



UNIVERSIDADE
LUSÓFONA

Robot Process Automation

Componente 1

Trabalho Final de curso

Relatório Intercalar 2º Semestre

Alexandre Santos de Sousa e Costa (a22007578)

João Pedro Cunha da Silva Eleutério (a22007237)

Rui Pedro Nobre Ribeiro

Trabalho Final de Curso | LEI | junho 2023

www.lusofona.pt

Direitos de cópia

Robotic Process Automation, Copyright de (Alexandre Santos de Sousa e Costa, João Pedro Cunha da Silva Eleutério), ULHT.

Com a realização do trabalho final de curso, em parceria com a CGI e realização do acordo assinado por ambas as partes relativamente à não divulgação de informação (NDA), todos os direitos são reservados à entidade externa (CGI). É proibida a publicação e distribuição deste relatório e quaisquer informações acerca do mesmo, tendo apenas autorização para realizar a partilha do documento em questão com os membros do júri e respectivos responsáveis, com o objectivo da avaliação do trabalho. Este termo aplica-se a todos os links associados a desenhos (Arquiteturas, Mockups, Protótipos, entre outros...) e repositório de código.

Confirmando-se a finalização da cadeira “TFC”, o respectivo relatório deverá ser eliminado, e os acessos aos links descritos, removidos.

Resumo

No âmbito da UC de TFC, foi proposto a execução de um trabalho que visa o desenho, orquestração e monitorização de processos, isto é, prevenir possíveis erros indesejáveis de RPA's (Robotic Process Automation). Estes são softwares que consistem na automação de processos de negócios, baseados em robôs ou em inteligência artificial.

As organizações começaram a usar RPA's para operações que não constituem o núcleo da organização. Porém, posteriormente, estes começaram a desempenhar um papel mais crucial dentro das empresas, oferecendo assim respostas mais funcionais, provocando uma dependência destes na mesma.

Apesar dos RPA's serem softwares previamente definidos e inteligentes, estes podem cometer erros cruciais para o bom funcionamento da empresa.

Este projeto consiste em monitorizar, orquestrar e desenhar um método que previne todas estas discrepâncias que são colocadas pelos RPA's. O objetivo deste trabalho final de curso foi criar um software que ofereça uma maior segurança à empresa, uma vez que estes erros podem ser notificados aos responsáveis de manutenção do sistema.

O produto final consiste uma plataforma (Process Monitoring) em que os operacionais conseguem corrigir os processos incorretamente executados pelo RPA, se não for chefe de equipa. Caso o operacional seja chefe de equipa, irá receber diariamente quantas tarefas tem a realizar no dia e quantas realizou, respetivamente, no dia anterior; tal como consegue visualizar os processos que a equipa tem a realizar; e adicionar skills aos restantes membros da equipa. Enquanto o analista, consegue verificar estatísticas sobre o bom ou mau funcionamento das tarefas atribuídas às equipas, tendo assim em conta um plano mais geral dos processos que são ou não feitos. Por último, o administrador consegue ver quantas tarefas por realizar ou realizadas cada utilizador tem, alterar a equipa ou o cargo de cada utilizador, criar equipas e apagá-las e mudar o chefe de equipa. Ainda, consegue atribuir os processos às equipas, adicionar e remover skills, e aceder a todas as páginas. (hierarquia: 1-admin; 2-analista; 3-operacional).

A plataforma tem uma página para reportar erros relativamente à plataforma, caso estes existam, sendo que os desenvolvedores da plataforma estarão contactáveis para a resolução de qualquer percalço.

Abstract

In the scope of the TFC UC, it was proposed the execution of a work that aims at the design, orchestration and monitoring of processes, i.e. preventing possible undesirable errors from RPA's (Robotic Process Automation). These are software that consist in the automation of business processes, based on robots or artificial intelligence.

Organizations started using RPA's for operations that are not the core of the organization. However, later on, these began to play a more crucial role within companies, thus offering more functional answers, causing a dependency on them within the company.

Although RPA's are pre-defined and intelligent software, they can make mistakes that are crucial for the proper functioning of the company.

This project consists in monitoring, orchestrating and designing a method that prevents all these discrepancies that may be posed by RPA's. The objective of this final course work is to create a software that offers greater security to the company, since these errors can be notified to those responsible for maintaining the system.

The final product is a platform (Process Monitoring) in which the operatives can correct the processes incorrectly executed by the RPA, if it is not a team leader. If the operative is a team leader, he/she will receive daily how many tasks he/she has to perform that day and how many he/she performed, respectively, the day before; he/she can also visualize the processes that the team has to perform; and add skills to the other team members. The analyst, meanwhile, can check statistics on how well or poorly the tasks assigned to the teams are working, thus having a more general plan of what processes are or are not getting done. Finally, the administrator can see how many outstanding or accomplished tasks each user has, change the team or role of each user, create teams, and delete them, and change the team leader. He can also assign processes to teams, and remove skills, and access all pages (hierarchy: 1-admin; 2-analyst; 3-operational).

The platform has a page to report bugs regarding the platform, if there are any, and the platform developers will be contactable for the resolution of any mishaps.

Índice

Resumo	iii
Abstract.....	iv
Índice.....	v
Lista de Figuras.....	vii
Lista de Tabelas	viii
1 Identificação do Problema	1
2 - Viabilidade e Pertinência.....	3
3 - Benchmarking	4
4 - Engenharia	6
4.1 Levantamento e análise de requisitos	6
4.1.1 Requisitos funcionais.....	6
4.1.2 Requisitos não funcionais.....	10
4.2 Diagramas de caso de uso	15
4.3 Diagramas de atividade (BPMN).....	17
4.4 Modelo de dados.....	18
4.5 Estrutura.....	19
4.6 User Stories	20
5 - Solução Proposta.....	21
5.1 Introdução.....	21
5.2 Apresentação da arquitetura.....	22
5.3 Tecnologias e Ferramentas Utilizadas	23
5.4 Implementação	23
5.5 Abrangência	24
6 - Método de planeamento	26
7 - Resultados.....	27
8 - Conclusão e trabalhos futuros.....	32
Bibliografia	34
Anexo 1 – RPA email	35
Anexo 2 - RPA formulário	39
Anexo 3 - Progresso de trabalho	46

Glossário49

Lista de Figuras

Figura 1 - Caso de Uso - Atribuição de tarefas de processos de envio de emails aos operacionais	15
Figura 2 - Caso de Uso - Análise de erros	16
Figura 3 - Caso de Uso - Uso do RPA	16
Figura 4 - Diagrama de atividade (BPMN)	17
Figura 5 - Modelo de dados (UML)	18
Figura 6 - Mapa Aplicacional	19
Figura 7 - Desenho da Arquitetura a implementar	22
Figura 8 - RPA Emails - Main	35
Figura 9 - RPA Emails - GetData	36
Figura 10 - RPA Emails - ProcessData	37
Figura 11 - RPA formulário - Main	39
Figura 12 - RPA formulário - GetData	40
Figura 13 - RPA formulário - ProcessData(1)	41
Figura 14 - RPA formulário - ProcessData(2)	42
Figura 15 - RPA formulário - ProcessData(3)	43
Figura 16 - RPA formulário - ProcessData(4)	44
Figura 17 - RPA formulário - ProcessData(5)	45
Figura 18 - Cronograma 1ª, 2ª e 3ª entrega	46
Figura 19 - Cronograma 3ª e 4ª entrega	47

Lista de Tabelas

Tabela 1 - Requisitos funcionais	10
Tabela 2 - Requisitos não funcionais	14
Tabela 3 - Testes e resultados	31

1 Identificação do Problema

As organizações começaram a usar RPA's por diversos motivos, sendo um deles a automatização de tarefas consideradas menos críticas e rotineiras para operações que não eram core da organização com alta eficiência, reduzindo a possibilidade de erros humanos e aumentando a eficiência dos processos empresariais. Um exemplo possível seria, o RPA, consoante as informações obtidas, enviar um email a uma lista de pessoas, que por si, não seria aleatória. Posteriormente, estes começaram a fazer tarefas consideradas mais cruciais dentro de uma empresa, o que fez com que começasse a haver uma maior dependência para o bom funcionamento da mesma.

Esta dependência deu-se por este software fazer uso da utilização de modelos de Inteligência Artificial para que o processo de tomada de decisão fosse mais inteligente e precisa, sendo menos propício a erros. Este procedimento fez com que não fosse necessária a presença da intervenção humana.

No entanto, podem existir falhas nos RPA's, podendo, assim, causar grandes prejuízos às empresas, o que se considera ser o problema ao qual este projeto se propõe a solucionar.

Imaginemos que um dos sistemas mudou e o robot continua a aceder ao campo previamente definido, isto significa que este continua a seguir os mesmos comandos podendo assim fornecer informação incorreta, não obtendo, desta forma, a resposta escolhida pelo software. Este problema é considerado muito grave, pois o RPA consegue efetuar milhares de operações por segundo, o que não resulta em apenas um erro, mas sim num número bastante elevado. É importante ressaltar que a ocorrência de falhas nos RPA's pode gerar impactos financeiros significativos para as empresas, além de prejudicar a eficiência e a produtividade dos processos empresariais. *((2) RPA: Oportunidade e inovação. Saiba como adotar esta tecnologia sem investimento inicial | LinkedIn).*

Para evitar esses problemas, é importante identificar as causas das falhas nos RPA's e encontrar maneiras de preveni-las ou minimizá-las. Isso pode envolver a implementação de medidas de segurança adicionais ou a realização de atualizações regulares nos sistemas conectados aos RPA's. Além disso, é fundamental monitorizar regularmente o desempenho dos RPA's e implementar procedimentos de correção rápida para lidar com quaisquer problemas que surjam.

Para a resolução deste tipo de problemas, existem várias partes que facultam a resolução do mesmo, como o administrador, que gere as equipas operacionais e processos; o

analista que analisa as estatísticas; os operacionais que são trabalhadores que fazem a correção quando existe um warning. Quando existe um erro, seja fatal error ou warning, é notificado o responsável do processo para emitir um alerta.

Com isto, o Back-End recebe a informação dos RPA's e dos operacionais através do Orchestrator; gera alarmística, análise e validação de logs, anteriormente enviadas pelo Orchestrator, e fazer configurações; ainda, são criados objetos e tabelas e a base de dados que servirá, posteriormente, para guardar toda a informação que é exposta.

2 - Viabilidade e Pertinência

A plataforma *Process Monitoring* tem como finalidade a monitorização e orquestração de processo, sendo uma tecnologia que traz grandes vantagens às empresas ao permitir a otimização dos processos de negócio e aumentar a escalabilidade e eficiência de diversos serviços IT, por meio da combinação de várias tecnologias e suas interações. Esta tecnologia fornece grandes vantagens às empresas, pois estas deixam de ter a necessidade de realizar manualmente certas funções, podendo assim, dedicar a sua mão de obra a outras áreas de maior importância. Ou seja, os RPA's fazem a grande parte das tarefas, mas com a plataforma, mesmo que esses dêem erro (mínimos ou desastrosos), é possível, facilmente, a visualização e correção dos mesmos por humanos (operacionais). Dependendo das skills dos operacionais e da equipa em que este se encontra, é possível realizar um determinado tipo de processo. As empresas, assim, obtêm um alto grau de automatização, resultando num aumento de produtividade, pois os RPA's executam as tarefas de forma mais diligente do que qualquer outro funcionário, existindo conseqüentemente maiores lucros e menores custos.

Todavia, os RPA's, são robôs, que efetuam milhares de operações por segundo, e quando cometem erros durante a execução de determinadas tarefas, os resultados podem ser desastrosos para a empresa. A detecção e correção desses erros é crucial para garantir o bom funcionamento das empresas que usam esse tipo de tecnologia, diminuindo possíveis prejuízos e aumentando a produtividade. Neste caso, em específico, é possível verificar uma funcionalidade na plataforma desenvolvida (*Process Monitoring*) que permite aos operacionais perceber e corrigir erros que possam ser prejudiciais à empresa.

Caso ocorram erros, e considerando a quantidade de operações que um RPA pode realizar, é necessário que haja uma forma de comunicação imediata ao responsável do ocorrido. Para isso, a plataforma envia um email ao chefe de equipa responsável por esse tipo de processo. No entanto, esta funcionalidade serve apenas para fins demonstrativos e, estes avisos são enviados, de tempo em tempo, para evitar repetições excessivas do mesmo erro, no mesmo dia.

Tendo por base o problema que foi identificado na secção anterior e a solução proposta nas secções seguintes, a pertinência deste trabalho inclui a boa funcionalidade de empresas que usem este tipo de robôs, como a diminuição de possíveis erros que possam ser desastrosos ou prejudiciais para as mesmas, sendo que a plataforma desenvolvida oferece meios de monitorização e correção de processos para a prevenção de possíveis danos.

3 - Benchmarking

Após a realização de uma pesquisa, no site *Gartner*, foram identificadas algumas empresas que trabalham com RPA's e que utilizam os mesmos conceitos que serão utilizados, entre outras. Sendo que, a nossa componente procura desenvolver uma aplicação que consiste em monitorizar o funcionamento normal dos RPA's, de maneira a que não haja erros ou complicações para a empresa. Se houver erros no funcionamento dos RPA's, é alertado o responsável do processo e é colocado numa lista de processos para um operacional ir solucioná-lo.

A *Appian Corporation*, ajuda a criar fluxos de trabalho rapidamente, com uma plataforma de automação de pouco código. A *Appian* é líder global em alguns mercados de tecnologia empresarial, incluindo no desenvolvimento de aplicações de pouco código, automação de processos digitais (DPA), sistemas inteligentes de gestão de processos de negócios (IBPMS) e gestão dinâmica de casos (DCM). Conseguem ainda rastrear todas as atividades de automação.

A *PegaSystems* utiliza a *Pega Platform* que torna mais simples para as empresas trabalharem de forma mais inteligente e adaptarem-se instantaneamente. Esta empresa usa alguns conceitos como: RPA atendido, ou seja, RPA's e humanos a trabalhar lado a lado; RPA's não assistido, ou seja, que não precisam de interação humana para funcionar normalmente; Visão raio-x, que é um IA para automação de autocorreção.

A *Automation Anywhere* oferece uma plataforma de força de trabalho digital mais sofisticada do mundo, tornando o trabalho humano mais automatizado. Esta empresa usa conceitos como: Automação de documentos, que visa extrair e incorporar dados de documentos ao fluxo de trabalho, para qualquer processo de negócios usando o IDP, RPA e IA - descoberta de processo, que identifica continuamente ineficiências e variações de processos em toda a empresa para reduzir o tempo de automação e transformação.

Para compor mais a nossa aplicação, utilizaram-se algumas técnicas de alarmística (enviar emails), embora estas técnicas devam ser realizadas por uma aplicação externa, como por exemplo, o *Grafana*, um software.

O *Grafana Cloud* é uma plataforma de observabilidade aberta e combinável que reúne métricas, logs e rastreamentos com visualizações do Grafana. Construído para ambientes nativos de nuvem e alimentado pelo melhor software de observabilidade de código aberto, o Grafana Cloud permite a concentração em permitir a observabilidade, sem a sobrecarga de construir, instalar, manter e dimensionar a pilha de observabilidade.

Para concluir, estas empresas são algumas das melhores do mercado nesta área, em que usam alguns conceitos que serão utilizados neste projeto para conseguir monitorizar processos e extrair dados para a base de dados. Pode-se verificar que a plataforma desenvolvida apresenta várias características compartilhadas com as plataformas referidas anteriormente, como é o caso da *PegaSystems*, que utiliza RPA's não assistidos; e da *Automation Anywhere*, no sentido em que a plataforma desenvolvida também tem como função identificar continuamente ineficiências e variações de processos em toda a empresa.

