IPG Politécnico da Guarda
Polytechnic of Guarda

# RELATÓRIO DE ESTÁGIO

Licenciatura em Engenharia Informática

Hugo Miguel Mendes Amaral

novembro | 2020

**Escola Superior de Tecnologia e Gestão**

Instituto Politécnico da Guarda

# RELATÓRIO DE ESTÁGIO

HUGO MIGUEL MENDES AMARAL

RELATÓRIO PARA A OBTENÇÃO DO GRAU DE LICENCIADO

EM ENGENHARIA INFORMÁTICA

Novembro 2020

# Agradecimentos

Em primeiro lugar gostaria de agradecer à Armis pela oportunidade da realização do estágio, mesmo numa altura crítica provocada pela pandemia de Covid-19. Dentro da Armis, agradeço a todos os que, de alguma forma, estiveram envolvidos na realização do projeto, em especial ao supervisor de estágio Orlando Neto, por toda a paciência para esclarecimento de dúvidas e orientação naquele que era um novo ambiente, e à Sofia Pires por toda a disponibilidade para ajuda à redação do relatório.

Agradeço também aos meus familiares, que acreditaram no meu gosto pela área, em especial ao meu pai Pedro Amaral, que sempre apoiou totalmente as minhas decisões e pelos conselhos e ânimo em momentos menos bons.

Um obrigado também ao meu orientador, professor José Figueiredo pelo gosto que tem pelo ensino na área da informática e por todas as sugestões de melhoramento ao relatório.

Gostaria ainda de incluir um agradecimento ao professor José Fonseca pela sua acessibilidade enquanto diretor de curso e preocupação demonstrada para com todos os alunos de Engenharia Informática do Instituto Politécnico da Guarda.

Por fim, mas não menos importante, um obrigado ao meu colega de projeto Ricardo Figueiredo por todo o companheirismo e apoio prestados ao longo da realização do estágio.

# Ficha de identificação

## Aluno

**Nome:** Hugo Miguel Mendes Amaral

**Número:** 1700149

**Licenciatura:** Engenharia Informática

## Estabelecimento de Ensino

Instituto Politécnico da Guarda (IPG)

Escola Superior de Tecnologia e Gestão (ESTG)

## Entidade Acolhedora do Estágio

**Nome:** Armis Sourcing, LDA

**Morada:** Rua do Freixo, 725, 4300-217 Porto

**Contacto Telefónico:** 226 002 295

## Supervisor de Estágio

**Nome:** Orlando Neto

**Função:** Chief People & Culture Officer

## Docente Orientador de Estágio

**Nome:** José Alberto Quitério Figueiredo

**Grau Académico:** Mestre

# Resumo

O uso das tecnologias da informação e comunicação está inevitavelmente cada vez mais presente no meio empresarial, estando estas diretamente ligadas a atividades internas e atividades de planeamento e gestão. Elas podem ser introduzidas para aperfeiçoar ou tornar mais eficiente um conjunto de tarefas já existentes.

O departamento de recursos humanos da Armis Group, de forma a substituir a entrada manual de dados em ficheiros Excel, pretende uma ferramenta que permita gerir a informação relativa ao histórico de colaboradores, diminuindo a duplicidade de operações e a multiplicação de ficheiros.

Dessa necessidade surge o Histórico de Colaboradores: uma solução que tem como objetivo a troca de processos manuais de inserção de dados em ficheiros Excel por um sistema informático online e automatizado.

Assim sendo, o presente documento descreve todo o processo de análise e desenvolvimento do projeto em contexto de estágio Histórico de Colaboradores realizado pelo aluno Hugo Miguel Mendes Amaral na instituição Armis Group.

**Palavras-Chave:** recursos humanos, histórico de colaboradores, Microsoft Teams, ASP.NET.

# Abstract

The use of information and communication technologies is inevitable more and more present in the business environment, which are directly linked to internal activities and planning and management activities. They can be introduced to improve or make more efficient a set of existing tasks.

The human resources department at Armis Group, in order to replace manual data entry in Excel files, needs a tool that allows managing the information related to the collaborator's history, reducing duplicated operations and file multiplication.

From this need, the Historical Collaborator's Data is born: a solution that aims to Exchange manual processes of data entry in Excel files by an online and automated computer system.

Therefore, this document describes the entire process of analysis and development of the project Historical Collaborator's Data made by the student Hugo Miguel Mendes Amaral at the company Armis Group.

**Keywords:** human resources, collaborator's history, Microsoft Teams, ASP.NET

# Índice geral

# Índice de figuras

# Índice de tabelas

# Lista de siglas e acrónimos

ASP – Active Server Pages

Azure AD – Azure Active Directory

CSOM – Client Side Object Model

CSS – Cascading Style Sheets

ERP – Enterprise Resource Planning

ESTG – Escola Superior de Tecnologia e Gestão

HTML – HyperText Markup Language

IDE – Integrated Development Environment

IPG – Instituto Politécnico da Guarda

JSON – JavaScript Object Notation

MIM – Microsoft Identity Manager

SDK – Software Development Kit

SQL – Structured Query Language

SSMS – SQL Server Management Studio

URL – Uniform Resource Locator

# 1 Introdução

O presente relatório descreve o desenvolvimento do projeto em contexto de estágio Histórico de Colaboradores realizado pelo aluno Hugo Miguel Mendes Amaral na instituição Armis Group, no âmbito da unidade curricular de Projeto de Informática.

## 1.1 Caraterização sumária da instituição

A Armis Group foi a instituição que tornou possível a realização deste estágio curricular, acolhendo-me durante 4 meses.

A Armis é um grupo empresarial português que se dedica ao desenvolvimento e implementação de soluções digitais complexas. O grupo atua em quatro principais áreas de negócio:

- ARMIS IT - Information Technology - Centra esforços na evolução das mais recentes tecnologias e no desenvolvimento de soluções como Portais, Aplicações *Mobile*, *Business Intelligence*, Segurança e Gestão de Identidades, Testes e Certificação de *Software*.
- ARMIS ITS - Intelligent Transport Systems - Responsável pela nova geração de soluções inteligentes na área dos transportes.
- ARMIS Digital Sport - Tem foco na conceção de soluções digitais especializadas na área do desporto.
- ARMIS OZONO – Voltada para *outsourcing* de serviços, processos e profissionais especializados, bem como na administração de operações e sistemas em ambiente de *cloud*.

Com a sede localizada na cidade do Porto e um escritório em Lisboa, a Armis Group abraça também projetos em países como o Egito, Rússia, Espanha, Angola e Irlanda. A nível internacional, a Armis conta com escritórios no Brasil, Macau e Países Baixos, sendo o último inaugurado recentemente.

## 1.2 Motivação

Como finalista do curso de engenharia informática, a entrada no mercado de trabalho é algo que gera algum inquietamento, ansiedade e, acima de tudo, preocupação. Dessa forma, um estágio não só proporciona um ambiente propício para a aquisição de novos conhecimentos, como também dá a oportunidade ao estagiário de obter o primeiro contacto no âmbito laboral e ser integrado numa equipa.

A escolha da Armis baseia-se no facto de esta ter ganho diversos prémios em anos recentes e no seu constante recrutamento nas áreas da segurança, desenvolvimento de aplicações *web* e PL/SQL. Além disso, a localização da sua sede na cidade do Porto teve também grande influência na tomada de decisão, dado que é uma cidade segura, servida por várias redes de transportes e, principalmente, com crescente oferta de trabalho.

## 1.3 Descrição do problema

O uso das tecnologias da informação está inevitavelmente cada vez mais presente no meio empresarial, estando estas diretamente ligadas a atividades internas e atividades de planeamento e gestão. Elas podem ser introduzidas para aperfeiçoar ou tornar mais eficiente um conjunto de tarefas já existentes possibilitando ter toda a informação num lugar só e acessível a pessoas autorizadas em vários dispositivos.

Partindo dessa premissa, o Histórico de Colaboradores surge como uma ferramenta, a ser usada pelo departamento de Recursos Humanos, que vem substituir a entrada manual de dados em ficheiros Excel por um sistema capaz de fazer a gestão de toda a informação relacionada com colaboradores e seu histórico na empresa, além de manter guardadas todas as operações feitas e respetivo autor. Esses dados devem ser armazenados de forma a que possam ser interpretados como válidos apenas num período específico, permitindo assim uma análise eficaz baseada no tempo.

## 1.4 Objetivos

O projeto em contexto de estágio tem como objetivo compor, juntamente com o colega e também estagiário curricular Ricardo Figueiredo, uma equipa de desenvolvimento para prototipação, análise, desenho, programação e implementação da aplicação Histórico de Colaboradores.

Esta aplicação, a ser importada para a plataforma unificada de comunicação e colaboração Microsoft Teams, deve permitir ao departamento de Recursos Humanos substituir a entrada manual de dados em ficheiros Excel por um sistema que permita gerir informação relativa ao histórico de colaboradores. O projeto pode ser dividido em três grandes fases:

1. Desenvolvimento da aplicação na plataforma Power Apps.
2. Integração com o Microsoft Identity Manager (MIM) e o Enterprise Resource Planning (ERP).
3. Integração com o People Hub.

A primeira fase consiste na elaboração da aplicação, passando por todas as fases de desenvolvimento de *software*, a saber: o planeamento, levantamento de requisitos, codificação e testes. Na segunda fase, a aplicação desenvolvida na fase anterior é integrada com as fontes externas de informação MIM e ERP. Dessa forma, todas as alterações feitas ao histórico de colaboradores a partir dessas ferramentas devem ser visíveis no Histórico de Colaboradores. Por fim, na terceira fase é feita a integração com o People Hub, uma solução desenvolvida pela Armis.

Como se trata de um projeto a dois, as tarefas deviam estar igualmente divididas. Assim, optou-se pela divisão clássica entre *back-end* e *front-end*, estando responsável pelas tarefas relativas ao *front-end* e o Ricardo Figueiredo pelas tarefas relativas ao *back-end*.

## 1.5 Estrutura do documento

O presente relatório está dividido em oito capítulos, cumprindo a seguinte ordem:

- Capítulo 1 – Expõe uma perspetiva geral do documento, incluindo uma descrição da instituição, descrição do problema e objetivos.
- Capítulo 2 – Descreve brevemente a função dos recursos humanos nas instituições e faz uma análise crítica aos procedimentos da empresa antes da realização do projeto.
- Capítulo 3 – Identifica e descreve a metodologia utilizada e explica como ela foi aplicada ao longo do estágio.
- Capítulo 4 – Apresenta todos os requisitos funcionais e não funcionais da aplicação, bem como os diagramas de contexto, casos de uso, classes e de sequências, as principais descrições dos casos de uso e o dicionário de dados.
- Capítulo 5 – Identifica e descreve brevemente todas as tecnologias utilizadas para a realização do projeto.
- Capítulo 6 – Explana o trabalho que foi efetuado, e o seu progresso para atingir os objetivos propostos.
- Capítulo 7 – Aborda todos os testes realizados à aplicação durante a fase de desenvolvimento e final.
- Capítulo 8 – Apresenta uma síntese do relatório e do projeto e algumas considerações adicionais, como as dificuldades encontradas.

# 2 Estado da Arte

Estado da arte pode ser definido como o estado atual de conhecimento ou avanço de um determinado assunto que está a ser tema de estudo. Nesta secção é apresentada a função dos gestores de recursos humanos de um ponto de vista geral e sua importância para as instituições na atualidade e, em particular, uma análise crítica e objetiva aos procedimentos presentes na empresa para a manutenção do histórico dos colaboradores. Além disso, será brevemente analisada uma ferramenta de auxílio à gestão de recursos humanos.

## 2.1 O que é a gestão de recursos humanos?

Recursos são definidos como algo, como por exemplo dinheiro, materiais, equipamentos, que podem ser usados para ajudar no crescimento de uma empresa, negócio ou país [3]. Os recursos humanos, o principal recurso de qualquer organização, estão diretamente ligados às pessoas dentro da instituição às quais se atribui o nome de colaboradores.

Na atualidade, a gestão estratégica de pessoas não pode ser vista como uma despesa para as empresas, mas sim como uma ferramenta de extrema importância para o desenvolvimento das organizações, focando a produtividade, relações interpessoais, mercado de trabalho e crescimento da organização. Dessa forma, o departamento dos recursos humanos é o responsável pelas atividades de planeamento e controlo de uma organização, visando construir e manter a relação entre colaboradores e a instituição, a fim de atingir tanto os objetivos da organização, como os objetivos do colaborador [2]. A gestão de recursos humanos inclui um amplo espetro de atividades que incluem planeamento, recrutamento, desenvolvimento, motivação, gestão de mudança, relacionamentos, avaliação de desempenho e apreciação (figura 1).


**Figura 1 - Funções da gestão de recursos humanos [4]**

- **Planeamento**

  O planeamento de recursos humanos é o processo que identifica as necessidades atuais e futuras de recursos humanos para que uma organização atinja os seus objetivos [5]. O objetivo deste planeamento é trazer uma visão realista para a administração de que a organização se adapta a mudanças tecnológicas, políticas, sociais e económicas mais rapidamente e efetivamente de acordo com a situação e necessidades de tempo. Individualmente, traz oportunidades de crescimento e desenvolvimente de habilidades, talento e soft skills [6].

- **Recrutamento**

  Recrutamento refere-se ao processo de identificação, atração, entrevista, seleção e contratação de colaboradores. Em outras palavras, envolve todos os passos

desde a identificação de uma necessidade de pessoal até o seu preenchimento. Um recrutamento desorganizado pode levar à contratação de colaboradores que não são ideais para a função.

- **Desenvolvimento**

  Desenvolvimento é a parte da gestão de recursos humanos que lida com o treino e desenvolvimento de habilidades dos colaboradores na organização. Ao contrário de outros recursos, os recursos humanos têm um potencial ilimitado. Cabe ao sistema de desenvolvimento de recursos humanos criar um ambiente que possa identificar, trazer à tona, treinar e usar continuamente as capacidades dos colaboradores. A rápida mudança das tecnologias veio aumentar ainda mais a importância desta área.

- **Motivação**

  O papel da gestão de recursos humanos na organização foca também assegurar um ambiente que valorize o potencial humano e que gere um clima favorável à motivação das pessoas, levando-as a contribuir com o seu melhor e retendo os seus talentos [1].

- **Gestão de mudança**

  A gestão de mudança é a aplicação de conhecimentos, recursos e ferramentas para lidar com eventuais mudanças. Envolve a definição e adoção de estratégias, procedimentos e tecnologias para suportar mudanças na área de negócio. O principal objetivo é implementar com sucesso essas novas estratégias, processos e tecnologias minimizando ao máximo os resultados negativos.

- **Relacionamentos**

  Dentro de uma organização os relacionamentos podem ser entre a própria organização e o colaborador e entre colegas de trabalho. É da responsabilidade dos gestores de recursos humanos preservar estes relacionamentos saudáveis.

Para os colaboradores serem produtivos, é necessário que estejam inseridos em um ambiente de trabalho que estimule a sua criatividade. Quando os colaboradores têm um bom relacionamento com os colegas de trabalho, isso fica evidente no seu desempenho e produtividade, havendo mais comunicação e cooperação.

- **Avaliação de desempenho**

  Avaliação de desempenho consiste no processo de avaliar quão eficazmente os colaboradores estão a cumprir as suas responsabilidades e a contribuir para o cumprimento dos objetivos da organização. Para uma avaliação efetiva, os gestores de recursos humanos devem estar informados sobre os resultados a esperar para uma função específica, acompanhar o comportamento e resultados do colaborador e comparar a expectativa com o comportamento e resultados observados. É também boa prática fornecer um *feedback* ao colaborador para que possa haver aprimoramento do seu desempenho se necessário.

- **Apreciação**

  A apreciação baseia-se em verificar e recompensar um colaborador, de acordo com o seu desempenho e resultados, e manter um registo disso. Esta área da gestão de recursos humanos está ligada com a motivação, uma vez que um bom sistema de recompensas manterá os colaboradores motivados a atingir os objetivos da organização. As recompensas podem ser de caráter financeiro, como aumento de salário base, bónus, seguro de saúde, carro de empresa, ou de caráter não financeiro, como oportunidades de desenvolvimento de carreira e promoções [7].

## 2.2 Análise crítica à solução existente

O Microsoft Excel consolidou-se como uma ferramenta essencial na maioria das organizações empresariais em todo o mundo. Ele pode ser usado para criar lembretes, controlar as tendências de vendas e outros dados de negócio. Além disso, o Excel

possui uma interface relativamente simples, o que permite aos utilizadores entender e realizar tarefas básicas neste *software*. Empresas das mais variadas dimensões usam o Excel para armazenar informação sobre os seus colaboradores. As folhas de cálculo atuam como uma base de dados, que permite à organização fazer a gestão e análise dos dados presentes.

O cenário descrito era o que estava vigente na Armis para fazer a manutenção do histórico de colaboradores: uma tabela, presente numa folha de cálculo do Excel, cujas linhas continham informação relacionada a um determinado colaborador. A figura 2, mostra parte desse ficheiro.

