

Aplicação para gestão de ordens de transporte

Tiago Manuel Almeida Costa

Relatório de projeto à Escola Superior de Tecnologia e Gestão para obtenção do grau de
mestre em Informática

Orientadores:

Prof. João Paulo Pereira

Bragança

Outubro de 2024

Resumo

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

O projeto desenvolvido teve como propósito desenvolver uma aplicação para gerir o transporte de ordens de transporte relativas a um determinado desenvolvimento.

Com esta aplicação pretende-se então agilizar o fluxo de trabalho do requerente do transporte e da equipa de transporte através da automatização de várias etapas que anteriormente seriam realizadas manualmente por ambas as partes.

Para o desenvolvimento deste projeto recorreu-se a uma metodologia Scrumban, consistindo em vários sprints, nos quais foram definidas diferentes tarefas com o objetivo de agilizar o processo de criação.

Relativamente aos dados utilizados no desenvolvimento da solução, os mesmos derivam de uma panóplia de dados já existentes no sistema de SAP ERP, como os dados de utilizadores e das ordens, como também de dados criados com o uso da aplicação, sendo estes no fim armazenados no ambiente com o auxílio de tabelas.

Em resumo, a solução alcançada produziu uma aplicação eficiente capaz de simplificar a gestão do transporte das ordens de transporte. A automação das etapas de release e transporte das ordens de transporte proporcionaram uma otimização ao fluxo de trabalho como um todo. Adicionalmente, a automatização da troca de emails ao longo do processo, além de aprimorar o fluxo, melhorou a produtividade de todos os intervenientes relacionados com o mesmo.

Palavras-chave: SAP, Low-Code No-Code, Ordens de Transporte, Webservices

Abstract

This report describes the project developed as part of the dissertation for the master's degree in Computer Science.

The purpose of the project was to develop an application to manage the transportation of transport requests for a particular development.

The aim of this application is to streamline the workflow of the transport requester and the transport team by automating various steps that would previously have been carried out manually by both parties.

A Scrumban methodology was used to develop this project, consisting of several sprints in which different tasks were defined with the aim of speeding up the creation process.

The data used to develop the solution was derived from a wide range of existing data in the SAP ERP system, such as user and order data, as well as data created using the application, which was then stored in the environment using tables.

In short, the solution produced an efficient application capable of simplifying the management of transport requests. Automating the stages of releasing and transporting transport requests has optimized the workflow as a whole. In addition, automating the exchange of emails throughout the process not only improved the flow, but also the productivity of all those involved.

Keywords: SAP, Low-Code No-Code, Transport Requests, Webservices

Índice Geral

Resumo	iii
Abstract.....	v
Índice Geral	vii
Lista de Siglas/Abreviaturas	ix
Índice de Figuras	xi
Índice de Tabelas	xiii
1. Introdução.....	1
1.1. Enquadramento.....	2
1.2. Objetivos.....	3
1.3. Estrutura do Documento.....	3
2. Contextualização e Estado de Arte.....	5
2.1. Contextualização	5
2.2. Estado de Arte	6
2.2.1. ChaRM	6
2.2.2. Enterprise Resource Planning.....	7
2.2.3. Low-Code No-Code	11
3. Ferramentas e Tecnologias Utilizadas.....	19
3.1. SAP ABAP.....	19
3.2. REST	20
3.3. JSON.....	20
3.4. SAP BTP	20
3.4.1. SAP Build Apps	21
3.4.2. SAP Intergration Suite.....	21
3.5. SMTP.....	21
3.6. Figma.....	22
3.7. Draw.io	22
4. Arquitetura e Modelação do sistema	23
4.1. Descrição do Sistema.....	23
4.2. Requisitos funcionais.....	24
4.3. Requisitos Não Funcionais	27
4.3.1. Requisitos Gerais.....	27
4.3.2. Tecnologias.....	27
4.4. Diagrama de Casos de Uso.....	28
4.5. Diagrama de Atividades	30

4.6.	Diagrama de Classes.....	31
4.7.	Modelação de dados	32
4.8.	Mockups	33
5.	Desenvolvimento	39
5.1.	Criação de objetos	39
5.1.1.	View ZV_E070_E07T	39
5.1.2.	Tabelas ZTRELEASE_REQK e ZTRELEASE_REQP	40
5.2.	Desenvolvimento da REST API.....	42
5.2.1.	Criação da ZCL_TRANSPORT_REQUEST	42
5.2.2.	Desenvolvimento da interface LIF_REST	43
5.2.3.	Desenvolvimento da classe LCL_REST	44
5.2.4.	Desenvolvimento da classe local LCL_REST_AUTH	46
5.2.5.	Desenvolvimento da classe local LCL_REST_TR	48
5.2.6.	Desenvolvimento da classe LCL_REST_REQUEST	50
5.3.	Integração da REST API no SAP Integration Suite	54
5.4.	Criação da aplicação no SAP Build Apps	55
5.4.1.	Integração da REST API	55
5.4.2.	App Variables	56
5.4.3.	Página de Login	56
5.4.4.	Página All Transport Request.....	57
5.4.5.	Página My Transport Request	58
5.4.6.	Página Transport Request Details.....	59
5.4.7.	Página Create Release Request.....	60
5.4.8.	Página Release Requests	61
5.4.9.	Página ReleaseRequestDetails.....	61
5.4.10.	Página Settings	63
6.	Demonstração da Aplicação	65
6.1.	Introdução.....	65
7.	Conclusões.....	69
7.1.	Conclusão	69

Lista de Siglas/Abreviaturas

OT Ordem de Transporte

TM Transport Management

ChARM Change Request Management

QA Quality Assurance

ERP Enterprise resource planning

SAP Systems Applications and Products

LCDP Low-code development platform

ABAP Advanced Business Application Programming

API Application Programming Interface

SMTP Simple Mail Transfer Protocol

Índice de Figuras

Figura 1: Quadrante Mágico para Cloud ERPs (reproduzido de [6]).....	9
Figura 2: Quadrante Mágico para Plataformas de Low-Code (reproduzido de [14])	16
Figura 3: Diagrama de Casos de Uso - Interação entre Utilizador e Aplicação.....	28
Figura 4: Diagrama de Casos de Uso - Interação entre Admin e Aplicação.....	29
Figura 5: Diagrama de Casos de Uso - Interação entre Aplicação e API.....	29
Figura 6: Diagrama de Atividades.....	31
Figura 7: Diagrama de Classes.....	32
Figura 8: Modelo de Dados.....	33
Figura 9: Mockup - Página de Login.....	34
Figura 10: Mockup - Página All Transport Requests.....	34
Figura 11: Mockup - Página My Transport Requests.....	35
Figura 12: Mockup - Página Transport Request Details.....	35
Figura 13: Mockup - Página Create Release Request.....	36
Figura 14: Mockup - Página Release Requests.....	36
Figura 15: Mockup - Página Release Request Details.....	37
Figura 16: Mockup - Página Settings.....	37
Figura 17: Teste dos endpoints.....	54
Figura 18: API Proxies.....	55
Figura 19: Configuração da REST API na aba Data.....	55
Figura 20: Integrações da REST API.....	56
Figura 21: Lógica de Login.....	57
Figura 22: Lógica do botão para criar pedidos de transporte.....	58
Figura 23: : Lógica da montagem da página Transport Request Details.....	59
Figura 24: Lógica do botão de criar pedido de transporte na página Transport Request Details.....	60
Figura 25: Lógica para criar pedidos de transporte.....	61
Figura 26: Lógica do botão editar.....	62
Figura 27: Lógica do botão cancelar.....	62
Figura 28: Lógica do botão confirmar.....	63
Figura 29: Lógica do botão rejeitar.....	63
Figura 30: Demonstração - Página de Login.....	66
Figura 31: Demonstração - Página My Transport Requests.....	66
Figura 32: Demonstração - Página Create Release Request.....	67
Figura 33: Demonstração - Email de criação.....	67
Figura 34: Demonstração - Página Release Request Details.....	68
Figura 35: Demonstração - Página de Settings.....	68

Índice de Tabelas

Tabela 1: Comparação entre ERPs	11
Tabela 2: Comparação das funcionalidades das plataformas LCDP (adaptado de [16])	18
Tabela 3: Campos da view ZV_E070_E07T	40
Tabela 4: Campos da tabela ZTRELEASE_REQK	41
Tabela 5: Campos da tabela ZTRELEASE_REQP	41
Tabela 6: Tipificações da interface LIF_REST	43
Tabela 7: Métodos HTTP definidos na interface LIF_REST	43
Tabela 8: Métodos auxiliares na interface LIF_REST	44
Tabela 9: Tipificações definidas na classe local LCL_REST_TR	48
Tabela 10: Tipificações da classe local LCL_REST_REQUEST	50

1. Introdução

Com as limitações das soluções fornecidas aos clientes pelos fabricantes de software, cabe a estes criar e desenvolver soluções que condigam às suas necessidades. Para tal é necessário existirem formas de armazenar esses desenvolvimentos e no caso de SAP não é diferente daí a origem das ordens de transporte.

As ordens de transporte são, de forma resumida, um pacote onde são guardadas as alterações realizadas em objetos (como programas, configurações ou até mesmo dados) e que são usadas para transportá-las entre diferentes sistemas SAP.

Apesar das ordens na sua forma básica parecem uma forma simples de armazenamento, elas têm uma importância enorme pois têm a capacidade de manterem a integridade e rastreabilidade das alterações, garantindo assim a estabilidade e consistência dos desenvolvimentos. Além disso, a documentação detalhada das modificações, possibilita um processo estruturado de revisão e aprovação, e consequente, a anulação de alterações caso necessário.

Assim, com o objetivo de manter as mais valias fornecidas pelas ordens de transporte criar-se-á uma aplicação com base no método de desenvolvimento Low-Code No-Code onde será possível gerir o transporte destas.

A motivação para a criação desta aplicação deve-se ao facto de que, atualmente, a maioria das etapas referentes a este processo são executados na sua grande maioria manualmente, desde a seleção/transporte de OTs, à troca de emails até à análise do pedido, o que torna o fluxo de trabalho ineficiente e pouco otimizado.

1.1. Enquadramento

Num cenário em constante transformação, a exigência por aprimorar o fluxo de trabalho torna-se evidente. Assim sendo, cada vez mais as empresas procuram soluções que simplifiquem as tarefas do fluxo, fazendo com que consequentemente aumentem a produtividade dos seus colaboradores.

Com esse propósito em mente para este projeto pretende-se desenvolver uma aplicação que permita a todos os intervenientes realizar a gestão do transporte de ordens de transporte de eficiente e otimizada.

A gestão dos transportes, como se encontra na sua forma atual, é um processo muito baseado na realização manual de múltiplas várias etapas. Dentro dessas várias etapas, as principais são o envio de emails seja no momento de criação do pedido ou até ao longo do processo devido à necessidade de mais informações e o transporte das ordens por parte da equipa de Transport Management, visto que dependendo do número de ordens a transportar será necessário verificar se estão liberadas e, caso não estejam, liberá-las. Além disso, o próprio momento de transporte requiere uma atenção redobrada aos detalhes, uma vez que será necessário selecionar as ordens corretas a transportar, mas também selecionar o sistema e o mandante, isto é, instância corretamente, senão poderão se originar problemas de integridade no sistema e mandante de destino.

Assim, como resultado final espera-se ter uma solução que possa oferecer às empresas a melhoria necessária para aprimorar o processo de transporte e as suas demais etapas.

1.2. Objetivos

O objetivo deste projeto será a criação de uma aplicação para a gestão do transporte de ordens de transporte (OT) relativas a um determinado desenvolvimento, agilizando assim o fluxo de trabalho do requerente do transporte como também da equipa de transporte através da automatização de várias etapas que anteriormente seriam realizadas manualmente por ambas as partes.

De maneira que este objetivo seja cumprido, o utilizador geral deverá ter a possibilidade de criar um pedido para a equipa de Transport Managment (TM) selecionando as OTs pretendidas, seja através de filtros ou de seleção direta.

Por sua vez, a equipa de TM deverá ter a capacidade de validar o pedido, seja autorizando-o ou rejeitando-o.

Além disso, para cumprir o principal objetivo será necessário alcançar a nível técnico diversas etapas. De forma mais detalhada tenciona-se:

- Desenvolver a arquitetura da aplicação, através da modelação e da criação das *mockups*;
- Desenvolvimento de uma API em SAP, disponibilizando diferentes *endpoints*;
- Integrar a API e os seus *endpoints* no SAP Integration Suite;
- Implementação de uma aplicação em SAP Build Apps.

1.3. Estrutura do Documento

A informação contida neste relatório encontra-se dividido em sete capítulos. Cada capítulo é dividido por secções para uma melhor definição dos mesmos. Todo o trabalho realizado está estruturado tal como se apresenta de seguida:

No capítulo 1 é introduzido o tema, bem como apresentado um enquadramento ao mesmo, citando também os objetivos estabelecidos.

No capítulo 2 é apresentada uma breve contextualização do tema, além de se abordar enunciado bem como o estado de arte. No capítulo 3 são levantados os conceitos teóricos refletidos neste projeto.

No capítulo 4 exibida a arquitetura e modelação desenvolvida. Assim sendo, são expostos os requisitos funcionais e não funcionais, diagramas, mockups e modelo de base de dados.

No capítulo 5 encontram-se pormenorizados os detalhes sobre o desenvolvimento e implementação.

No capítulo 6 apresenta-se uma breve demonstração da aplicação

No último capítulo é apresentada uma breve conclusão do trabalho.

2. Contextualização e Estado de Arte

Neste capítulo será apresentada a contextualização do projeto como também será abordado o estado de arte .

2.1. Contextualização

Como referido no capítulo anterior, após a conclusão de um desenvolvimento é necessário realizar o transporte das alterações para os restantes ambientes, desde a fases de desenvolvimento e de testes até à implementação destas em produção.

Atualmente, o processo de transporte de ordens é um processo muito burocrático e demorado devido as múltiplas normas existentes para mitigar potenciais riscos para o ambiente de produção caso os transportes não sejam devidamente planeados e controlados. A principais razões para um processo longo e demorado são as contantes trocas de emails entre os intervenientes e o tempo necessário para realizar a liberação e o transporte das próprias ordens.

Então, dada a necessidade de automatizar o processo de transporte das ordens, este projetivo visa permitir a todos os intervenientes realizar a gestão dos transportes de maneira eficiente e otimizada com a necessidade mínima de intervenção.

2.2. Estado de Arte

De maneira a cumprir o objetivo proposto para este projeto, isto é, automatizar o processo de transporte das ordens de transporte foram realizadas várias pesquisas acerca do tema de gestão de transportes com objetivo de apresentar uma análise do estado da arte relacionado ao tema através da revisão de novas soluções ou de soluções já existentes.

Das várias pesquisas realizadas, a única que vai de encontro com o escopo de SAP, foi o SAP ChaRM.

Além disso, serão também abordados os principais conceitos teóricos deste projeto, ERPs e programação Low-Code No-Code.

2.2.1. ChaRM

SAP ChaRM, acrónimo de Change Request Management, é uma ferramenta que gere as atividades realizadas durante a alteração de objetos, desde a fases iniciais de desenvolvimento e de testes até à implementação destas no sistema de produção.

Esta ferramenta permite aos utilizadores acompanhar os pedidos de alteração, disponibilizando a estes a capacidade de conceder autorizações para a alteração de código, supervisionar as alterações realizadas e manter documentação para comprovar as aprovações dadas.

Das múltiplas funcionalidades disponibilizadas pelo ChaRM, a que mais é realçada é o transporte de ordens de transporte, que como já mencionada, são essencialmente pacotes que contêm as alterações de código. Estes transportes facilitam a passagem do código da fase de desenvolvimento para a fase de Quality Assurance (QA) e, finalmente, para o ambiente de produção.

Como estas alterações podem ocorrer a qualquer altura, a gestão das alterações também acarreta potenciais riscos para o ambiente de produção caso não sejam devidamente planeadas e controladas.

Desta forma, ao fazer o controlo dos transportes, o ChaRM garante uma transição perfeita e controlada das alterações realizadas, aumentando assim a eficiência e reduzindo os riscos associados às alterações.