(Grafana Cloud | Observability platform overview, sem data)

(Mitchell, Gartner Magic Quadrant for Robotic Process Automation (RPA) 2022, 2022)

(Robotic Process Automation (RPA) Platform & Solutions | Appian)

(Robotic process automation (RPA) software | Pega)

(RPA Workspace | Automation Platform | Automation Anywhere)

4 - Engenharia

4.1 Levantamento e análise de requisitos

Os requisitos do trabalho final de curso descrevem como o back-end foi elaborado. Estes foram desenvolvidos no início da elaboração do projeto e estão divididos em requisitos funcionais e não funcionais.

4.1.1 Requisitos funcionais

ID	Classificação	Requisito	MoSCoW	Cumprimento
RF1	Alarmística	É necessário ter um Sistema que envia um e-mail ao chefe de equipa quando de um processo obter um erro fatal	Must Have	Implementado totalmente
RF2	Alarmística	É necessário ter um Sistema que envia um e-mail ao chefe de equipa quando de um processo obter um warning.	Must Have	Implementado totalmente
RF3	Alarmística	É necessário ter um Sistema que envia e-mail ao chefe (operacional) de cada equipa de operacionais com o número de tarefas que a mesma tem de realizar nesse dia, juntamente com o resumo do dia de trabalho anterior.	Should Have	Implementado totalmente
RF4	Alarmística	É necessário ter um Sistema que envia um e-mail ao responsável de um processo de negócio, quando um determinado RPA ou Humano está com um nível de objetivo abaixo do esperado.	Must Have	Não implementado
RF5	Base de Dados	Criar uma base de dados que armazena os dados dos processos de negócio.	Must Have	Implementado totalmente
RF6	Base de Dados	A base de dados necessita de ser flexível e estar preparada para eventuais alterações nos templates dos processos.	Must Have	Implementado totalmente
RF7	Base de Dados	A base de dados necessita de estar preparada para armazenar os dados do sistema de login, armazenando o e-mail e a respetiva palavra-passe.	Must Have	Implementado totalmente
RF8	Base de Dados	A base de dados necessita de estar preparada para armazenar os dados do que os RPA's vão produzir que são os logs.	Must Have	Implementado totalmente
RF9	Base de Dados	A base de dados necessita de estar preparada para armazenar os dados que o humano produz que são os logs.	Must Have	Implementado totalmente

RF10	Base de Dados	A base de dados necessita de estar preparada para armazenar os dados que extrai do UiPath Orchestrator.	Must Have	Implementado totalmente
RF11	Base de Dados	A base de dados deve conseguir guardar as correções efetuadas pelos operacionais.	Must Have	Implementado totalmente
RF12	Base de Dados	A base de dados deve conseguir guardar os reports de erros no sistema gerados pelos utilizadores.	Could Have	Novo e implementado totalmente
RF13	Base de Dados	A base de dados deve conseguir guardar as tarefas que cada equipa de operacionais tem pendentes.	Must Have	Implementado totalmente
RF14	Base de Dados	A base de dados deve conseguir guardar as tarefas que cada operacional tem pendente.	Must Have	Implementado totalmente
RF15	Base de Dados	A base de dados deve conseguir guardar o tipo das tarefas.	Must Have	Implementado totalmente
RF16	Base de Dados	A base de dados deve conseguir guardar KPI's	Must Have	Implementado totalmente
RF17	Base de Dados	A base de dados deve conseguir guardar o tipo dos Processos.	Must Have	Implementado totalmente
RF18	Base de Dados	A base de dados deve conseguir guardar os utilizadores da aplicação.	Must Have	Implementado totalmente
RF19	Base de Dados	A base de dados deve conseguir guardar processos de configuração.	Must Have	Implementado totalmente
RF20	Base de Dados	A base de dados deve conseguir guardar tarefas de configuração.	Must Have	Implementado totalmente
RF21	Base de Dados	A base de dados deve conseguir guardar todos os processos criados e extraídos do orchestrator	Must Have	Implementado totalmente
RF22	Base de Dados	A base de dados deve conseguir guardar todas as tarefas de um processo criadas e extraídas do orchestrator	Must Have	Implementado totalmente
RF23	Base de Dados	A base de dados deve conseguir guardar labels para serem atribuídas a processos para filtragens do Front-End	Must Have	Implementado totalmente
RF24	Base de Dados	A base de dados deve conseguir guardar equipas de trabalho	Must Have	Implementado totalmente
RF25	Base de Dados	A base de dados deve conseguir guardar os tipos de logs	Must Have	Implementado totalmente

RF26	Estatística	É necessário ter um Sistema que guarda o tempo de execução da tarefa no ficheiro de logs de cada processo.	Must Have	Modificado
RF27	Estatística	É necessário ter um Sistema que guarda o local/plataforma onde a tarefa foi executada no ficheiro de logs de cada processo.	Must Have	Modificado
RF28	Estatística	É necessário ter um Sistema que guarda a mensagem da tarefa no ficheiro de logs de cada processo.	Must Have	Modificado
RF29	Estatística	É necessário ter um Sistema que guarda o tipo de log da operação no ficheiro de logs de cada processo.	Must Have	Modificado
RF30	Estatística	É necessário ter um Sistema que guarda o ID do processo no ficheiro de logs de cada processo.	Must Have	Modificado
RF31	Estatística	É necessário ter um Sistema que guarda o nome do processo de configuração no ficheiro de logs de cada processo.	Must Have	Modificado
RF32	Log In	É necessário ter um Sistema de envio de um código para o e-mail de um utilizador quando este pretende fazer a recuperação das credenciais.	Must Have	Implementado totalmente
RF33	Sistema	É necessário ter um Sistema que organize os processos pelas suas prioridades.	Must Have	Retirado
RF34	Sistema	É necessário ter um Sistema que capta as informações fornecidas pelos RPA's.	Must Have	Implementado totalmente
RF35	Sistema	É necessário ter um Sistema que valida e filtra os logs fornecidos pelos RPA's.	Must Have	Implementado totalmente
RF36	Sistema	É necessário ter um Sistema que analise os logs fornecidos pelos RPA's.	Must Have	Implementado totalmente
RF37	Sistema	Após analisar os logs é necessário verificar se o nível de objetivo dos RPA's é superior a X%.	Must Have	Não implementado
RF38	Sistema	Após analisar os logs é necessário verificar se o nível de objetivo dos operacionais é superior a X%.	Must Have	Não implementado
RF39	Sistema	É necessário ter um Sistema que introduza na base de dados o ciclo de vida do processo com todas as suas etapas.	Should Have	Implementado totalmente
RF40	Sistema	Quando os RPA's estão muito tempo abaixo do objetivo e após ser enviado o email, é posto no	Could Have	Não implementado

		estado "parado".		
RF41	Sistema	É necessário ter um Sistema que após o e-mail ser enviado ao humano, quando ocorre um erro fatal, coloca o processo em estado de abortado.	Should Have	Modificado e implementado totalmente
RF42	Sistema	É necessário ter um Sistema que envie a cópia dos documentos de um determinado processo quando esse obtém um erro.	Must Have	Implementado totalmente
RF43	Sistema	Quando enviado um email ao humano responsável, por causa de um RPA, esse email é enviado com possíveis lugares de erro.	Should Have	Implementado totalmente
RF44	Sistema	É necessário ter um Sistema que organize as tarefas a serem realizadas consoante o nível de prioridade.	Should Have	Implementado totalmente
RF45	Sistema	É necessário ter um Sistema que capte as informações fornecidas pelos humanos.	Must Have	Implementado totalmente
RF46	Sistema	É necessário ter um Sistema que valide e filtre os logs fornecidos pelos humanos.	Must Have	Implementado totalmente
RF47	Sistema	É necessário ter um Sistema que analise os logs fornecidos pelos humanos.	Must Have	Não implementado
RF48	Sistema	É necessário ter um Sistema que calcule o número de tarefas que cada equipa de operacionais tem pendentes.	Must Have	Implementado totalmente
RF49	Sistema	É necessário ter um Sistema que gera uma lista de tarefas que cada equipa de operacionais tem pendentes.	Must Have	Implementado totalmente
RF50	Sistema	É necessário ter um Sistema que gera uma lista de tarefas que cada operacional tem pendentes.	Must Have	Implementado totalmente
RF51	Sistema	É necessário ter um Sistema que faça a conexão com o UiPath Orchestrator	Must Have	Implementado totalmente
RF52	Sistema	É necessário ter um Sistema que extrai dados dos RPA's do UiPath Orchestrator	Must Have	Implementado totalmente
RF53	Sistema	É necessário ter um Sistema que transforme a informação dos RPA's do UiPath Orchestrator	Must Have	Implementado totalmente
RF54	Sistema	É necessário ter um Sistema que transforme a informação dos processos do UiPath Orchestrator	Must Have	Implementado totalmente
RF55	Sistema	É necessário ter um Sistema que transforme a informação das tarefas de um processo do UiPath	Must Have	Implementado totalmente

		Orchestrator		
RF56	Sistema	É necessário ter um Sistema que transforme a informação dos logs de um RPA do UiPath Orchestrator	Must Have	Implementado totalmente
RF57	Sistema	É necessário ter um Sistema que extrai dados dos processos do UiPath Orchestrator	Must Have	Implementado totalmente
RF58	Sistema	É necessário ter um Sistema que extrai dados das tarefas de um processo do UiPath Orchestrator	Must Have	Implementado totalmente
RF59	Sistema	É necessário ter um Sistema que permita ao operacional reservar uma tarefa que se encontrava na lista de tarefas a realizar pela sua equipa	Must Have	Implementado totalmente
RF60	RPA's	É necessário que o RPA gere logs informativos do que está a realizar.	Could Have	Implementado totalmente

Tabela 1 - Requisitos funcionais

4.1.2 Requisitos não funcionais

ID	Classificação	Requisito	Capacidades	Cumprimento
RNF-1	Confiabilidade e	Os RPA's têm de ter um objetivo de x%.	Para conseguir manter uma boa produtividade é bom manter uma certa assertividade, se baixar colocar o robô em revisão.	Não implementado
RNF-2	Ético	Ninguém consegue ver dados de outros utilizadores.	De uma forma ética só o utilizador sabe as suas credenciais onde não estão visíveis a ninguém.	Implementado totalmente
RNF-3	Segurança	É necessário ter um Sistema de encriptação de ficheiros quando é necessário enviá-los para o front-end.	Para efetuar a encriptação iremos utilizar a função do MD5, fornecendo assim uma maior segurança no envio de ficheiros confidenciais entre o back-end e o front-end.	Não implementado

RNF-4	Sistema	A tabela das estatísticas da base de dados deve ser atualizada de x em x horas.	Para efetuar a atualização das estatísticas, iremos efetuar um loop no código que quando muda a hora este irá efetuar as atualizações necessárias.	Retirado
RNF-5	Sistema	Os códigos enviados por e-mail apenas devem ser válidos durante x mins.	Para efetuar uma maior segurança, o código enviado ao utilizador só terá 15 minutos para utilizar esse Código.	Implementado totalmente
RNF-6	Sistema	Após ser enviado o email ao responsável do processo em que o RPA está envolvido a alertar da assertividade , têm x horas para tentar resolver o problema senão o RPA é colocado no estado "parado".	Para que o RPA não continue com uma assertividade baixa para a automação do processo para não haver mais enganos e possíveis prejuízos.	Não implementado
RNF-7	Sistema	UiPath Orchestrator API	A API Orchestrator é uma API REST compatível com o padrão OData versão 4.0 que expõe entidades e seus relacionamentos usando regras de acesso, navegação e modificação bem	Implementado totalmente

			definidas.	
RNF-8	Tecnologia	SQLite	SQLite é uma biblioteca em linguagem C que implementa uma base de dados SQL embutida. Programas que usem a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo SGBD separado.	Implementado totalmente
RNF-9	Tecnologia	UiPath	UiPath é uma empresa global de software que fabrica software de automação de processos robóticos.	Implementado totalmente
RNF-10	Velocidade	Os e-mails enviados quando o processo não é executado com sucesso não podem exceder x mins após a ocorrência do erro.	Para efetuar o envio do e-mail, após ter sido lançado o alerta de que o processo obteve um erro é iniciado um contador que quando mais próximo no tempo limite para o envio do e-mail vai aumentando a prioridade com que esta tarefa tem de ser executada.	Implementado totalmente

RNF-11	Velocidade	Os e-mails enviados aos utilizadores quando estes pretendem fazer a recuperação das credenciais não podem exceder os x mins.	Após ter sido enviado o email o utilizador tem apenas 15 minutos para utilizar o código único e de única utilização para recuperar as credenciais.	Implementado totalmente
RNF-12	Velocidade	Após o RPA estar abaixo da assertividade, o envio de um email ao responsável do referente RPA, deve demorar no máximo x minutos.	Para efetuar o mínimo de erros na sua automação é enviado um email a alertar do erro e pedido de resolução.	Não implementado
RNF-13	RPA's	É necessário que o RPA seja criado em UiPath Studio	Consoante o cliente o RPA tem de ser desenvolvido em UiPath Studio.	Adicionado e implementado totalmente
RNF-14	RPA's	É necessário que um RPA seja publicado no UiPath Orchestrator	Consoante o cliente tem de ser utilizado o UiPath Orchestrator como plataforma gestora de RPA's.	Adicionado e implementado totalmente
RNF-15	RPA's	É necessário que o processo que o RPA executa seja atribuído ao Folder no UiPath	Para o bom funcionamento e organização do UiPath Orchestrator é necessário que os RPA's sejam todos executados no mesmo folder.	Adicionado e implementado totalmente

		Orchestrator		
--	--	--------------	--	--

Tabela 2 - Requisitos não funcionais

Justificação dos novos requisitos, de modificação de requisitos antigos e de exclusão de requisitos:

RF4 -> Não implementado, devido ao facto de não ter sido consumada a parte dos objetivos dos humanos e, pelo facto, dos RPA's não terem sido colocados na aplicação, estando apenas no UiPath Orchestrator.

RF25 ao RF30 -> Foram modificados, pois as estatísticas foram implementadas através do *kibana* e este necessita das informações que foram especificadas, de forma a conseguir implementar todos os gráficos e estatísticas apresentadas no Front-End.

RF32 -> O requisito foi retirado, pois os processos deixaram de ter uma prioridade.

RF36, RF39, RFN1, RNF6 e RNF12 -> Estes requisitos não foram implementados, devido ao facto dos RPA's não estarem presentes na nossa solução, estando apenas no UiPath Orchestrator.

RF37 -> Requisito não implementado, devido ao facto de não ter sido implementado a parte dos objetivos dos humanos (operacionais).

RF40 -> O requisito foi alterado, pois o estado do processo deixou de Waiting para Aborted, fazendo assim a distinção entre os processos. Waiting, significa que se encontram à espera que o humano vá executar as suas tarefas e, o estado Aborted, que significa que o processo obteve um erro fatal.

RNF3 -> O requisito não foi implementado, pois não foi necessário o envio de nenhum ficheiro para o Front-End.

RNF4 -> O requisito foi retirado, pois deixou de existir a tabela de estatísticas na nossa base de dados porque estas foram calculadas através do *kibana*.

RNF13 -> Este requisito foi adicionado de forma a especificar qual a plataforma onde os RPA's têm de ser criados para poderem ser monitorizados pela solução desenvolvida.

RNF14 e RNF15 -> Estes requisitos foram adicionados de forma a especificar como os RPA's devem ser publicados e executados para a nossa solução conseguir extrair os dados produzidos por estes robôs.

4.2 Diagramas de caso de uso

Um caso de uso é um conjunto de ações efetuadas por um ator e as correspondentes interações com o sistema de forma a alcançar um objetivo. Os atores, por norma, são humanos, mas, no entanto, podem ser sistemas externos à aplicação.