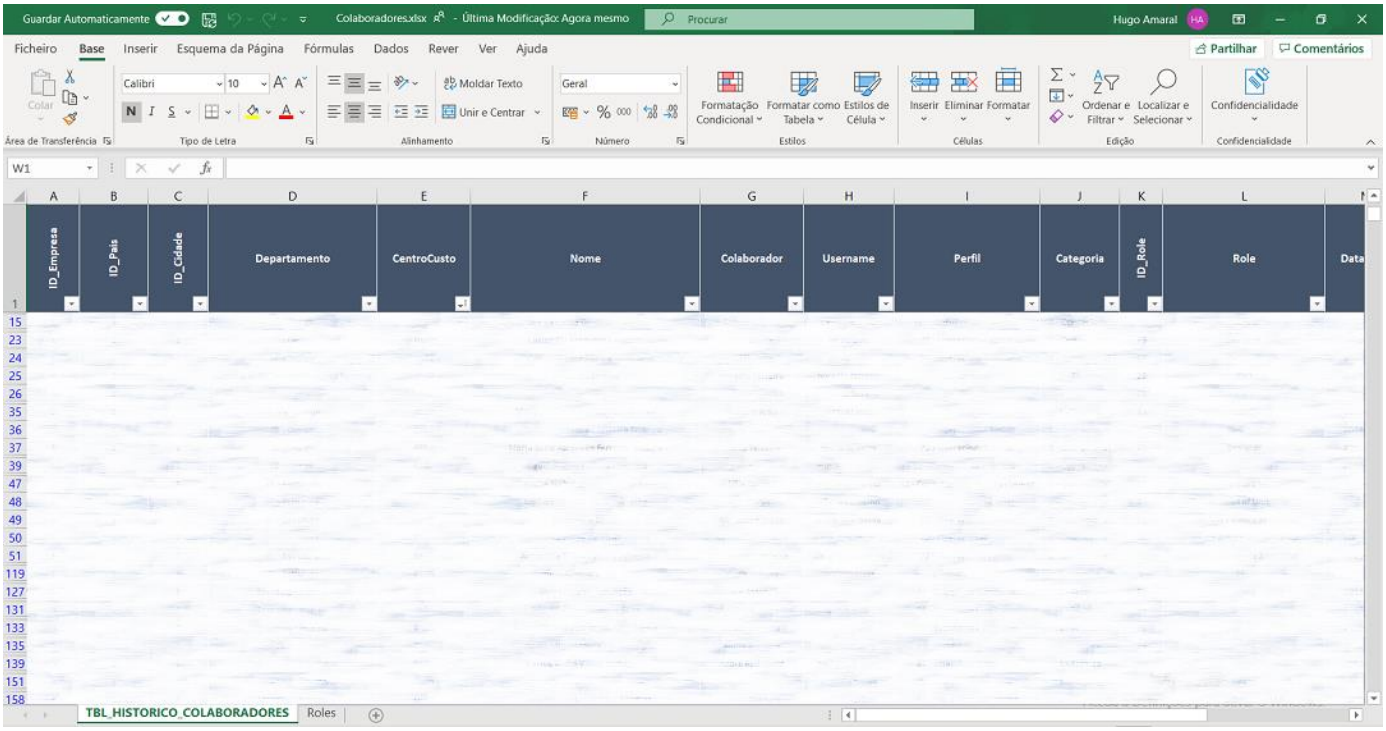


**Figura 2 - Ficheiro Excel do histórico de colaboradores (dados distorcidos por motivos de confidencialidade)**

Apesar de ser de acessível e de simples utilização, estas folhas de cálculo podem trazer alguns problemas, com tendência a agravarem-se conforme o crescimento da organização. Assim como muitos programas, o uso do Excel é um processo manual. Isso significa que o conteúdo das células está sempre suscetível a erros humanos, como escrever ou copiar valores errados. Segundo a Forbes, quase 90% das folhas de cálculo

possuem erros devido à entrada manual de dados [8]. Além disso, a redundância de dados presente no Excel dificulta a garantia de informações precisas, coesas e coerentes aquando uma alteração é necessária. Outro grande obstáculo é a falta de suporte para colaboração: apenas um utilizador pode atualizar a folha de cálculo ao mesmo tempo. Dessa forma, torna-se obrigatório o uso de pastas partilhadas ou anexos de e-mail para a partilha destes ficheiros. Por fim, o Excel carece também de medidas e níveis de segurança. Por ser opcionalmente protegido por uma *password,* apenas dá a possibilidade aos colaboradores de realizar todos ou nenhum tipo de operações, não havendo níveis intermédios de autorização.

O Histórico de Colaboradores surge como uma ferramenta para colmatar todos estes obstáculos: uma aplicação que mantenha a informação organizada e disponível sempre que necessário, valide todas as entradas de dados, ofereça suporte para colaboração entre pessoas autorizadas e agilize o processo de gestão do histórico de colaboradores.

## 2.3 Aplicações existentes

Atualmente, existem inúmeras ferramentas que facilitam o trabalho de gestão de recursos humanos. Elas, de um modo geral, oferecem funcionalidades que vão desde o processamento de salários até à avaliação de desempenho.

Depois de uma pesquisa profunda e análise de várias ferramentas, a aplicação cujos objetivos se assemelham mais com o Histórico de Colaboradores é a NAVHR [25]. Esta solução, assim como será o Histórico de Colaboradores, é integrada no ERP e visa acompanhar o desenvolvimento de cada colaborador dentro da organização. Além disso, integra também uma ferramenta de gestão interna desenvolvida pela própria empresa, o HyColaborador, que permite acesso facilitado a várias informações sobre os colaboradores. Em conformidade, o Histórico de Colaboradores pretende igualmente ser integrado com o People Hub da Armis. No entanto, a NAVHR vai mais além e inclui também sistemas de segurança e higiene no trabalho, gestão de equipamentos do colaborador e alertas por data.

Com esta análise, conclui-se que a integração da solução com outros sistemas externos é de extrema importância para gerir, processar e manter organizada todas a informação referente à gestão de recursos humanos de uma organização.

# 3 Metodologia

A definição de uma metodologia no desenvolvimento de *software* é um passo crucial para se obter *software* de qualidade e cumprir corretamente todos os requisitos. Entre as possíveis abordagens, optou-se pela metodologia ágil. Os métodos ágeis são métodos de desenvolvimento em incrementos, que se concentram no desenvolvimento rápido de *software* e lançamentos frequentes, reduzindo as despesas gerais do processo, minimizando a documentação e produzindo código de alta qualidade. O manifesto da metodologia ágil [9] é composto por 4 valores principais:

- "Indivíduos e interacções mais do que processos e ferramentas"
- "Software funcional mais do que documentação abrangente"
- "Colaboração com o cliente mais do que negociação contratual"
- "Responder à mudança mais do que seguir um plano"

Dentro das metodologias ágeis optou-se pelo seguimento do modelo SCRUM. Este capítulo irá descrever a metodologia enunciada e especificar de que forma foi aplicada no desenvolvimento do projeto.

## 3.1 Metodologia aplicada - SCRUM

SCRUM [10] é uma *framework* ágil de desenvolvimento de produtos. Esta abordagem define uma estratégia flexível e holística de desenvolvimento de produtos. Em vez de uma abordagem sequencial, os produtos são progressivamente desenvolvidos e melhorados de forma iterativa e incremental, o que faz dele um modelo eficaz para projetos dinâmicos e suscetíveis a mudanças de requisitos, sejam eles novos ou apenas requisitos modificados. A figura 3 representa graficamente o funcionamento do SCRUM.

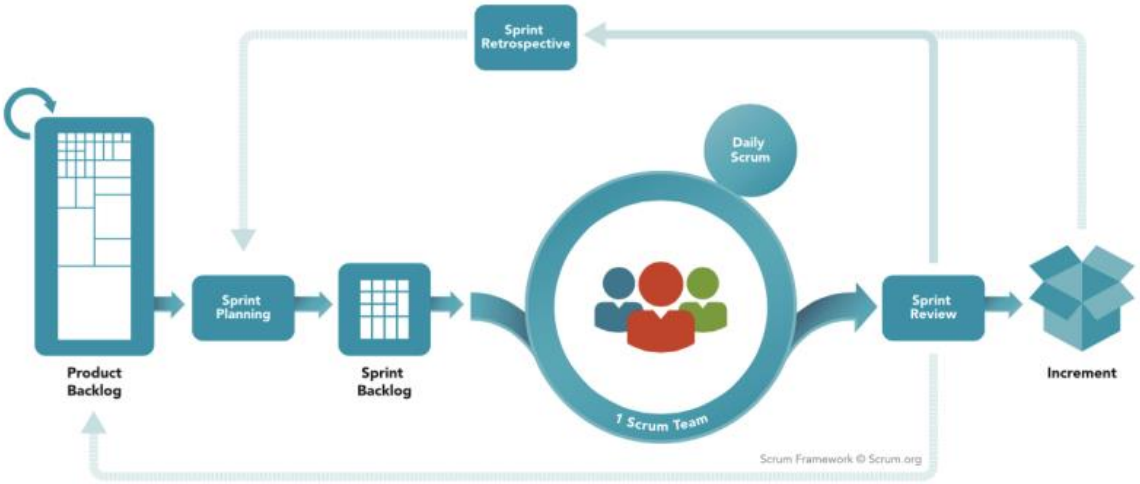


**Figura 3 - Funcionamento do SCRUM [10]**

O ciclo do SCRUM tem o seu progresso baseado numa série de iterações, cada uma com duração de 1 a 4 semanas, chamadas *Sprints*. Antes de cada *Sprint* realiza-se uma reunião de planeamento, chamada *Sprint Planning*, onde alguns requisitos do *Product Backlog* são trazidos para o *Sprint Backlog*. No decorrer da fase seguinte (execução do *Sprint)* a equipa de desenvolvedores reúne-se diariamente (D*aily Scrum*), durante o máximo 15 minutos, para avaliar o seu progresso. No final de cada *Sprint* é realizada uma revisão do produto do *sprint* (*Sprint Review*) em conjunto com o cliente e *stakeholders*[1]. Por fim, a equipa procede a uma reunião de retrospetiva do *Sprint*, que lhes permite melhorar o processo futuramente, e o ciclo recomeça.

## 3.2 Aplicação da metodologia SCRUM no projeto

A adoção do SCRUM teve auxílio de "gestão à vista" a partir do uso de um quadro *kanban* (figura 4).

O painel "A Fazer", exibido no quadro, representa o *Product Backlog*. Inicialmente foram lá colocadas todas as tarefas a cumprir e, posteriormente, acrescentadas novas tarefas à medida que estas iam surgindo. O *Sprint Backlog* está representado pelo painel "Em Andamento".

[1] Stakeholders – Todas as pessoas que têm interesse na realização do projeto

Cada tarefa presente no quadro cumpria o seguinte *template*:

1. Prioridade
2. Título
3. Descrição
4. Responsável

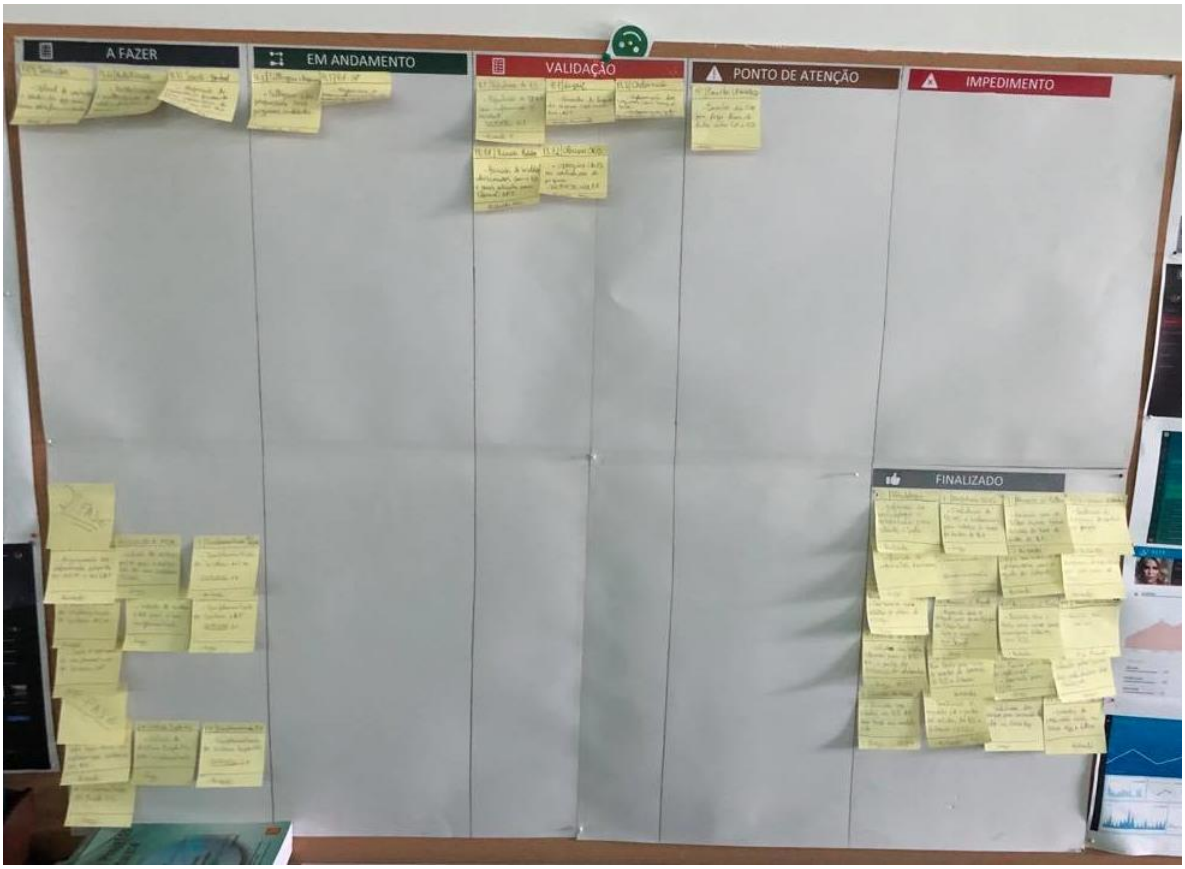


**Figura 4 - Quadro Kanban do projeto**

Durante cada *Sprint* eram feitas reuniões diárias de curta duração, juntamente com o supervisor, onde se esclareciam dúvidas, se relatava o realizado no dia anterior e os alvos a atingir no próprio dia. Quando todas as tarefas do *Sprint* estavam cumpridas, eram movidas para o painel "Validação" e fazia-se um *Sprint Review*. Por fim, se as tarefas satisfizessem os requisitos, eram colocadas no painel "Finalizado". Se, por alguma razão, não fosse possível o cumprimento de uma tarefa, ela era transferida para os painéis "Ponto de Atenção" ou "Impedimento", dependendo do motivo.

# 4 Análise de Requisitos

Os requisitos são o ponto de partida para toda a definição de um sistema e, consequentemente, são fatores decisivos no desenvolvimento do produto final. Eles expressam as características do *software* do ponto de vista da satisfação das necessidades do utilizador [11], ou seja, são características que o software ou o sistema a desenvolver deverá cumprir.

Este capítulo será dedicado tanto à análise de requisitos funcionais como de não funcionais, serão apresentados os diagramas de contexto, casos de uso, classes e de sequências, as principais descrições dos casos de uso e o dicionário de dados.

## 4.1 Requisitos funcionais

Os requisitos funcionais são aqueles que descrevem o comportamento do sistema, ou seja, são aqueles que descrevem as funcionalidades, as expectantes que o sistema desempenhe [11].

Dessa forma, o sistema deverá possibilitar:

- Gerir Áreas de Negócio
- Gerir Departamentos
- Gerir Empresas
- Gerir Colaboradores
- Gerir Países
- Gerir Cidades
- Gerir Cargos
- Gerir Perfis
- Gerir Escritórios
- Gerir o Histórico de Colaboradores
- Consultar Centros de Custo
- Consultar Preços de Perfil

- Alterar a ordem de visualização das listas por arrastar os itens
- Inserir contratos e adicionar o respetivo documento em listas do SharePoint
- Autenticação a partir da conta do Office365
- Obter uma lista de colaboradores não registados a partir do Azure Active Directory (Azure AD)
- Guardar o autor de cada operação de inserção e edição

## 4.2 Requisitos não funcionais

Os requisitos não funcionais não estão diretamente ligados com as funções fornecidas pelo sistema. Antes, preocupam-se com os padrões de qualidade como confiabilidade, desempenho, segurança, facilidade de utilização, legibilidade, entre outros [11].

Os requisitos não funcionais são:

- O sistema deve estar em inglês
- O sistema deve correr na plataforma unificada de comunicação e colaboração Microsoft Teams

## 4.3 Diagrama de Contexto

O diagrama de contexto é um diagrama de fluxo de dados e contém apenas um processo, representando todo o sistema, que estabelece o contexto e os limites do sistema a ser modelado. Ele identifica os fluxos de informação entre o sistema e entidades externas, isto é, os atores. Os atores, por sua vez, serão o gestor de recursos humanos, o MIM, o ERP e o People Hub. A figura 5 representa o diagrama de contexto referente ao projeto.

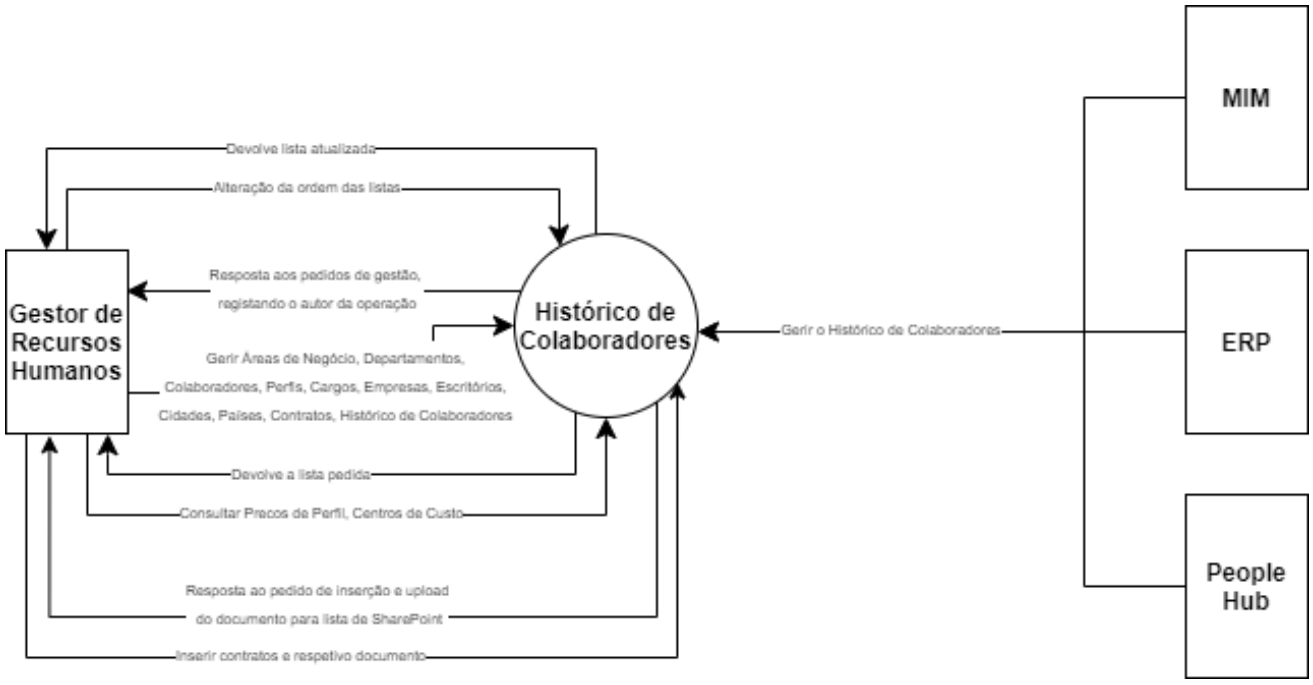


**Figura 5 - Diagrama de Contexto**

## 4.4 Diagrama de casos de uso

Os casos de uso são muito úteis para auxiliar na análise de requisitos do sistema. Em grande número de casos, são a primeira coisa a fazer ao abordar um projeto. Cada caso de uso corresponde a um potencial requisito. O diagrama de casos de uso organiza os requisitos na perspetiva dos utilizadores, e faz os relacionamentos entre os atores e os casos de uso. Além disso reflete também o comportamento que o sistema deve disponibilizar. A figura 6 representa o diagrama de casos de uso.