2.2.2. Enterprise Resource Planning

ERP, Enterprise Resource Planning, pode ser descrito como uma panóplia de ferramentas de gestão disponíveis a uma empresa que permitem relacionar clientes e fornecedores numa cadeia de abastecimento completa, aplicando processos comprovados para a tomada de decisões e proporcionando elevados graus de integração entre os diferentes setores de uma empresa [1]

Por outras palavras, o ERP pode ser considerado como um sistema que permite a partilha de informações e comunicação entre os vários departamentos através de uma base de dados única e central, interligando os recursos permitindo, assim, geri-los em tempo real [2].

Assim, com a necessidade de adquirir métodos de planeamento cada vez mais eficientes e, simultaneamente, com aumento da concorrência a nível empresarial, o surgimento de sistemas de ERP tornou-se decisivo. Com o crescimento exponencialmente de um mercado mais competitivo, as empresas enfrentam hoje um enorme desafio para manter o controlo sobre todos os seus processos e informações, enquanto se tentam destacar do resto do mercado oferecendo custos reduzidos e mantendo os lucros elevados. Para isso, é vital contar com ferramentas dedicadas que possam mensurar cada detalhe do negócio de forma automatizada e em tempo real, garantindo a agilidade e eficiência do trabalho diário da organização [3].

Vantagens e desvantagens de um sistema ERP

Aquando da escolha de um sistema ERP é necessário verificar se o mesmo se adequa às funcionalidades pretendidas pela empresa e se, conseqüentemente, este trará benefícios,

tais como, a melhora da qualidade de serviço e do aumento no corte de gastos. Além disso, também é importante que o supra citado tenha a capacidade de customização conforme as necessidades, estratégias e a estrutura da empresa [4].

Assim, tendo em conta o que foi referido anteriormente serão listados abaixo os possíveis benefícios que um sistema ERP pode prover [5]:

- **Acesso a informação fiável:** Base de dados central e com informação precisa;
- **Evita informação e operações redundantes:** Os módulos acedem à mesma informação através de uma base de dados comum, evitando assim a entrada de dados duplicados;
- **Minimiza a recuperação e atrasos nos relatórios:** Redução do tempo de ciclo de entrega;
- **Redução dos custos:** Poupança de tempo com melhoria do controlo sobre toda a empresa;
- **Adaptação facilitada:** Mudança nos processos de negócio facilitada para adaptação e reestruturação;
- **Maior escalabilidade:** Design modular e estruturado com *add-ons*;
- **Alcance global:** Módulos alargados como o CRM e SCM.

No entanto, para colher os benefícios de um ERP, as organizações precisam também de ultrapassar certos problemas e desvantagens, sendo estas [5]:

- **Consumo de tempo:** Minimizar os assuntos sensíveis e consenso nas políticas internas;
- **Custo:** O custo pode variar de milhares a milhões de dólares. O custo de reengenharia dos processos de negócio pode ser extremamente dispendioso;
- **Conformidade dos módulos:** A arquitetura do sistema tem de estar de acordo com o processo de negócio e os objetivos da empresa;
- **Dependência do vendedor:** Um único vendedor versus vendedores múltiplos. Escolher o que melhor se compromete a um suporte de longa duração;
- **Características e complexidade:** Diferentes ERPs podem ter diferentes características e módulos, por isso é vital que as empresas analisem cuidadosamente as suas necessidades;

- **Escalabilidade e alcance global:** Analisar o investimento feito pelo vendedor em investigação e desenvolvimento e comprometimento a longo prazo com o produto;
- **Capacidade do ERP alargado:** Considerar se existe a possibilidade da instalação de *add-ons* em módulos alargados como o CRM e SCM.

Sistemas ERP

Atualmente existem múltiplas ferramentas ERP no mercado. Cada uma destas ferramentas tem as suas próprias características, tornando-as únicas. Esta diversidade permite que os utilizadores escolham a plataforma que melhor se adapta às suas necessidades e objetivos.

Na Figura 1 é apresentado o quadrante mágico da Gartner dos Cloud ERPs mais cotados no mundo.



Figura 1: Quadrante Mágico para Cloud ERPs (reproduzido de [6])

Como é possível ver na figura anterior, os cloud ERPs com maior destaque são o SAP S/4 Hana, Oracle Fusion Cloud e o Microsoft Dynamics.

SAP ERP

O SAP ERP é o principal produto da SAP AG, uma empresa alemã líder no desenvolvimento de software empresarial. Este sistema integrado de gestão empresarial transaccional, com mais de meio século de história, é geralmente implementado em grandes corporações, pois fornece as mais recente tecnologias, arquiteturas e metodologias quando introduz os seus novos produtos, substituindo, no entanto, gradualmente as tecnologias e abordagens mais antigas o que é algo que as grandes empresas almejam devido à sua grande escala o que demonstra a robustez da solução [7].

O sistema SAP vê uma empresa como um todo, em que cada área específica corresponde a um modulo em SAP. Desta forma, é possível melhorar o desempenho da organização, minimizando tempo, custos e recursos.

Com a implementação completa de um sistema SAP ERP, a empresa ganha controlo total sobre todas as atividades através de um sistema centralizado, aumentando assim a sua produtividade.

Oracle Fusion Cloud ERP

Oracle Fusion Cloud ERP, desenvolvido pela Oracle, é um conjunto de aplicações de ERP na *cloud*, concebidas para ajudar as empresas a gerir e automatizar os seus processos.

Este ERP tira partido de tecnologias como IA, *machine learning* e a análise de dados para otimizar processos e fornecer informações em tempo real.

Por ser uma solução baseada na *cloud*, promove uma redução de custos, mais especificamente, na infraestrutura que seria acomodar caso o mesmo fosse *on-premise*.

Com a sua arquitetura modular e flexível, o Oracle Fusion Cloud ERP permite às empresas escalar as operações e integrar-se com outros serviços sem problemas.

Microsoft Dynamics

Microsoft Dynamics é uma solução ERP versátil que oferece uma gama de aplicações integradas para a gestão empresarial. Este ERPD abrange áreas-chave como as finanças, *supply chain* e recursos humanos. Uma característica que se destaca é a sua integração perfeita com outros produtos Microsoft, promovendo a colaboração e a consistência dos dados em toda a empresa.

O Microsoft Dynamics é caracterizado pela sua interface visual de fácil utilização e adaptabilidade, o que o torna uma escolha apelativa para as organizações [8].

Comparação das plataformas

Nesta secção será apresentada uma comparação das soluções levantadas anteriormente com base em características e funcionalidades analisadas e levantadas por Shahzad [8]

Tabela 1: Comparação entre ERPs

Módulos/Métricas	SAP	Oracle	Microsoft Dynamics
Supply Chain Management (SCM)	<ul style="list-style-type: none"> - Integração de análise avançada - Automação de processos - Soluções para mitigação de riscos 	<ul style="list-style-type: none"> - Flexibilidade e escalabilidade - Análise em tempo real - Logística otimizada e cumprimento de encomendas 	<ul style="list-style-type: none"> - XIntegração com Microsoft Office - Interface intuitiva - XMenor complexidade para empresas de menor dimensão
Customer Relationship Management (CRM)	<ul style="list-style-type: none"> - Integração profunda com outros produtos SAP - Interações personalizadas com clientes - Recomendado a grandes empresas 	<ul style="list-style-type: none"> - Alta flexibilidade e escalabilidade - IA para análises preditivas 	<ul style="list-style-type: none"> - Integração com Office - Interface intuitiva - Foco na melhoria das relações dos clientes e dos processos de negócio
Finance and Controlling	<ul style="list-style-type: none"> - Controlo financeiro avançado - Relatórios extensivos - Conformidade com normas internacionais 	<ul style="list-style-type: none"> - Gestão financeira flexível e escalável - Integração com outras aplicações Oracle 	<ul style="list-style-type: none"> - Fácil de usar - Integração com ferramentas do Microsoft Office
Manufacturing and Material Management	<ul style="list-style-type: none"> - Planeamento de produção em tempo real - Integração com módulos SCM e MES 	<ul style="list-style-type: none"> - Alta flexibilidade e escalabilidade - Planeamento e controlo de produção - Gestão de qualidade 	<ul style="list-style-type: none"> - Adequado para empresas de menor dimensão - Integração rápida com outros processos empresariais
Market Share	19%	13%	16%
Custo total de aquisição (em \$)	\$2.09 milhões	\$2.38 milhões	\$2.06 milhões
Período de implementação	23.1 meses	24.5 meses	23.6 meses

2.2.3. Low-Code No-Code

Com o aumento contínuo da necessidade de novos requisitos e com a consequente falta de recursos especializados, seja por estes se encontrarem sobrecarregados ou pela falta

efetiva deles, as empresas têm verificado a necessidade de respostas rápidas e mais flexíveis para os problemas com que se deparam.

Assim, de maneira a resolverem os vários problemas que surgem é necessário encontrar uma forma de desenvolver soluções, sejam elas qual forem, de uma forma mais rápida. É aqui então onde surge a necessidade do uso de plataformas low-code, visto que estas exigem menos uso código (ou até mesmo nenhum) diminuindo por consequência o tempo de desenvolvimento das mesmas.

Low-code, segundo da Cruz [9], pode ser definido como “um conceito (..) que transforma representações gráficas em software funcional, permitindo a qualquer pessoa ser o programador (...)”. Por sua vez, Dushnitsky [10] complementam afirmando que com o aparecimento de plataformas low-code, o desenvolvimento de soluções pode ser realizado com pouco ou até mesmo nenhum código, tornando assim o seu uso quase inexistente.

Além disso, com o decorrer do tempo essas plataformas também oferecem um melhor custo-benefício seja na implementação de novas soluções, devido à diminuição substancial no tempo de desenvolvimento por causa do menor grau de complexidade necessário, ou a fácil manutenção de soluções longo prazo pois será mais fácil examinar o “código” já existente [10].

Como caso de exemplo, Dushnitsky [10] apresentam a plataforma Shopify que permite aos seus utilizadores, independentemente do nível de conhecimento prévio, desenvolver a sua plataforma de e-commerce. Esclarecem ainda que segundo a análise realizada, start-ups criadas no Shopify começam com um número menor de recursos, sejam eles financeiros ou humanos, comparando as suas congéneres de e-commerce, contudo elas conseguem alcançar um nível semelhante de sucesso, pelo facto de que do valor criado pelos seus funcionários é superior as dos seus concorrentes.

Citizen Development

Com o surgimento dos desenvolvimentos em *low-code*, surgiu também um novo tipo de programadores, os *citizen developers*.

O *Citizen Development*, é uma solução disruptiva para colmatar a lacuna existente no mundo atual em termos de conhecimentos técnicos, especialmente no que diz respeito aos programadores profissionais. Combinado com ao desenvolvimento *low-code*, dá poder

àqueles que não escrevem código, mas que têm uma boa compreensão das necessidades e desafios empresariais. São estas pessoas que podem agora criar aplicações que ajudam a responder melhor aos requisitos do seu trabalho quotidiano, reduzindo simultaneamente os riscos de erros humanos e oferecendo melhores experiências aos clientes.

Desenvolvimento Low-code vs Desenvolvimento tradicional

Nos últimos anos, tem-se observado um aumento significativo na popularidade do desenvolvimento low-code. Este fenómeno pode ser atribuído a uma variedade de fatores, contudo, com esse crescimento em popularidade, a comparação entre o desenvolvimento de soluções em low-code e da forma tradicional tem aumentado entre os programadores. De forma a compreender ambas as formas de desenvolvimento pretende-se explorar um pouco das duas abordagens, examinando as suas vantagens e desvantagens.

O desenvolvimento em plataformas low-code permite aos utilizadores criar aplicações usando interfaces gráficas, em vez de escrever código. Entre as suas vantagens encontram-se [11]:

- **Maior agilidade:** Faz com que todas as funcionalidades de solução funcionem sem qualquer tipo de problema em diferentes tipos de dispositivos.
- **Redução de custos:** Com a capacidade de criar mais aplicações em menos tempo, os custos diminuem. Além disso, reduz a necessidade de mais programadores, reduzindo os custos de contratação.
- **Maior produtividade:** Como o desenvolvimento *low-code* permite que mais aplicações sejam construídas em menos tempo. Com o desenvolvimento *low-code*, o tempo deixa de ser uma barreira para a inovação real.
- **Melhor experiência do cliente (UX):** Com o desenvolvimento *low-code*, as organizações podem adaptar-se rapidamente às mudanças do mercado ou às necessidades dos clientes.
- **Gestão de risco e governação eficazes:** O desenvolvimento *low-code* permite uma mudança rápida, para que possa cumprir os requisitos regulamentares e manter-se à frente dos prazos.
- **Mudança fácil:** O desenvolvimento *low-code* facilita a alteração das aplicações, adaptando-as a novos requisitos. Sem exigir codificação complexa, o

desenvolvimento com pouco código facilita a mudança imediata quando necessário.

- **Transformação mais rápida:** O desenvolvimento com pouco código elimina a complexidade da criação de aplicações empresariais excelentes e modernas. E a complexidade reduzida significa uma navegação mais suave.

No entanto, as LCDPs têm limitações no que diz respeito ao [12]:

- **Menor nível de personalização:** O desenvolvimento low-code pode não oferecer a mesma capacidade de personalização comparada ao desenvolvimento tradicional. Isso pode resultar em aplicações que não são totalmente adaptadas às necessidades específicas de uma empresa ou usuário.
- **Limite nas opções de hospedagem fornecidas pela plataforma:** Com o desenvolvimento em plataformas *low-code* fica-se limitado às opções fornecidas pelo fornecedor da plataforma, o que pode limitar a flexibilidade e controlo sobre o ambiente em que queremos hospedar as soluções desenvolvidas.
- **Interfaces com aparência padrão:** As aplicações desenvolvidas em plataformas *low-code* podem ter uma aparência “padrão” que não se destaca ou não reflete a marca da sua empresa.
- **Difícil ou impossível integração com sistemas *legacy*:** As plataformas *low-code* podem ter dificuldades para se integrar com sistemas legados, o que pode ser um grande desafio se a sua empresa depende de tais sistemas.
- **Escalabilidade limitada:** Embora as plataformas *low-code* possam ser ótimas para criar aplicações rapidamente, elas podem não ser capazes de escalar eficientemente à medida que as empresas crescem.

Por outro lado, o desenvolvimento tradicional envolve a escrita manual de código e o uso de linguagens de programação¹. Entre as suas vantagens encontram-se [11]:

- **Funcionalidades ilimitadas:** O desenvolvimento tradicional permite criar praticamente qualquer tipo de funcionalidade, além de permitir diversos tipos de integração l.
- **Controlo total:** Quem opta por desenvolver aplicações da maneira tradicional possui o controlo total sobre todos os aspetos da solução a ser desenvolvida pois detêm a totalidade do código fonte criado e, portanto, controlam a arquitetura, a segurança e a integrabilidade da aplicação, entre outros aspetos.

- **Processo de desenvolvimento suave:** Ao recorrer ao desenvolvimento tradicional é necessário ter um processo de desenvolvimento bem definido e estabelecido e, para isso, normalmente espera-se o uso das melhores práticas de DevOps, garantindo assim que a solução desenvolvida seja facilmente mantida e atualizada.

No entanto, tal como as LCDPs, o desenvolvimento tem limitações no que diz respeito à:

- **Custos elevados:** A programação tradicional requer um orçamento elevado, seja pelo longo tempo de desenvolvimento ou pela necessidade de recursos especializados.
- **Longo tempo de desenvolvimento:** O desenvolvimento de aplicações da maneira tradicional demora mais tempo em comparação com o desenvolvimento em plataformas *low-code*. Apesar de existirem várias bibliotecas e *frameworks* que podem acelerar o processo desenvolvimento, a maior parte do código é realizado manualmente, o que obviamente demora muito tempo.
- **Requer recursos qualificados:** Para criar uma aplicação é necessária uma equipa com um elevado número de elementos. Habitualmente nessas equipas podem se encontrar programadores, engenheiros de controlo de qualidade, designers entre outros especialistas.

Em resumo, tanto o desenvolvimento low-code como o tradicional têm os seus méritos. A escolha entre um e outro deve ser baseada nas necessidades específicas do projeto, no tempo disponível para o desenvolvimento e na experiência da equipa de desenvolvimento.