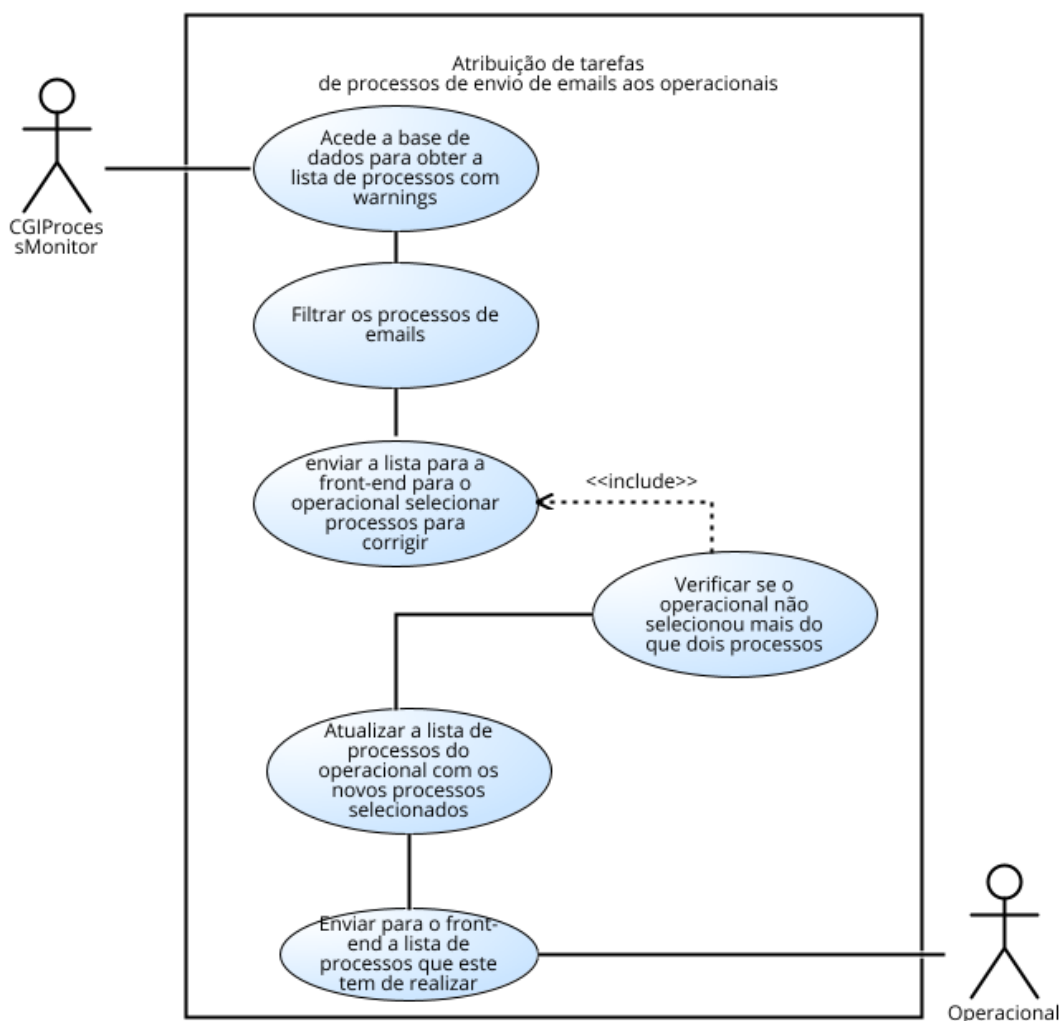


Figura 1 - Caso de Uso - Atribuição de tarefas de processos de envio de emails aos operacionais

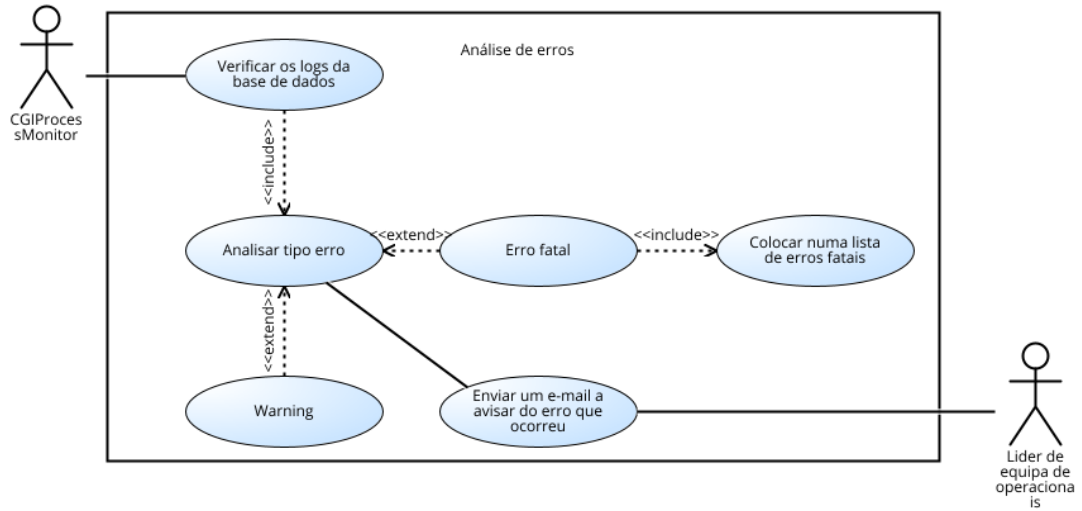


Figura 2 - Caso de Uso - Análise de erros

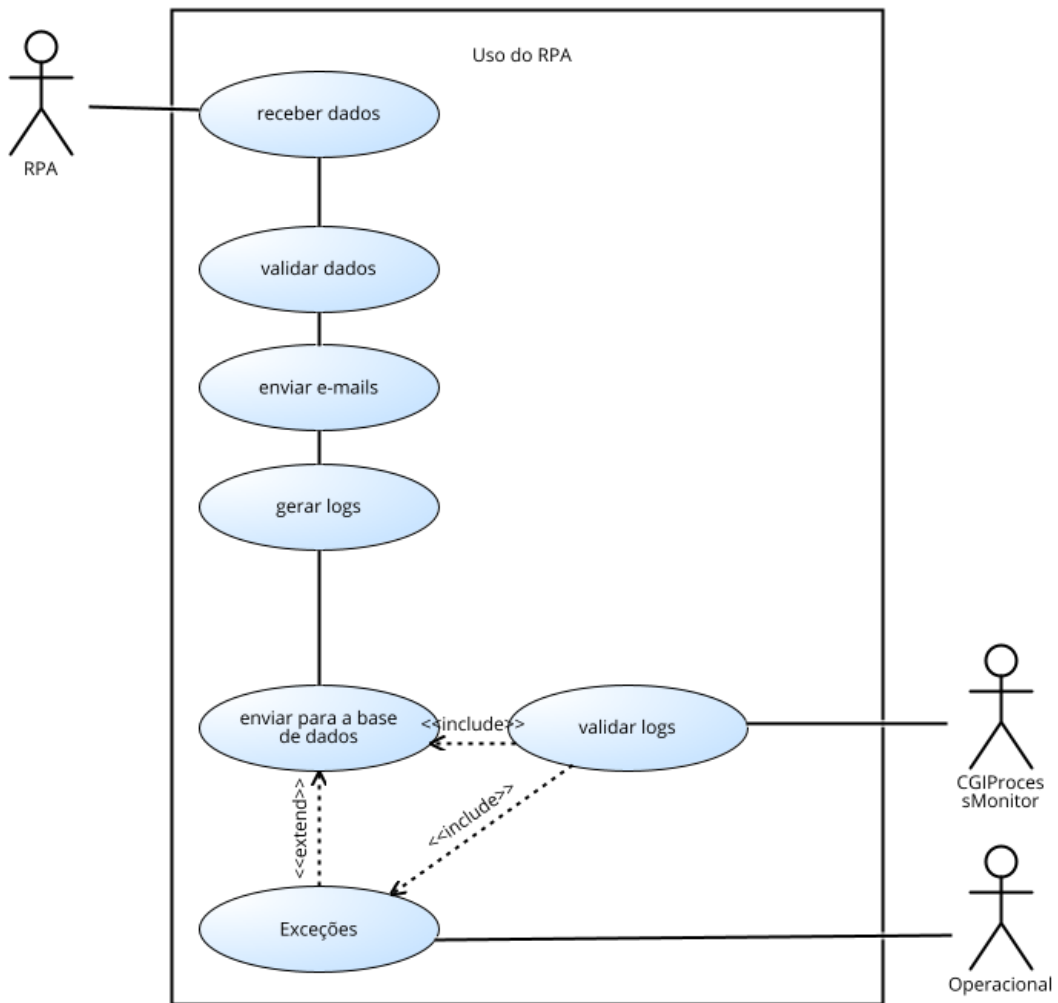


Figura 3 - Caso de Uso - Uso do RPA

4.3 Diagramas de atividade (BPMN)

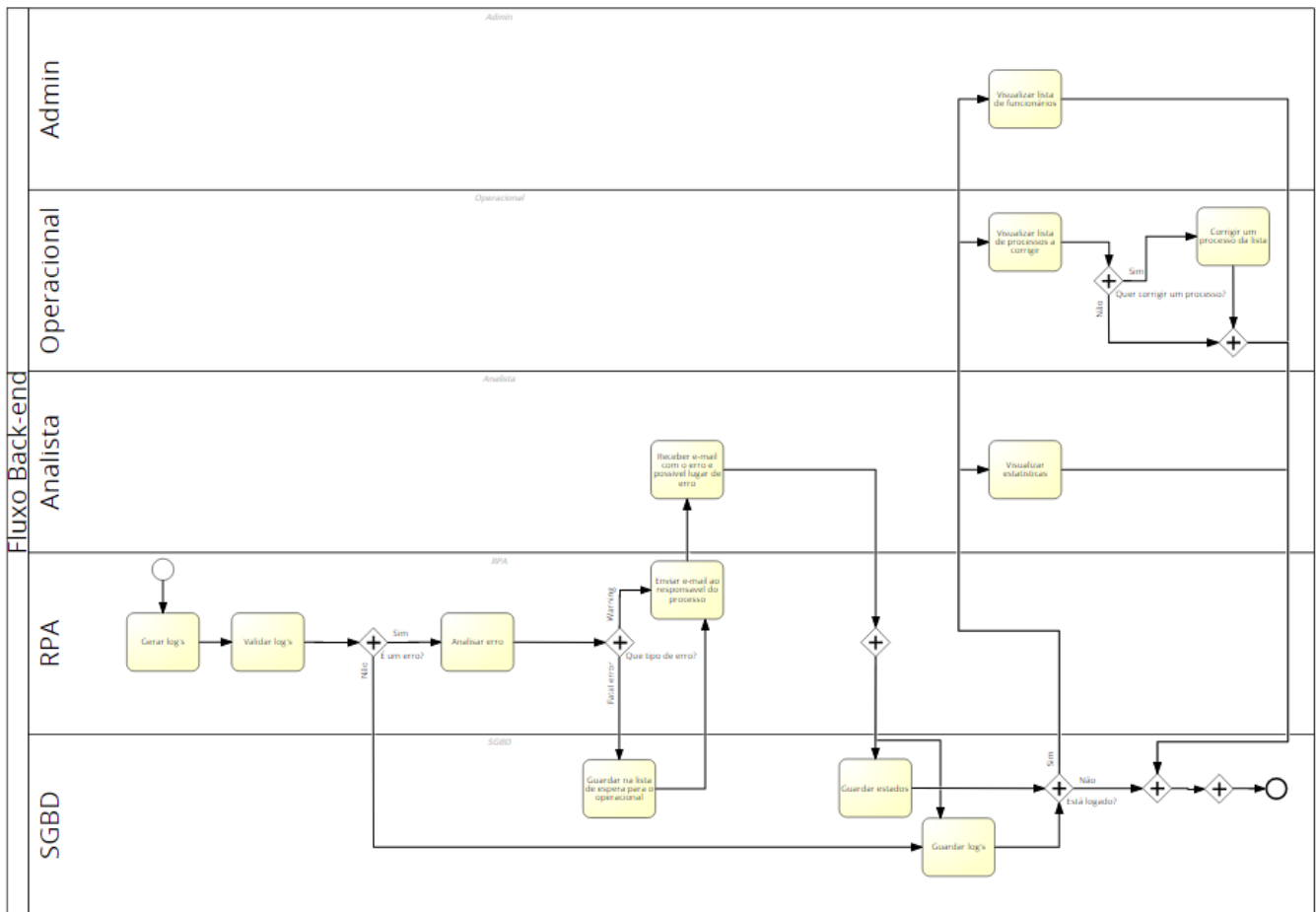


Figura 4 - Diagrama de atividade (BPMN)

4.4 Modelo de dados

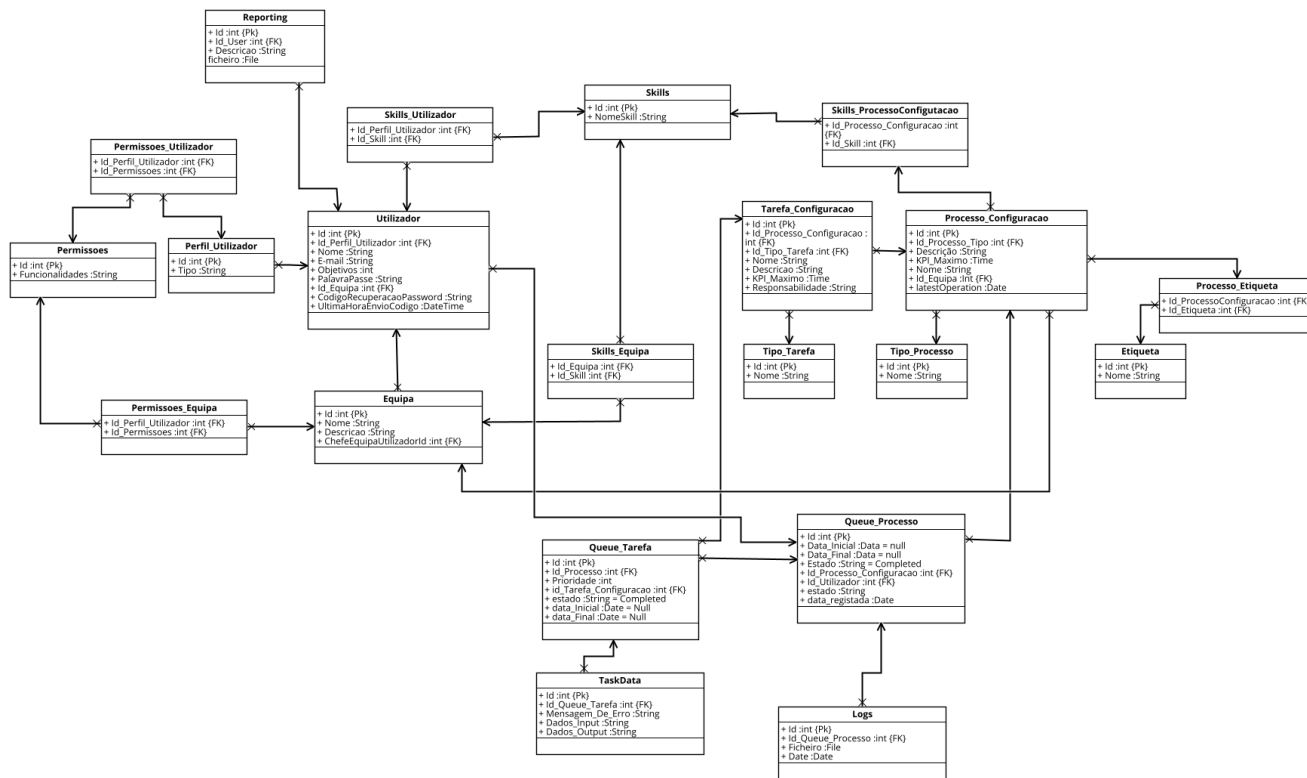


Figura 5 - Modelo de dados (UML)

A Figura 5 - Modelo de dados (UML) representa o modelo de dados desenvolvido pelo back-end que serve de suporte para a criação da base de dados da aplicação.

