento de aviso de pagamentos temporalmente mais eficiente que a abordagem manual” (1), foram elaborados diagramas representativos da mesma (4.2). A solução desenvolvida trata-se de uma solução faseada, composta por dois processos independentes, e permite tratar vários avisos de pagamento de uma única execução. Por sua vez, como verificado no capítulo 6, a solução desenvolvida é de facto temporalmente mais eficiente que a abordagem manual;
- Quanto ao objetivo “Implementação da 1ª fase (**etapas 1-6**) da solução desenhada para automatização do processo, para 5 clientes pré-selecionados pela organização Inapa”(2), a solução concebida foi implementada recorrendo à ferramenta UiPath, e os seus detalhes podem ser consultados na secção 5.1. A implementação da solução focou-se na automatização e adaptação do processo existente às necessidades automáticas, reduzindo a necessidade de intervenção manual no processo em questão;
- Para o objetivo “Desenvolvimento de mecanismos de visibilidade, transparência e controlo sobre os ficheiros validados e tratados (detalhados em 3.2.1)” (3), foram pensados 3 mecanismos complementares. Por um lado, recorrendo a ficheiros *Excel Process Summary* e *Payment Advice*, é possível manter sempre registado que avisos de pagamento foram considerados para cada execução e o seu estado, e quais os detalhes e validações efetuados para cada aviso de pagamento, respetivamente. Por fim, foi ainda desenvolvida uma solução de consumo dos ficheiros *Process Summary* de modo a auxiliar os colaboradores Inapa a rapidamente avaliar a prioridade de tratamento dos ficheiros em erro.

Tendo em conta o exposto, considera-se que todos os objetivos previamente enunciados foram plenamente atingidos.

7.2 Limitações e Trabalho Futuro

No decorrer do desenvolvimento do projeto foram notadas algumas limitações, especialmente no que toca ao processamento textual de informação não estruturada dos ficheiros *PDF*. Um exemplo específico passa pela implementação do subprocesso responsável por esta extração de informação (5.1.3.2) em que no caso de não serem recuperados pelo menos dois campos, torna-se imensamente difícil perceber quais dos campos vieram com os valores em falta. Outra limitação do processo passa pela possibilidade de existência de vários movimentos do mesmo cliente para a mesma data com exatamente o mesmo montante. Uma vez que os ficheiros *PDF* estão subdivididos em diretórios por cliente, a coexistência de dois ficheiros com o mesmo nome não é possível, sendo necessário que só depois de um ser processado é que se possa copiar o outro.

Em termos de trabalho futuro, destacam-se a implementação da segunda fase da automatização do processo, a automatização da recolha e da mudança do nome dos ficheiros *PDF* dos avisos de pagamento (atualmente recolhida do email e realizada manualmente pelos colaboradores), a aplicação de mecanismos de comunicação em tempo real entre as aplicações *Dashboard* e *Folder Monitor*, e a extensão da automatização do processo para mais clientes Inapa.

7.3 Apreciação Final

Considerando todo o trabalho desenvolvido desde o estudo, incluindo a análise e implementação da solução desenhada, considera-se que a automatização parcial do processo de tratamento de avisos de pagamento foi realizada com sucesso, sendo uma abordagem temporalmente mais eficiente que a contraparte manual, como comprovado em 6.

8 Bibliografia

altexsoft, 2022. *Non-functional Requirements: Examples, Types, How to Approach*. [Online]
Available at: <https://www.altexsoft.com/blog/non-functional-requirements/>
[Acedido em 21 fevereiro 2023].

Automation Anywhere, 2020. *Automation Anywhere Version 11.3*. [Online]
Available at: <https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/aae-architecture-implementation/client-overview.html>
[Acedido em 06 02 2023].

Beers, B., 2022. *P-Value: What It Is, How to Calculate It, and Why It Matters*. [Online]
Available at: <https://www.investopedia.com/terms/p/p-value.asp>
[Acedido em 11 fevereiro 2023].

Bhandari, P., 2022. *Control Variables | What Are They & Why Do They Matter?*. [Online]
Available at: <https://www.scribbr.com/methodology/control-variable/>
[Acedido em 8 fevereiro 2023].

Capterra, s.d. *ABBYY FlexiCapture*. [Online]
Available at: <https://www.capterra.com/p/177446/ABBYY/>
[Acedido em 04 fevereiro 2023].

Capterra, s.d. *Compare software and service providers, and read verified reviews*. [Online]
Available at: <https://www.capterra.com/>
[Acedido em 04 fevereiro 2023].