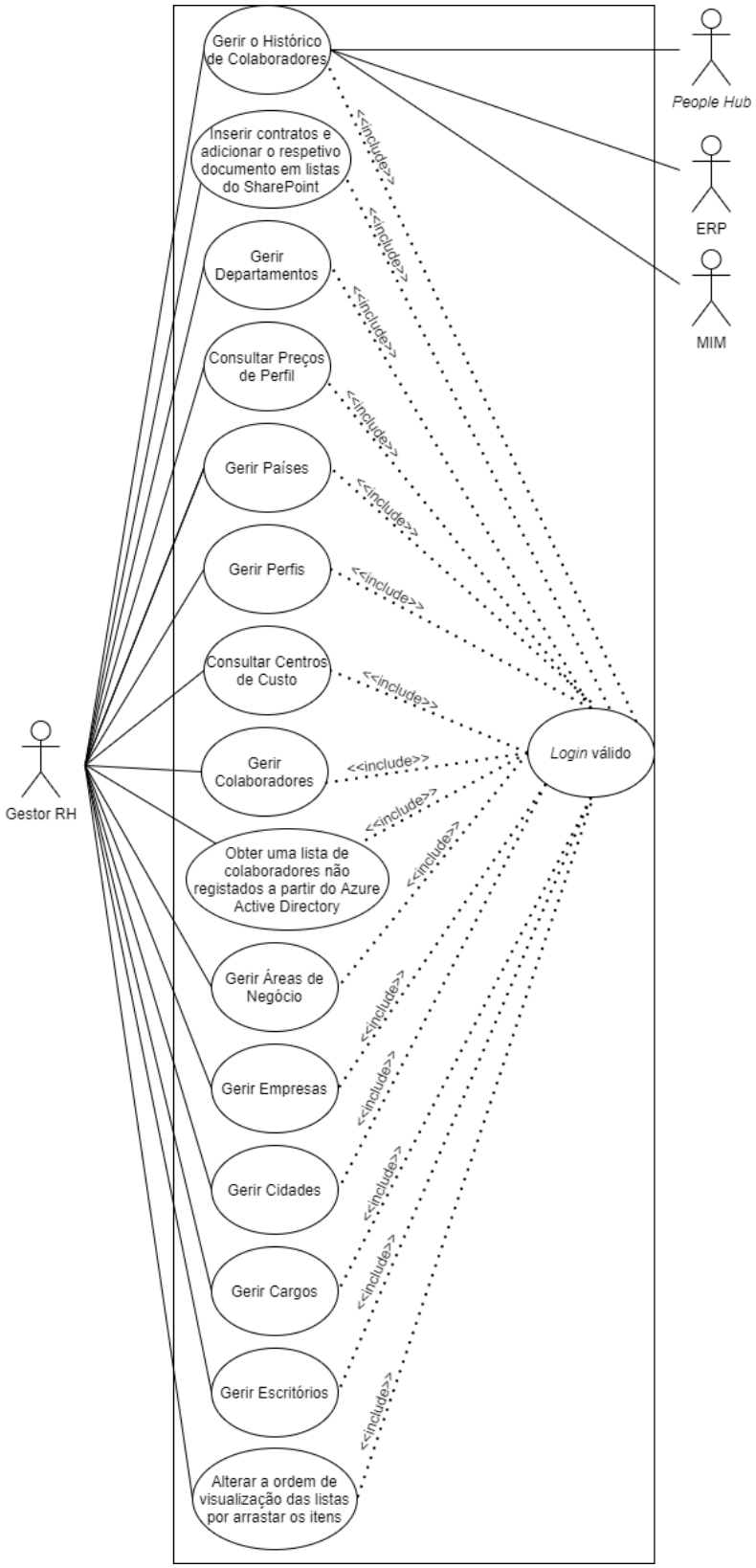


**Figura 6 - Diagrama de casos de uso**

## 4.5 Descrição dos casos de uso e diagramas de sequência

Uma descrição de um caso de uso representa detalhadamente e passo a passo a interação entre o ator e o sistema para um determinado caso e uso. Ela descreve os resultados de uma ação realizada e especifica os diferentes caminhos que podem ser seguidos por definir um caminho principal e um caminho alternativo.

A descrição será feita adotando o seguinte padrão:

- Nome: Nome do caso de uso;
- Descrição: Breve descrição do caso de uso em foco;
- Ator(es): Ator(es) que participam no caso de uso;
- Pré-Condição: Descreve condições que devem ser verdadeiras ou atividades que devem ser concluídas antes da execução do caso de uso;
- Caminho Principal: Descrição detalhada das ações do utilizador e respetivas respostas do sistema que ocorrerão durante a execução do caso de uso sob condições normais esperadas;
- Caminho Alternativo: Descrição de condições de erro que podem ocorrer durante a execução do caso de uso e como o sistema deve responder a essas condições

Nas tabelas 1 e 2 é apresentada a descrição para os casos de uso “Gerir o Histórico de Colaboradores (Adicionar)” e “Inserir contratos e adicionar o respetivo documento em listas do SharePoint”, respetivamente.

| **Nome** | **Gerir o Histórico de Colaboradores (Adicionar)** |
|---|---|
| **Descrição** | Caso de uso que trata de inserir um histórico de colaborador |
| **Atores** | Gestor de Recursos Humanos |
| **Pré-Condição** | *Login* válido |
| **Caminho Principal** | 1. O ator seleciona a opção "Criar novo histórico"<br>2. O sistema abre uma nova página com o formulário para a inserção dos dados<br>3. O ator seleciona um colaborador da lista<br>4. O sistema preenche o formulário com as informações do histórico mais recentes relativas ao colaborador escolhido, caso exista.<br>5. O ator preenche/altera os campos que deseja e submete.<br>6. O sistema guarda o histórico na base de dados e redireciona o ator para uma página de detalhes. |
| **Caminho Alternativo** | 4. A) Se algum campo não for preenchido ou inválido é mostrada uma mensagem de erro por baixo do campo.<br>B) Caso o histórico anterior não tenha data de fim de vigência é mostrado um novo pop-up para o ator atualizar essa mesma data. |

**Tabela 1 - Descrição do caso de uso "Gerir o Histórico de Colaboradores (Adicionar)"**

| | |
|---|---|
| **Nome** | **Inserir contratos e adicionar o respetivo documento em listas do *Sharepoint*** |
| **Descrição** | Caso de uso que trata de inserir contratos e efetuar o *upload* do respetivo documento para uma lista do *Sharepoint* |
| **Atores** | Gestor de Recursos Humanos |
| **Pré-Condição** | *Login* válido |
| **Caminho Principal** | 1. O ator seleciona a opção "Adicionar contrato"<br>2. O sistema abre uma nova janela com o formulário para a inserção dos dados<br>3. O ator seleciona a opção "Escolher documento"<br>4. O sistema abre um pop-up onde o ator poderá escolher o documento presente no sistema de gestão de ficheiros do sistema operativo.<br>5. O ator seleciona o ficheiro, preenche os restantes campos e submete.<br>6. O sistema guarda as informações na base de dados, faz upload do ficheiro para uma lista do *Sharepoint*, e mostra ao autor uma mensagem de sucesso. |
| **Caminho Alternativo** | 6. A) Se algum campo não for preenchido ou inválido é mostrada uma mensagem de erro por baixo do campo.<br>B) É mostrada uma mensagem de erro ao ator se houver qualquer outra falha no processo. |

**Tabela 2 - Descrição do caso de uso "Inserir contratos e adicionar o respetivo documento em listas do Sharepoint"**

Outra excelente forma de representar o comportamento de um caso de uso é através de diagramas de sequência. Como o nome indica, os diagramas de sequência descrevem como e em que ordem os objetos num sistema funcionam. As figuras 7 e 8 mostram os diagramas de sequência relativos aos casos de uso enunciados.

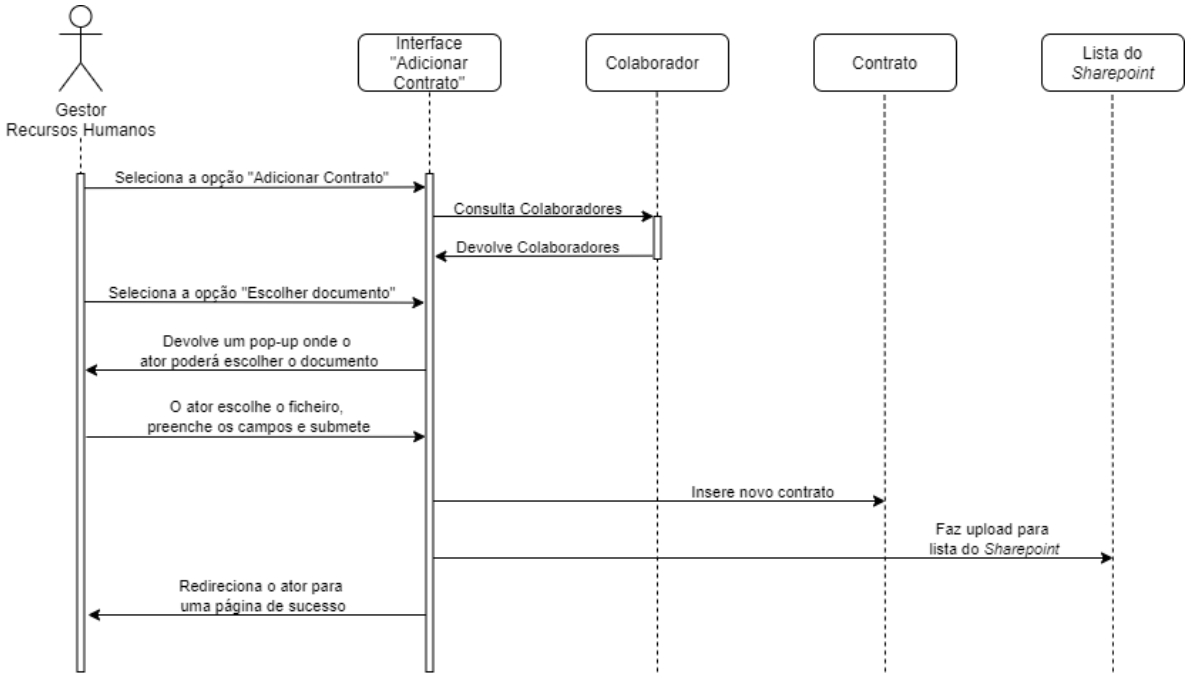


**Figura 7 - Diagrama de sequência do caso de uso "Inserir contratos e adicionar o respetivo documento em listas do Sharepoint"**

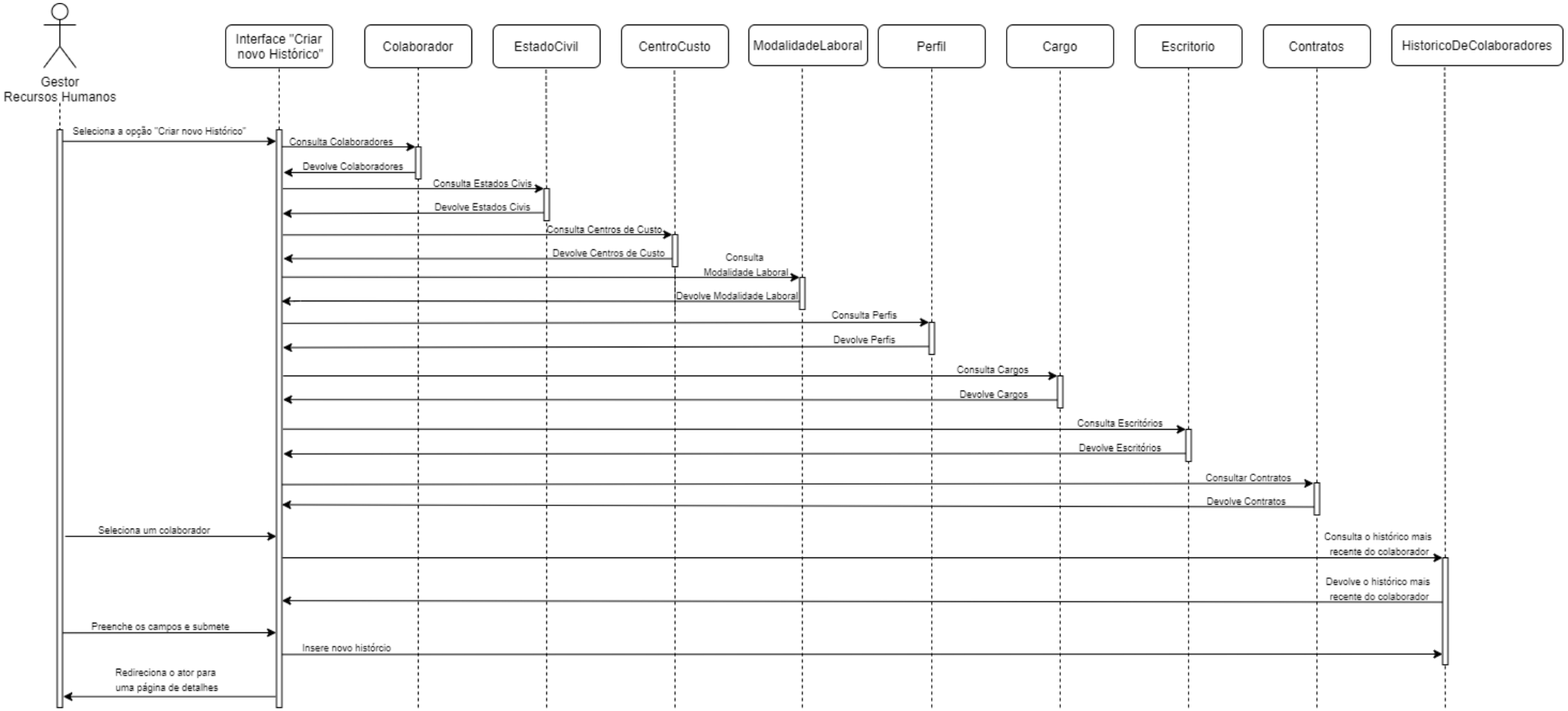


**Figura 8 - Diagrama de Sequência do caso de uso "Gerir o Histórico de Colaboradores (Adicionar)"**

## 4.6 Diagrama de classes

Um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Cada classe do diagrama representa uma tabela na base de dados e deve ter um nome (substantivo no singular), atributos e principais operações.

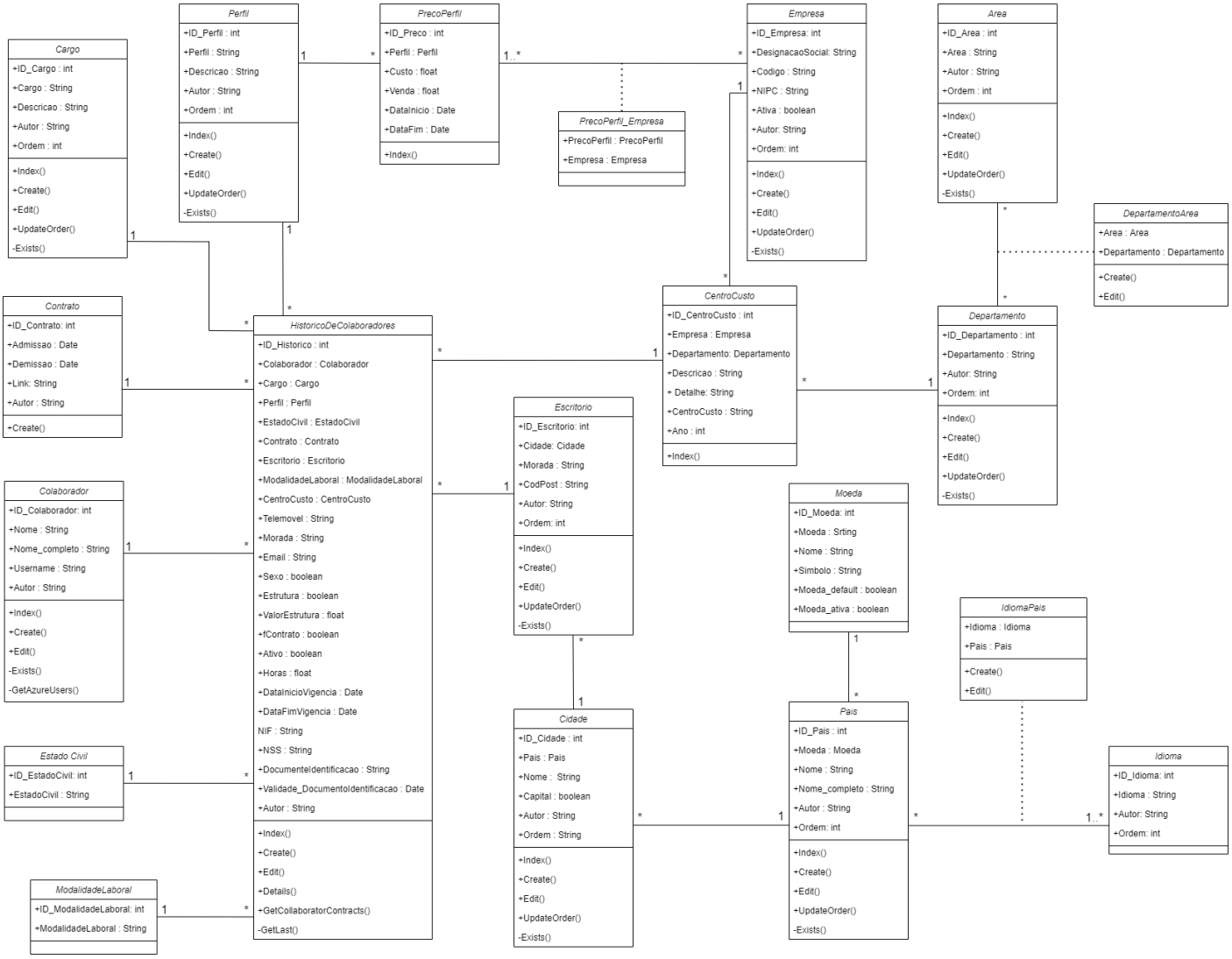


**Figura 9 - Diagrama de classes**

O diagrama de classes está representado pela figura 9. A classe "HistoricoDeColaboradores" é a classe principal. Ela usa dados de várias outras classes, que servem para fazer a sua manutenção. É digno de nota também que as classes em que são esperadas as operações "*Create*()" e/ou "*Edit*()" têm o atributo "Autor", de forma a guardar o responsável por tais operações.