4.5 Estrutura

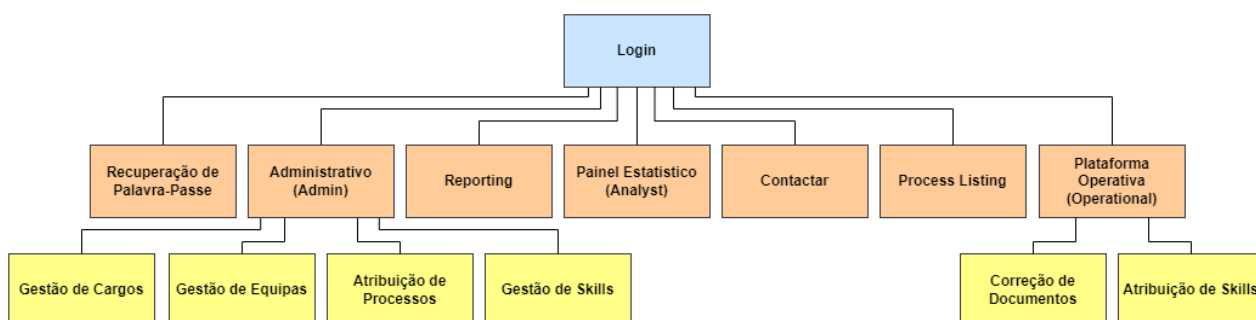


Figura 6 - Mapa Aplicacional

A estrutura aplicacional da Figura 6 - Mapa Aplicacional representa o mapa realizado pela componente 2 (Front-end), à qual o back-end dá suporte. É possível dar suporte aos utilizadores como em casos de recuperação de palavras-passe, gestão de equipas e atribuições de processos por parte do administrador. É possível, também, dar suporte ao painel de estatísticas, fornecendo logs dos rps e dos operacionais; auxiliar o process listing, enviando os processos de negócio anteriormente criados pelos logs; e, ainda, é possível facilitar o suporte à plataforma operativa, dando, por um lado, uma lista de processos para que os operacionais da equipa (caso não sejam chefes de equipa), possam escolher ou corrigir tarefas, por outro, uma lista das suas próprias tarefas pendentes. Caso sejam chefes de equipa conseguem ver a lista pendente da equipa e alterar skills dos membros.

4.6 User Stories

User stories são descrições simples do que o sistema deve fazer, descrevendo assim as suas funcionalidades. Estes seguem o esquema de Como (ator) ...devo (ação)...para (finalidade)...

- Como dono do processo, preciso de uma base de dados que guarde todos os logs feitos pelo robô para ter informação do mesmo.
- Como dono do processo, preciso de uma base de dados que guarde todos os logs do operacional para ter informações do mesmo.
- Como utilizador, preciso de uma base de dados para guardar as credenciais para conseguir fazer login.
- Como dono do processo, preciso de uma base de dados que guarde o ciclo de vida do processo de negócio para ficar com o seu registo e as suas informações.
- Como utilizador, preciso de uma base de dados para guardar os reports de erro que faço para que alguém consiga corrigi-lo.
- Como operacional, preciso de uma base de dados para guardar as tarefas a realizar da minha equipa para poder escolher e depois corrigir.
- Como operacional, preciso de uma base de dados para guardar as minhas tarefas a realizar para poder corrigir.
- Como RPA, preciso de uma base de dados para guardar os logs feitos por mim para terem mais informações nas tarefas.
- Como operacional, preciso que a aplicação leia os logs feitos por mim para ficar registado na base de dados o que realizei.
- Como RPA, preciso que a aplicação consiga extrair os dados do orchestrator para ter acesso aos logs.
- Como utilizador, preciso que a aplicação me envie um email de recuperação de palavra-passe para conseguir redefinir a minha palavra-passe, caso já não a saiba ou queira trocar.
- Como RPA, preciso de produzir logs segundo as regras estabelecidas para poder ser monitorizado nesta solução.
- Como RPA, preciso de registar, nos meus logs, tudo o que produzo para que esta informação seja extraída para esta solução.

5 - Solução Proposta

5.1 Introdução

Link Repositório GitHub: <https://github.com/josesobral22005813/CGI-Process-Monitoring-302-303.git>

Link Vídeo Youtube: https://youtu.be/bEmfPKPD_1A

O trabalho final de curso encontra-se na fase de desenvolvimento, onde já existe o estabelecimento da solução definida para a resolução do problema da monitorização e orquestração dos processos que são desempenhados pelos RPA's.

Primeiramente, foi construída uma base de dados na qual foi armazenada informações provenientes da api do orchestrator, da UiPath, tendo de ser feitas as devidas verificações e validações das mesmas e dados para o bom funcionamento da plataforma como equipas, utilizadores, skills, permissões, etc.

Considerando que este projeto está dividido em 2 grupos, a parte de backend realizado pelos elementos deste relatório Robot Process Automation - Componente 1, enquanto a parte de frontend realizada pelo grupo de Robot Process Automation - Componente 2. Estes encontram-se a produzir a camada de apresentação que inclui dashboard, gerir os utilizadores, mostrar as estatísticas, etc...

Posteriormente, a solução encontrada foi a construção de uma plataforma que tem como função monitorizar RPA's, que acedendo à base de dados, tem acesso aos registos dos mesmos (logs), de forma a conseguir obter uma melhor performance e mais segura destes robôs. Esta plataforma, depois de obter os dados, faz uma análise, a fim de conseguir perceber se ocorreu algum problema nos diversos RPA's que estão a ser monitorizados. Caso exista um warning, a plataforma coloca o processo numa lista de espera para que um operacional (humano) o consiga solucionar. Por outro lado, caso exista um erro fatal, é colocado numa lista de processos com o estado de "aborted" e é enviado um email ao responsável do processo de forma a alertá-lo com o possível ponto de erro ou local onde o mesmo está a decorrer - usamos este tipo de alarmística como exemplo, pois o que deveria ser implementado é a utilização de um software externo.

Vão ser referidos nos seguintes capítulos, tópicos como a arquitetura do projeto que mostra as camadas e que ferramenta foi utilizada em cada uma; as tecnologias e ferramentas

utilizadas; de que forma foi feita a implementação da plataforma; e por último a abrangência do projeto.

5.2 Apresentação da arquitetura

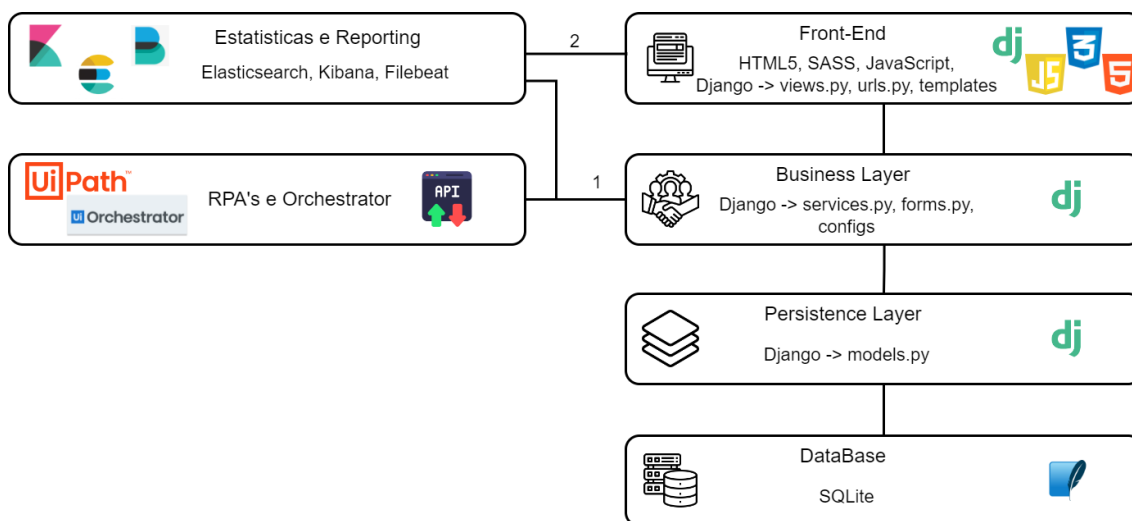


Figura 7 - Desenho da Arquitetura a implementar

A ligação 1 da arquitetura compreende as API Calls feitas pela Business Layer aos componentes de orquestração de processos, com os objetivos de extrair logs, utilizar a aplicação, as estatísticas e o report. A ligação 2 compreende a deteção de logs pelo Filebeat para que os mesmos sejam inseridos no elasticsearch e kibana.

- Presentation layer - É o front-end.
- Security layer - onde existe proteção contra SQL injection.
- RPA's e Orchestrator – onde se recolhem e gerem os logs dos rpa's.
- Business layer - onde está presente o código onde se faz as estatísticas, alarmística, análise e validação de dados, chamadas da API Orchestrator e fazer as configurações.
- Persistence layer - onde se criam os objetos e tabelas.
- DataBase layer - onde está a base de dados que é criada com SQLite.

5.3 Tecnologias e Ferramentas Utilizadas

As tecnologias e as ferramentas utilizadas foram o *Django* (framework full-stack Web), *UiPath Studio*, *UiPath Orchestrator*, *IDE PyCharm*, *Signavio*, *Trello* e o *SQLite*.

Muito sucintamente, no início usou-se o *UiPath Studio* para a criação de um protótipo de RPA. Depois foi utilizado o *Signavio* na ajuda para a documentação (diagramas), o *TeamGantt* para realizar o cronograma, o *Trello* para gerir o projeto e o *SQLite* para a criação da base de dados.

De seguida, usou-se o *PyCharm* para desenvolver o código em Python e, para a extração de dados do *UiPath Orchestrator* que guardava automaticamente na base de dados os mesmos.

Por último, o *Django* foi utilizado para processamento, modelação e organização da base de dados.

5.4 Implementação

A aplicação Web do sistema foi implementada usando o *Django*, framework utilizado para um desenvolvimento rápido para web, escrito em Python, que foi utilizado, no Back-End, para criar o sistema de monitorização e processos, pois foi utilizado na UC de Programação Web.

O *UiPath Studio* foi utilizado para o desenvolvimento de RPA's de teste, ou seja, os robôs que irão validar documentos e, posteriormente, os que irão ser monitorizados. Por outro lado, o *UiPath Orchestrator* foi utilizado para monitorizar e gerir os RPA's, anteriormente criados no *UiPath Studio*. O *UiPath* foi somente utilizado para a criação de RPA's a pedido do cliente. E para esta tarefa foi requerido um curso que a *UiPath* disponibiliza de 38 horas.

Foi usado o *Pycharm* como IDE (ambiente de desenvolvimento integrado), usado para programação em Python, para fazer as estatísticas, alarmística, análise e validação dos dados provenientes do orchestrator e todo o suporte para Front-End. Mais concretamente, este software, fornece análise de código, um depurador gráfico, que testa a unidade integrada que ajuda nos sistemas de controlo de versão e suporta desenvolvimento web com *Django* (software também utilizado no decorrer do projeto).

Signavio, o *Trello* e o *TeamGantt* são aplicações Web que serviram para o desenvolvimento dos diagramas, organização das tarefas e atribuição das mesmas e desenvolvimento do cronograma para a documentação, ou seja, o relatório.

Por último, foi, também, utilizado *SQLite*, uma biblioteca em linguagem C que

implementa uma base de dados SQL embutida. Programas que usem a biblioteca *SQLite* podem ter acesso a banco de dados SQL sem executar um processo SGBD separado.

Está, em [Anexo 1 – RPA email](#) e em [Anexo 2 - RPA formulário](#), protótipos de RPA's que foram usados para testar a aplicação desenvolvida.

Algumas componentes que requerem mais esforço são:

- A criação da base de dados, pois nunca tinha sido criada uma base de dados de raiz pela equipa do projeto, e, por esse mesmo motivo, existia a ausência de pensamento crítico.

- O orquestrador do UiPath foi complicado, pois nunca tinham sido feitas conexões em python, o que revelou algumas dificuldades em usar/organizar o mesmo.

5.5 Abrangência

Perante esta proposta de solução ao problema do trabalho final de curso, foram lecionadas diversas cadeiras, ao longo dos 3 anos do curso de engenharia informática, nas quais foram adquiridos os conhecimentos necessários para a realização desta solução, sendo elas a cadeira de Base de Dados, Engenharia de Software, Inteligência Artificial, Linguagens de Programação I, Linguagens de Programação II, Data Science e Sistemas de Suporte à Decisão.

A cadeira de *Base de dados* forneceu o conhecimento necessário para a manipulação de maneira direta e/ou indireta dos dados que são produzidos pelos diversos RPA's e que têm de ser armazenados, aos quais efetuamos o tratamento e as devidas validações e verificações.

A unidade curricular de *Engenharia de Software* foi bastante útil no que diz respeito à organização da resolução do problema, fornecendo diversos mecanismos de pensamento e formas de administrar um projeto.

A disciplina de *Inteligência Artificial* fornecerá competências na utilização de diversas tecnologias, modelos e técnicas existentes, o que é bastante útil para a criação de RPA's com uma melhor capacidade de tomada de decisão, retirando, desta forma, parcial ou totalmente, a intervenção humana em determinadas tarefas.

As cadeiras de *Linguagens de Programação I* e de *Linguagens de Programação II* foram bastante importantes, pois forneceram os fundamentos que são necessários para uma boa programação de RPA's.

A unidade curricular de *Data Science* forneceu uma visão de como fazer o tratamento e análise de dados, o que permitiu formular hipóteses que justificam os mesmos, pois é algo bastante útil na formulação das razões para as falhas dos RPA's.

A disciplina de *Sistemas de Suporte à Decisão* fez com que houvesse um primeiro contacto com ferramentas de reporting e de construção de dashboards de reporting com a finalidade de, futuramente, poder-se utilizar os conhecimentos adquiridos para a área de deteção e comunicação dos problemas dos RPA's.

A disciplina de *Programação Web* fez com que conhecêssemos o padrão de arquitetura MVC e que nos deu conhecimentos sobre Django, o que nos deu algumas bases para a realização deste projeto.

6 - Método de planeamento

No decorrer do projeto final de curso, é crucial ter um planeamento detalhado das etapas a serem realizadas e dos prazos correspondentes, para garantir o cumprimento dos objetivos e a entrega dentro do prazo estabelecido. Nesse sentido, apresenta-se no [Anexo 3 - Progresso de trabalho](#), todos os calendários utilizados ao longo da execução do trabalho, elaborado com a utilização da aplicação Team Gantt e a metodologia Gantt. Estes cronogramas abrangem as etapas das entregas. Com uma divisão clara das entregas e das tarefas, o cronograma permite visualizar o tempo necessário para a realização de cada etapa e a conclusão bem-sucedida das mesmas. Foi utilizado, também, para organização das tarefas o Trello a plataforma web em que foi gerido pelas duas componentes do projeto para destacar as tarefas por realizar, a realizar, em teste, realizadas e rejeitadas. O grupo teve dificuldades em realizar no tempo previsto para esta entrega por causa de outras UCs, mas foi possível concluir tudo o que tinha sido agendado para a entrega.

A primeira entrega foi feita dentro do prazo estipulado, sendo que as tecnologias foram pouco aprofundadas e foi recomendado a criação de pequenos protótipos, por isso, na segunda entrega, feita dentro do prazo, foram corrigidos os mesmos aspetos. No entanto, esta tinha pouco desenvolvimento no que diz respeito à inovação e pertinência do projeto. Na terceira entrega do relatório, já mais próximo do que era pretendido, foram corrigidos todos os erros dados anteriormente, sendo que houve uma melhoria no resumo e user stories.