Kermanchi [13], no seu artigo explora as diferenças entre a experiência dos programadores ao usar plataformas de desenvolvimento tradicionais comparativamente a plataformas de desenvolvimento low-code. No seu estudo concluiu que a experiência prévia em programação parece influenciar fortemente o desempenho dos desenvolvedores, as suas experiências e as suas preferências de ferramentas. Além disso, a maioria dos desenvolvedores permanece cética quanto à escalabilidade e manutenção de aplicações desenvolvidas usando LCDPs.

Plataformas Low-code

No mundo do desenvolvimento low-code, existe uma grande diversidade de plataformas low-code. Cada uma destas plataformas tem as suas próprias características, tornando-as

únicas. Esta diversidade permite que os utilizadores escolham a plataforma que melhor se adapta às suas necessidades e objetivos.

Na Figura 3 é mostrado o quadrante mágico da Gartner das plataformas low-code mais cotadas no mundo.

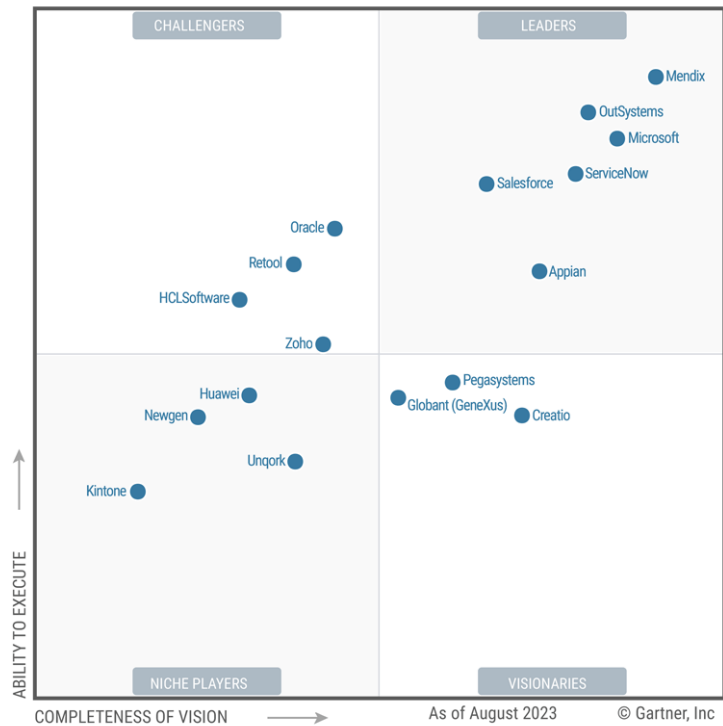


Figura 2: Quadrante Mágico para Plataformas de Low-Code (reproduzido de [14])

Como é possível ver na figura anterior algumas das plataformas *low-code* líderes de mercado com maior destaque são a Mendix, OutSystems e Microsoft Power Apps.

De seguida serão apresentadas algumas informações sobre estas plataformas, além de um comparativo as três plataformas.

Mendix

A Mendix é uma plataforma concebida para impulsionar a criação de aplicações em todas as fases do ciclo de vida do desenvolvimento, desde a ideia inicial até ao desenvolvimento, implementação e gestão contínua das aplicações, seja na *cloud* ou localmente. A Mendix disponibiliza duas opções de programação: no-code, que permite a modelação visual, e low-code, que oferece ferramentas extensíveis e integráveis para apoiar equipas multifuncionais que trabalham de forma colaborativa. Isto permite que especialistas em negócios, como analistas, colaborem com programadores especializados

para alcançar uma grande agilidade nos negócios e uma entrega acelerada. Além disso, a arquitetura nativa da *cloud* da Mendix e as suas ferramentas de automação facilitam a implementação, gestão e monitorização de recursos altamente disponíveis.

Outsystems

OutSystems é uma plataforma *low-code* utilizada para o desenvolvimento de aplicações web e mobile que possibilita aos programadores desenvolver novas soluções de alta qualidade com rapidez e eficiência sem terem de lidar com a complexidade inerentes às linguagens de programação tradicionais.

OutSystems apresenta-se num ambiente de desenvolvimento visual orientado por modelos e auxiliado por inteligência artificial (IA), garantindo assim que as aplicações desenvolvidas nela sejam construídas em dias ou semanas em vez de meses ou anos. Os serviços disponíveis na plataforma, assistidos por IA, proporcionam uma automação que aprimora todo o ciclo de vida das aplicações. Com isso, podem ser implementadas com um simples clique e geridas com uma facilidade sem precedentes.

Microsoft Power Apps

O Microsoft Power Apps é uma plataforma *low-code* em *cloud* que oferece um ambiente de programação rápida, serviços e conectores, além de uma plataforma de dados. O objetivo desta LCDP consiste em criar aplicações empresariais sem codificação manual.

São oferecidas duas abordagens diferentes no desenvolvimento de uma aplicação.

A primeira abordagem é designada por aplicações de tela e permite aos utilizadores desenvolver uma aplicação a partir do zero. Após a ligação às várias fontes de dados disponibilizadas, a aplicação pode ser criada através do *drag-and-drop* conforme o que é desejado.

Na segunda abordagem, designada por aplicações condicionadas por modelo, o foco é colocado nos componentes. A conceção da aplicação final é em grande parte pré-determinada pelos componentes utilizadas e não pode ser personalizada. Além disso, este tipo de abordagem utiliza uma plataforma de dados subjacente denominada de Dataverse, que armazena os dados em tabelas e pode ser utilizado na aplicação. [15]

Comparação das plataformas

Nesta subsecção será apresentada uma comparação das plataformas levantadas anteriormente com base em características e funcionalidades normalmente disponibilizadas em várias das plataformas.

Tabela 2: Comparação das funcionalidades das plataformas LCDP (adaptado de [16])

Ferramenta	Funcionalidades de Desenvolvimento					Tecnologia de Base de Dados			Tipo de app alvo				Funcionalidades da app gerada		Deploy ment e Operações	
	Linguagem	Suporte à escrita de código	Hot reload	3rd party plugins	Editor Visual	SQL Database	Graph Database	Key-value DB	Web	iOS	Android	Desktop	Live Collaboration	REST API	Severless/Cloudnativ	Monitoração
Mendix	Java	X		X	X	X			X	X	X			X	X	X
Outsystems	C#,Javascript	X		X	X	X	X	X	X	X	X				X	X
Powerapps+ Azure	PowerFx	X	X	X	X	X	X		X				X	X	X	X

3. Ferramentas e Tecnologias Utilizadas

Neste capítulo apresentar-se-á as ferramentas e tecnologias essenciais para a realização deste projeto apresentando uma descrição de cada uma delas. Serão exploradas as linguagens de programação , protocolos e também plataformas usadas.

3.1. SAP ABAP

SAP ABAP mais bem conhecido pela sua abreviação ABAP, acrónimo de *Advanced Business Application Programming*, é uma linguagem de programação desenvolvida pela empresa alemã SAP para a implementação de soluções no ambiente SAP.

ABAP é considerada como uma linguagem multiparadigma, isto é, podem-se utilizar princípios de programação procedural e de orientação a objetos. Embora seja a principal linguagem de programação do SAP, os programas escritos com ABAP podem ser executados juntamente com aqueles baseados em outras linguagens de programação, como Java, JavaScript e SAPUI5.

Desta forma, SAPABAP cumpre um papel fundamental no SAP, visto que permite às empresas desenvolver as suas próprias soluções customizadas não estando assim dependentes da SAP para responder às suas necessidades.

3.2. REST

REST, *Representational State Transfer*, refere-se a um conjunto de restrições arquiteturais que geram sistemas distribuídos eficientes, confiáveis e escaláveis. Um sistema é denominado RESTful quando adere a todas essas restrições [17].

REST API, conhecida também por RESTful API, é uma interface de programação de aplicações (API) que adere aos princípios da arquitetura. As API REST oferecem um modo flexível e simples de integrar aplicações [18].

3.3. JSON

JSON, *JavaScript Object Notation* (em português, Notação de Objetos JavaScript), é uma formatação leve de troca de dados, de leitura e escrita fácil para humanos e de fácil interpretação e criação para máquinas. É baseado em um subconjunto da linguagem de programação JavaScript. JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados [19].

3.4. SAP BTP

SAP BTP, *Business Technology Platform*, é uma plataforma que combina automação, desenvolvimento de aplicações, inteligência artificial, *data analytics*, e integração tudo num só único ambiente, oferecendo assim um portfólio integrado de tecnologias SAP.

O SAP BTP foi introduzido pela primeira vez em 2021, quando a SAP decidiu agrupar vários serviços e produtos numa única plataforma que as empresas poderiam utilizar.

O SAP BTP inclui duas das tecnologias utilizadas neste projeto, o SAP Build Apps e SAP Integration Suite.

3.4.1. SAP Build Apps

O SAP Build Apps é uma plataforma de desenvolvimento *low-code* para criar aplicações para *mobile* nativas e para *Web*, incluindo também capacidades de *backend*.

O SAP Build Apps permite a qualquer pessoa desenvolver aplicações *fullstack* de uma forma visual. Os utilizadores do negócio podem criar aplicações de nível empresarial que funcionam perfeitamente em todos os sistemas operativos, sem escrever uma única linha de código, enquanto os programadores reduzem o esforço necessário na criação de modelos de dados e na codificação da lógica de complexa, beneficiando-se assim com um aumento significativo de produtividade [20].

3.4.2. SAP Intergration Suite

O SAP Integration Suite é uma plataforma de integração que ajuda as empresas a ligarem e integrarem facilmente as suas aplicações, dados e processos dentro e fora da sua organização. O Integration Suite disponibiliza uma coleção pacotes de integração pré-construídos e de melhores práticas que simplificam e aceleram as integrações ajudando assim os programadores [21].

Os principais componentes oferecidos pelo SAP Integration Suite incluem a integração na *cloud*, a gestão de APIs, Open Connectors, o Integration Advisor e o SAP API Business Hub.

3.5. SMTP

SMTP, isto é, Protocolo de Transferência de Correio Simples, é um protocolo standard para transmitir correio eletrónico através de uma rede. Assim como outros protocolos de

rede, o SMTP permite que computadores e servidores troquem dados independentemente do hardware ou software usado. Desta forma, o SMTP padroniza a maneira como o email viaja do remetente ao destinatário, tornando possível a entrega generalizada destes [22].

3.6. Figma

Figma é um editor online de design gráfico focado na criação de interfaces gráficas (UI) e experiência de usuário (UX) com ênfase colaborativa. A ferramenta permite às equipas ter acesso em simultâneo, trabalhando assim todos num mesmo projeto ao mesmo tempo [23].

Com uma interface intuitiva, o Figma permite que os utilizadores construam ecrãs e fluxos de interação, além de possibilitar feedback em tempo real e integração fácil com outras ferramentas de design. Por ser baseado na web, o Figma pode ser acessado a partir de qualquer lugar, facilitando o trabalho remoto e colaborativo.

3.7. Draw.io

O draw.io é uma ferramenta personalizada para criar diagramas e gráficos. A ferramenta fornece uma vasta gama de formas e centenas de componentes gráficos que permitem criar diagramas únicos. Além disso, visto ser uma ferramenta web, o draw.io está totalmente integrado no Google Drive, o que permite guardar automaticamente os gráficos criados [24].

Devido à vasta gama de componentes oferecidos na criação de diagramas e gráficos, esta solução permite criá-los de forma rápida e eficiente.

4. Arquitetura e Modelação do sistema

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

Nesse sentido, são demonstrados os requisitos funcionais e não funcionais, o diagrama de classes, o diagrama de atividades, o diagrama de casos de uso e as *mockups*, tendo como objetivo ilustrar com mais clareza e detalhe o funcionamento da aplicação, bem como as atividades que possam ser realizadas.

4.1. Descrição do Sistema

O sistema é uma aplicação móvel projetada para a gestão do transporte de ordens de transporte. O público-alvo da aplicação inclui desenvolvedores e equipa funcional de determinados desenvolvimentos e projetos, para liberar as ordens relativas a estes últimos, porém também ajuda a equipa de TM a gerir os pedidos de forma eficiente.

A aplicação permitirá aos utilizadores criar pedidos para o transporte de ordens, de forma eficiente e intuitiva. A solução proposta visa automatizar as etapas manuais referentes a este processo, desde a transporte de ordens, à troca de emails até à análise do pedido, pretendendo assim tornar o fluxo de trabalho otimizado e eficiente.

4.2. Requisitos funcionais

Ver ordens de transporte

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores ver as ordens de transporte, devolvendo uma listagem das mesmas .

Ver detalhes da ordem de transporte

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores ver os detalhes de cada ordem de transporte.

Criar pedidos de transporte

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores criar pedidos de transporte das ordens de transporte selecionadas.

Editar pedidos de transporte

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores editar pedidos de transporte já criados.

Cancelar pedidos de transporte

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;

- **Descrição:** A aplicação deve permitir aos utilizadores cancelar pedidos de transporte já criados.

Ver detalhes dos pedidos de transporte

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores ver os detalhes dos pedidos de transporte criados.

Aceitar pedidos de transporte

- **Ator:** Admin;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores admin aceitar os pedidos de transporte realizados.

Rejeitar pedidos de transporte

- **Ator:** Admin;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores admin rejeitar os pedidos de transporte realizados.

Ver definições

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;
- **Descrição:** A aplicação deve permitir aos utilizadores ver as definições da aplicação.

Alterar definições

- **Ator:** Qualquer utilizador da aplicação;
- **Tecnologia:** Aplicação *Low-Code*;

- **Descrição:** A aplicação deve permitir aos utilizadores alteração as definições da aplicação.

Envio de email automático ao criar pedido de transporte

- **Ator:** SAP;
- **Tecnologia:** SAP;
- **Descrição:** O sistema deve permitir o envio de emails aquando da criação do pedido de transporte.

Envio de email automático ao cancelar pedido de transporte

- **Ator:** SAP;
- **Tecnologia:** SAP;
- **Descrição:** O sistema deve permitir o envio de emails aquando do cancelamento do pedido de transporte.

Envio de email automático ao aceitar o pedido de transporte

- **Ator:** SAP;
- **Tecnologia:** SAP;
- **Descrição:** O sistema deve permitir o envio de emails ao aceitar o pedido de transporte.

Envio de email automático ao rejeitar o pedido de transporte

- **Ator:** SAP;
- **Tecnologia:** SAP;
- **Descrição:** O sistema deve permitir o envio de emails ao aceitar o pedido de transporte.

4.3. Requisitos Não Funcionais

4.3.1. Requisitos Gerais

Performance e Confiabilidade

A aplicação deverá ter uma boa performance tendo em vista as ações do utilizador e, além disso, deverá apresentar informação em tempo real.

Utilidade

A aplicação deve ser fácil de utilizar e deve garantir a redução do tempo de transportes de ordens, que é um processo lento e propicio a erros.

Interface Gráfica

A aplicação deve oferecer uma interface graficamente simples, mas agradável e fácil de entender.

4.3.2. Tecnologias

SAP

Software utilizado para desenvolver a API.

SAP ABAP

Linguagem de programação utilizada para implementar a API no SAP ERP

SAP Build Apps

Solução utilizada para o desenvolvimento da aplicação.

4.4. Diagrama de Casos de Uso

O diagramas de casos de uso permite visualizar as interações entre um sistema e os seus utilizadores, ajudando a identificar as suas principais funcionalidades e como os utilizadores interagem com este.

Na Figura 3 é possível observar todas as funcionalidades disponíveis na aplicação para um utilizador comum, sendo a mais importante a criação dos pedidos de transporte.