# 5 Tecnologias

Para o desenvolvimento do projeto, foi necessário escolher as tecnologias e ferramentas que mais se adequavam às necessidades. Este capítulo irá apresentar sucintamente as tecnologias escolhidas.

- **Microsoft Teams**

  Microsoft Teams [13] é uma plataforma unificada de comunicação e colaboração que combina *chat*, videoconferências, armazenamento de ficheiros e integração de aplicações. O serviço integra-se no pacote do Office365 e apresenta extensões que podem ser integradas a produtos que não são da Microsoft. Todo o trabalho de colaboração e comunicação foi realizado a partir desta plataforma.

  O Microsoft Teams permite também incorporar aplicações *web* personalizadas nos seus separadores, através de um *iframe* que aponta para o domínio da aplicação desenvolvida. A figura 10 contém um exemplo de um separador personalizado.

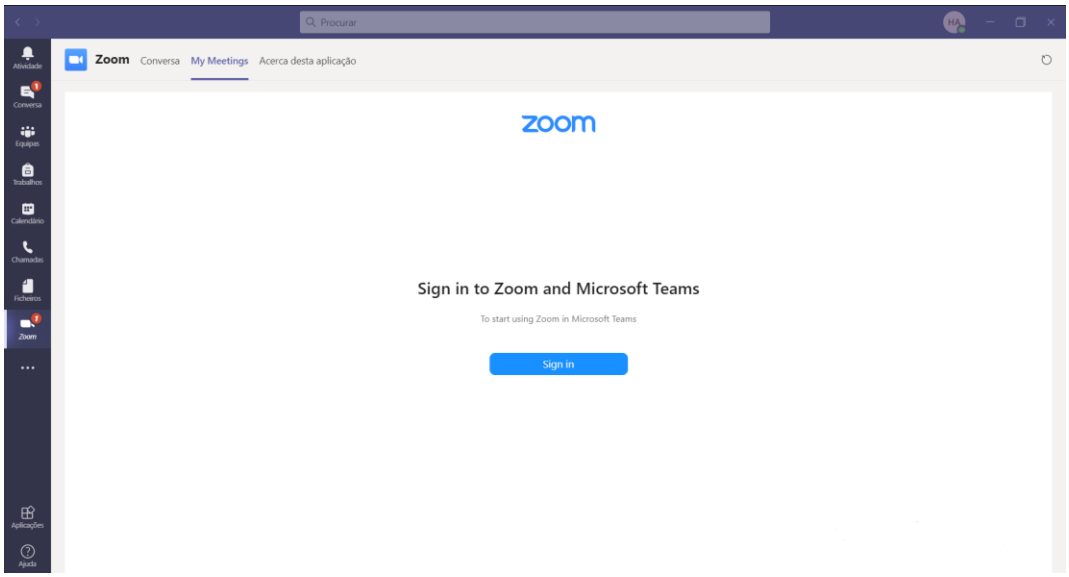


**Figura 10 - Separador no Microsoft Teams do Zoom**

- **SQL Server Management Studio**

  O SQL Server Management Studio (SSMS) é um IDE para gerir infraestruturas SQL. Ele fornece ferramentas para configurar, controlar e administrar instâncias do SQL Server e bases de dados. Todo o trabalho de modelação e desenho de base de dados e carregamento de tabelas foi feito a partir desta ferramenta.

- **Power Apps**

  O Power Apps [12] é um conjunto de aplicações, serviços, conectores e plataformas de dados que fornece um ambiente de programação rápida de aplicações empresariais. Esta plataforma permite que as aplicações criadas se liguem facilmente a dados empresariais armazenados em várias origens, como por exemplo o SharePoint, Microsoft 365, Dynamics 365, SQL Server, através de conectores disponíveis. A figura 11 mostra o ambiente de desenvolvimento do Power Apps.

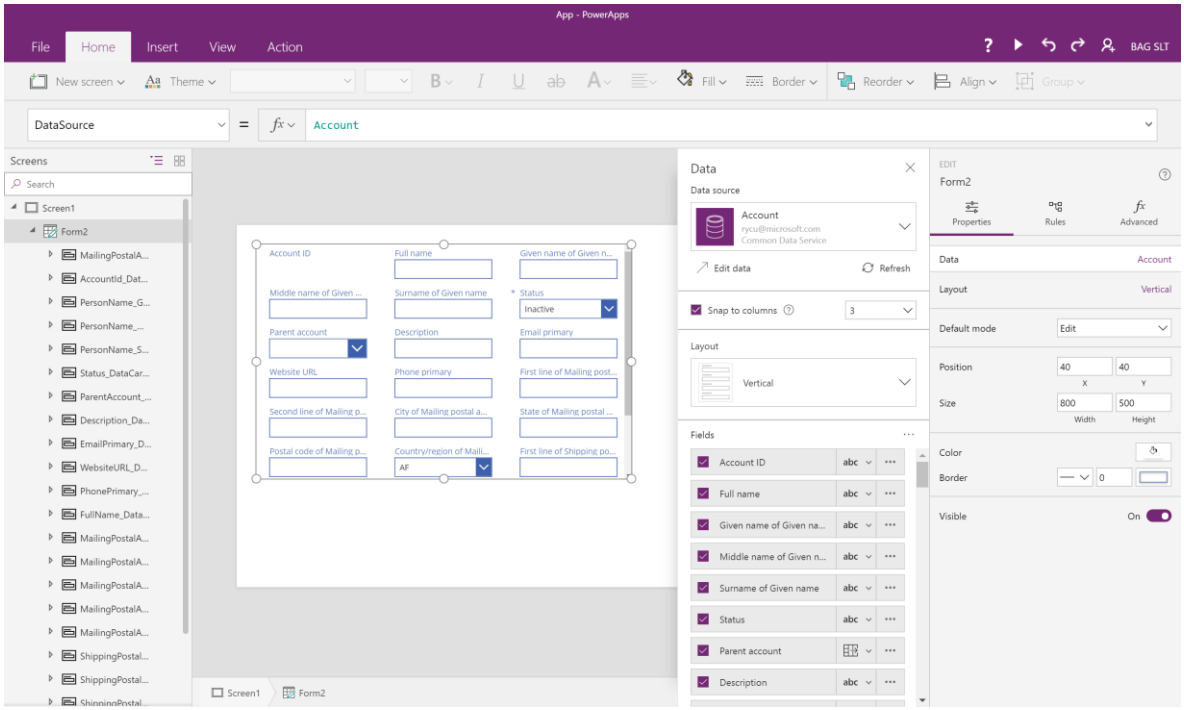


**Figura 11 - Ambiente de desenvolvimento do Power Apps**

O que difere o Power Apps de outras plataformas é que, embora possa ser usado por desenvolvedores, também pode ser usado por colaboradores não técnicos, dado que a maior parte da escrita de código é feita através de um mecanismo de *drag and drop.* Além disso, as aplicações desenvolvidas estão disponíveis em várias plataformas, incluindo *web*, iOS, Android e Windows [12].

Contudo, apesar de todas estas vantagens, o Power Apps traz consigo algumas limitações. Além de não permitir mais que um desenvolvedor a trabalhar simultaneamente na mesma aplicação, só consegue receber apenas 500 registos da origem de dados por defeito, podendo este número ser estendido até no máximo 2000 a troco de performance [14]. O custo de alguns conectores *premium*, nomeadamente do SQL Server, pode ser considerado também uma limitação.

- **Power Automate**

  O Power Automate [15] é um serviço baseado em *cloud* que permite criar fluxos de trabalho que automatizam processos e tarefas manuais em diferentes aplicações e serviços. Pode ser usado para sincronizar ficheiros entre aplicações, receber notificações, recolher dados, entre outros. No caso do projeto, recorreu-se a este serviço para realizar o upload de ficheiros para listas do SharePoint a partir do Power Apps.

- **SharePoint**

  O SharePoint [16] é uma plataforma de aplicações *web* da Microsoft, com utilização na criação de portais e intranets empresariais, gestão de conteúdos, gestão documental e publicação de aplicações *web*. É acessível através de quase qualquer dispositivo, sendo apenas necessário um *browser*, e facilmente integrado com outras ferramentas do Microsoft Office como o Word, Excel, PowerPoint e Power Apps. Dentro do SharePoint, aproveitou-se a versatilidade das suas listas que permitem recolher e armazenar diferentes tipos de dados, facilitando a organização da informação.

- **C#**

  C# é uma linguagem de programação orientada a objetos, desenvolvida pela Microsoft no ano 2000, e possui características semelhantes ao Java e C++. É usada por várias razões, mas a sua popularidade deve-se principalmente pelos seus pontos fortes em desenvolver aplicações *desktop* para Windows, jogos e aplicações *web* [17].

- **ASP.NET**

  ASP.NET [18] é uma *framework open source* criada pela Microsoft para a construção de aplicações *web* e *web services* com plataforma .NET, ampliando-a com ferramentas e bibliotecas específicas para esse fim, e cujo código pode ser escrito em C#.

  Dentro do ASP.NET optou-se pela versão ASP.NET Core [18], que é a versão multiplataforma do ASP.NET, que lhe permite correr em diversos sistemas operativos. Esta *framework* destaca-se pela sua performance e pela possibilidade de utilização de todo o ecossistema de bibliotecas e pacotes do .NET.

- **HTML**

  *Hypertext Markup Language*, ou vulgarmente chamada como HTML, é a linguagem de marcação padrão para páginas *web* [19]. Os elementos de uma página HTML são representados por *tags* e podem ser assistidos por outras tecnologias, como CSS e JavaScript.

- **CSS**

  Enquanto HTML é usado para estruturar os elementos uma página, o CSS vem especificar o estilo desses elementos, como cores, fontes, posição e tamanho. Além disso, permite também adicionar efeitos e animações. Geralmente esta

personalização é feita em ficheiros externos, separando completamente o código CSS do HTML, por questões de organização e reutilização.

- **JavaScript**

  O JavaScript é uma linguagem de programação do lado cliente, ou seja, é processada pelo próprio *browser*. Com o JavaScript é possível proporcionar uma maior interatividade com o utilizador, uma vez que ele cria e/ou altera dinamicamente o conteúdo de uma página *web*. O JavaScript é também uma linguagem de programação orientada a objetos, ou seja, ela trata todos os elementos da página como objetos distintos [20].

- **Git**

  O Git é um sistema de controlo de versões *open source* que permite armazenar código em repositórios, acompanhar mudanças e reverter para uma versão anterior quando necessário. Ele dá a possibilidade de criar cópias do código (*branches*) que podem ser trabalhadas em paralelo com a versão principal (*master*). Cada alteração é guardada numa nova versão através de *commits*. Após todas as alterações estarem concluídas num determinado *branch*, faz-se *merge* de volta para o *master*. A figura 12 ilustra o funcionamento do Git.

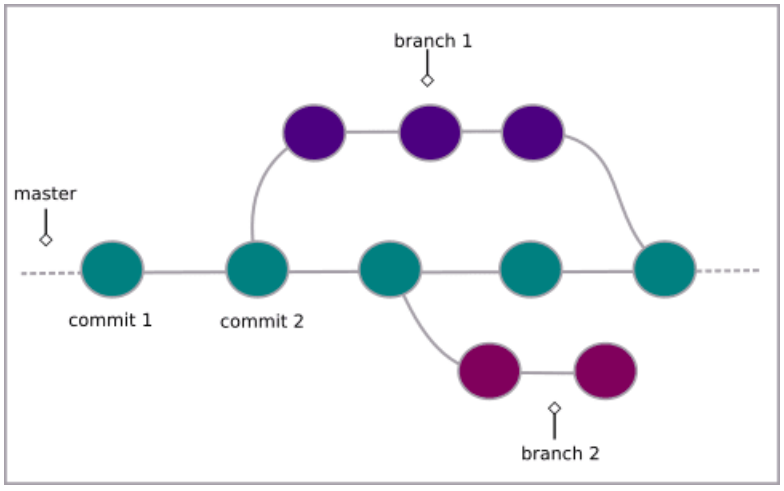


**Figura 12 - Funcionamento do Git**

# 6 Implementação

Este capítulo irá descrever todo o processo de implementação da solução. Numa primeira fase observou-se que as tecnologias usadas até então, não atendiam às necessidades. Por essa razão, procedeu-se à adoção de novas tecnologias, o que permitiu uma maior dimensão dos dados e melhor desempenho da aplicação. Dessa forma, este capítulo descreverá também como se percebeu que a primeira abordagem não era a mais indicada e as razões pelas quais se decidiu seguir outra direção.

## 6.1 Desenho e modelação da base de dados

O processo de implementação começou com o desenho e modelação da base de dados. Esta é uma fase importante no ciclo de vida de desenvolvimento de aplicações, dado que permite que a aplicação funcione bem desde o seu início.

Assim, o processo começou com a análise do ficheiro Excel usado para fazer a manutenção do histórico de colaboradores (mostrado no capítulo 2.2). Esta análise permitiu o levantamento de algumas tabelas e atributos. Para complementar, foram tidas várias conversas com os potenciais utilizadores da base de dados, de forma a compreender e registar os seus requisitos. Depois de várias iterações, e de acordo com a análise efetuada, surgiu o modelo Entidade-Relacionamento final (figura 13) e as tabelas foram criadas no SQL Server.

Visto que alguma informação já se encontrava armazenada noutras bases de dados, a última fase desta etapa passou pela criação de procedimentos para inserir os registos nas tabelas criadas. O código referente a alguns destes procedimentos pode ser encontrado nos anexos 9.2.1 e 9.2.2.

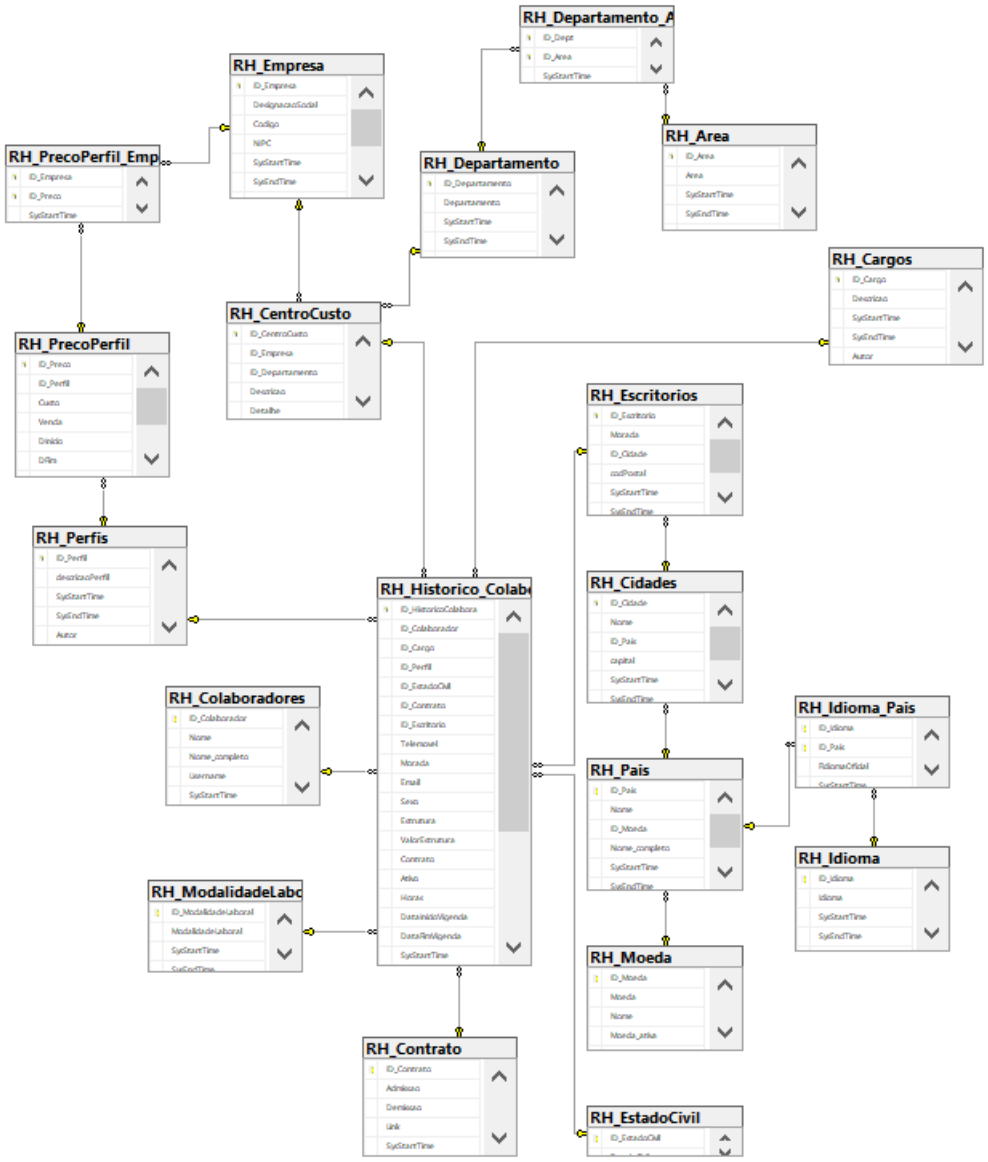


**Figura 13 - Modelo ER**

## 6.2 Implementação com o Power Apps

Numa primeira abordagem, optou-se por desenvolver a aplicação na plataforma Power Apps pela sua facilidade de aprendizagem e rápido desenvolvimento.