7 - Resultados

O plano de testes e validação é uma fase crucial do processo de desenvolvimento de projetos. Este plano permite demonstrar que a solução construída é aplicável, pertinente e relevante, e que contribui efetivamente para a solução do problema em questão. Com testes abrangentes, é possível validar a qualidade da solução, o seu funcionamento e operação em contexto produtivo, garantindo que esta cumpra os requisitos definidos.

Na tabela seguinte, apresentamos como realizamos os nossos testes e validações, tendo como finalidade demonstrar a funcionalidade, requisitos, descrição, dados de input e os resultados esperados comparados com os resultados obtidos.

ID	Título	Funcionalidade	Descrição	Identificação de Requisitos	Resultado Esperado	Resultado Obtido
1	Email devido a erro fatal	Alarmística	Aceder ao UiPath Assistant e correr um RPA que gera um erro fatal.	RF1, RF43.	O chefe de equipa deve receber o e-mail com a informação sobre o processo que obteve o erro e a informação presente nos logs correspondente ao erro.	OK
2	Email devido a um warning	Alarmística	Aceder ao UiPath Assistant e correr um RPA que gera um warning.	RF2.	O chefe de equipa deve receber o e-mail com a informação sobre o processo que obteve o warning.	OK
3	Email com o número de tarefas que uma equipa tem para realizar	Alarmística	Correr a aplicação e aceder ao e-mail de um chefe de equipa.	RF3.	O chefe de equipa deve receber um e-mail com a informação sobre o número de tarefas que a sua equipa tem para realizar no presente dia e quantas tarefas realizou no dia anterior.	OK

4	Verificar se a base de dados guardou o processo de negócio	Base de dados	Correr a aplicação e aceder ao http://127.0.0.1:8000/admin/	RF5, RF10, RF39, RF51, RF52, RF53, RF54, RF57.	Ao aceder à página http://127.0.0.1:8000/admin/ deverá verificar se, após correr os RPA's, foram registados os respetivos processos de negócio.	OK
5	Verificar se a base de dados guardou os logs produzidos pelos RPA	Base de dados	Correr um RPA e verificar se, após a aplicação extrair os dados do orchestrator, os logs dos RPA's foram guardados na base de dados	RF8, RF21, RF22, RF55, RF56, RF58	Ao aceder à página http://127.0.0.1:8000/admin/ deverá verificar se foi criado um ficheiro “.log” no processQueue correspondente a instância de um RPA. Abrir o ficheiro indicado que deverá estar preenchido com as ações do RPA.	OK
6	Verificar se a base de dados guardou os logs produzidos pelos Humanos	Base de dados	Aceder à página de operacionais e realizar uma tarefa.	RF9, RF11, RF45.	Ao aceder a página http://127.0.0.1:8000/admin/ deverá verificar se foi criado um ficheiro “.log” no processQueue correspondente a instância de um RPA. Abrir o ficheiro indicado que deverá estar preenchido com as ações que o operacional realizou.	OK
7	Verificar os erros do sistema	Base de dados	Após um utilizador reportar um erro no sistema verificar se este foi guardado na base de dados.	RF12.	Após ser reportado um erro no sistema deveria aparecer junto ao sininho no canto superior direito uma notificação com as informações que foram preenchidas quando o erro foi reportado.	OK

8	Guardar a lista de tarefas de uma equipa	Base de dados	Após correr um RPA que gera trabalho para os operacionais (RPA que gera warnins) deverá aparecer mais um processo à lista da equipa que está atribuída a esse processo de negócio.	RF13.	Aceder à página de operacionais e verificar se, após ter sido corrido um RPA que gerou trabalho, aparece um novo elemento na lista de trabalho da equipa.	OK
9	Guardar a lista de tarefas de um operacional	Base de dados	Após correr um RPA que gera trabalho para os operacionais (RPA que gera warnins) deverá aceder à página de operacionais e escolher um processo para resolver.	RF14.	Aceder à página de operacionais e verificar se, após ter sido corrido um RPA que gerou trabalho e escolhido um processo para resolver, apareceu esse processo na sua lista de trabalho.	OK
10	Verificar se foram guardados os processos configuração	Base de dados	Após correr um RPA deverá aceder à página http://127.0.0.1:8000/admin/	RF19.	Ao aceder à página http://127.0.0.1:8000/admin/ deverá aparecer o processo de configuração do processo que foi executado pelo RPA.	OK
11	Verificar se foram guardadas as tarefas configuração	Base de dados	Após correr um RPA deverá aceder à página http://127.0.0.1:8000/admin/	RF20.	Ao aceder à página http://127.0.0.1:8000/admin/ deverá aparecer as tarefas de configuração do processo que foi executado pelo RPA.	OK

12	Verificação de logs	Estatísticas	Após ser executado um RPA e os dados por este produzidos serem registados na aplicação, deverá aceder ao ficheiro de logs criado para este processo e verificar se cada ação executada ao longo desse processo está registada nesse ficheiro tendo a data de execução da tarefa, o local onde foi realizada essa tarefa, o tipo de log, a mensagem do log, o id do processo e o nome do processo de configuração.	RF26, RF28, RF30, RF32.	RF27, RF29, RF31,	Ao aceder ao ficheiro de logs que foi gerado para o processo deverá ser visível os dados que o RPA gerou, mostrando a data de execução de cada tarefa, o local onde foi extraída a operação (UiPath Orchestrator), o tipo de log (warm, info ou error), a mensagem, o id do processo e o nome do processo.	OK
13	Envio de e-mail de recuperação de palavra-passe	Log In	Aceder ao ecrã de recuperar palavra-passe e introduzir um e-mail válido.	RF32.		Após proceder à recuperação da palavra-passe o utilizador deveria receber um e-mail com o código de recuperação de palavra-passe.	OK
14	Verificar se o processo ficou "aborted"	Sistema	Executar RPA com erro fatal; aceder ao ecrã 'Lista de processos' e ver o processo de negócio gerado ficou em "aborted".	RF41.		Após correr o RPA na lista de processos clique no processo de negócio gerado pelo RPA e verifique se o estado é "Aborted".	OK
15	Verificar a cópia de documentos	Sistema	Aceder à página de correção de tarefas, selecionar uma e clicar em corrigir.	RF42, RF59.		Após entrar no formulário de correção de erros é suposto aparecer um pdf.	OK

16	Verificar o número de tarefas que o utilizador tem pendentes	Sistema	Aceder à página de correção de tarefas.	RF48.	Verificar o número de tarefas por corrigir.	OK
17	Verificar a lista de tarefas que a equipa tem pendente	Sistema	Aceder à página de correção de tarefas.	RF49.	Verificar a lista que a equipa tem por corrigir.	OK
18	Verificar a lista de tarefas que o utilizador tem pendente	Sistema	Aceder à página de correção de tarefas.	RF50.	Verificar a lista que o utilizador tem por corrigir.	OK
19	Recuperação de palavra-passe	Sistema	Recuperar a palavra-passe e esperar mais que 15 minutos para ver se o código irá parar de funcionar.	RNF-5.	Quando colocado o código após o tempo (15 minutos) irá aparecer um erro.	OK
20	Colocar um RPA na plataforma	RPA's	Criar um RPA teste e colocá-lo no orchestrator adicionando na pasta 'CGIProcessMonitoring' um processo.	RNF-15.	Quando adicionado o processo no orchestrator é possível correr no UiPath Assitant.	OK

8 - Conclusão e trabalhos futuros

Depois de ser concluído o projeto, foi discutido em equipa que estudos futuros podem abranger a colocação de RPA's em VM's, utilizando serviços de nuvem, para que o projeto se torne mais semelhante a um projeto empresarial. Ou seja, hospedar os RPA's em ambientes virtuais, em vez de executá-los em máquinas físicas, o que, por sua vez, implica a aproveitação da tecnologia de virtualização para criar instâncias virtuais em que os RPA's são executados. Com isto, existem vários serviços em nuvem, como é o caso de Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, que podem ser utilizados dependendo das necessidades sugeridas pelo projeto.

Sugere-se a ampliação dos exemplos de RPA's além da leitura no Microsoft Excel, podendo-se explorar outras funcionalidades e integrações. Por exemplo, a criação de RPA's que leiam e validem pdf's/imagens, como por exemplo, balancetes, cartões de cidadão, declarações amigáveis, etc. Isto facilitaria a automatização de diversos processos que não estejam apenas restritos a uma leitura simples como é o caso da leitura utilizada em excel. Com isto, poderia haver uma maior abrangência de recolha de informação, mesmo que estas imagens estejam com pouca qualidade, pdf's desformatados, etc.

Para estudos futuros e, como não foi desenvolvida essa tarefa neste projeto, seria benéfico que utilizadores consigam corrigir e interagir com os reports que já estão a ser notificados. Pois, no projeto desenvolvido esta interação não existe e as notificações têm só carácter visual, o que não permite ao utilizador retificar. Poderiam, também, ser desenvolvidos auditores, ou seja, utilizadores que supervisionam as correções dos operacionais e tarefas aleatórias dos RPA's, para assim proporcionar a assertividade dos mesmos. Com isto, é possível notificar o chefe de equipa ou o responsável do processo do RPA quando é verificada a assertividade abaixo do threshold. Será benéfico, para a aplicação, a inserção dos RPA's na base de dados como utilizadores para usufruir da assertividade. Poderia ser também feito o desenho dos processos de negócio dentro da aplicação, numa das páginas do admin. Criar uma plataforma extra e enviar os ficheiros de correção por ele.

Para o bom funcionamento e motivação das equipas que utilizam a plataforma foi pensado que, ao longo da execução das tarefas, as mesmas poderiam ser anotadas e pontuadas. Sendo que, a pontuação final resultaria no reconhecimento do utilizador, podendo este usufruir de algum bónus.

O trabalho foi realizado e discutido em diversas reuniões com o cliente para que houvesse um resultado fidedigno e dentro dos objetivos e parâmetros estipulados. O cliente

declarou o bom alcance do “MVP”, sendo que foram adicionados extras pela equipa como é o caso do reporting, das skills e criação de equipas. A versão final oferece ao cliente uma plataforma que o mesmo pode usar para diminuir a frequência de erros que possam levar a danos à empresa. Sendo esta uma ferramenta eficaz, fácil e autoexplicativa que inclui todos os tipos de utilizadores. Os testes foram realizados pelo grupo do Robot-Process-monitor-Comp2 que é o FrontEnd e que não encontraram nenhum erro ao realizá-los.

Bibliografia

- (2) *RPA: Oportunidade e inovação. Saiba como adotar esta tecnologia sem investimento inicial* | LinkedIn. Obtido 24 de novembro de 2022, de https://www.linkedin.com/pulse/rpa-oportunidade-e-inova%C3%A7%C3%A3o-saiba-como-adotar-esta-tecnologia-gil/?trk=pulse-article_more-articles_related-content-card&originalSubdomain=pt
- Mitchell, C. (2022, agosto 4). *Gartner Magic Quadrant for Robotic Process Automation (RPA) 2022*. CX Today. <https://www.cxtoday.com/data-analytics/gartner-magic-quadrant-for-robotic-process-automation-rpa-2022/>
- Robotic Process Automation (RPA) Platform & Solutions* | Appia. Obtido 24 de novembro de 2022, de <https://appian.com/platform/complete-automation/robotic-process-automation-rpa.html>
- Robotic process automation (RPA) software* | Pega. Obtido 25 de novembro de 2022, de <https://www.pega.com/products/platform/robotic-process-automation>
- RPA Workspace | Automation Platform | Automation Anywhere*. Obtido 25 de novembro de 2022, de <https://www.automationanywhere.com/products/rpa-workspace>
- SQLite*. (2021, February 19). Wikipedia. <https://pt.wikipedia.org/wiki/SQLite>

Anexo 1 – RPA email

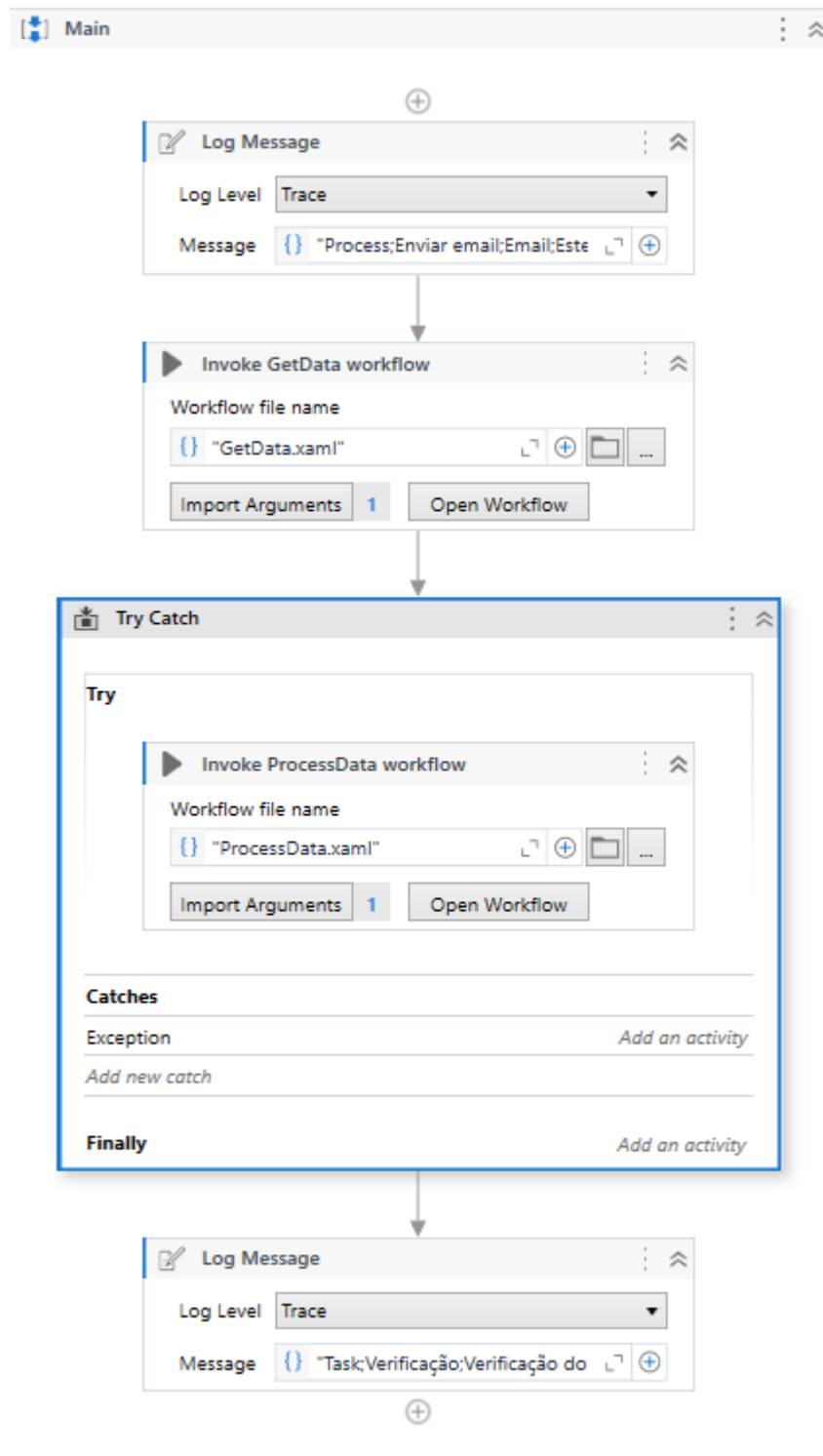


Figura 8 - RPA Emails - Main

Log Trace Inicial: "Process;Enviar email;Email;Este processo vai extrair dados de uma tabela de excel e utiliza-los para enviar emails;02:00:00;ler excel,enviar email"