Capterra, s.d. *Tipalti*. [Online]
Available at: <https://www.capterra.com/p/236980/Tipalti/#prosCons>
[Acedido em 04 fevereiro 2023].

Castro, B. A., s.d. *Como fazer diagrama de processos? Passo a passo com exemplos*. [Online]
Available at: <https://blog.zeev.it/como-fazer-diagrama-de-processos/>
[Acedido em 18 fevereiro 2023].

CiGen, 2020. *Measuring RPA: 10 Performance Metrics for Assessing Robotic Process Automation Benefits*. [Online]
Available at: <https://www.cigen.com.au/measuring-rpa-10-performance-metrics-assessing-rpa-benefits/>
[Acedido em 7 fevereiro 2022].

Deloitte, s.d. *Perspectivas Robotic Process Automation (RPA)*. [Online]
Available at: <https://www2.deloitte.com/pt/pt/pages/strategy-operations/articles/robotic-process-automation.html>

[Acedido em 01 Fevereiro 2023].

Enríquez, J., Jiménez, A., Domínguez, F. & García, J., 2020. Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study. 4 Março, p. 1.

Eschberger-Friedl, T., 2018. *The fuzzy front end of the innovation process*. [Online]
Available at: <https://www.lead-innovation.com/en/insights/english-blog/fuzzy-front-end>
[Acedido em 17 fevereiro 2023].

Jayawardene, P. D., 2021. *4+1 Architectural view model in Software*. [Online]
Available at: <https://medium.com/javarevisited/4-1-architectural-view-model-in-software-ec407bf27258>
[Acedido em 27 04 2023].

Jiang, L. & Zhang, K., 2018. Apparel brand overlap based on customer perceived value and eye-tracking technology. *Tsinghua Science and Technology*, vol. 23, no. 1, fevereiro, p. 48.

Johns Hopkins Medical Institutions, 2006. *FBL5N - View Open Receivables*. [Online]
Available at: <https://ssc.jhmi.edu/accountsreceivable/training/fbl5n.html>
[Acedido em 14 fevereiro 2023].

Kambil, A. & Ari, G., 1996. Re-Inventing Value Propositions. janeiro, p. 3.

Koen, P. A., 2007. The Fuzzy Front End for Incremental, Platform, and Breakthrough Products. 15 novembro, p. 81.

Koen, P. A. et al., 2002. Fuzzy Front End : Effective Methods , Tools , and Techniques. pp. 19-20.

Koen, P. A., Bertels, H. M. & Kleinschmidt, E. J., 2014. Managing the Front End of Innovation—Part II. p. 26.

Koen, P. A., Bertels, H. M. & Kleinschmidt, E., 2014. Managing the Front End of Innovation—Part I.: *Research-Technology Management*.

Kralj, K., s.d. *5 Reasons You Need Dependency Injection in Your C# Code*. [Online]
Available at: <https://methodpoet.com/dependency-injection-benefits/>
[Acedido em 01 05 2023].

Lavrakas, P., 2008. *Encyclopedia of Survey Research Methods*. s.l.:SAGE.

Mine Arc Systems, 2019. *What is a Controlled Environment?*. [Online]
Available at: <https://minearc.com/what-is-a-controlled-environment/?cn-reloaded=1>
[Acedido em 8 fevereiro 2023].

Minitab, 2019. *Como escolher entre um teste não paramétrico e um teste paramétrico*. [Online]
Available at: <https://blog.minitab.com/pt/como-escolher-entre-um-teste-nao-parametrico-e-um-teste-parametrico>

[Acedido em 11 fevereiro 2023].

Minitab, s.d. *How are dependent and independent samples different?*. [Online]
Available at: <https://support.minitab.com/en-us/minitab/21/help-and-how-to/statistics/basic-statistics/supporting-topics/tests-of-means/how-are-dependent-and-independent-samples-different/>

[Acedido em 11 fevereiro 2023].

Osterwalder, A., Pigneur, Y., Bernarda, G. & Smith, A., 2014. Em: *Value Proposition Design: How to Create Products and Services Customers Want*. s.l.:Wiley, pp. 8-9.

Paixão, J. R., 2019. *O que é SOLID: O guia completo para você entender os 5 princípios da POO*. [Online]

Available at: <https://medium.com/desenvolvendo-com-paixao/o-que-%C3%A9-solid-o-guia-completo-para-voc%C3%AA-entender-os-5-princ%C3%ADpios-da-poo-2b937b3fc530>

[Acedido em 29 04 2023].

Rossum, s.d. *About Rossum*. [Online]

Available at: <https://rosum.ai/company/>

[Acedido em 02 fevereiro 2023].