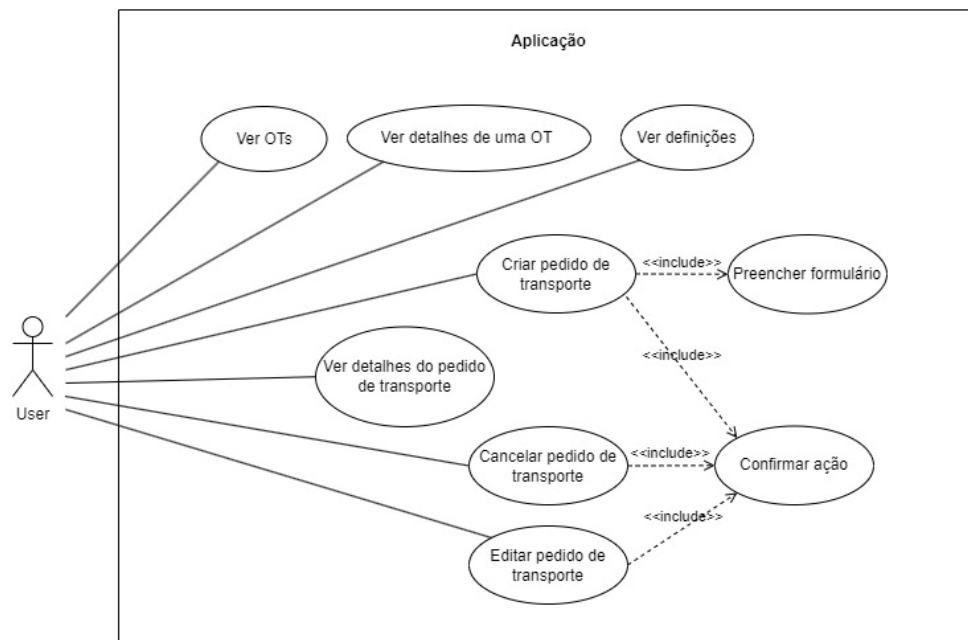


Figura 3: Diagrama de Casos de Uso - Interação entre Utilizador e Aplicação

Na Figura 4 são apresentadas todas as funcionalidades que um admin, mais especificamente, a equipa de TM pode realizar. Diferente do utilizador comum, o admin pode aceitar e rejeitar o pedido de transporte.

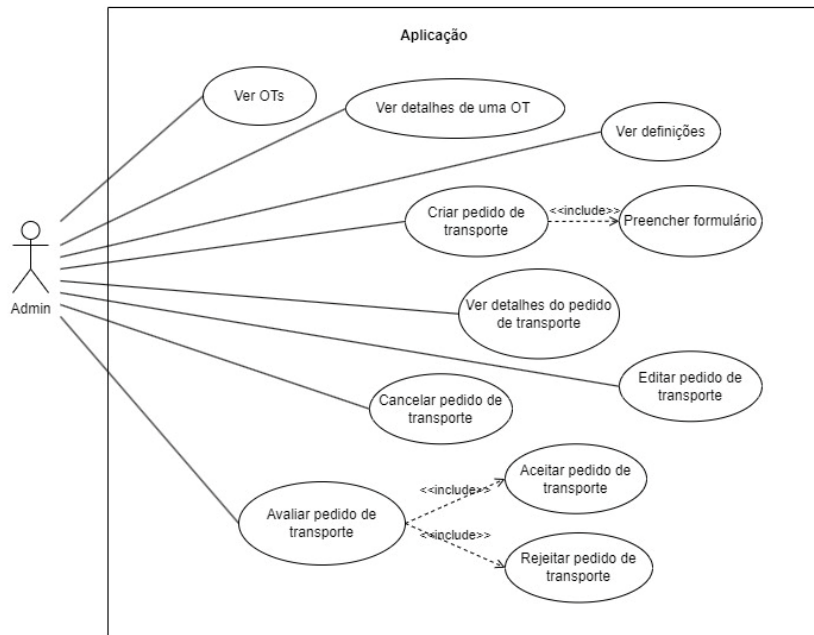


Figura 4: Diagrama de Casos de Uso - Interação entre Admin e Aplicação

Na Figura 5 é apresentado todos os métodos HTTP utilizados pela aplicação, de forma mais específica.

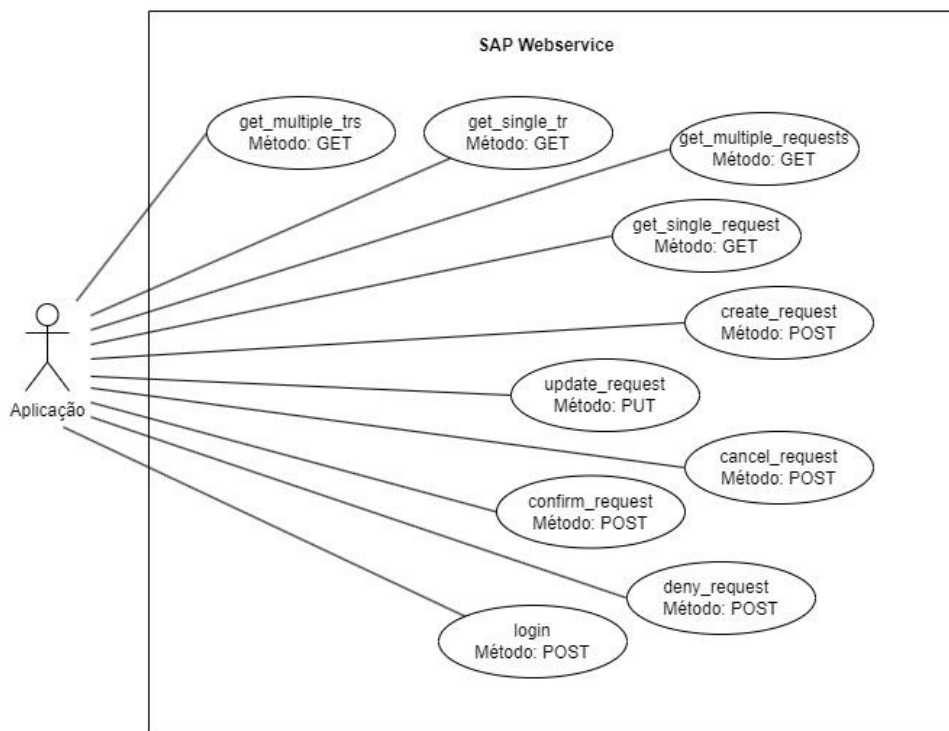


Figura 5: Diagrama de Casos de Uso - Interação entre Aplicação e API

4.5. Diagrama de Atividades

O diagrama de atividade permite ilustrar com clareza a sequência e a lógica das atividades de um sistema.

Na Figura 6, é possível ver o fluxo completo da aplicação com os seus diferentes intervenientes. Este começa pelo utilizador ao efetuar o login inserindo o nome e a respetiva password. Por sua vez, a aplicação e a API validam se os dados introduzidos se encontram corretos, senão volta à etapa inicial.

De seguida, o utilizador navega para umas das páginas onde são disponibilizadas as ordens de transporte e seleciona as ordens de transporte que pretende transportar. Com as ordens selecionadas o mesmo cria o pedido de transporte preenchendo os campos obrigatórios, sendo estes o tema, a razão e as ordens de transporte que, conseqüentemente, serão validados pela aplicação. Caso a aplicação os valide, os mesmos dados introduzidos serão validados pela API, que no caso desta os validar resultará no envio de um email para o utilizador e para a equipa de TM.

Com o pedido já criado, a equipa de TM pode então começar a validar o pedido. Após um determinado tempo de análise esta enviará o veredicto da análise.

Ao receber o veredicto, a API verifica-o e conforme o mesmo altera o status para aceite ou rejeitado. Além disso, caso o pedido de transporte seja aceite serão liberadas e transportadas as ordens de transporte.

Por fim, enviar-se-á um email com o veredicto final para todos os intervenientes.

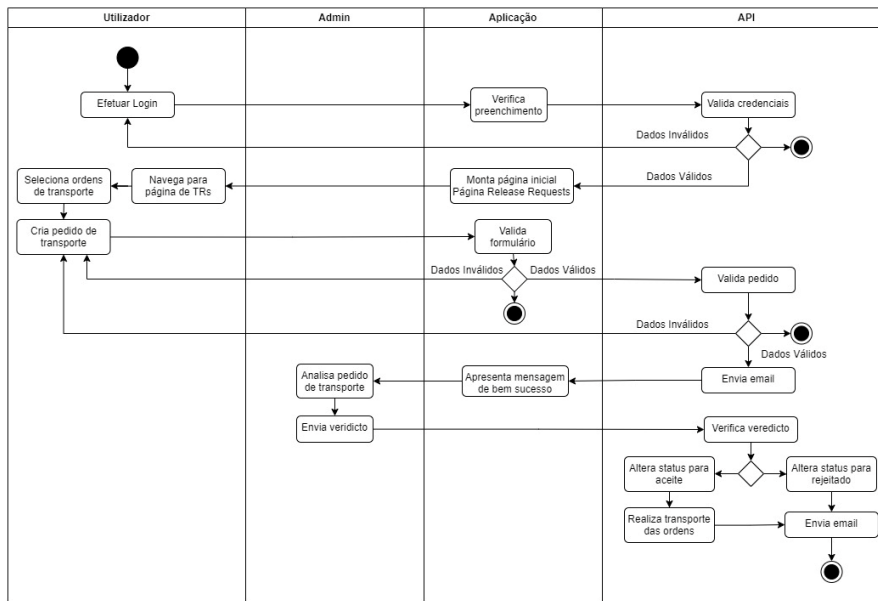


Figura 6: Diagrama de Atividades

4.6. Diagrama de Classes

Na Figura 7, encontra-se ilustrado o diagrama de classes referente ao desenvolvimento. Este tipo de diagrama é importante uma vez que exhibe claramente a estrutura do sistema, modelando as suas classes, atributos, operações e relações entre objetos.

A classe Utilizador representa o utilizador da aplicação podendo este visualizar ordens de transporte, mas também criar pedidos de transporte. Além disso, com os pedidos já criados este pode também editá-los e cancelá-los.

A classe Admin representa a equipa de TM sendo esta proveniente da herança da classe Utilizador. Adicionalmente às operações fornecidas pela classe Utilizador, esta oferece a possibilidade de aceitar e rejeitar pedidos de transporte.

A classe Release Request representa os pedidos de transporte criados pelo utilizador, permitindo a este saber o tema e a razão pela qual o mesmo foi criado.

A classe Transport Request representa as ordens de transporte que o utilizador pode consultar e criar pedidos de transporte sobre.

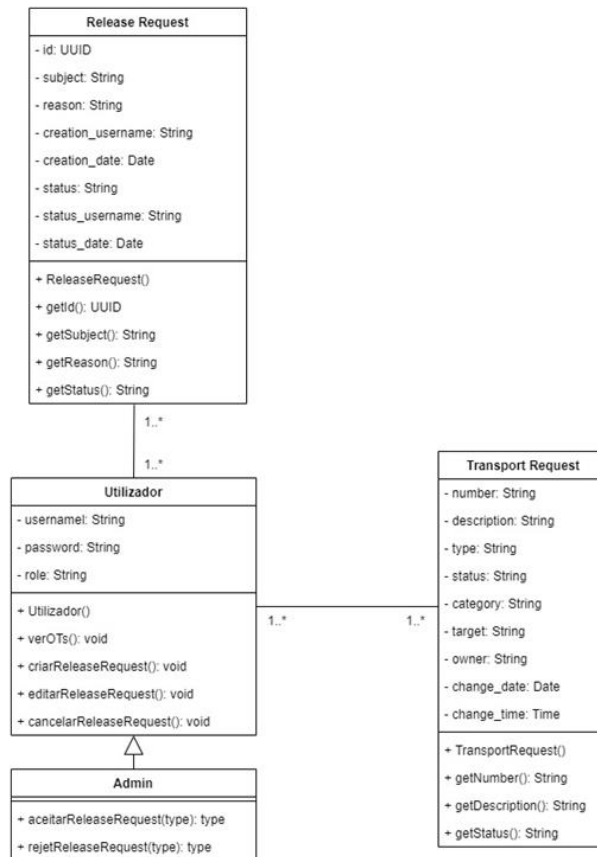


Figura 7: Diagrama de Classes

4.7. Modelação de dados

A modelação de dados é uma etapa importante no desenvolvimento de qualquer aplicação visto que visa definir a estrutura dos dados de uma base de dados. Um modelo de dados bem definido ajuda na criação de uma base de dados simplificada que elimina redundâncias e reduz os requisitos de armazenamento.

Na Figura 8 é possível verificar no modelo as tabelas usadas na base dados. As tabelas ZTRELEASE_REQK, ZTRELEASE_REQP e a visão ZV_E070_E07T foram criadas especificamente para esta aplicação. As ZTRELEASE_REQK e ZTRELEASE_REQP funcionam como uma relação de pai e filho onde a primeira conterá as informações de cabeçalho do pedido de transporte, enquanto a segunda terá a informação dos itens, neste caso das ordens de transporte do pedido. A visão ZV_E070_E07T funciona como uma tabela lógica dinâmica para a obtenção de dados relativos com as ordens de transporte

que contém apenas os campos necessários das tabelas E070, E07T e E070C, todas estas relacionadas.

As tabelas `USR02` e `AGR_USERS` são tabelas standard que contém as informações relativas dos utilizadores disponíveis no sistema. A primeira será usada para verificar as credenciais dos utilizadores enquanto a segunda contém os roles, isto é, as autorizações do utilizador.

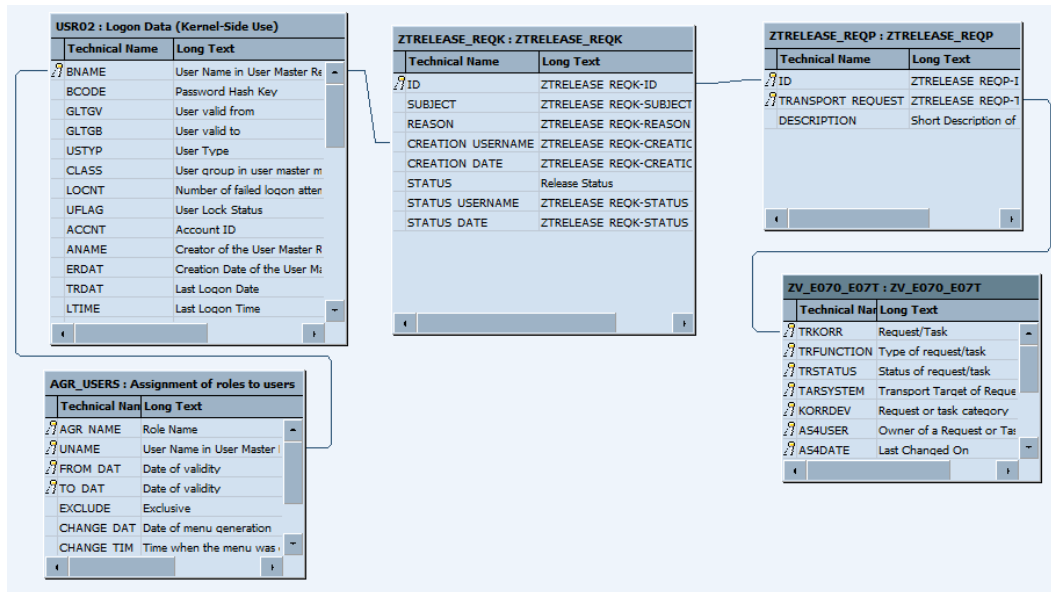


Figura 8: Modelo de Dados

4.8. Mockups

As *mockups* são uma maneira de representar visualmente o design de uma aplicação transmitindo assim a aparência e a estrutura do projeto, possibilitando a compreensão do projeto final.

Desta forma, apresentar-se-á, nesta secção, as *mockups* referentes às páginas da aplicação desenvolvidas, sendo estas, a página de Login, All Transport Requests, My Transport Requests, Transport Request Details, Create Release Request, Release Requests e Release Request Details.

Na página de Login, o utilizador poderá inserir os seus dados dos ambientes SAP de maneira a entrar na aplicação.

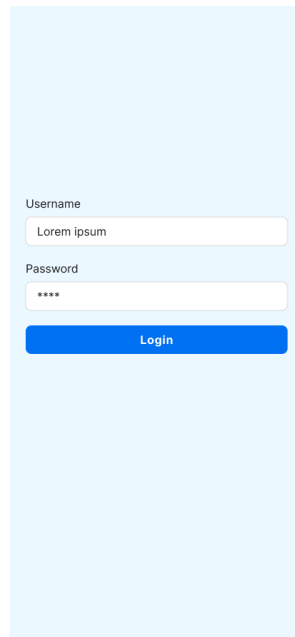


Figura 9: Mockup - Página de Login

Nas páginas All Transport Requests e My Transport Requests são ambas duas páginas que irão apresentar a listagem de ordens de transporte, daí o design igual, porém na primeira exibir-se-á todas as ordens existentes enquanto a segunda apenas do utilizador. De maneira a ilustrar os diferentes formatos das páginas numa será apresentado quando o modo de seleção está desativo e outro quando está ativo.