Log Trace Final: "Task;Verificação;Verificação do preenchimento do envio de emails;HUMANO-Preencher o email com os dados do excel de forma correta;01:55:00;5"

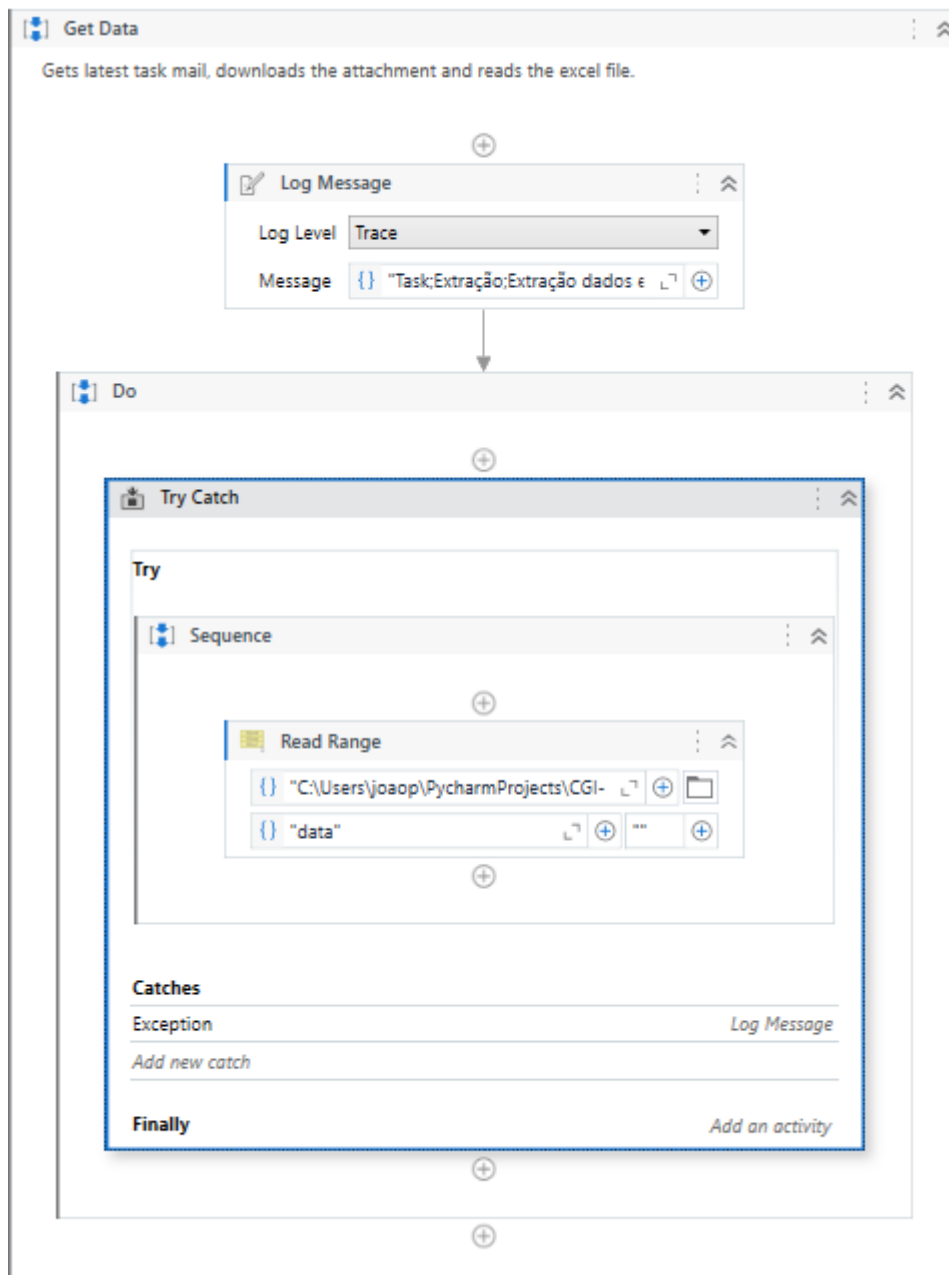


Figura 9 - RPA Emails - GetData

Log Trace: "Task;Extração;Extração dados excel;RPA-Esta tarefa acede a uma tabela excel e extrai os dados da mesma;00:04:00;3"

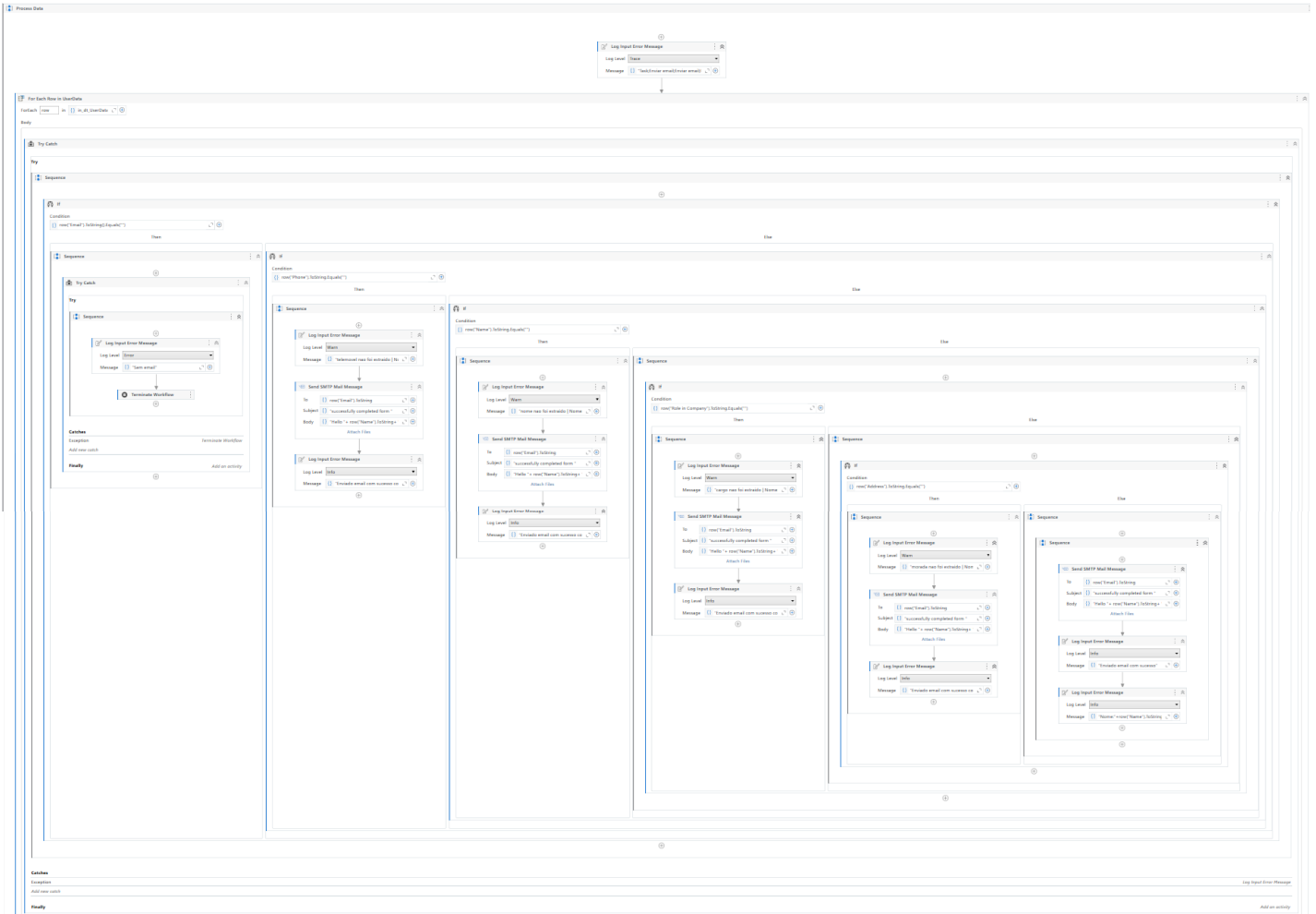


Figura 10 - RPA Emails - ProcessData

(Quando não tem email) Log Error: "Sem email"

(Quando não tem telemovel) Log Warn: "telemovel nao foi extraido | Nome:"+row("Name").ToString+";Funcao na empresa:"+row("Role in Company").ToString+";Morada:"+row("Address").ToString+";Email:"+row("Email").ToString+";Numero de telefone:"

(Quando não tem nome) Log Warn: "nome nao foi extraido | Nome:;Funcao na empresa:"+row("Role in Company").ToString+ ";Morada:"+row("Address").ToString+ ";Email:"+row("Email").ToString+";Numero de telefone:"+row("Phone").ToString

(Quando não tem cargo de funcionário) Log Warn: "cargo nao foi extraido |
Nome:"+row("Name").ToString+";Funcao na empresa:;Morada:"
+row("Address").ToString+";Email:"+row("Email").ToString+";Numero de telefone:"
+row("Phone").ToString

(Quando não tem morada) Log Warn: "morada nao foi extraido |
Nome:"+row("Name").ToString+";Funcao na empresa:"+row("Role in
Company").ToString +";Morada:;Email:"+row("Email").ToString+ ";Numero de
telefone:"+ row("Phone").ToString

(Quando está tudo bem) Log Info: "Nome:"+row("Name").ToString+";Funcao na
empresa:"+row("Role in Company").ToString+ ";Morada:"+row("Address").ToString+
";Email:"+row("Email").ToString+";Numero de telefone:"+row("Phone").ToString

Anexo 2 - RPA formulário

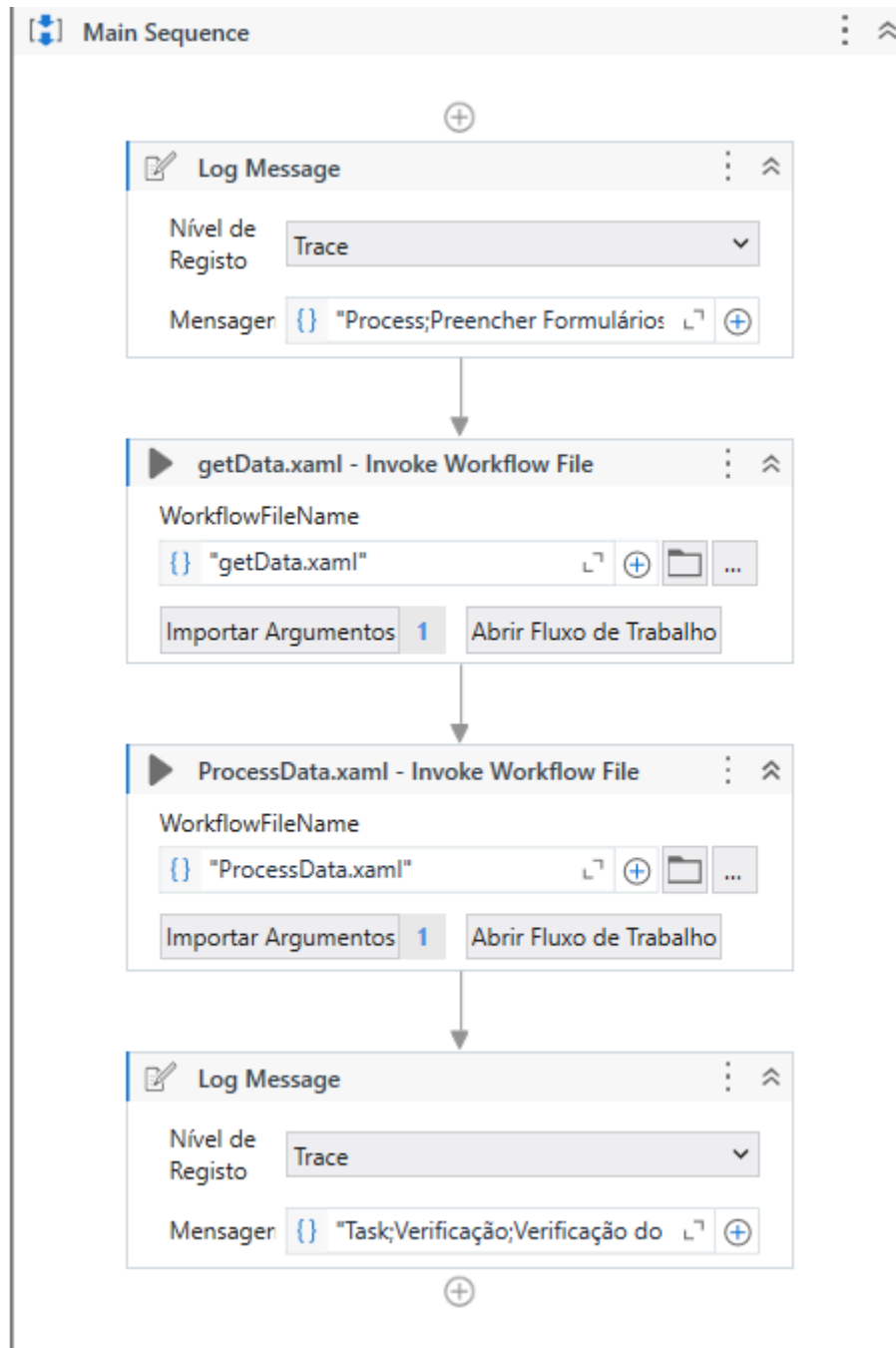


Figura 11 - RPA formulário - Main

Log Trace inicial:"Process;Preencher Formulários;Formulários;Este processo vai extrair dados de uma tabela de excel e utiliza-los para preencher um formulário;02:00:00;ler excel"

Log Trace Final:"Task;Verificação;Verificação do preenchimento de formulário;HUMANO-Preencher o formulário com os dados do excel de forma correta;01:55:00;5"

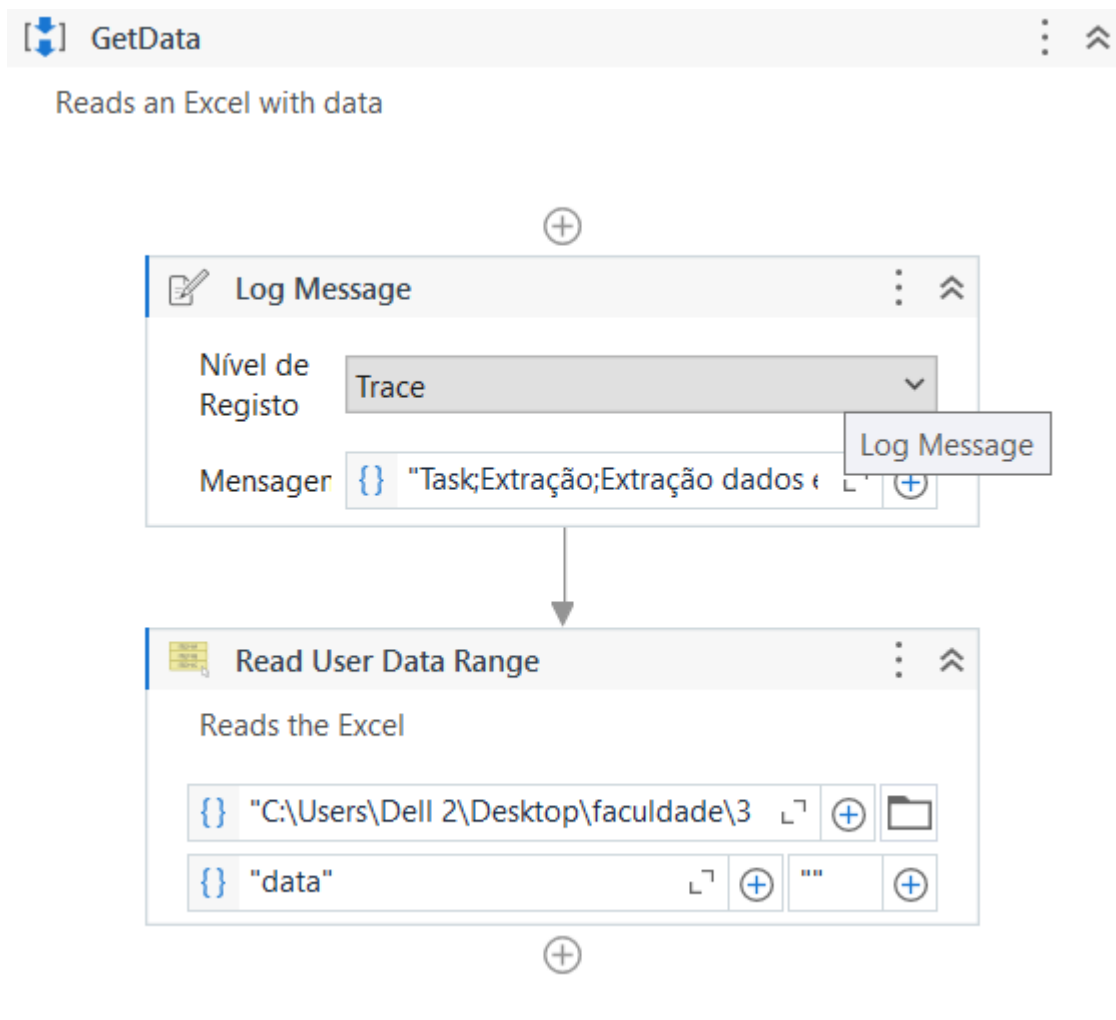


Figura 12 - RPA formulário - GetData

Log Trace: \"Task;Extração;Extração dados excel;RPA-Esta tarefa acede a uma tabela excel e extrai os dados da mesma;00:01:00;3\"