Rossum, s.d. *The Rossum Plataform*. [Online]

Available at: <https://rosum.ai/platform/>

[Acedido em 02 fevereiro 2023].

SAP SE, s.d. *SAP*. [Online]

Available at: <https://www.sap.com/portugal/index.html>

[Acedido em 31 01 2023].

SAP, 2006. *VF03 - Display Billing Document*. [Online]

Available at: <http://web.mit.edu/cao/www/SB2002/CR/VF03.htm>

[Acedido em 14 fevereiro 2023].

SE80, s.d. *FEBA SAP Postprocess Electronic Bank Statmt Transaction Code*. [Online]

Available at: <https://www.se80.co.uk/sap-tcodes/?name=feba>

[Acedido em 14 fevereiro 2023].

Seng, J. K. A. K. L., 2019. *Multimodal Emotion and Sentiment Modeling From Unstructured Big Data: Challenges,Architecture, & Techniques*. 25 Julho, p. 1.

Techlorean, 2019. *SAP ELECTRONIC BANK STATEMENT: BASIC PROCESS AND OVERVIEW*. [Online]

Available at: <https://techlorean.com/2019/12/18/electronic-bank-statement-basic-process-and->

overview/

[Acedido em 14 fevereiro 2023].

Teodorczuk, B., 2021. *How To Measure RPA Success? A Guide to Robotic Process Automation Metrics*. [Online]

Available at: <https://flobotics.io/blog/rpa-successs-metrics/>

[Acedido em 7 fevereiro 2023].

Tipalti, s.d. *Transform the way your finance team works*. [Online]

Available at: <https://tipalti.com/en-eu/brand->

[ptr/?cq_src=google_ads&cq_cmp=18209983159&cq_con=145902955852&cq_term=tipalti&cq_med=&cq_plac=&cq_net=g&cq_pos=&cq_plt=gp&utm_source=google&utm_medium=cpc&utm_term=tipalti&utm_campaign=G_S_EU_Brand_General&utm_adgroup=Tipal](https://tipalti.com/en-eu/brand-ptr/?cq_src=google_ads&cq_cmp=18209983159&cq_con=145902955852&cq_term=tipalti&cq_med=&cq_plac=&cq_net=g&cq_pos=&cq_plt=gp&utm_source=google&utm_medium=cpc&utm_term=tipalti&utm_campaign=G_S_EU_Brand_General&utm_adgroup=Tipal)

[Acedido em 04 fevereiro 2023].

Tripathi, J., 2023. *Top 10 Invoice Processing Automation Software in 2023*. [Online]

Available at: <https://www.docsumo.com/blog/best-invoice-automation-software>

[Acedido em 02 fevereiro 2023].

UiPath, s.d. *Extract Structured Data*. [Online]

Available at: <https://docs.uipath.com/activities/other/latest/user-guide/extract-data>

[Acedido em 24 05 2023].

Vajgel, B. et al., 2021. *Development of Intelligent Robotic Process Automation: A utility Case Study in Brazil*, s.l.: s.n.

Visual Paradigm, s.d. *BPMN Activity Types Explained*. [Online]

Available at: <https://www.visual-paradigm.com/guide/bpmn/bpmn-activity-types-explained/>

[Acedido em 16 fevereiro 2023].

VISURE, s.d. *O que são casos de teste? Como escrever casos de teste relacionados a software?*.

[Online]

Available at: <https://visuresolutions.com/pt/what-are-test-cases-how-to-write-software-related-test-cases>

VISURE, s.d. *O que são Requisitos Funcionais: Exemplos, Definição, Guia Completo*. [Online]

Available at: <https://visuresolutions.com/pt/blog/functional-requirements/>

[Acedido em 16 fevereiro 2023].

Walpita, P., 2019. *Software Architecture Patterns — Layered Architecture*. [Online]

Available at: <https://priyalwalpita.medium.com/software-architecture-patterns-layered-architecture-a3b89b71a057>

[Acedido em 29 04 2023].

Widianta, M. & Rizaldi, T., 2018. Comparison of Multi-Criteria Decision Support Methods (AHP, TOPSIS, SAW & PROMENTHEE) for Employee Placement. *Journal of Physics: Conference Series Vol. 953*, pp. 5-6.

Woodall, T., 2003. Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis. janeiro.

Zach, 2021. *Is Time An Interval or Ratio Variable? (Explanation & Example)*. [Online]
Available at: <https://www.statology.org/is-time-interval-or-ratio/>
[Acedido em 11 fevereiro 2023].