Visando a integração com fontes externas de informação, era essencial que a informação fosse armazenada numa base de dados. Apesar de a plataforma oferecer um conector para o SQL Server, o seu custo [21] trazia despesas incomportáveis para a

Armis. Por essa razão, recomendou-se a utilização listas de SharePoint como origem de dados para o Power Apps e a construção de um *web service* para fazer o fluxo para a base de dados. A figura 14 representa a arquitetura descrita.

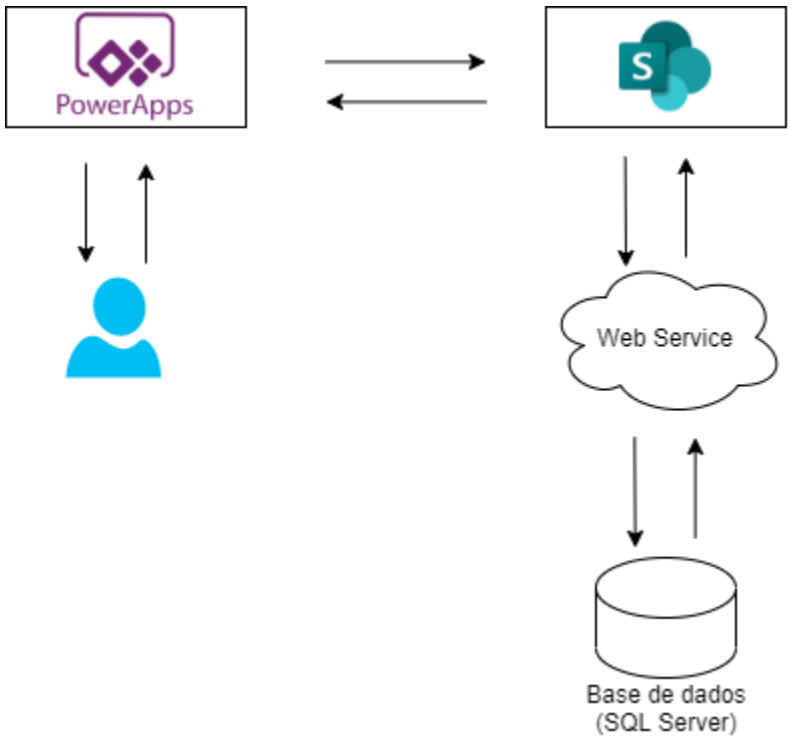


**Figura 14 - Arquitetura da solução**

Antes de proceder à elaboração da aplicação, foi necessário o agendamento de uma reunião com a equipa de *design*. Nesta reunião, depois de explicado o contexto da aplicação, foram recebidas instruções sobre como os vários elementos deveriam estar distribuídos e organizados de modo a tornar a experiência mais *user-friendly*. Foi disponibilizado também um ficheiro do Adobe XD que apresentava todos os detalhes dos ecrãs: tipos e tamanhos de letra, ícones, cores, imagens, entre outros.

Posterirormente, inicializou-se a construção da aplicação no Power Apps e do *web service* em C# para o fluxo de dados com a base de dados. A figura 15 apresenta a interface que permitia fazer a gestão das cidades. Aqui, não só era possível consultar a lista de cidades completa, como também era possível a pesquisa e aplicação de filtros.

A criação de uma cidade nova (figura 16) e edição de existente são feitas a partir deste mesmo ecrã através de um *pop-up*. O mesmo se aplica para as restantes interfaces.

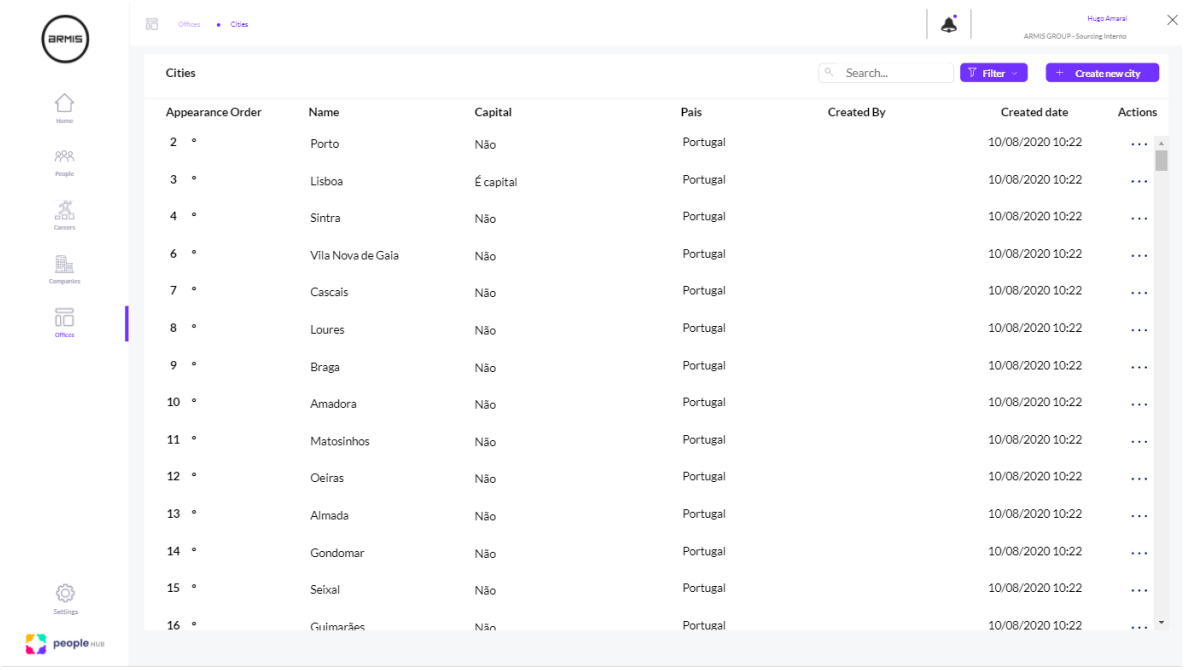


**Figura 15 - Interface para gestão de cidades**

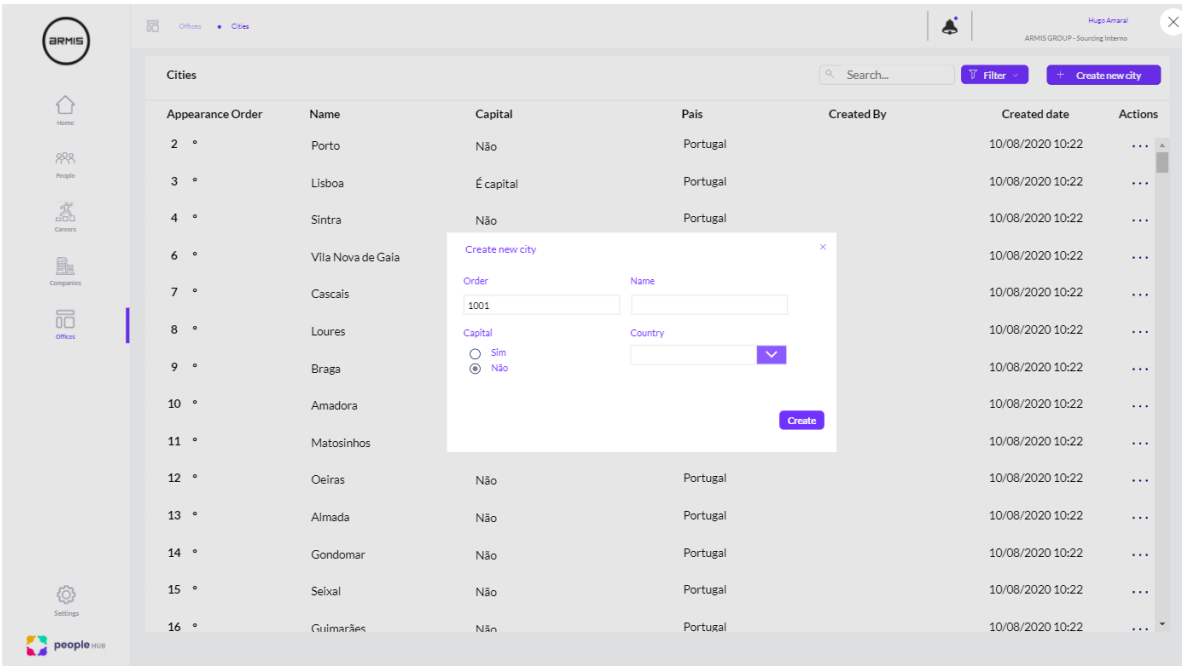


**Figura 16 - Pop-up para criação de uma nova cidade**

Contudo, como foi mencionado no capítulo 5.3, existe uma limitação na plataforma que apenas lhe permite consultar no máximo 2000 registos, ignorando os remanescentes. Isso representa um grave obstáculo ao Histórico de Colaboradores, não só porque a longo prazo a tabela que armazena informações relativas ao histórico irá eventualmente ultrapassar esse valor, como também porque já existiam tabelas cujo número de registos excedia os 2000. Assim, junto com o supervisor, resolveu-se trocar a tecnologia a fim de contornar esta limitação. Além de permitir uma maior de dimensão dos dados, esta troca veio também melhorar o desempenho e aparência da aplicação e trazer mais flexibilidade.

## 6.3 Implementação com o ASP.NET Core

Um dos principais objetivos do projeto relacionava-se com a implementação da solução no Microsoft Teams e, portanto, a escolha da tecnologia desta segunda abordagem ao projeto deveria permitir fazê-lo. Como o Microsoft Teams permite integrar aplicações *web* através de um *iframe* que aponta para o domínio da aplicação, optou-se pela criação de uma aplicação *web* com o ASP.NET Core seguindo o modelo MVC.

### 6.3.1 Arquitetura MVC

A arquitetura MVC [22], sigla para *model-view-controller*, separa a aplicação em três principais componentes lógicos: o modelo (*model*), a vista (*view*) e o controlador (*controller*). Por estar organizada desta forma, esta arquitetura ajuda na escrita de código mais organizado e, consequentemente, mais sustentável. Cada um destes componentes é construído para lidar com aspetos específicos da aplicação:

- Modelo – representa a estrutura de dados da aplicação. É responsável pela leitura e escrita de dados além de conter as regras de validação, ou seja, está ligado à manipulação de dados.

- Vista – é a parte que trata de apresentar os resultados de forma visual aos pedidos do utilizador.

- Controlador – os pedidos são encaminhados para o controlador, o qual é responsável por trabalhar com o modelo para executar ações pedidas pelo utilizador. O controlador responde a esses pedidos escolhendo a vista para mostrar ao utilizador, juntamente com os dados necessários.

A figura 17 ilustra o funcionamento da arquitetura MVC.

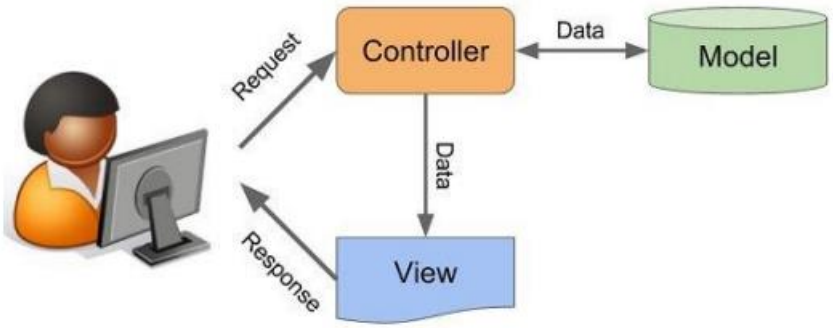


**Figura 17 - Arquitetura MVC**

## 6.3.2 Criação de um separador no Microsoft Teams

Os separadores são uma maneira simples de exibir conteúdo no Microsoft Teams, essencialmente por incorporar uma página *web*. Existem dois tipos de separadores – equipa e pessoal. Um separador para equipas fornece conteúdo para canais e *chats* em grupo. Por outro lado, um separador pessoal tem como alvo utilizadores individuais. Este tipo de separador pode ser fixo na barra esquerda de navegação para fácil acesso e oferece uma experiência mais próxima de *websites* tradicionais. Por estas razões, optou-se pela criação de um separador pessoal.

Um separador é criado no Microsoft Teams através de um ficheiro JSON chamado *manifest* [23]. Este ficheiro, que deve estar de acordo com a estrutura disponibilizada pela Microsoft, contém informação que permite ao Microsoft Teams apresentar

corretamente a aplicação aos utilizadores. Para auxiliar neste processo, o Microsoft Teams oferece uma ferramenta, o App Studio (figura 18), que simplifica a criação do *manifest*, permitindo especificar os detalhes da aplicação (como nome, descrição, desenvolvedores, ícones) e os recursos. Estando este ficheiro criado, ele é facilmente carregado para o Microsoft Teams para testes ou publicação.

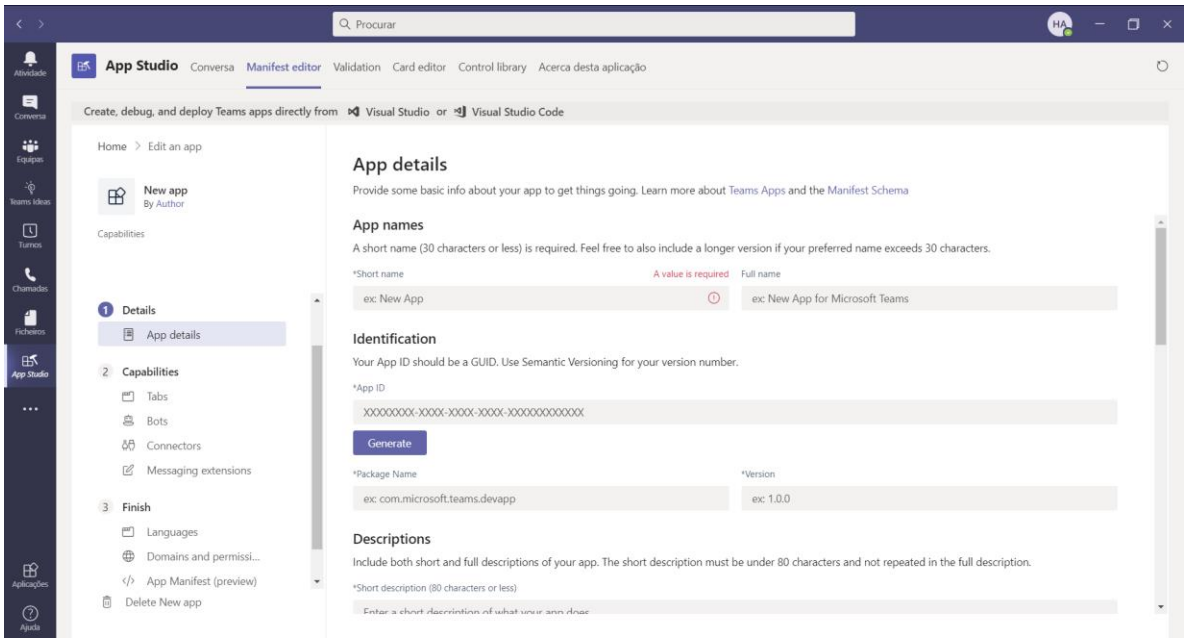


**Figura 18 - App Studio**

### 6.3.3 Autenticação

Muitas das aplicações que se podem usar dentro do Microsoft Teams requerem autenticação e autorização para obter acesso ao serviço. Ora, visto que o Histórico de Colaboradores é uma ferramenta que visa apenas ser usada pelo departamento de recursos humanos da Armis, optou-se por usar a autenticação através do Azure AD, onde os colaboradores estão registados e usam essa mesma conta para aceder ao Microsoft Teams.

O Azure AD, como a maioria dos fornecedores de identidade, não permite que o seu conteúdo seja colocado num *iframe* [24]. Isto significa que é necessário adicionar um *pop-up* para hospedar o fonecedor de identidade. Por essa razão, o fluxo de autenticação

não deve ser iniciado automaticamente, mas sim ser desencadeado por uma ação do utilizador, caso contrário o bloqueador de *pop-ups* irá disparar. Para isso, foi adicionado um botão de *login* à página inicial para que o utilizador se autentique antes de realizar qualquer ação. Está página pode ser vista na figura 19.