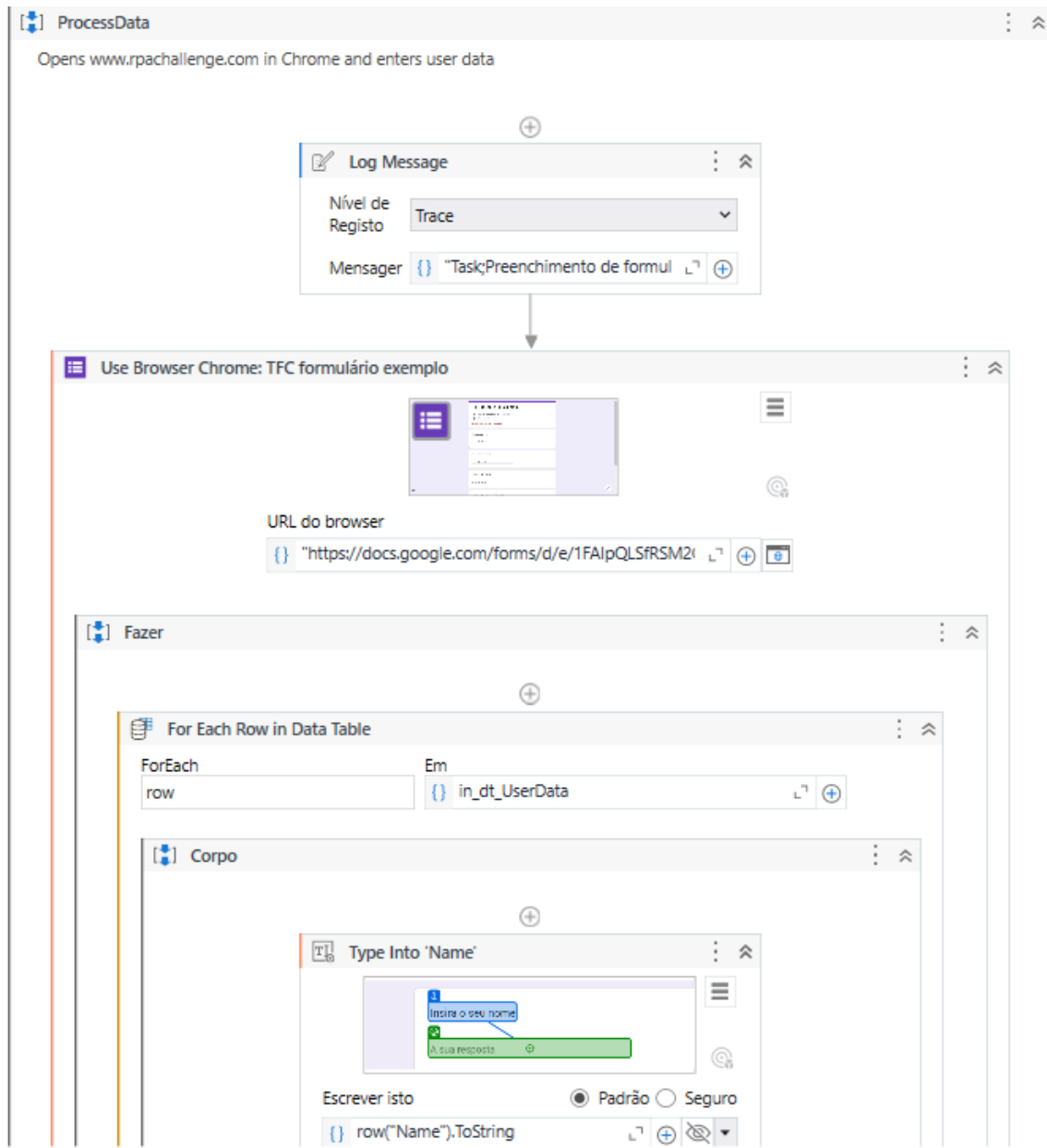


Figura 13 - RPA formulário - ProcessData(1)

Log Trace: "Task;Preenchimento de formulários;Preenchimento do formulário TFC formulário exemplo;RPA-Preenchimento do formulário com o primeiro e ultimo nome de uma pessoa, o seu email, a sua morada e o seu número de telemovel;00:04:00;4"

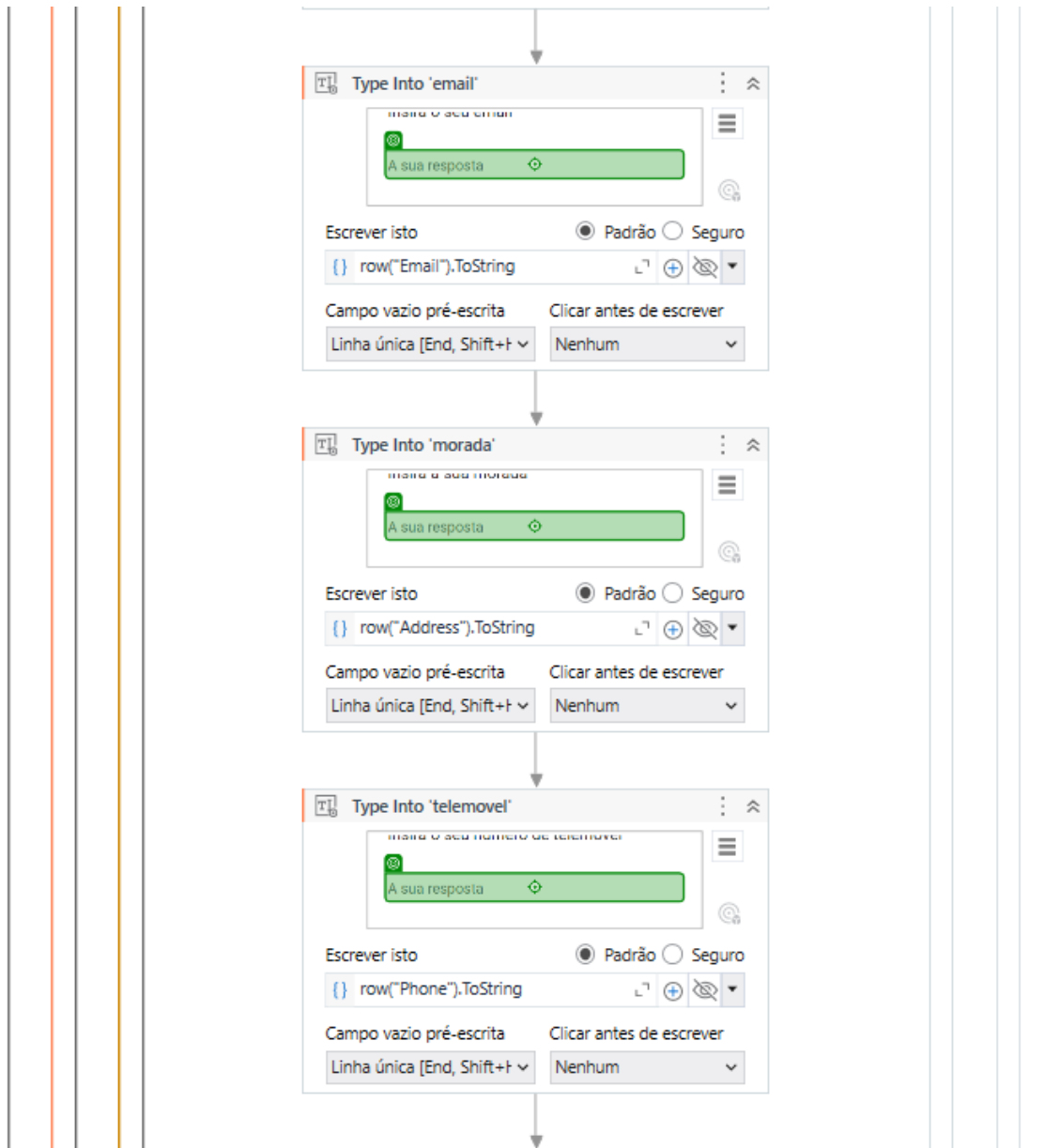


Figura 14 - RPA formulário - ProcessData(2)

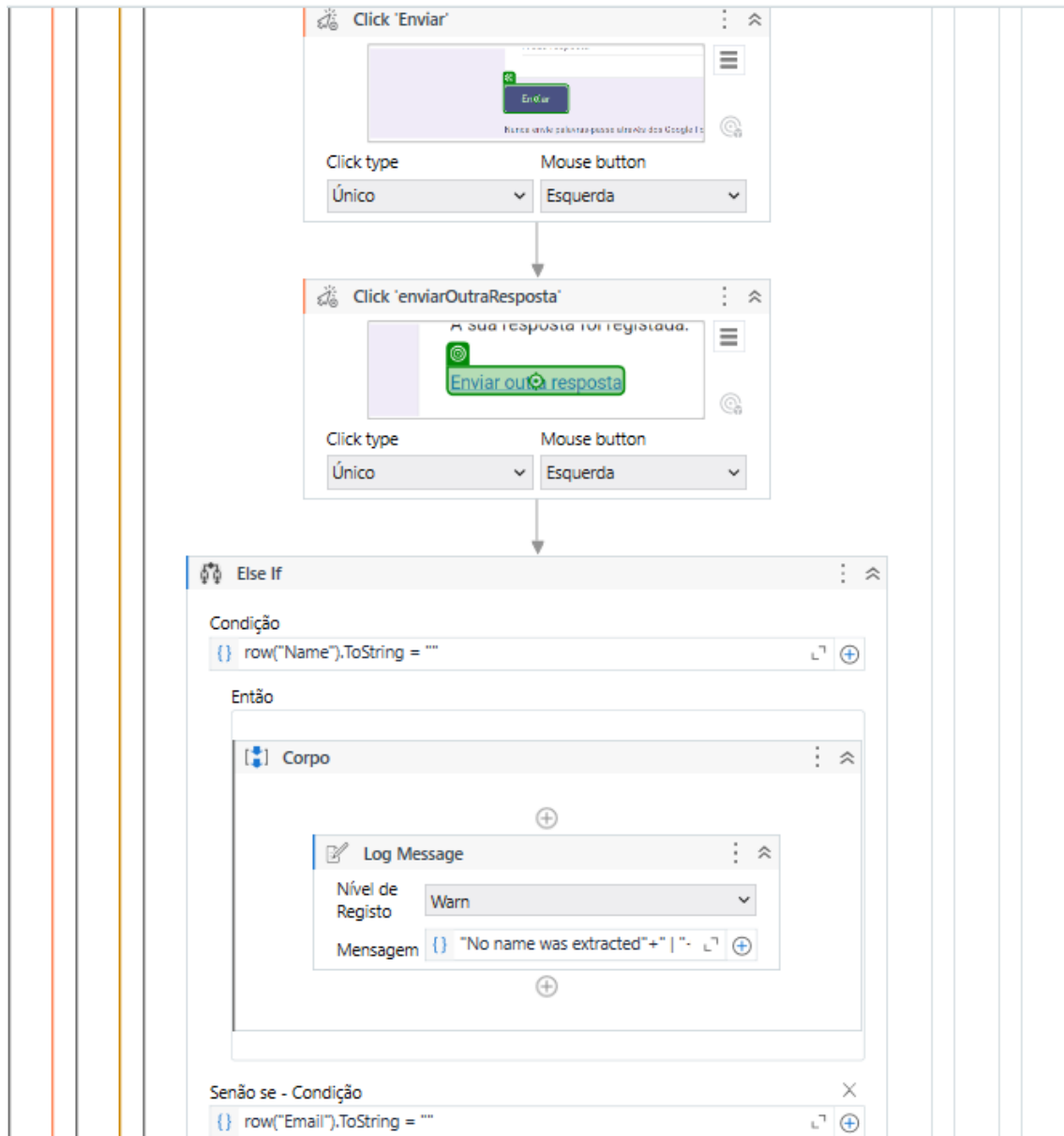


Figura 15 - RPA formulário - ProcessData(3)

Log Warn:"No name was extracted"+ " | " + "Nome:" + row("Name").ToString + ";Email:" + row("Email").ToString + ";Morada:" + row("Address").ToString + ";Número de telemovel:" + row("Phone").ToString

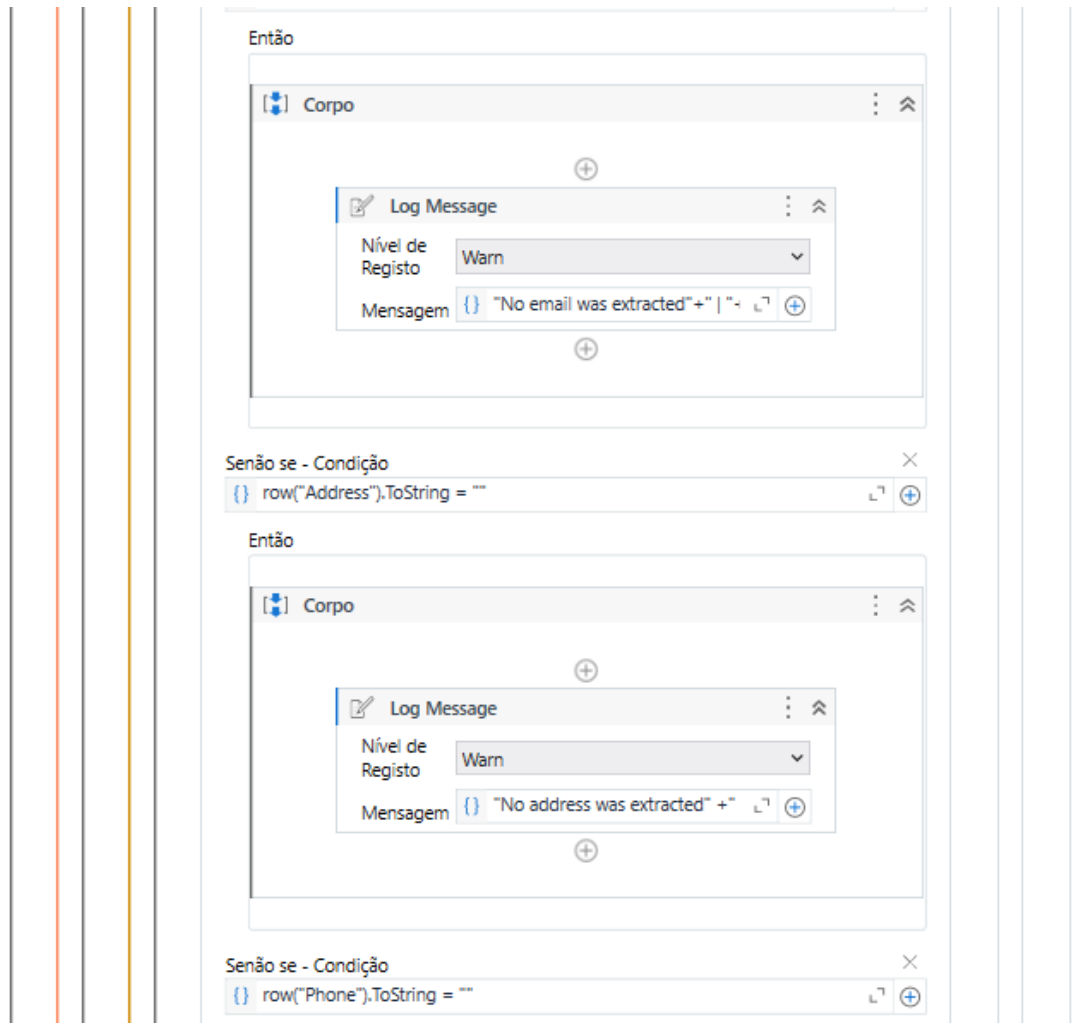


Figura 16 - RPA formulário - ProcessData(4)