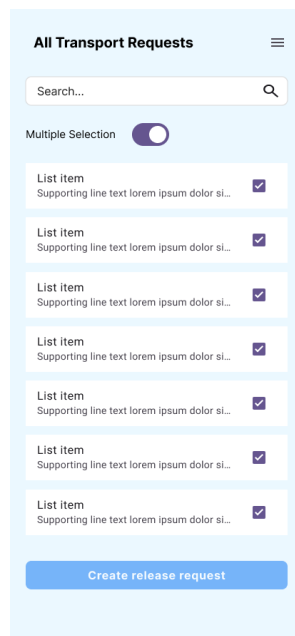


Figura 10: Mockup - Página All Transport Requests

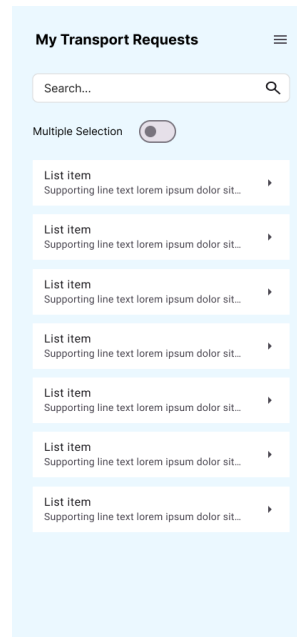


Figura 11: Mockup - Página My Transport Requests

A página Transport Request Details, como o nome sugere, serve para apresentar os detalhes da ordem, além de ser possível a criação de pedidos de transporte.

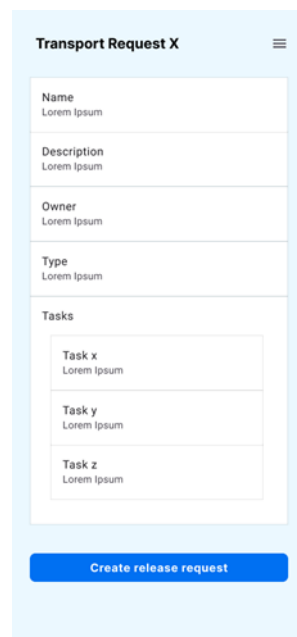


Figura 12: Mockup - Página Transport Request Details

A página Create Release Request serve para a criação dos pedidos de transporte de ordens. Nesta o utilizador deverá preencher os campos obrigatórios para que a criação seja possível.

Create Release Request

Subject
Lorem ipsum

Reason
Lorem ipsum

Transport Requests

TR 1
Lorem ipsum

TR 2
Lorem ipsum

TR 3
Lorem ipsum

TR 4
Lorem ipsum

Create release request

Figura 13: Mockup - Página Create Release Request

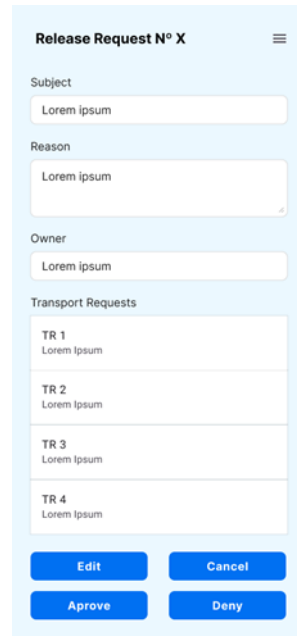
Na página Release Request apresentar-se-á uma listagem com todos os pedidos de transporte do utilizador que ainda estão abertos.

-
- Release Requests**
- List item
Supporting line text lorem ipsum dolor sit... ▶
- List item
Supporting line text lorem ipsum dolor sit... ▶
- List item
Supporting line text lorem ipsum dolor sit... ▶
- List item
Supporting line text lorem ipsum dolor sit... ▶
- List item
Supporting line text lorem ipsum dolor sit... ▶
- List item
Supporting line text lorem ipsum dolor sit... ▶
- List item
Supporting line text lorem ipsum dolor sit... ▶

Figura 14: Mockup - Página Release Requests

Por sua vez, na página Release Request Details estarão disponíveis todos detalhes do pedido. Além disso, dependendo do role do utilizador a aceder estarão disponíveis diferentes botões.

Se for um utilizador comum a aceder apenas estarão disponíveis os botões de editar e cancelar o pedido. Contudo caso o utilizar tenha o role de admin, isto é, um integrante da equipa de TM terá disponível os botões de confirmar e negar o pedido de transporte.



Release Request Nº X

Subject
Lorem ipsum

Reason
Lorem ipsum

Owner
Lorem ipsum

Transport Requests

TR 1
Lorem Ipsum

TR 2
Lorem Ipsum

TR 3
Lorem Ipsum

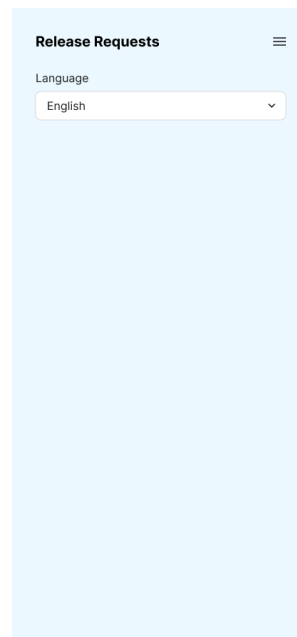
TR 4
Lorem Ipsum

Edit Cancel

Approve Deny

Figura 15: Mockup - Página Release Request Details

Na página Settings, será possível alternar entre os idiomas disponíveis na aplicação, inglês e português.



Release Requests

Language
English

Figura 16: Mockup - Página Settings

5. Desenvolvimento

Neste capítulo será abordado todos os passos realizados para o desenvolvimento da app. Inicialmente, serão apresentados a criação dos objetos base para o desenvolvimento da API em SAP. De seguida, será apresentado o desenvolvimento da API e seus *endpoints* explicando cada método HTTP usado. Após o desenvolvimento da API, será abordado a integração da mesma realizada no SAP Integration Suite. Por fim, será relatado o desenvolvimento da aplicação em SAP Build Apps.

5.1. Criação de objetos

Nesta secção abordar-se-á o desenvolvimento dos objetos criados para auxiliar na REST API e no SAP Build Apps.

5.1.1. View ZV_E070_E07T

A *view* ZV_E070_E07T foi criada com o simples objetivo de facilitar a coleta das ordens de transporte.

As *views* são objetos utilizados para visualizar/obter os registos de várias tabelas num único local. Uma *view* é semelhante a uma tabela presente na base de dados, contudo esta

não contém quaisquer registos, em vez disso, atua como uma tabela virtual onde os registos apresentados derivam dos registos das tabelas definidas.

As tabelas definidas na *view* são a tabela E070, contém todos os dados de cabeçalho das ordens de transporte e *tasks*, a E07T, contém as descrições de todas as OTs e *tasks* e os idiomas em que as mesmas se encontram, e a tabela E070C, que contém os sistemas de origem e destino. De maneira a relacionar as tabelas referidas, foi usado o campo TRKORR visto ser a única chave presente em todas.

Entre os vários campos presentes nas tabelas os únicos selecionados foram os campos presentes na Tabela 3.

Tabela 3: Campos da view ZV_E070_E07T

Campo	Tabela de origem	Descrição
TRKORR	E070	Número da Ordem/Task
TRFUNCTION	E070	Tipo da Ordem/Task
TRSTATUS	E070	Status da Ordem/Task
TARSYSTEM	E070	Sistema de destino
KORRDEV	E070	Categoria da Ordem/Task
AS4USER	E070	Dono da Ordem/Task
AS4DATE	E070	Data da última alteração
AS4TIME	E070	Hora da última alteração
STRKORR	E070	Ordem principal
LANGU	E07T	Idioma da descrição
AS4TEXT	E07T	Descrição
CLIENT	E070C	Mandante do sistema de origem
TARCLIENT	E070C	Mandante do sistema de destino

5.1.2. Tabelas ZTRELEASE_REQK e ZTRELEASE_REQP

Em SAP, os dados são frequentemente organizados em tabelas, e essas tabelas podem ser amplamente categorizadas em dois tipos: tabelas de cabeçalho e tabelas de itens.

Por um lado, as tabelas de cabeçalho são usadas para armazenar dados gerais que se aplicam ao documento. Dentro desses dados estão incluídas informações tais como o número do documento, a data de criação, a pessoa que o criou, entre outras informações.

Por outro lado, as tabelas de itens são usadas para armazenar dados específicos relacionados a itens individuais dentro do documento. Nesses dados estão incluídas informações tais como o número do item, a quantidade do item, o preço do item, entre outras informações.

Juntas, as tabelas de cabeçalho e itens fornecem uma visão abrangente de um documento em SAP. A tabela de cabeçalho fornece uma visão geral de alto nível, enquanto a tabela de itens fornece informações detalhadas sobre cada item pertencente ao documento. Com este tipo de dinâmica, a gestão e seleção dos dados nos sistemas SAP torna-se mais eficientes.

Assim, de maneira a manter este tipo de dinâmica foram criadas as tabelas ZTRELEASE_REQK e ZTRELEASE_REQP, para a gestão dos pedidos de transporte criados pelos utilizadores. A tabela ZTRELEASE_REQK servirá como a tabela de cabeçalho contendo os dados gerais para do pedido e a ZTRELEASE_REQP servirá como tabela de itens contendo as ordens de transporte do pedido.

A tabela ZTRELEASE_REQK foi definida tal como está apresentada na Tabela 4.

Tabela 4: Campos da tabela ZTRELEASE_REQK

Campos	Elemento de Dados	Categoria dos Dados	Descrição
MANDT	MANDT	CLNT (3)	Mandante
ID	Z_TAC_ID	NUMC (10)	Número do pedido
SUBJECT	Z_TAC_SUBJECT	CHAR (255)	Assunto
REASON	Z_TAC_REASON	CHAR (255)	Razão
CREATION_USERNAME	Z_TAC_USERNAME	CHAR (12)	Utilizador de criação
CREATION_DATE	Z_TAC_CREATION_DATE	DATS (8)	Data de criação
STATUS	Z_TAC_REJECTED	CHAR (1)	Status
STATUS_USERNAME	Z_TAC_USERNAME	CHAR (12)	Utilizador de alteração
STATUS_DATE	Z_TAC_CREATION_DATE	DATS (8)	Data de alteração

A tabela ZTRELEASE_REQP foi definida tal como está apresentada na Tabela 5.

Tabela 5: Campos da tabela ZTRELEASE_REQP

Campos	Elemento de Dados	Categoria dos Dados	Descrição
MANDT	MANDT	CLNT (3)	Mandante
ID	Z_TAC_ID	NUMC (10)	Número do pedido
TRANSPORT_REQUEST	Z_TAC_TRANSPORT_REQUEST	CHAR (20)	Ordem de transporte
DESCRIPTION	AS4TEXT	CHAR (60)	Descrição da ordem de transporte

5.2. Desenvolvimento da REST API

Nesta secção abordar-se-á o desenvolvimento da REST API usada na conexão do SAP ERP com o SAP Build Apps.

5.2.1. Criação da ZCL_TRANSPORT_REQUEST

Inicialmente, para começo de desenvolvimento da REST API foi necessário criar a classe global ZCL_TRANSPORT_REQUEST, via transação SE24.

Na classe foi definida a interface standard IF_HTTP_EXTENSION. Esta interface garante que o *Internet Communication Framework* (ICF), camada entre o *Internet Communication Manager* (ICM), que recebe e envia pedidos de HTTP, e o SAP Web Application Server, que funciona tanto como *webserver* e *webclient*, interaja de forma coerente com os *handlers* dos pedidos HTTP. Então, visto que a classe criada tem como principal intuito servir de *handler* para os de pedidos HTTP é de caráter obrigatório a definição desta interface.

O único método pertencente a esta interface é o HANDLE_REQUEST. Este método é chamado pelo gestor do ICF quando é realizado um pedido. De maneira que o *handler* possa aceder aos dados, tanto de um pedido realizado como aos da resposta a ser enviada, é necessário passar uma referência à interface IF_HTTP_SERVER, visto que esta tem referência a ambos.

Assim, de antemão foi definido que este método serviria como *handler* principal da API e que, dependendo do *path* chamado, redirecionaria para um *handler* específico. Esse *handler* específico seria um método de uma classe própria, que seria definida localmente na classe ZCL_TRANSPORT_REQUEST.

Optou-se pela definição local das classes após considerar-se que a definição global das mesmas exporia os seus respetivos códigos, visto que as mesmas poderiam vir a ser chamadas em outros objetos. Assim ao optar pela definição local, as classes desenvolvidas apenas poderiam ser chamadas na classe onde se encontram, neste caso a classe ZCL_TRANSPORT_REQUEST.

Para auxiliar o método `HANDLE_REQUEST` foi criado o método `GET_REST`. Neste obter-se-ia o *path* do pedido realizado e, dependendo do mesmo, criar-se-ia um objeto da respectiva classe local a ser chamada.

Assim e tal como foi referido anteriormente, com o objeto em mãos realizar-se-ia, no *handler* principal, a chamada ao respetivo *handler* da respetiva classe.

5.2.2. Desenvolvimento da interface LIF_REST

Começou-se a desenvolver a interface `LIF_REST` nela foram definidos várias tipificações que serão usados pelo resto do código desenvolvido.

Inicialmente começou-se por desenvolver tipificações para serem utilizados como estruturas:

Tabela 6: Tipificações da interface `LIF_REST`

Tipificação	Descrição
<code>TY_RESPONSE</code>	Usada como body de resposta para envio de mensagem
<code>TY_RESULT</code>	Usada como como body de resposta para envio de dados

Além disso, foram definidos vários métodos que vão ser usados pelas várias implementações em que a implementar-lhe-ão. Existem dois tipos de classes de métodos que foram definidos, os métodos de *request* usados para as chamadas às API e os métodos auxiliares para o processamento dos pedidos realizados à API .

Métodos de Request:

Tabela 7: Métodos HTTP definidos na interface `LIF_REST`

Métodos	Descrição
<code>GET</code>	Processa os pedidos <code>GET</code>
<code>HEAD</code>	Processa os pedidos <code>HEAD</code>
<code>POST</code>	Processa os pedidos <code>POST</code>
<code>DELETE</code>	Processa os pedidos <code>DELETE</code>
<code>OPTIONS</code>	Processa os pedidos <code>OPTIONS</code>

Métodos Auxiliares:

Tabela 8: Métodos auxiliares na interface LIF_REST

Métodos	Descrição
HANDLE_REQUEST	Funciona como Handler dos pedidos realizados a API redirecionando para o respetivo método de request
RESULT_TO_CDATA	Usado para converter os dados de resposta, estejam eles em qualquer tipo de dados ou formato, em formato JSON
PREPARE_RESPONSE	Prepara os dados de resposta a serem enviados na resposta
SET_RESPONSE	Define a resposta a ser enviada ao pedido realizado
GET_REQUEST_BODY	Obtem o body do request realizado

5.2.3. Desenvolvimento da classe LCL_REST

Com a interface LIF_REST em mãos, fez-se a implementação da classe local LCL_REST com referência à mesma.

Esta classe foi criada com intuito de servir de superclasse para as restantes classes locais a serem desenvolvidas. Optou-se por prosseguir por este caminho devido a vários dos métodos usados entre as múltiplas classes serem iguais, evitando assim a repetição de código, além de facilitar a implementação das mesmas.

Na definição da classe, os métodos auxiliares provenientes da interface foram definidos em duas categorias diferentes, métodos finais ou métodos abstratos. De todos os métodos auxiliares apenas um foi definido como abstrato, tendo este sido o SET_RESPONSE. A razão para tal decisão deve-se ao facto de a resposta a ser enviada poder ser diferente dependendo da classe onde se encontra.

Para além dos diferentes tipos de objetos provenientes da interface, definiu-se, como *protected*, o objeto GO_SERVER fazendo referência à interface IF_HTTP_SERVER, para que este esteja disponível globalmente na classe. Tal como referido na secção anterior, esta interface tem referências tanto aos dados provenientes do pedido realizado como aos da resposta a ser enviada.

Após a definição da classe seguiu-se para a implementação dos métodos.

Construtor

Neste método apenas é atribuído o objeto importado ao objeto GO_SERVER.

Método handle_request

Neste método será realizado o tratamento de cada pedido recebido.