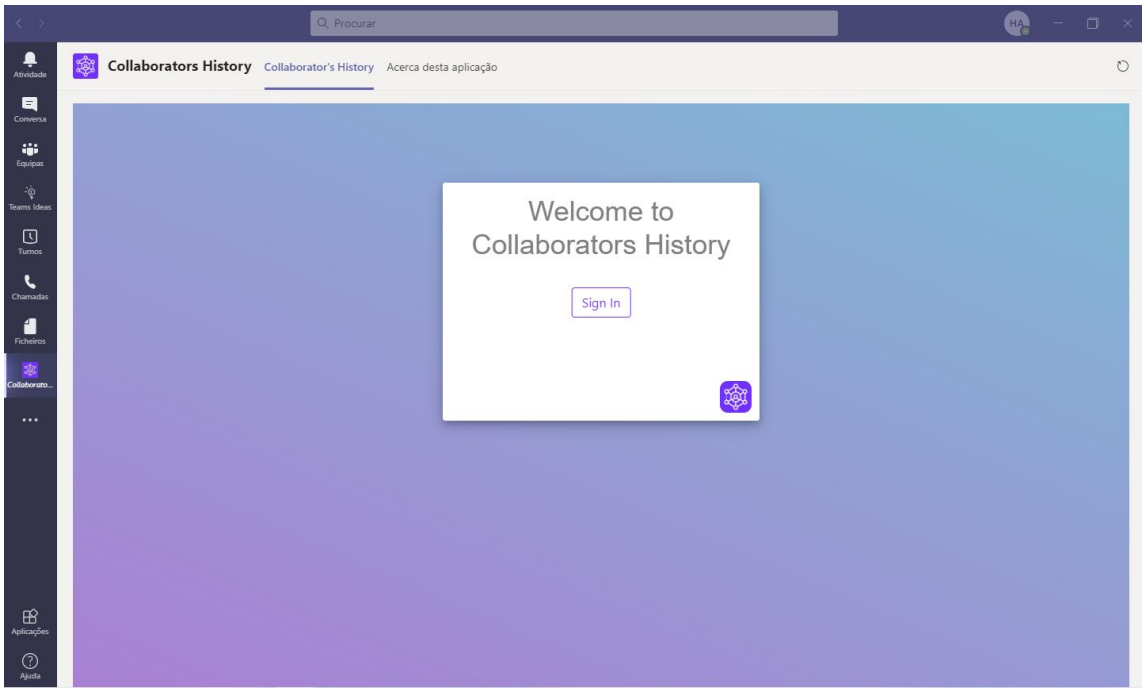


**Figura 19 - Página de autenticação**

O Azure AD usa o padrão de autenticação OAuth2, assim como muitos fornecedores de identidade. Para facilitar esse fluxo, a Microsoft disponibiliza um SDK JavaScript para o Microsoft Teams. Dessa forma, o fluxo de autenticação é o seguinte [24]:

1. O utilizador interage com a página, normalmente com um botão rotulado "*Login*" ou "*Sign in*".
2. A página constrói o URL para a página inicial da autenticação. Para simplificar a experiência de autenticação para o utilizador, o parâmetro "login_hint" é definido como o seu *email*. Dessa forma, não é necessário preencher o campo do *email* e, se ele tiver sido autenticado recentemente, o Azure AD usará as credenciais armazenadas em cache. Se esse for o caso, o *pop-up* apenas piscará brevemente e desaparecerá.

3. A página chama o método `microsoftTeams.authentication.authenticate()` e regista as funções `successCallback` e `failureCallback.`
4. O Microsoft Teams abre a página de início de autenticação num *pop-up*. A página gera então uma *string* aleatória, guarda-a para validação futura e redireciona o utilizador para o *endpoint* do fonecedor de identidade.
5. No *site* do fornecedor, o utilizador insere as suas credenciais e permite acesso à página.
6. O fornecedor redireciona o utilizador para a página de fim de autenticação com um *token*.
7. A página de fim de autenticação verifica se a *string* aleatória gerada corresponde ao valor guardado anteriormente, e chama o método `microsoftTeams.authentication.notifySuccess()`, que por sua vez chama a função `successCallback` registada no passo 3.
8. O pop-up é fechado e o *token* recebido é guardado nas *cookies*.
9. Este *token* é enviado para o servidor e validado. Se for válido, o utilizador é autenticado.

A figura 20 representa graficamente o fluxo de autenticação descrito.

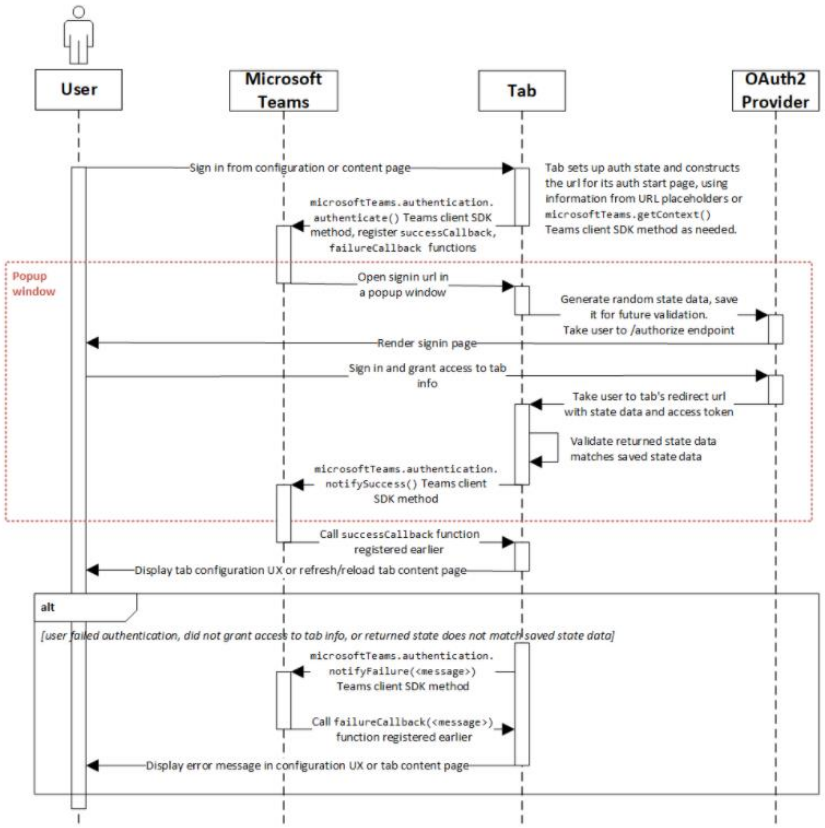


**Figura 20 - Diagrama de Sequência do fluxo de autenticação no Teams. Adaptado de [24]**

### 6.3.4 Gestão de tabelas corporativas

No projeto, as tabelas corporativas representam uma grande parte da aplicação. São elas que, direta ou indiretamente, fornecem informações que permitem fazer a manutenção do histórico de colaboradores. A figura 21 retrata a interface que possibilita a gestão de uma destas tabelas, especificamente das cidades [2].

[2] Para efeitos de demonstração, apenas se mostra as interfaces relativas ao módulo das cidades, dado que os demais são idênticos. A lista dos restantes módulos pode ser encontrada no anexo 9.1.

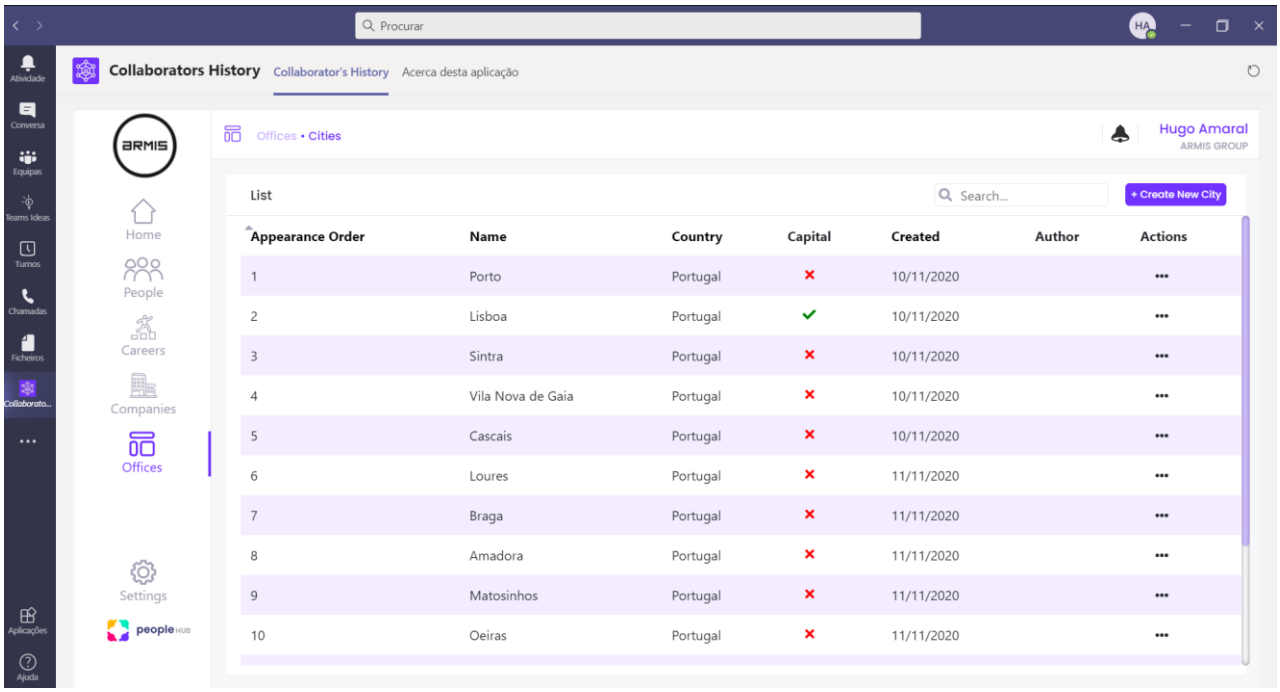


**Figura 21 - Interface para gestão das cidades**

Salvo algumas exceções, todas estas tabelas tinham uma coluna que representava a sua ordem de aparecimento. Esse valor pode ser alterado arrastando o item para a posição desejada. Esta ação teria de alterar não só a valor da ordem do registo pretendido, como também o valor dos registos abaixo da posição desejada.

Por defeito, quando uma página é aberta, a lista vem ordenada pela ordem de aparecimento. Para aperfeiçoar a consulta, ela pode ser ordenada por outros campos, ascendente ou descendentemente, carregando no cabeçalho pretendido: um *click* para ascendente e dois para descendente. A figura 22 mostra a lista das cidades ordenada pelo nome, de forma descendente. Além disso, é possível também fazer pesquisas por meio da barra colocada para este efeito.

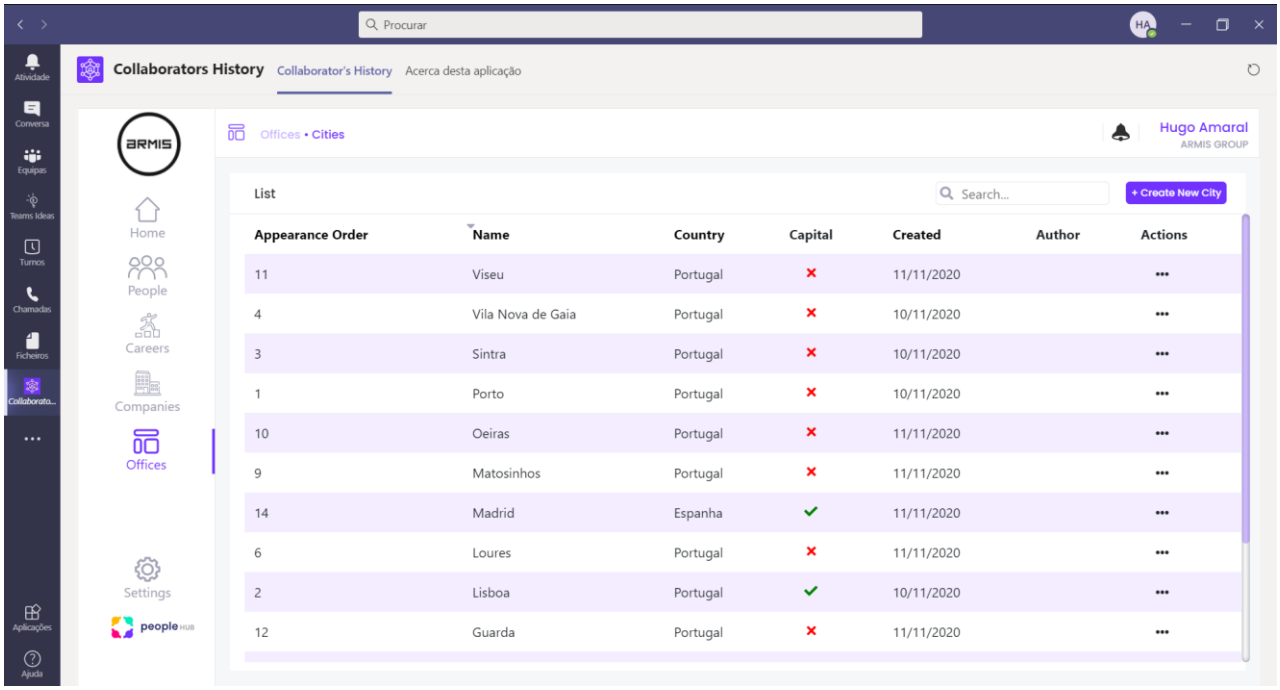


**Figura 22 - Lista de cidades ordenada por nome, descendentemente**

Quanto ao formulário de criação, optou-se por implementar o método utilizado na abordagem anterior: um *pop-up* (figura 23). Desta forma evita-se a navegação para outra página com este propósito, melhorando consideravelmente o desempenho da aplicação. O mesmo vale para os formulários de edição.

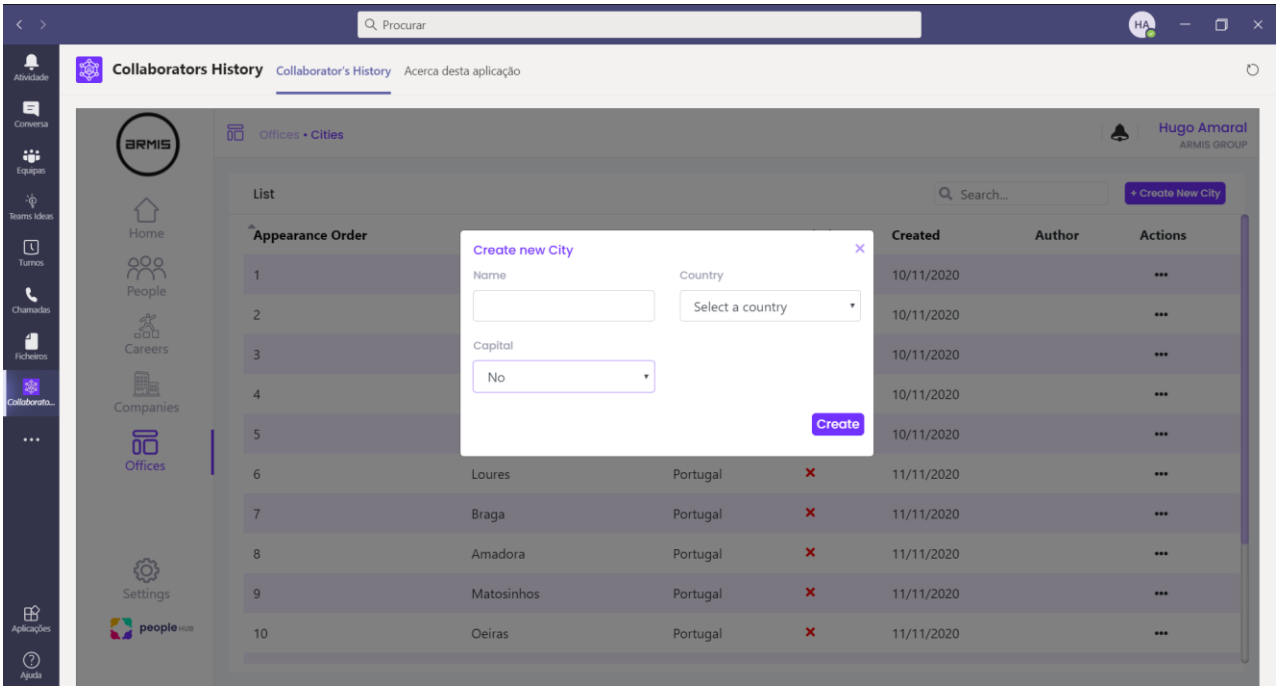


**Figura 23 - *Pop-up* para criação de uma cidade**

Das tabelas corporativas, destaca-se ainda o módulo dos contratos. Este difere dos demais, dado que a sua gestão envolve também o *upload* de ficheiros para uma lista do SharePoint.

O uso da biblioteca CSOM do .NET tornou possível a sua implementação. Esta biblioteca permite consultar, atualizar e gerir dados no SharePoint. Porém, para aceder ao SharePoint, é requerido o fluxo de autenticação tradicional do Azure (sem ser através de um pop-up). Por essa razão, quando a opção para adicionar contrato é selecionada, é aberta uma nova janela do *browser* que autentica o utilizador e redireciona-o para o formulário de criação de contrato (figura 24).