Log Warn: "No email was extracted"+" | "+ "Nome:" + row("Name").ToString + ";Email:" + row("Email").ToString + ";Morada:" + row("Address").ToString + ";Número de telemovel:" + row("Phone").ToString

Log Warn:"No address was extracted" + " | " + "Nome:" + row("Name").ToString + ";Email:" + row("Email").ToString + ";Morada:" + row("Address").ToString + ";Número de telemovel:" + row("Phone").ToString

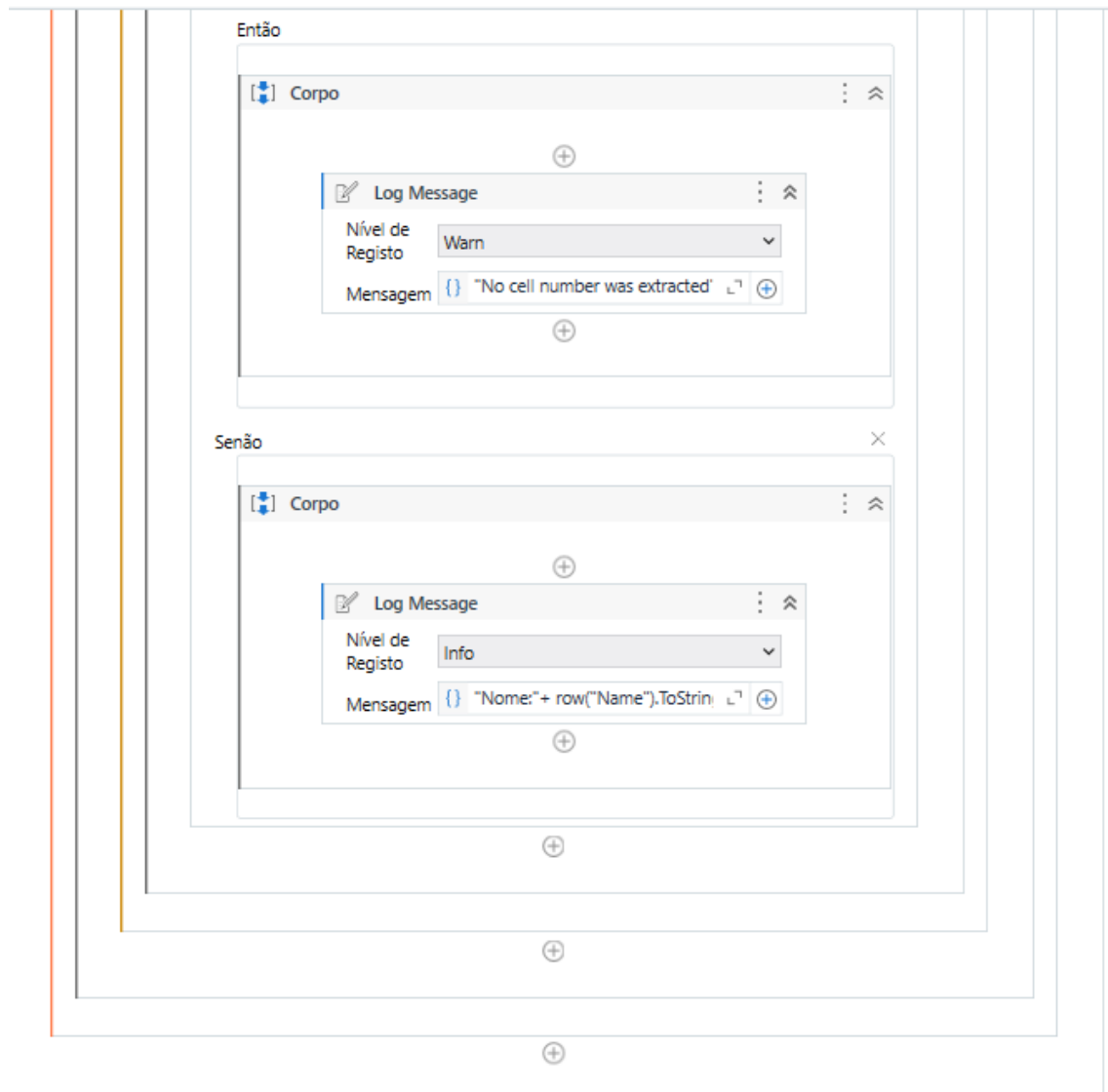


Figura 17 - RPA formulário - ProcessData(5)

Log Warn:"No cell number was extracted" + " | "+ "Nome:"+ row("Name").ToString +
";Email:" + row("Email").ToString + ";Morada:" + row("Address").ToString + ";Número de
telemovel:" + row("Phone").ToString

Log Info:"Nome:"+ row("Name").ToString + ";Email:" + row("Email").ToString +
";Morada:" + row("Address").ToString + ";Número de telemovel:" +
row("Phone").ToString

Anexo 3 - Progresso de trabalho

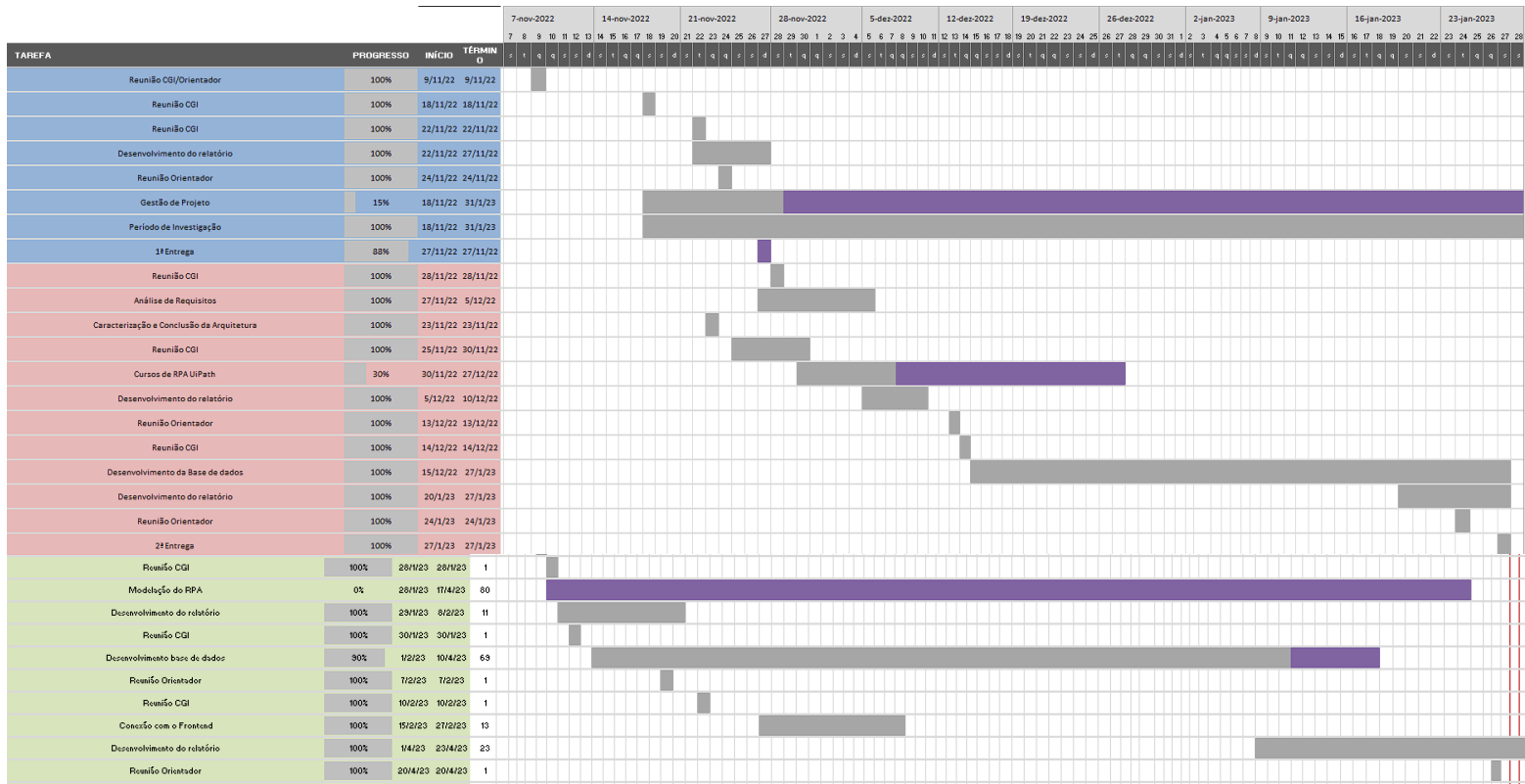


Figura 18 - Cronograma 1ª, 2ª e 3ª entrega

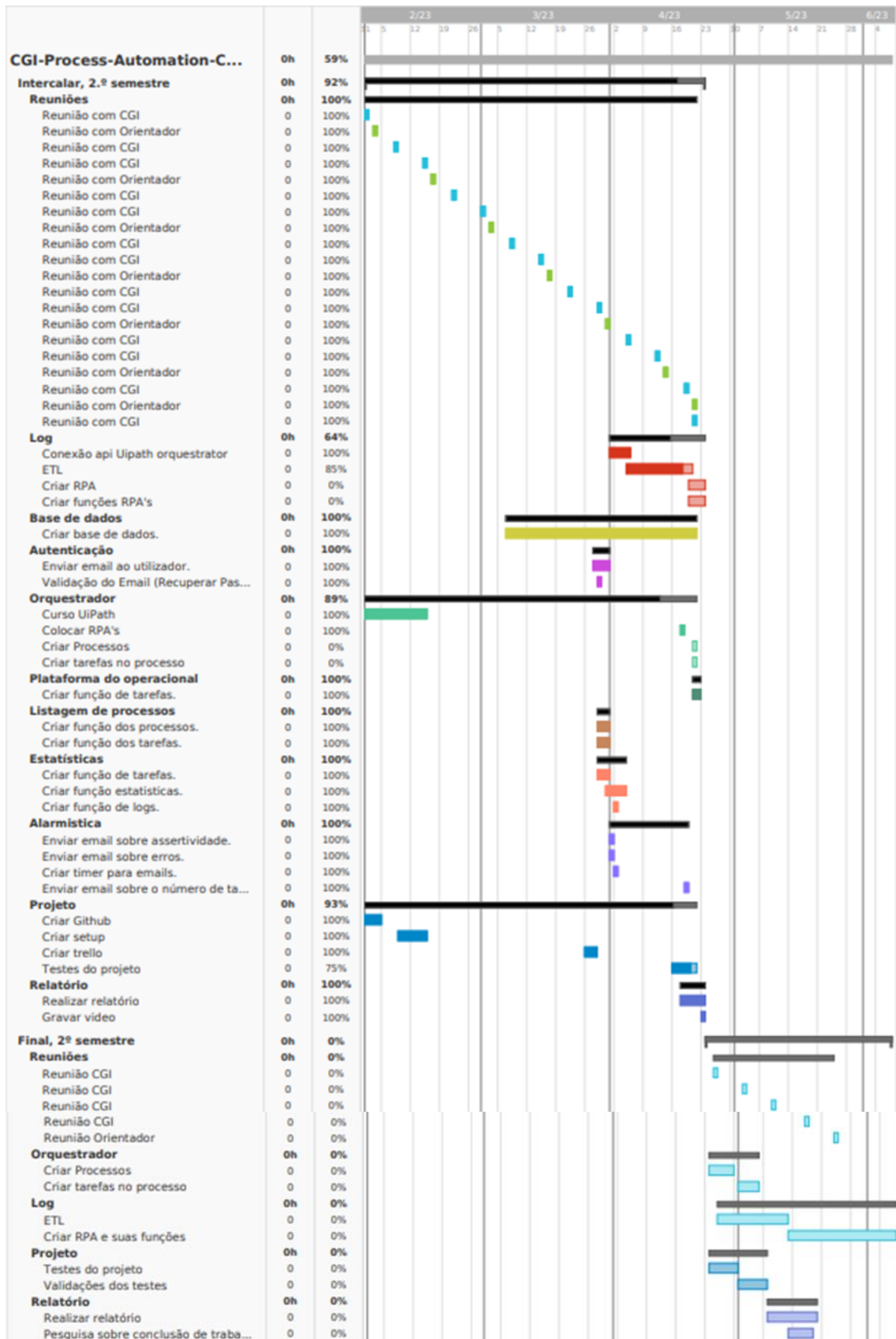


Figura 19 - Cronograma 3ª e 4ª entrega

Este anexo contém o cronograma da entrega anterior, sem edições, onde a cor azul representa a primeira entrega - intercalar 1º semestre; a rosa, a segunda entrega - intermédio 1º semestre; e a verde a terceira entrega - intercalar 2º semestre e na segunda imagem com o título de “intercalar, 2º semestre”; e a presente entrega com o título de Final, 2º semestre . Durante estas quatro entregas, o grupo esforçou-se por realizar tudo o que tinha proposto, mas algumas tarefas ficaram incompletas, como o curso de UiPath e a gestão do projeto.

Para uma melhor visualização dos cronogramas, optámos por mudar para o TeamGantt, permitindo uma visualização mais detalhada e clara das diferentes etapas e tarefas que tínhamos de realizar. Embora não soubéssemos exatamente o que esperar no desenvolvimento do projeto na última entrega, conseguimos alterar o cronograma para ficar mais detalhado e com melhor visualização. Conseguimos, também, concluir as tarefas que foram propostas. Apesar de alguns problemas colaterais, o grupo superou as suas expectativas, dedicando-se bastante ao trabalho. Com as reuniões semanais com o cliente, pudemos obter feedback constante sobre o desenvolvimento do projeto e as suas validações.

Glossário

LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso
RPA	Robotic Process Automation
IDP	Intelligent Document Processing
IA	Inteligência Artificial
DPA	Automação de processos digitais
IBPMS	Sistemas inteligentes de gestão de processos de negócios
DCM	Gestão dinâmica de casos
MVCC	Controle de Concorrência Multiversões
IDE	Ambiente de desenvolvimento integrado
SGBD	Sistema de gestão de base de dados
VM	Virtual Machine(máquina virtual)