Para tal, inicialmente, será obtido do objeto go_server, o método HTTP chamado. Com o método obtido, será chamado o método correspondente na classe.

Após a execução da lógica do método HTTP correspondente, utilizar-se-ia do método set_response para definir a resposta a ser enviada.

Método result_to_cdata

Neste método serão convertidos os dados de resposta, estejam eles em qualquer tipo de formato, em JSON com o uso do método serialize da classe /ui2/cl_json.

Para este método funcionar é necessário importar para o mesmo a resposta a ser enviada, para o parâmetro data, e a *naming convention*, para o parâmetro pretty_name. No parâmetro pretty_name determinou-se que apenas usar-se-ia *camel case*.

Método prepare_response

O objetivo deste método é preparar a resposta a ser enviada. Para isso será retornado o valor do código HTTP, a mensagem referente ao código HTTP e a resposta a ser enviada, após a mesma ser obtida do método result_to_cdata.

Método get_request_body

Neste método obter-se-á o body do pedido realizado.

Primeiramente, com o método get_cdata retornar-se-á o body do pedido realizado, contudo, o mesmo encontrar-se-á em formato JSON. Então, para que este converta o JSON no formato de uma estrutura ou tabela SAP é necessário utilizar o método deserialize da classe /ui2/cl_json.

Métodos HTTP

Nos métodos HTTP foi implementada uma lógica standard igual para todos eles. Nesta lógica apenas retornar-se-á o código HTTP 405, tal como a mensagem correspondente, caso o método não seja redefinido nas subclasses criadas.

5.2.4. Desenvolvimento da classe local LCL_REST_AUTH

A classe LCL_REST_AUTH, tal como o nome sugere, terá como finalidade autenticar as credenciais dos utilizadores que utilizam a APP.

Começou-se então por definir a classe LCL_REST_AUTH como herdeira da superclasse LCL_REST.

De seguida, dos métodos herdados da superclasse redefiniu-se os métodos set_response, options e post.

Além disso, na secção privada, foi definido o método login, que verificará as credenciais enviadas, e a tipificação local ty_login_body que será utilizada para definir a estrutura que receberá o body do pedido.

Após a definição da classe seguiu-se a implementação dos métodos.

Método login

Tal como foi referido anteriormente o método login tem como propósito verificar as credenciais inseridas pelos utilizadores ao tentar aceder na APP.

No método iniciou-se, através do método get_request_body, pela obtenção do body enviado com as credenciais do utilizador.

Após a obtenção das mesmas, estas são confrontadas, por meio do módulo de função standard SUSR_USER_PASSWORD_PUT, com as credenciais dos utilizadores existentes na máquina.

Caso as credenciais correspondam, verificar-se-á o role do utilizador, podendo estes variar entre o role Z_PRJ_TRANSPORT_REQUEST_ADMIN (Admin) e o Z_PRJ_TRANSPORT_REQUEST_USER (Utilizador geral). Para obter o *role* do utilizador é necessário seleccionar a tabela standard AGR_USERS com o nome do mesmo.

Finalmente, após a análise do role, será preparada a resposta fazendo uso do método `prepare_response`. Nela estarão presentes apenas o *username* e o *role* do utilizador.

Método post

Por sua vez, o método `post` tem como intuito redirecionar as diferentes chamadas realizadas dependendo do *path* chamado.

Inicialmente, pretendia-se que o mesmo verifica-se três possíveis *paths*, sendo estes, o de login, logout e de signup. Contudo após verificar-se os requisitos da aplicação concluiu-se que apenas seria utilizado o método de login, deixando assim os outros de lado.

Apesar disso, a lógica inicial manteve-se caso futuramente esta seja utilizada.

Método options

O método `options` é utilizado para descrever as diversas opções de comunicação disponíveis no serviço. Os browsers enviam um pedido `OPTIONS` para descobrir os métodos HTTP suportados e outras opções suportadas antes de enviarem o pedido propriamente dito. Além disso, os pedidos `OPTIONS` permitem aos clientes obter parâmetros e requisitos para recursos específicos e capacidades do servidor sem atuar sobre o recurso ou solicitar o recurso.

Nesta implementação foi incluído na resposta o *header* `Access-Control-Allow-Methods` com os métodos HTTP disponibilizados, neste caso apenas o `POST` e o `OPTIONS`, o *header* `Access-Control-Allow-Origin`, o *header* mais importante para a especificação do CORS (Cross-Origin Resource Sharing), e o *header* `Access-Control-Max-Age`, que define por quanto tempo os resultados obtidos na chamada dos métodos HTTP é mantida.

Método set_response

O método `set_reponse` tem como finalidade definir a resposta a ser enviada.

Um pouco à semelhança do método `options`, neste método definir-se-ão os *headers* `Access-Control-Allow-Methods` e `Access-Control-Allow-Origin`.

Além disso, foi definido o *header* `Content_Type`, que indica o formato utilizado na resposta, como `'application/json'`.

Por fim, foram definidos o *body* de resposta e o código HTTP tal como a mensagem correspondente.

5.2.5. Desenvolvimento da classe local LCL_REST_TR

A classe LCL_REST_TR destina-se a disponibilizar aos utilizadores as ordens de transporte presentes no sistema.

Inicialmente, começou-se então por definir a classe LCL_REST_TR como herdeira da superclasse LCL_REST.

De seguida, à semelhança da classe LCL_REST_AUTH, foram redefinidos os métodos `set_response` e `options`. Nestes métodos o código desenvolvido foi semelhante à anteriormente referida.

Visto que o objetivo da classe é disponibilizar as OT presentes no sistema, foi necessário também redefinir o método `GET`.

Na secção privada, foram definidos os métodos `get_multiple_trs` e `get_single_tr` que têm como finalidade obter todas as ordens de transporte e obter uma específica dependendo do número selecionado, respetivamente. Além disso, foram definidas várias tipificações como se pode verificar pela Tabela 9.

Tabela 9: Tipificações definidas na classe local LCL_REST_TR

Tipificação	Tipo de Uso	Descrição
TY_COLLECTION	Estruturas	Usada na definição da tipificação TY_T_COLLECTION e no método <code>get_single_tr</code> para a obtenção da OT selecionada presente no <i>body</i> do pedido
TY_T_COLLECTION	Tabelas	Usada no método <code>get_multiple_trs</code> para a obtenção de múltiplas OTs
TY_TASKS	Estruturas	Usada na definição da tipificação TY_T_TASKS de uma task
TY_T_TASKS	Tabelas	Usada no método <code>get_multiple_trs</code> para a obtenção de todas as tasks de uma OT
TY_SINGLE	Estruturas	Usada no método <code>get_single_tr</code> para a obtenção de uma OT específica

Por fim, após a conclusão da definição da classe seguiu-se a implementação dos métodos já mencionados.

Método get

O método `get` tem como intuito redirecionar as diferentes chamadas realizadas dependendo se é realizada alguma *query*.

Caso seja enviado o parâmetro `number` no pedido realizado, realizar-se-á a chamada do método `get_single_tr` de maneira a obter a ordem específica.

Assim, caso acontece o contrário, será executado o método `get_multiple_trs`.

Método get_single_tr

Este método tem como finalidade selecionar as informações duma OT específica dependendo do número enviado no campo de importação `iv_request`.

À semelhança do que acontece no método `get_multiple_trs`, aqui também é realizado um *select* à *view* `ZV_E070_E07T`, porém desta vez com o objetivo de obter apenas as informações da ordem especificada.

Caso o *select* anterior ocorra sem problemas é realizado uma nova seleção à `ZV_E070_E07T`, contudo, desta vez com o objetivo de selecionar todas as *tasks* da ordem. Para tal é necessário utilizar o campo `STRKORR`, campo que contém a ordem `pai`, atribuindo-lhe o valor da ordem.

Por fim, com o método `prepare_response`, prepare-se a resposta a ser enviada.

Método get_multiple_trs

O método `get_multiple_trs`, tal como já referido na introdução desta secção, terá o objetivo de disponibilizar ao utilizador todas as ordens de transporte presentes no servidor.

Para alcançar o propósito mencionado foi realizado um `SELECT` à *view* `ZV_E070_E07T`.

Assim, dependendo do sucesso da seleção realizada, preparar-se-á a resposta a ser enviada através do método `prepare_response`.

5.2.6. Desenvolvimento da classe LCL_REST_REQUEST

A classe LCL_REST_REQUEST terá finalidades um pouco distintas dependendo do role do utilizador.

Caso seja um utilizador comum a classe deverá permitir a este criar pedidos para o transporte das ordens selecionadas. Além disso, deverá também permitir visualizar os pedidos criados por si e eliminar os que já não deseja.

Por outro lado, a caso seja um integrante da equipa de TM, a classe deverá permitir visualizar todos os pedidos criados pelos utilizadores, além de permitir que este os atualize seja validando-os ou rejeitando-os.

Assim, começou-se por defini-la como herdeira da superclasse

De seguida, à semelhança das restantes classes, foram redefinidos os métodos set_response e options. Nestes métodos o código desenvolvido foi semelhante ao que já tinha sido anteriormente referido, com a exceção do método options, em que foram adicionados apenas mais métodos HTTP no header Access-Control-Allow-Methods.

Além dos métodos já referidos, foi necessário também redefinir os métodos get, post, put e delete, visto que a classe deverá disponibilizar essas funcionalidades aos utilizadores.

Na secção privada, foram definidos os métodos get_multiple_requests e get_single_request que têm como finalidade, respetivamente, obter todas os pedidos de transporte e obter um pedido específico dependendo do número selecionado. Nesta também foi definido o método send_email que terá como intuito realizar o envio de email.

Além disso, foram também definidas várias tipificações como se pode comprovar na Tabela 10.

Tabela 10: Tipificações da classe local LCL_REST_REQUEST

Tipificação	Tipo de Uso	Descrição
TY_REQUEST_HEADER	Estruturas	Usada na definição da tipificação TY_T_REQUEST_HEADER e no método get_single_request para a obtenção de um pedido em específico
TY_T_REQUEST_HEADER	Tabelas	Usada no método get_multiple_requests para a obtenção de múltiplas requests
TY_REQUEST_ITEM	Estruturas	Usada na definição da tipificação TY_T_REQUEST_ITEM

TY_T_REQUEST_ITEM	Tabelas	Usada no método <code>get_multiple_requests</code> para a obtenção de todas as ordens de um pedido
TY_REQUEST_ITEM_DATA	Estruturas	Usada na definição da tipificação <code>TY_T_REQUEST_ITEM_DATA</code>
TY_T_REQUEST_ITEM_DATA	Tabelas	Usada na definição da tipificação <code>TY_T_REQUEST_DATA</code>
TY_REQUEST_DATA	Estruturas	Usada na definição da tipificação <code>TY_T_REQUEST_DATA</code>
TY_T_REQUEST_DATA	Tabelas	Usado nos bodies de envio e resposta

Por último, após a conclusão da definição da classe seguiu-se a implementação dos métodos já mencionados.

Método `get`

O método `get` tem como finalidade redirecionar as diferentes chamadas realizadas dependendo se é realizada alguma *query*.

Por um lado, se for enviado o parâmetro `number` no pedido realizado, realizar-se-á a chamada do método `get_single_request` de maneira a obter a ordem específica.

Caso acontece o contrário, será executado o método `get_multiple_requests`.

Método `get_multiple_requests`

O método `get_multiple_requests`, tal como já referido na introdução desta secção, terá o propósito de disponibilizar ao utilizador todos os pedidos de liberação e transporte criados por si, caso seja um utilizador genérico ou todos os pedidos existentes ainda por validar, caso seja um membro da equipa de *Transport Management*.

Iniciamente, verifica-se se foi enviado o parâmetro `username` no pedido realizado. Se o mesmo se encontrar preenchido, significa que o utilizador que pretende obter os pedidos é um utilizador genérico e, por essa razão, serão apenas seleccionados os pedidos criados por si mesmo. Contudo, se o parâmetro estiver vazio, será realizado uma seleção de todos os pedidos existentes ainda não validados.

De seguida, caso a seleção anterior seja bem sucedida, realizar-se-á uma seleção à tabela `ZTRELEASE_REQP`, com o objetivo de seleccionar todas as OT seleccionadas no pedido.

Por fim, prepare-se a resposta a ser enviada através do método `prepare_response`.

Método `get_single_request`

Este método tem como finalidade selecionar as informações de um pedido específico dependendo do número enviado no campo de importação `iv_request`.

À semelhança do que acontece no método `get_multiple_requests`, aqui também é realizado um *select* à tabela `ZTRELEASE_REQK`, porém desta vez com o objetivo de obter apenas as informações do pedido especificado.

Caso a seleção anterior ocorra sem problemas é realizado uma nova seleção porém desta vez à tabela `ZTRELEASE_REQP`, com o objetivo de selecionar todas as OT selecionadas no pedido.

No final, com o método `prepare_response`, prepare-se a resposta a ser enviada.

Método `send_email`

Neste método, tal como o nome sugere, pretende-se enviar o envio de emails.

Inicialmente, através do método `create_persistent` da classe `standard cl_bc`, criou-se o pedido de envio de email, sendo este retornado como instância da classe.

De seguida, com a instância `lo_request` criada definiu-se o assunto do pedido através do método `set_message_subject`, além de defini-lo como de saída automática com o método `set_send_immediately`. Ambos os métodos mencionados apenas se encontram disponíveis para instanciações da classe.

Com o método `create_internet_address` da classe `cl_cam_address_bcs`, foram criados o remetente e o destinatário do email. Este método retorna uma instância, neste caso, `lo_sender` e `lo_recipient`, respetivamente, que em conjunto com o método `set_sender` e `add_recipient` podem ser definidos na instância `lo_request`, como remetente e destinatário.

Após a definição do remetente e destinatário, é importante criar o body do email, por outras palavras, o documento a ser enviado. Para tal será necessário utilizar o método `create_document` da classe `cl_document_bcs`, onde é necessário importar o tipo de documento, o assunto do email e o texto a ser enviado. Consequentemente, este retornará uma instância do documento criado, sendo esta definida como `lo_document`. Adicionalmente para ajudar na construção do body foram criados quatro métodos adicionais para definir o body dependendo do tipo de ação a ser tomada, entre as quais se encontram a criação, cancelamento, rejeição e confirmação dos pedidos de transporte.

Com a instância em mãos, a mesma será definida como o body a ser enviado na `lo_request` através do método `set_document`.

Por fim, com o método `send` da `lo_request`, realiza-se o envio-o do email.

Método post

O método `post` tem como intuito possibilitar aos utilizadores criar pedidos de transporte.

Inicialmente, começou-se, com o uso do método `get_request_body`, por obter os dados enviados no body.

De seguida, tendo os dados provenientes do body, preparou-se os dados que seriam inseridos na tabela de cabeçalho `ZTRELEASE_REQK`.

Após a conclusão do passo anterior, realizou-se a tentativa da inserção do registo. Caso o registo seja inserido com sucesso, repetir-se-á os passos já mencionados, porém, desta para a tabela de itens `ZTRELEASE_REQP`.

No fim, com o método `prepare_response`, prepare-se a resposta a ser enviada.

Método put

O método `put` tem como finalidade redirecionar as diferentes chamadas realizadas dependendo da ação definida no parâmetro `action`. Atualmente existem quatro ações definidas com os seus próprios métodos, sendo estas as ações de edição, cancelamento, rejeição e aprovação.

O método de edição fará a atualização das alterações realizadas no pedido de transporte guardando-as nas tabelas `ZTRELEASE_REQK` e `ZTRELEASE_REQP`.

Os métodos de cancelamento e rejeição farão a alteração dos status do pedido conforme o método onde se encontra.

Por sua vez, o método de confirmação lado além de atualizar o status do pedido fará também o processo de liberação e transporte das OTs.

Assim, primeiramente, com as ordens presentes no body inicia-se o processo de liberação, sendo necessário utilizar a BAPI standard `BAPI_CTREQUEST_RELEASE`.

Caso a liberação das ordens seja um sucesso, partimos de imediato para o transporte destas com o uso do módulo de função `TMS_MGR_IMPORT_TR_REQUEST`.

Por último, se os transportes forem bem sucedidos, atualizar-se-á a tabela ZTRELEASE_REQK.

Em cada um destes métodos será enviado um email específico à ação realizada.