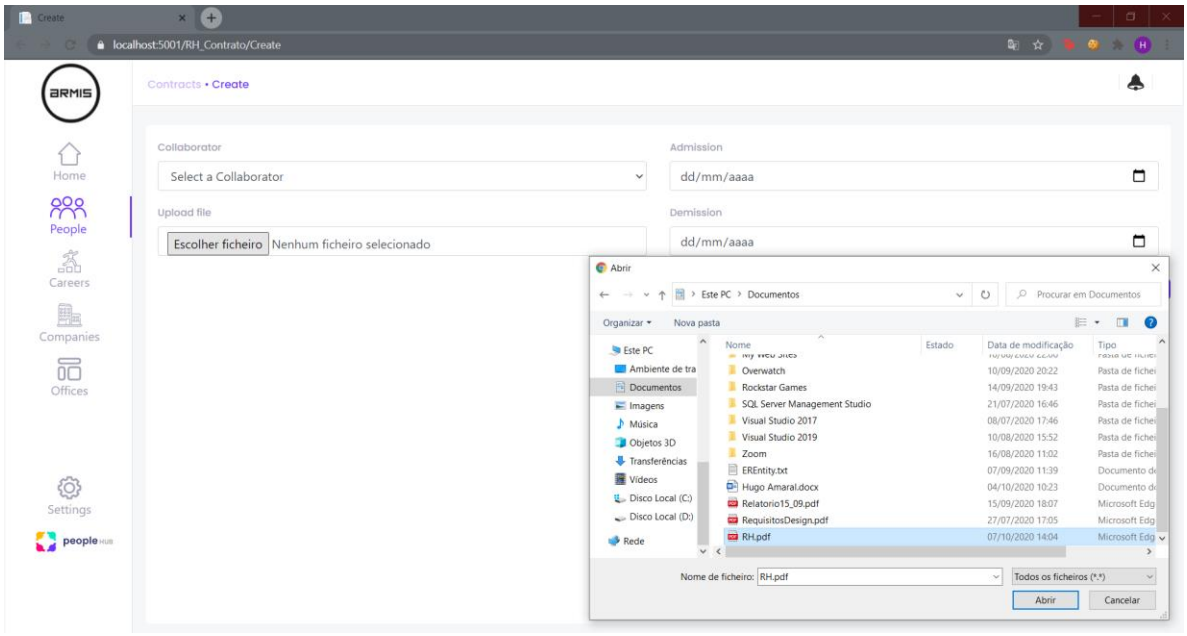


**Figura 24 - Formulário de criação de contratos**

Quando um contrato é criado, ele é inserido dentro de uma pasta com o nome do colaborador na lista do SharePoint para melhor organização dos ficheiros. Se o colaborador ainda não tiver uma pasta, ela é criada. Por sua vez, o ficheiro é renomeado para uma junção da data de admissão com o nome completo do colaborador, distinguindo os vários contratos dentro da mesma pasta. A figura 25 apresenta uma pasta de um colaborador com contratos na lista do SharePoint. A nível de base de dados é guardado um link que aponta para o contrato inserido, além da restante informação.

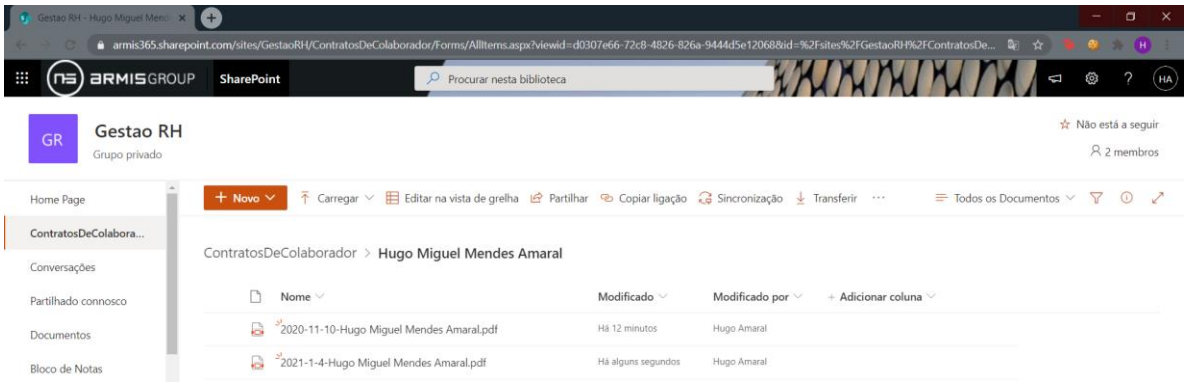


**Figura 25 - Pasta de um colaborador com contratos na lista do SharePoint (ficheiros fictícios)**

### 6.3.5 Histórico de Colaboradores

O histórico de colaboradores é o foco principal do projeto. Todos os restantes módulos servem para fazer a sua manutenção. Assim sendo, é essencial que o processo de consulta, adição e alteração de um histórico seja o mais otimizado e eficiente possível. Foi com isso bem presente que se implementou o histórico de colaboradores. A figura 26 mostra a interface para a sua gestão.

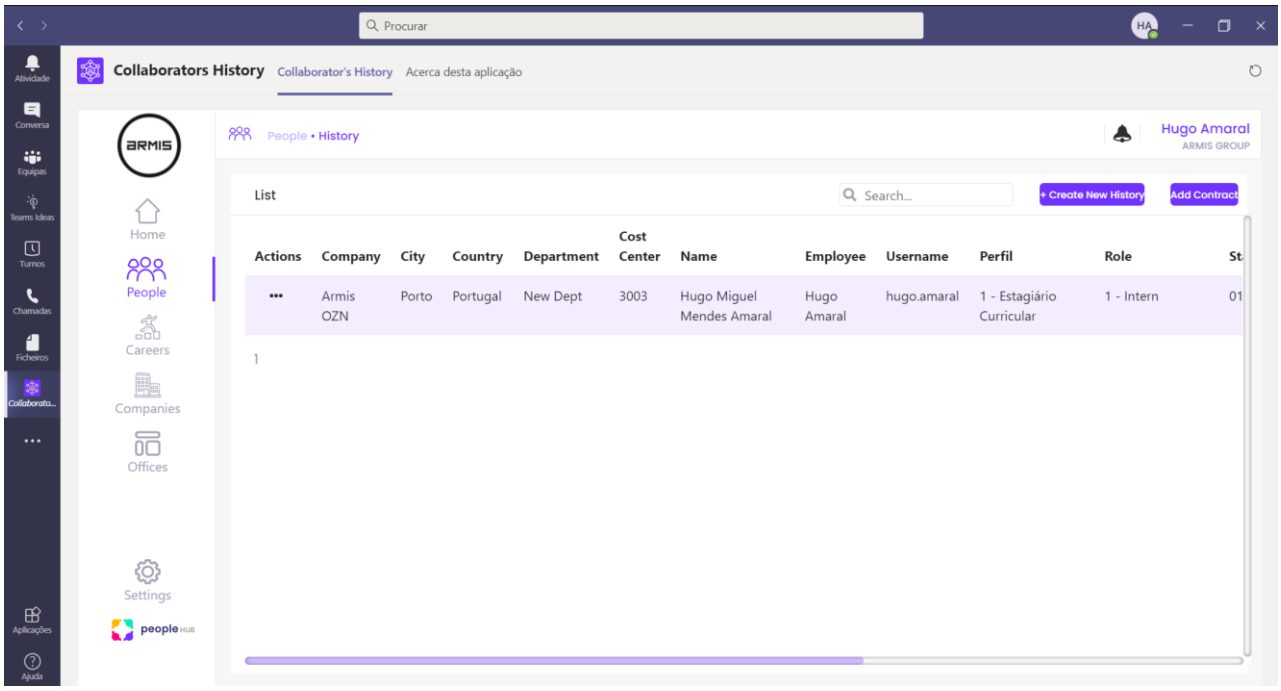


**Figura 26 - Interface para gestão do histórico de colaboradores (dados fictícios)**

Como nesta lista a ordem dos registos não tem qualquer relevância, eles não podem ser arrastados, à semelhança dos outros módulos. Logo, ao invés de ter presente na página uma lista completa, foi implementado um mecanismo de paginação. Isto fornece aos utilizadores opções de navegação adicionais. Além disso, visto que o histórico possui um elevado número de registos, este método melhora substancialmente o desempenho da aplicação uma vez que apenas são carregados um número limitado de registos de cada vez.

Para tornar a pesquisa do histórico mais eficaz, além de uma barra de pesquisa, foram colocados filtros que aparecerem ao se carregar no cabeçalho da lista, como mostra a

figura 27. Cada filtro pode aplicar uma ou mais opções (por exemplo Lisboa e Porto nas cidades).

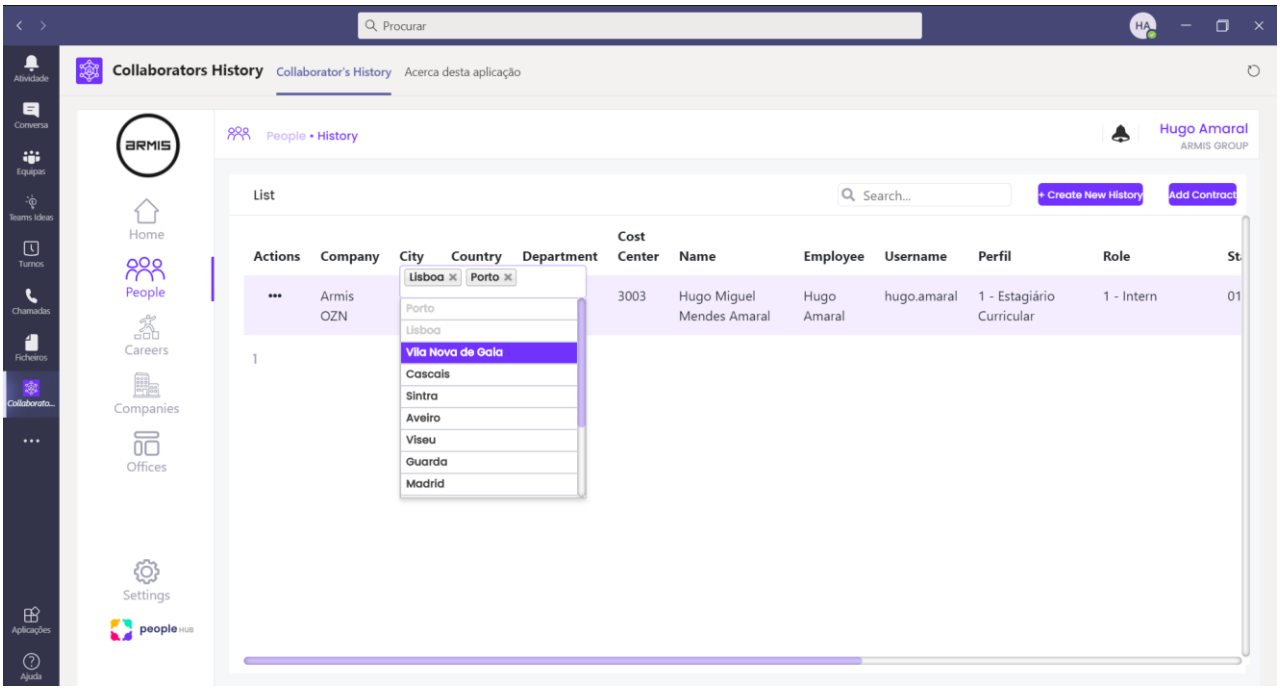


**Figura 27 - Filtro no cabeçalho do histórico (cidades)**

Em relação aos formulários para criação e edição, decidiu-se colocá-los numa página separada, em vez de num *pop-up*, devido à sua dimensão. Para melhor organização, estes formulários foram divididos em duas partes:

- Informação pessoal – Informação pessoal do colaborador raramente sujeita a alterações, no entanto, exista essa possibilidade (como estado civil);
- Informação empresarial – Informação que diz respeito ao percurso do colaborador na empresa regularmente sujeita a alterações (como o cargo).

Para facilitar a criação de um novo histórico, o formulário é preenchido automaticamente com as informações mais recentes do colaborador escolhido no *dropdown*. Dessa forma, evita-se o preenchimento de grande parte dos campos e a experiência de criação é melhorada exponencialmente. A figura 28 mostra o formulário de criação de um novo histórico.

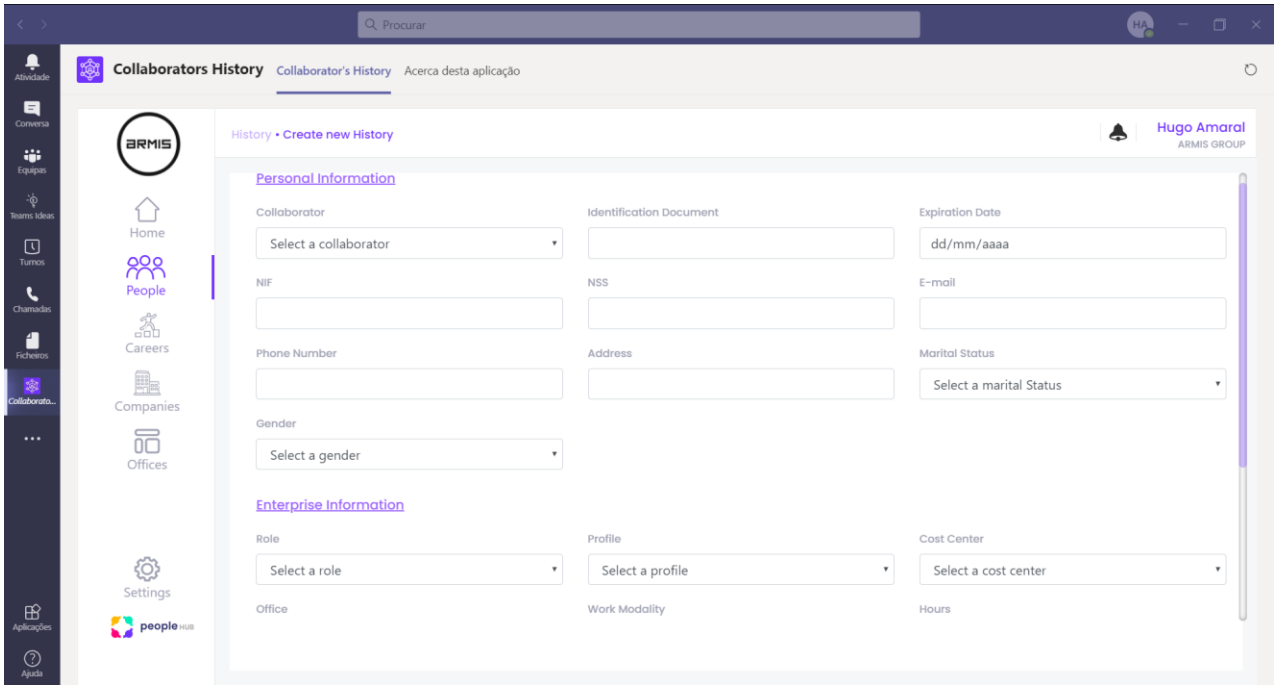


**Figura 28 - Parte do formulário de criação de novo histórico**

Por vezes, ao criar-se um histórico, a sua data de fim de vigência é ainda desconhecida. Assim, é possível adicionar um histórico sem a data de fim de vigência. No entanto, aquando a criação do próximo se essa mesma data estiver vazia, será exibido um *pop-up* que obriga o utilizador a inserir uma data de conclusão do último histórico (figura 29). Isso não só facilita ao utilizador a gestão das datas, como previne que se deixe um histórico em "aberto".

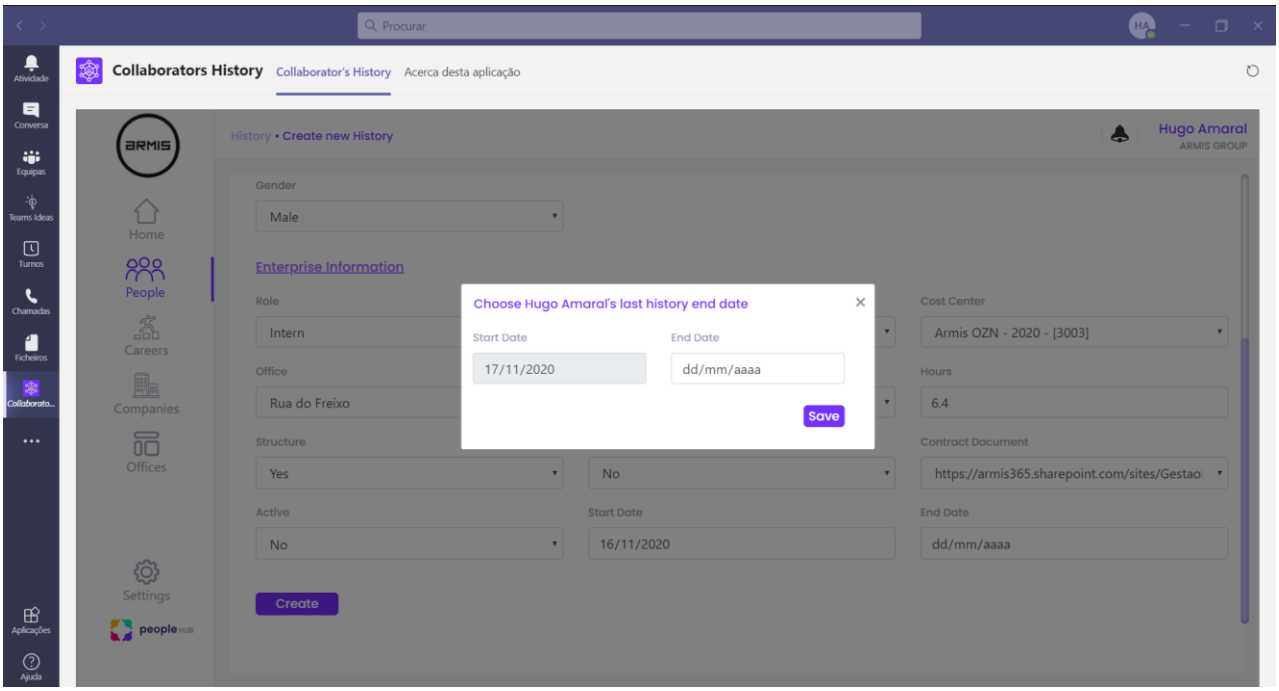


**Figura 29 - *Pop-up* para inserir a data de fim de vigência do último histórico, caso esteja vazia**

Por fim, quando a criação de um novo histórico é feita com sucesso, o utilizador é redirecionado para uma página de detalhes, visível na figura 30. O código completo do controlador do módulo do histórico de colaboradores encontra-se no anexo 9.2.3.