5.3. Integração da REST API no SAP Integration Suite

Após a criação da API, testou-se os *endpoints* via *browser* e percebeu-se que a ligação era insegura, resultado de a máquina onde o *webservice* tem origem, ter um certificado próprio, por outras palavras, *self-signed*, não gerado por uma entidade certificadora.

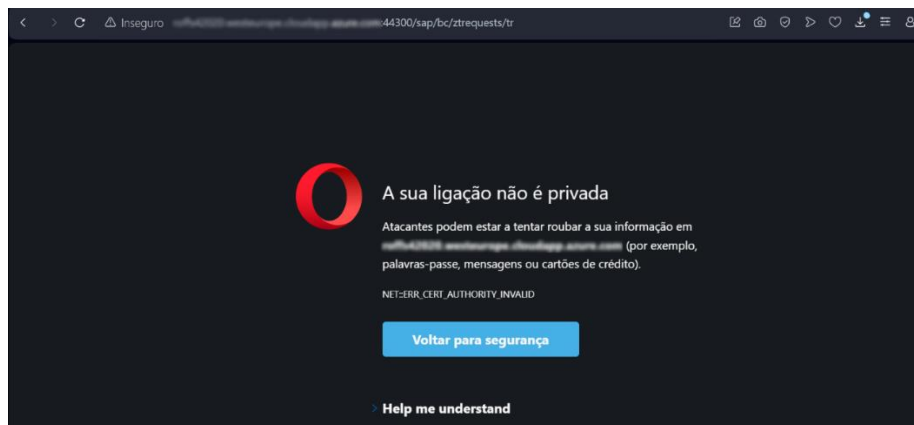


Figura 17: Teste dos endpoints

Apesar do problema poder ser ultrapassado ao testar no navegador, o mesmo afeta o normal funcionamento da aplicação no telemóvel, visto que não é obtida nenhuma resposta do *webservice*, não sendo possível obter de dados.

Com isto, decidiu-se então integrar os *endpoints* via SAP Integration Suite, através de API Proxies. Um API Proxy pode ser definido como uma representação discreta de uma API, visto funcionar como um intermediário entre o *front-end* e o *backend*, resumidamente, funciona como uma interface para um serviço já existente.

Após configurar os dados necessário criados três API Proxies, como é possível verificar na imagem abaixo.

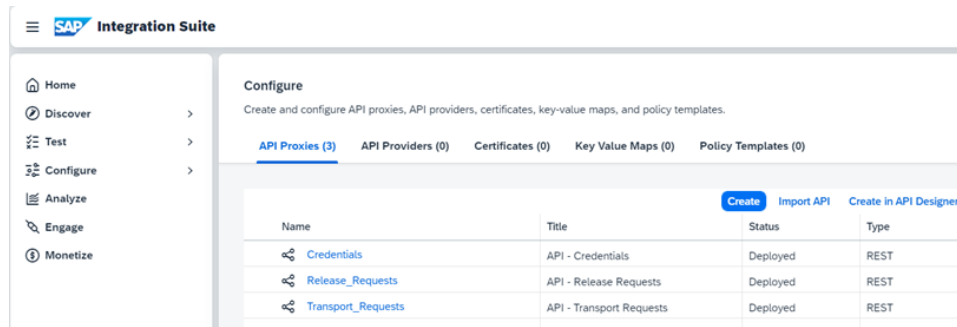


Figura 18: API Proxies

5.4. Criação da aplicação no SAP Build Apps

5.4.1. Integração da REST API

Após a criação da REST API, passou-se à integração desta na aplicação de maneira a disponibilizar todas as informações necessárias para o funcionamento básico da aplicação.

Para criar a integração é necessário preencher a aba base com um nome, descrição e URL da API.

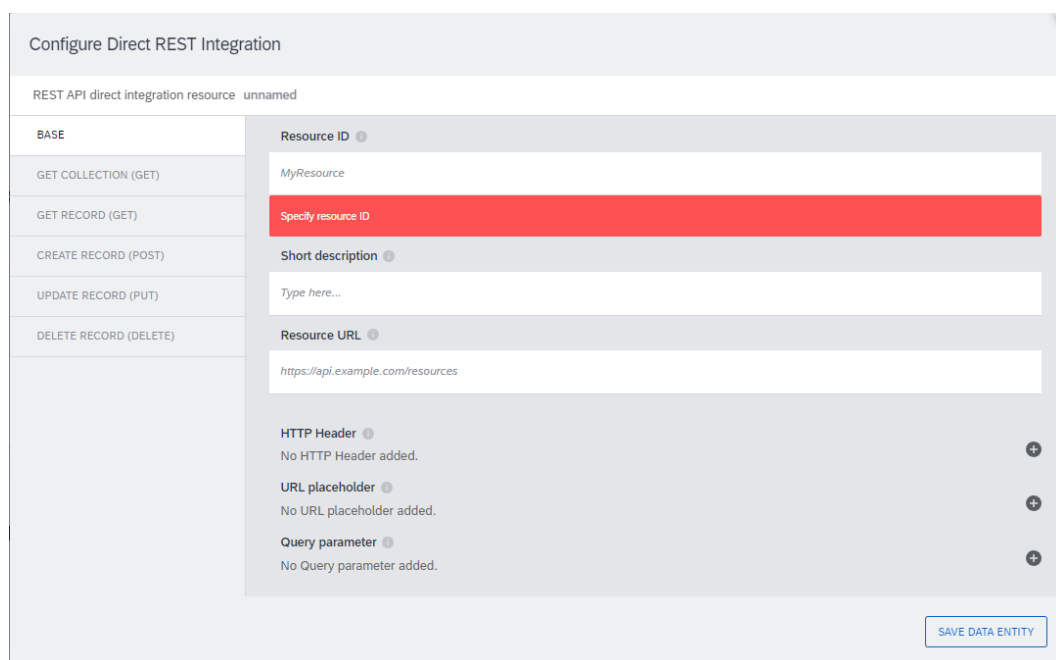


Figura 19: Configuração da REST API na aba Data

Com os dados já preenchidos foram disponibilizadas três integrações, como é possível verificar na imagem abaixo, com base nas classes locais e os métodos já apresentadas.

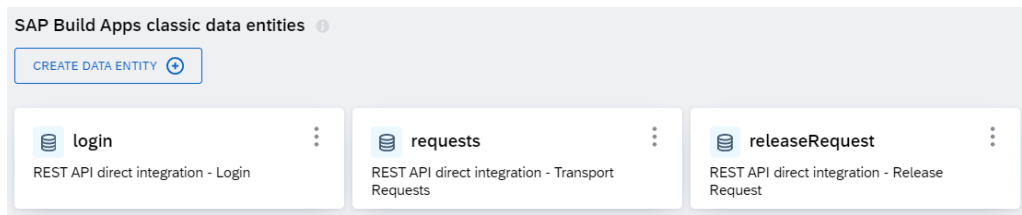


Figura 20: Integrações da REST API

5.4.2. App Variables

No SAP Build Apps existem vários tipos de variáveis, mas uma das mais importantes é são as app variables.

As app variables existem globalmente na aplicação, o que significa que as mesma podem ser acedidas a partir de qualquer página da aplicação.

Criar uma app variable no SAP Build Apps é simples, apenas é preciso navegar para a página variable ao clicar no *toogle* disponível em todas as páginas no canto superior direito. A partir daí, para criar uma variável basta clicar em "Add App Variable".

Então, tendo em conta os passos mencionados anteriormente foram criadas duas app variables, a `currentUser` e a `menuOpen`.

A variável `currentUser`, tal como o nome sugere, é utilizada para guardar o utilizador atual, isto é, o utilizador que efetuou log in. A variável é um objeto com duas propriedades, `role` e `username`.

A variável `menuOpen` é um simples booleano que tem como finalidade abrir ou fechar o menu dependendo do seu valor. Optou-se por definir esta variável como global uma vez que a mesma será utilizada em todas páginas, além de que caso a mesma fosse definida apenas localmente ao entrar nas diferentes páginas seria necessário importar o valor atual.

5.4.3. Página de Login

Na página de login apenas foi criada a page variable `credentials`. Esta variável é um objeto com duas propriedades, `username` e `password`.

Na página foram adicionados dois inputs, uma para o utilizador e outro para a password, e um botão, para a verificação dos dados inseridos.

Ambos os inputs foram associados com a respetiva propriedade da page variable credentials.

No botão foi inserido a seguinte lógica, presente na imagem, que será executada quando surge um clique.

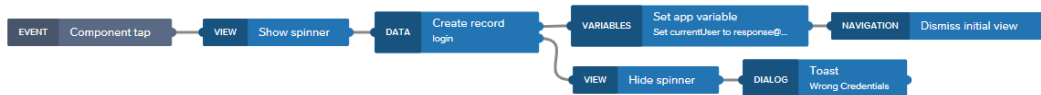


Figura 21: Lógica de Login

Após o clique irá surgir um *spinner* na página que apenas afeta visualmente a aplicação dando ao utilizador a sensação de que o log in está a ser processado.

De seguida, usa-se a flow function Create Record com o intuito de realizar um pedido POST ao data resource de login atribuindo-lhe as propriedades da variável credentials.

Posto isto, o processo tomará dois caminhos distintos dependendo do resultado do Create Record.

Por um lado, se o POST for bem sucedido atribuir-se-á a resposta da chamada à app variable currentUser e, de imediato, através da função “Dismiss initial view”, navegar-se-á para a página inicial da app, à Display Requests.

Por outro, se ocorrer um erro na tentativa de realizar login, o *spinner* inserido inicialmente desaparecerá e, conseqüentemente, será exibido na tela uma mensagem de erro apresentada via uma notificação toast.

5.4.4. Página All Transport Request

Nesta página foram criadas 3 page variables, sendo elas a search, switch e selectedRequests.

A variável search está associada ao input presente, o qual funciona como filtro, e que por consequência será usada na lista apresentada na página.

A variável switch é um booleano que está associado ao componente de *toggle* presente na própria página. Ao contrário do que aparenta esta variável é de extrema importância

uma vez que a mesma é usada para alternar a lista entre modo de seleção única para modo múltiplo de OTs.

A variável `selectedRequests` é uma lista de objetos que tem como intuito final conter todas as OTs selecionadas no modo de seleção múltipla.

Além disso, foi criada a data variable `requests` para obter a coleção de registo da API `Requests` através do método `GET`.

Como já referido anteriormente, nesta página foi adicionado um `input` que funcionará como filtro, podendo pesquisar pelo respetivo nome da OT ou pela descrição da mesma.

Além disso, foram incorporados uma lista onde apresentar-se-á todas as ordens de transporte disponíveis e um *toggle* que permitirá, como já mencionado, alternar a lista entre o modo de seleção única e múltipla.

Por último, foi introduzido um botão para criar o pedido de transporte das ordens de transporte apenas quando forem selecionadas múltiplas ordens, ou seja, o mesmo estará somente visível quando a lista estiver preenchida.

Para tal foi necessário implementar a seguinte lógica onde é necessário transformar a lista de OTs numa lista em JSON, pois o SAP Build Apps não permite que existam page parameters em forma de lista.

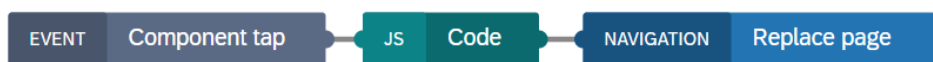


Figura 22: Lógica do botão para criar pedidos de transporte

Após a transformação, o JSON será passado como parâmetro no componente “Replace Page” para se navegar para a página `Create Request`.

5.4.5. Página `My Transport Request`

A página `My Transport Request` é uma replica da página `All Transport Requests`, porém como o nome sugere na mesma apenas apresentar-se-á as ordens de transporte relacionadas ao utilizador em questão.

Esta decisão foi tomada devido a duas principais razões. A primeira deve-se ao facto das limitações de espaço presentes na página para a inserção de novos filtros, visto que com uma área tão reduzida julgou-se preferível que ao invés de adicionar um novo filtro

apenas para filtrar o utilizador, a criação da página seria superior pois em diversos casos as ordens de transporte não são criadas pelos principais intervenientes de um determinado desenvolvimento o que então acabaria por tornar esse filtro pouco rigoroso e inútil.

A outra razão seria facilitar a experiência do utilizador, porque o mesmo poderia aceder de forma rápida às suas ordens de transporte. Apesar do que foi citado acima, caso as ordens sejam realmente provenientes do utilizador em questão, a forma mais veloz do mesmo aceder a estas é ao abrir uma página onde apenas essas sejam apresentadas.

Assim, para proceder a essa filtragem fez-se uso do parâmetro extra na lista onde é possível adicionar um filtro adicional, que neste caso em específico foi o nome do utilizador.

5.4.6. Página Transport Request Details

Nesta página foram criadas duas variáveis distintas, sendo estas a page parameter number e a data variable SingleTR.

Por um lado, a variável number terá o valor do número da ordem de transporte seleccionada para detalhe.

Por sua vez, foi criada a data variable SingleTR que usará a variável já citada para obter um registo único da API Requests, isto é, os detalhes da ordem de transporte, através do método GET.

Inicialmente, nesta página após ser acionado o evento de “Page mounted”, ou seja, após a página for inicializada será apresentado um *spinner* até que a data referente à ordem de transporte seja obtida através da API. Após a mesma ser obtida o *spinner* desaparecerá exibindo a página normalmente.

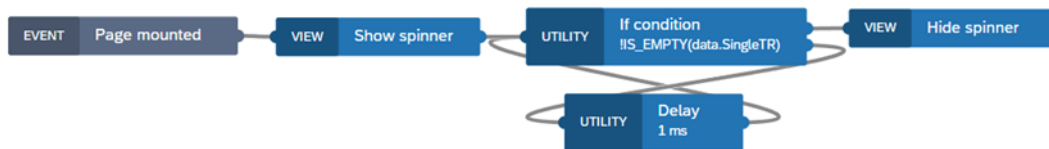


Figura 23: : Lógica da montagem da página Transport Request Details

Além disso, com o intuito de exibir toda a informação possível relacionada com a ordem de transporte foram adicionados também múltiplos containers onde a mesma será

apresentada. Dentro da informação referida será exposto o número, a descrição, o dono, o tipo e as *tasks* da ordem de transporte.

Por último, foi introduzido um botão para criar o pedido de transporte da ordem de transporte. Para tal foi necessário implementar a seguinte lógica onde é necessário transformar a informação da OT numa lista em JSON, pois o SAP Build Apps não permite que existam page parameters em forma de lista.

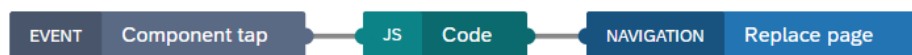


Figura 24: Lógica do botão de criar pedido de transporte na página Transport Request Details

Após a transformação, o JSON será passado como parâmetro no componente “Replace Page” para se navegar para a página Create Request.

5.4.7. Página Create Release Request

Nesta página existem três variáveis distintas, sendo estas a page parameter list, a page variable form e a data variable releaseCreate.

A variável list foi criada com o objetivo de ter disponível as ordens de transporte selecionadas na etapa anterior. Relativamente à tipificação, foi necessário que esta fosse uma string, pois como já referido anteriormente o Build Apps não permite que existam page parameters em forma de lista.

Por sua vez, a variável form é um objeto que como o nome sugere será utilizada no formulário existente na página. Esta contém os atributos subject, reason e transportRequests, sendo os dois primeiros do tipo string e o último uma lista.

Por último, a data variable releaseCreate será usada para criar um registo na API Requests através do método POST.

Como layout esta página apresenta os dois inputs, um com o assunto e outro com a razão da criação do pedido de transporte. Ambos os inputs foram associados com o respetivo atributo do objeto form.

Além dos componentes já referidos foi adicionado uma lista com todas as ordens de transporte selecionadas, sendo esta associada ao atributo transportRequests.

Por último foi adicionado um botão para criar o pedido de transporte. No botão, após premido, são realizadas validações aos campos presentes na página com o intuito de verificar se estes estão preenchidos. No caso da validação ser um sucesso é montado a data variable `releaseCreate` com os dados que foram preenchidos na página e, de seguida, é realizada a criação do registo.

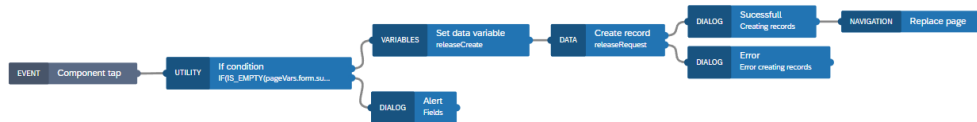


Figura 25: Lógica para criar pedidos de transporte

5.4.8. Página Release Requests

Nesta página apenas foi criada a data variable `releaseRequestCollectionperUser`. Esta variável funciona como uma page variable, porém a sua estrutura é baseada na data entity a que está associada, neste caso a entidade `releaseRequest`. Além disso, dentro destas variáveis é possível escolher o tipo específico dependendo do método HTTP referente, sendo neste caso o método GET da API `releaseRequest`.

Na página serão apresentados todos os pedidos de transporte de OTs realizados pelo utilizador em específico em forma de lista.

Na lista é possível clicar em cada pedido para verificar os seus detalhes. Ao clicar, o evento de tap que foi definido no componente, efetua a navegação para uma nova página, página essa que tem como nome de `ReleaseRequestDetails`.

5.4.9. Página ReleaseRequestDetails

Nesta página foram criadas três variáveis distintas, sendo estas a page parameter `releaseRequestID`, a page variable `form` e a data variable `releaseRequestSingle`.

A variável `releaseRequestID` terá o valor do ID do pedido seleccionado para detalhe.

A variável `form` é um objeto que como o nome sugere será utilizada no formulário existente na página. Esta contém os atributos `subject`, `reason`, `owner` e `transportRequests`, sendo os três primeiros do tipo string e o último uma lista. De forma a alimentar esta

variável, no momento de montagem da página, ser-lhe-ão atribuídos valores da variável `releaseRequestSingle`.

Por sua vez, a `releaseRequestSingle` fará uso da variável mencionada inicialmente de maneira a obter o registo específico da API `releaseRequest` através do método GET.

No layout da página foram adicionados três inputs, sendo estes para o assunto, a razão e dono relativos à criação do pedido de transporte. Todos os inputs foram associados com a respetivo atributo da variável `form`.

Além dos componentes já referidos foi adicionado uma lista com todas as ordens de transporte seleccionadas, sendo esta associada ao atributo `transportRequests` da `form`.

Por último foram adicionados quatro botões principais, podendo estes se dividirem em dois grupos distintos dependendo dos roles do utilizador. No primeiro grupo encontram-se os botões disponíveis para todos os utilizadores, sendo estes os botões de editar e cancelar. No segundo grupo teremos os botões de confirmar e rejeitar estando estes apenas visíveis à equipa de TM.

Apesar das diferentes funções cada um dos botões mencionados tem uma estrutura de lógica muito semelhante. De maneira a facilitar e simplificar a explicação será apresentada uma imagem com a lógica de todos e, por fim, um resumo geral da lógica.

Botão editar

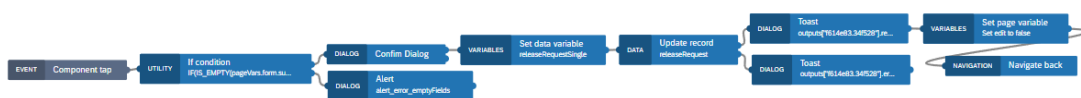


Figura 26: Lógica do botão editar

Botão cancelar



Figura 27: Lógica do botão cancelar

Inicialmente, o valor predefinido pelo sistema, mais especificamente pela variável de sistema `systemVars.i18n.currentLanguage`, é o Inglês, porém dependendo do valor selecionado levará a que o valor desta seja alterado por consequência.

6. Demonstração da Aplicação

Neste capítulo será apresentada uma breve demonstração da aplicação. Será apresentado um passo a passo do processo necessário para um utilizador criar um pedido de transporte e, no final, este seja aceite e, conseqüentemente, as ordens deste sejam liberadas e transportadas.

6.1. Introdução

De forma a apresentar todas as funcionalidades da aplicação iniciaremos com as credenciais do admin, ou seja, da equipa de TM. Caso as credenciais inseridas estejam erradas surgiram uma notificação toast a referir o mesmo.

Por outro lado, caso estas estejam corretas o utilizador será redirecionado para a aplicação.

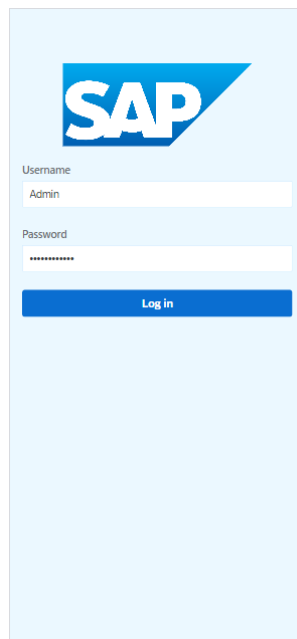


Figura 30: Demonstração - Página de Login

Após o login, o utilizador pode navegar para a página My Transport Request para seleccionar as ordens que pretende transportar. Para seleccionar as ordens o utilizador pode aceder aos detalhes de uma ordem e criar o pedido, porém caso o utilizador pretende seleccionar múltiplas ordens pode ativar o modo de seleção múltipla.

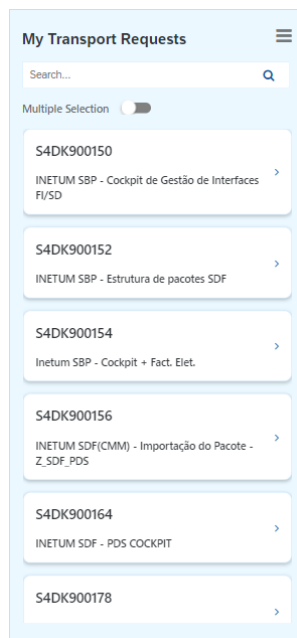


Figura 31: Demonstração - Página My Transport Requests

Após seleccionar a primeira ordem o mesmo segue para a criação do pedido de transporte, onde terá de preencher os campos subject e reason visto estes serem ambos obrigatórios.

Ao submeter a criação do pedido, os campos anteriormente referidos serão verificados e caso estejam a vazio será exibida uma mensagem toast a referir a falta de preenchimento.

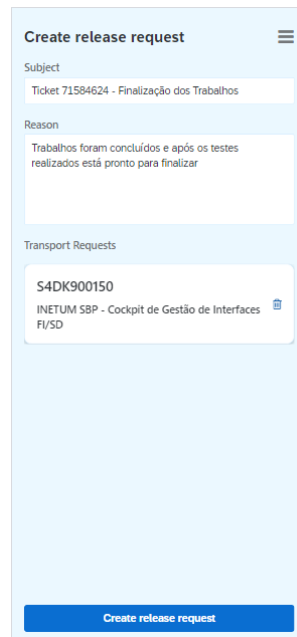


Figura 32: Demonstração - Página Create Release Request

Após a criação bem sucedida, o criador de pedido e a equipa de TM recebem um email a informar da criação do pedido, onde serão exibidas as informações inseridas no momento de criação.



Figura 33: Demonstração - Email de criação

Navegando para a página de detalhes do pedido o admin poderá editar e cancelar o pedido, visto ser ele quem o criou, além de ter disponível as opções de confirmar e negar o pedido, uma vez que este tem autorizações de admin.

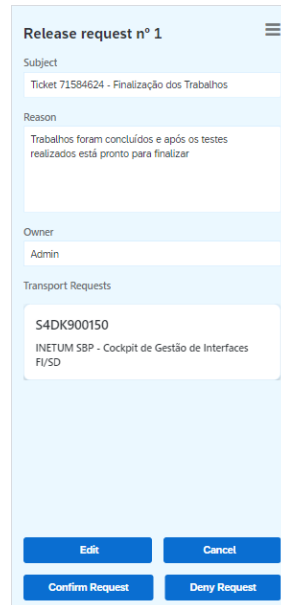


Figura 34: Demonstração - Página Release Request Details

Após cada uma destas opções, os registos relativos ao pedido serão atualizados, além de que, excluindo a opção de editar, será enviado também um email.

À parte do processo de criação de pedidos de transporte, os utilizadores podem alterar o idioma em que a aplicação está a ser executada.

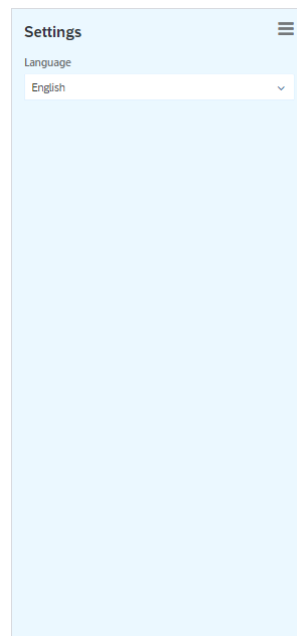


Figura 35: Demonstração - Página de Settings

7. Conclusões

7.1. Conclusão

Em conclusão, a solução desenvolvida teve como principal objetivo desenvolver uma aplicação que permitisse a todos os intervenientes realizar a gestão do transporte de ordens de transporte, agilizando assim o fluxo de trabalho do requerente do transporte como também da equipa de transporte através da automatização de várias etapas que anteriormente seriam realizadas manualmente por ambas as partes.

Numa primeira fase do projeto, houve alguma dificuldade na aprendizagem do SAP Build Apps e do SAP Integration Suite, visto serem duas tecnologias com que se nunca trabalhou sendo necessário seguir algumas Learning Journeys fornecidas pela SAP de maneira a aprender sobre ambas as tecnologias.

Ao tentar testar a aplicação no telemóvel, surgiu também o problema de a aplicação surgir sem dados por não ser obtida nenhuma resposta por parte do *webservice*, pois a máquina onde o *webservice* tem origem, tem um certificado *self-signed*, sendo necessário integrar a API no SAP Integration Suite.

Apesar de maior parte dos problemas ter sido resolvido sugeriram dois no qual não foi possível corrigir devido a problemas da máquina SAP utilizada.

O primeiro problema é a falha no envio de emails aquando da criação, cancelamento, confirmação e rejeição dos pedidos de transporte, visto que o protocolo SMTP não se encontra configurado, contudo, o envio de emails nas etapas mencionadas encontra-se mesmo assim implementado, sendo possível verificar este na transação SOST em SAP.

O segundo problema é a falha no transporte das ordens, uma vez que no ecossistema onde a máquina se encontra existe apenas esta máquina não sendo possível o transporte para outros ambientes como seria normalmente realizado. Tal como mencionado no problema anterior, apesar desta funcionalidade também não se encontrar ativa, o código encontra-se totalmente desenvolvido.

A solução implementada cumpriu todos os aspetos idealizados e propostos. Os objetivos propostos para este projeto consistiram no desenvolvimento da arquitetura da aplicação, através da modelação e da criação das *mockups*, o desenvolvimento de uma API em SAP, a integração da API e dos seus *endpoints* no SAP Integration Suite, a implementação de uma aplicação em SAP Build Apps.

Além dos objetivos relativos ao desenvolvimento terem sido alcançados, o objetivo principal deste projeto, isto é, a automatização dos processos no transporte de ordens, mais especificamente, a troca de emails e a liberação e transporte das ordens foi também alcançado apesar dos problemas com o SMTP e com a falta de mais ambientes no ecossistema.

Resta salientar que, como trabalho proposto, sugiro a melhoria da interface para web, uma vez que apesar da principal forma de uso da aplicação seja a aplicação móvel, no browser o design encontra-se pouco responsivo apesar do próprio SAP Build Apps tentar torná-la responsiva.

Além disso, para trabalho futuro poderia ser adicionada uma funcionalidade onde seria disponibilizada a possibilidade de verificar a integridade das ordens de transporte, analisando os objetos presentes nesta e ao verificar os *logs* de liberação e transporte.

Bibliografia

- [1] T. F. Wallace e M. H. Kremzar, *ERP: Making It Happen: The Implementers' Guide to Success with Enterprise Resource Planning*, John Wiley & Sons, 2002.
- [2] A. a. E. M. Aytaç, “The Role of ERP in Advanced Managerial Accounting Techniques: A Conceptual Framework,” *Business and Economics Research Journal*, vol. 8, nº 1, pp. 83-100, 03 2017.
- [3] Z. Lin, “Comparison on the Implementations of ERPs with Different Versions,” *BCP Business & Management*, vol. 34, pp. 1003-1009, 12 2022.
- [4] M. Rashid, L. Hossain e J. Patrick, “The Evolution of ERP Systems: A Historical Perspective,” em *Enterprise Resource Planning: Global Opportunities and Challenges*, I. G. Publishing, 2002, pp. 1-16.
- [5] A. C. R. Gomes, “Avaliação da implementação do SAP ERP na logística: estudo de caso,” 2015.
- [6] Oracle, “What are analysts saying about Oracle Fusion Cloud ERP?,” What are analysts saying about Oracle Fusion Cloud ERP?, [Online]. Available: <https://www.oracle.com/erp/gartner-magic-quadrant/#form>.
- [7] T. Orosz, “Analysis of SAP Development tools and methods,” em *2011 15th IEEE International Conference on Intelligent Engineering Systems*, Poprad, Slovakia, 2011.
- [8] U. Shahzad, “A comparative analysis of ERP system providers,” 2023.
- [9] M. da Cruz, H. de Paula, B. Caputo, S. Mafra, P. Lorenz e J. Rodrigues, “OLP—A RESTful Open Low-Code Platform,” *Future Internet*, vol. 13, nº 10, 2021.
- [10] G. Dushnitsky e B. K. Stroube, “Low-code entrepreneurship: Shopify and the alternative path to growth,” *Journal of Business Venturing Insights*, vol. 16, p. e00251, 2021.
- [11] “What Are The Benefits of Low-Code: Low-Code Basics,” [Online]. Available: <https://appian.com/low-code-basics/benefits.html>. [Acedido em 20 10 2023].
- [12] K. Kalinin, “Lowcode/Nocode vs Traditional Development: How to Choose?,” 18 07 2023. [Online]. Available: <https://topflightapps.com/ideas/no-code-low-code-vs-traditional-development/>. [Acedido em 20 10 2023].
- [13] A. Kermanchi, “Developer Experience in Low-Code Versus Traditional Development Platforms - A Comparative Experiment,” 2022.
- [14] “Appian Named a Leader in the Low-Code Gartner® Magic Quadrant™ for 2023,” 2023. [Online]. Available: https://appian.com/learn/resources/resource-center/google/2023/gartner-magic-quadrant-for-low-code-2023.html?gclid=CjwKCAjwysipBhBXEiwApJOcuwkFyrVp51B7qYDzvkBUBxBgaEl4Yfn7EL4Z9-48EvUFjnoM1jVEURoC4KIQAuD_BwE. [Acedido em 20 10 2023].

- [15] F. a. T. G. Gürcan, “Using Microsoft PowerApps, Mendix and OutSystems in Two Development Scenarios: An Experience Report,” *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 67-72, 2021.
- [16] J. C. a. J. N. a. R. B. a. W. A. Kirchhof, “Navigating the Low-Code Landscape: A Comparison of Development Platforms,” em *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2023.
- [17] “REST,” Mozilla, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/REST>. [Acedido em 10 2024].
- [18] “What is a REST API?,” IBM, [Online]. Available: <https://www.ibm.com/topics/rest-apis>. [Acedido em 10 2024].
- [19] “Introdução ao JSON,” [Online]. Available: <https://www.json.org/json-pt.html>. [Acedido em 10 2024].
- [20] “What Is SAP Build Apps?,” [Online]. Available: <https://help.sap.com/docs/build-apps/service-guide/what-is-sap-build-apps>. [Acedido em 10 2024].
- [21] “What is SAP Integration Suite?,” [Online]. Available: <https://help.sap.com/docs/integration-suite/sap-integration-suite/what-is-sap-integration-suite>. [Acedido em 10 2024].
- [22] “What is the Simple Mail Transfer Protocol (SMTP)?,” [Online]. Available: <https://www.cloudflare.com/learning/email-security/what-is-smtp/>. [Acedido em 10 2024].
- [23] “Figma,” [Online]. Available: <https://www.figma.com/>. [Acedido em 10 2024].
- [24] “10 reasons why you should use draw.io,” [Online]. Available: <https://drawio-app.com/blog/10-reasons-why-you-should-use-draw-io/>. [Acedido em 10 2024].
- [25] V. H. G. Coutinho, “Relatório de Estágio,” 2019.