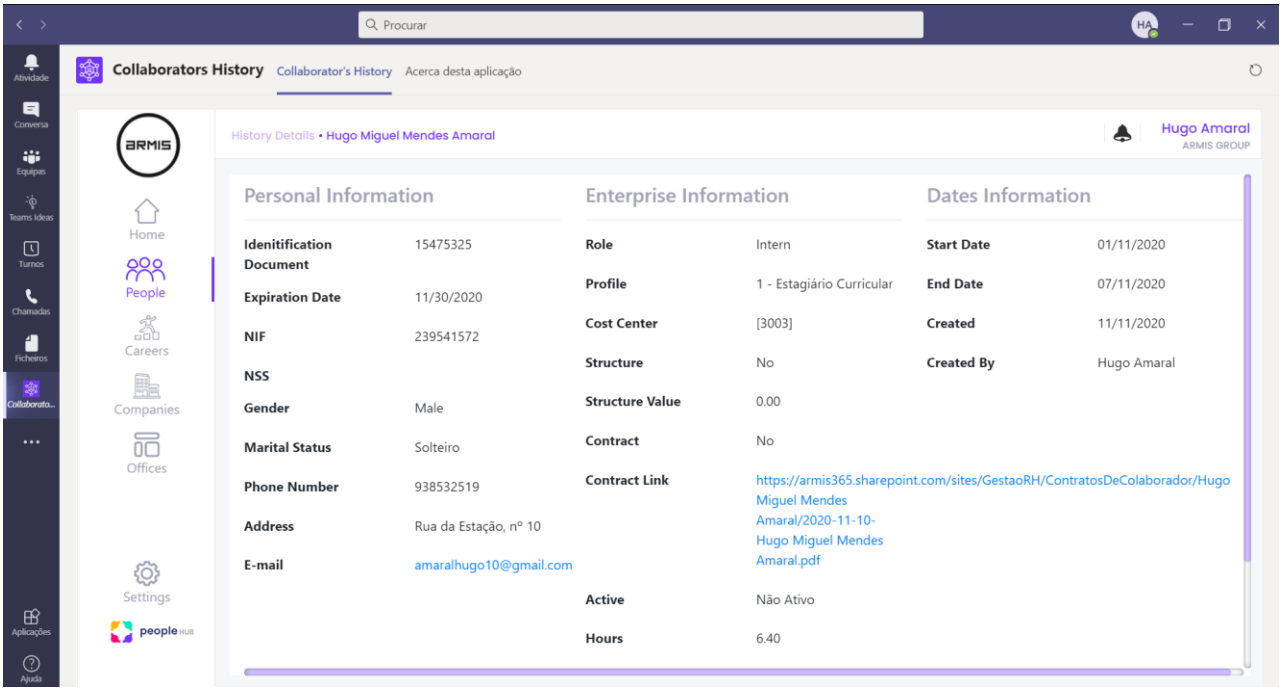


**Figura 30 - Página de detalhes de um histórico**

# 7 Testes

Para garantir a máxima qualidade da solução, foram realizados vários testes às funcionalidades da aplicação, durante o desenvolvimento e no final do projeto concluído. Estes testes ajudaram detetar *bugs* e a garantir o bom funcionamento de todas as funcionalidades da aplicação. Na figura 31 é demonstrado um exemplo de um teste realizado à criação de um histórico de colaborador. Neste teste verifica-se que não é possível inserir um novo histórico se todos os campos obrigatórios não forem preenchidos.

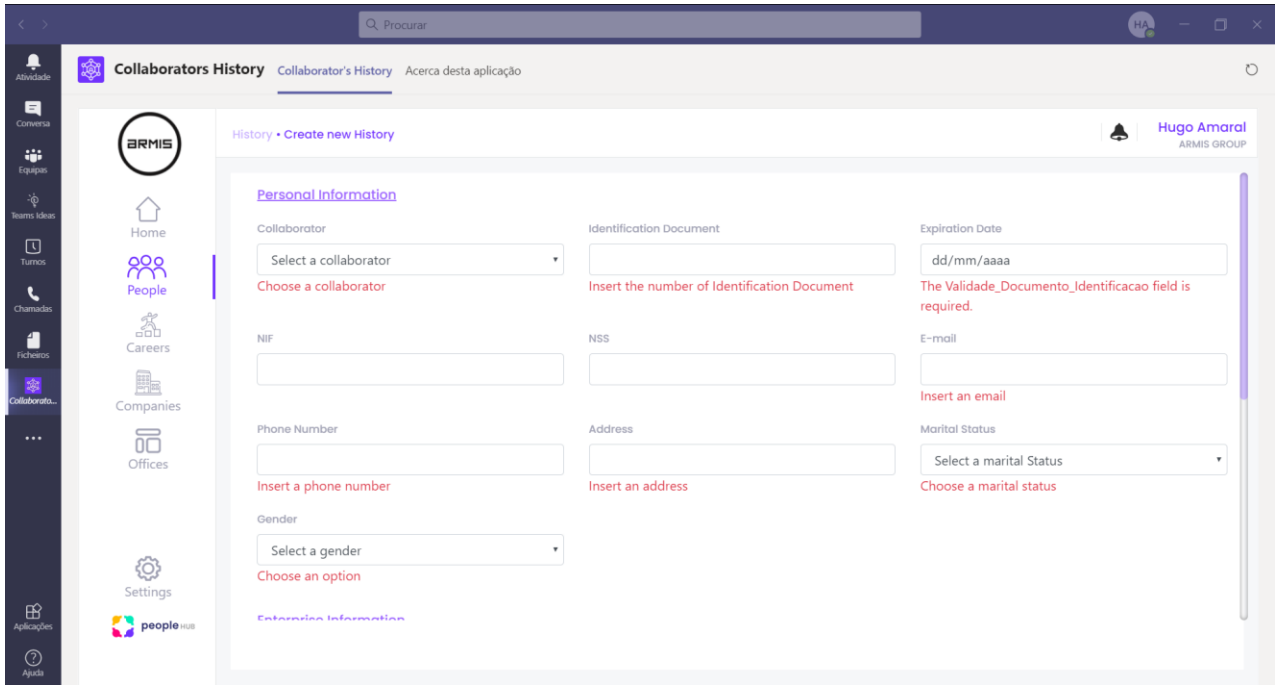


**Figura 31 - Teste à criação de um histórico de colaborador**

Além disso, ainda foram realizados testes utilizando as ferramentas de desenvolvedor do Google Chrome, como observar a responsividade da aplicação, desativar o JavaScript e ver os tempos de resposta. Estes testes permitiram melhorar a segurança e desempenho da aplicação. A figura 32 apresenta um teste realizado ao pedido para obter a lista de todos os históricos de colaboradores. Para tornar os testes mais realistas, limitou-se a velocidade da ligação para *Slow 3G*, de forma a simular ligações à internet

mais lentas. Como é observável na figura, para essa velocidade o pedido à lista de todos os históricos levou 2.81 segundos. Testes como este foram aplicados a diversos métodos da aplicação.

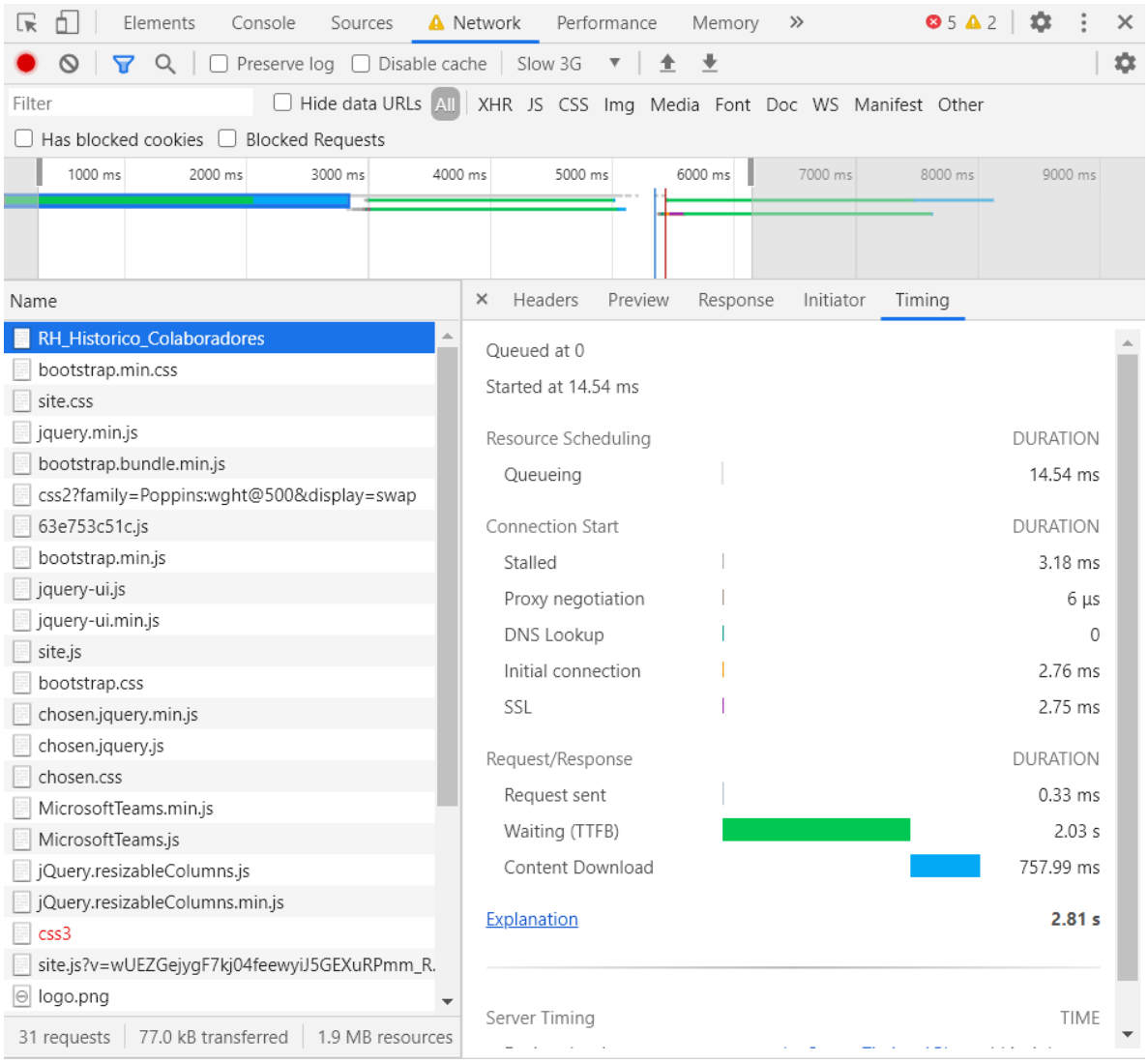


**Figura 32 - Teste de tempo de resposta ao pedido da lista de todos os históricos de colaboradores para a velocidade de ligação Slow 3G**

Em adição a estes testes, as entregas regulares auxiliaram na identificação e prevenção de algumas lacunas. A ajuda dos responsáveis do departamento de recursos humanos foi crucial nesse aspeto.

# 8 Conclusão

Este projeto teve como objetivo implementar uma solução integrada na plataforma unificada de comunicação e colaboração Microsoft Teams que permita ao departamento de Recursos Humanos substituir a entrada manual de dados em ficheiros Excel por um sistema que permita gerir informação relativa ao histórico de colaboradores, oferecendo mais suporte para colaboração e evitando a redundância de informação.

É seguro dizer que todas as três fases do projeto foram planeadas com sucesso, mas infelizmente, por falta de tempo, apenas se concluiu completamente a primeira fase, ou seja, o desenvolvimento da aplicação. Como trabalho futuro, a conclusão das restantes fases (integração com as fontes externas de informação MIM, ERP e People Hub) seriam definitivamente funcionalidades interessantes e úteis a adicionar.

É importante realçar também que, ainda que a primeira abordagem ao projeto não tenha dado certo, não se considera como perdido o tempo gasto no seu desenvolvimento. Pelo contrário, encara-se a oportunidade de trabalho com tecnologias nunca experienciadas como uma mais valia.

As maiores dificuldades residiram certamente no processo de autenticação e autorização para a aplicação no Microsoft Teams. A preciosa ajuda e orientação de alguns profissionais da Armis foi de extrema importância para contornar este e outros obstáculos.

Vale lembrar que o estágio possibilitou um contacto não só com profissionais experientes da área das tecnologias da informação e comunicação, mas também com departamentos de diferentes áreas, nomeadamente Design e Recursos Humanos. Isto permitiu a aquisição de novos conhecimentos e ter uma perspetiva interna do funcionamento de uma empresa inserida no desenvolvimento de soluções digitais complexas.

Por fim, trabalhar na Armis foi uma experiência muito gratificante, porque além de todos os momentos positivos, as dificuldades encontradas ao longo deste percurso

favoreceram uma clara evolução como futuro profissional e, mais importante, como pessoa.

# Referências Bibliográficas

[1] Lidiane Vieira, “Humanidades E Inovação,” THE IMPORTANCE OF PERSONNEL MANAGEMENTIN, 08 04 2015.

[2] A. Trost, INTRODUCTION INTO HUMAN RESOURCES MANAGEMENT, 2013.

[3] M. Rundell, Macmillan English Dictionary for Advanced Learners, 2002.

[4] “HRExam,” [Online]. Available: https://www.hrmexam.com/2019/05/17/difference-between-personnel-management-and-human-resource-management/. [Acedido em 05 10 2020].

[5] “Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Strategic_human_resource_planning. [Acedido em 05 10 2020].

[6] Nnamdi Azikiwe University, “The Effect of Human Resources Development on Organizational Productivity,” 10 2013.

[7] “https://www.toppr.com/guides/business-studies/directing/incentives/,” [Online]. [Acedido em 07 10 2020].

[8] S. Leung, “Forbes,” *Sorry, Your Spreadsheet Has Errors (Almost 90% Do),* 13 09 2014.

[9] M. p. o. D. Á. d. Software. [Online]. Available: http://agilemanifesto.org/iso/ptpt/manifesto.html. [Acedido em 17 10 2020].

[10] [Online]. Available: http://www.scrumportugal.pt/scrum/. [Acedido em 17 10 2020].

[11] F. N. R. Machado, Análise e Gestão de Requisitos de Software Onde nascem os sistemas, Saraiva Educação S.A, 2018.

[12] “Docs Microsoft,” Microsoft, [Online]. Available: https://docs.microsoft.com/pt-pt/powerapps/powerapps-overview. [Acedido em 02 11 2020].

[13] “Wikipedia,” [Online]. Available: https://pt.wikipedia.org/wiki/Microsoft_Teams. [Acedido em 03 11 2020].

[14] “Docs Microsoft,” Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/delegation-overview. [Acedido em 03 11 2020].

[15] “Docs Microsoft,” Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/power-automate/getting-started. [Acedido em 03 11 2020].

[16] “Wikipedia,” [Online]. Available: https://pt.wikipedia.org/wiki/Microsoft_SharePoint. [Acedido em 03 11 2020].

[17] “W3Schools,” [Online]. Available: https://www.w3schools.com/cs/cs_intro.asp. [Acedido em

04 11 2020].

[18] “Microsoft,” [Online]. Available: https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet. [Acedido em 04 11 2020].

[19] “W3Schools,” [Online]. Available: https://www.w3schools.com/whatis/whatis_html.asp. [Acedido em 04 11 2020].

[20] “Mozilla,” [Online]. Available: https://developer.mozilla.org/pt-PT/docs/Web/JavaScript/O_que_%C3%A9_o_JavaScript. [Acedido em 04 11 2020].

[21] “Powerapps,” Microsoft, [Online]. Available: https://powerapps.microsoft.com/pt-pt/pricing/. [Acedido em 08 11 2020].

[22] “Chrome Developer,” [Online]. Available: https://developer.chrome.com/apps/app_frameworks. [Acedido em 09 11 2020].

[23] “Docs Microsoft,” Microsoft, [Online]. Available: https://docs.microsoft.com/en-us/microsoftteams/platform/resources/schema/manifest-schema. [Acedido em 09 11 2020].

[24] “Docs Microsoft,” Microsoft, [Online]. Available: https://docs.microsoft.com/pt-br/microsoftteams/platform/tabs/how-to/authentication/auth-tab-aad. [Acedido em 09 11 2020].

[25] “Hydra,” Hydra, [Online]. Available: https://www.hydra.pt/erp-solucoes-gestao/dynamics-365-business-central/gestao-recursos-humanos/. [Acedido em 18 11 2020].

# 9 Anexos

## 9.1 Lista de módulos da aplicação

- Colaboradores
- Histórico de Colaboradores
- Cargos
- Perfis
- Preço de Perfil
- Contratos
- Centros de Custo
- Empresas
- Áreas de Negócio
- Departamentos
- Escritórios
- Cidades
- Países

## 9.2 Código

### 9.2.1 Procedimento para carregar tabela do histórico

```
USE [RH]
GO
/****** Object:  StoredProcedure [dbo].[insereHistorico] ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO


CREATE   procedure [dbo].[insereHistorico] as
declare @ID_Colaborador int;
declare @ID_Dept int;
declare @ID_Empresa int;
declare @ID_Cargo int;
declare @ID_Perfil int;
declare @ID_CentroCusto varchar(15);
```

```
declare @ID_Escritorio int;
declare @ID_Contrato int;
declare @NomeCompleto varchar(150);
declare @Contrato int;
declare @email varchar(100);
declare @estrutura bit;
declare @vEstrutura numeric(18,2);
declare @horas numeric(4,2);
declare @Dinicio datetime;
declare @Dfim datetime;
declare @ID_Cidade int;
declare @role varchar(50);
declare @perfil varchar(50);
declare @username varchar(50);
declare @centroCusto varchar(50);
declare @departamento varchar(100);
declare @ano int;
declare @admissao date;
declare @demissao date;
declare @IdDepartamento int;

declare c_historico cursor for
    select Departamento, ID_Empresa, ID_Cidade, CentroCusto, Username,
Perfil,
        Role, DataInicio, DataFim, Horas, Estrutura, ValorEstrutura,
Contrato, Admissao, Demissao
        from TBL_HISTORICO_COLABORADORES;

open c_historico;
fetch next from c_historico into @departamento, @ID_Empresa, @ID_Cidade,
@centroCusto, @username, @perfil, @role, @Dinicio, @Dfim,
@horas, @estrutura, @vEstrutura, @Contrato, @admissao, @demissao;

while @@FETCH_STATUS = 0
begin
    if(@ID_Empresa = 7)
    begin
        Set @ID_Empresa = 6;
    end
    if(@ID_Empresa = 9)
    begin
        Set @ID_Empresa = 5;
    end

    Set @ID_Escritorio = (select ID_Escritorio from RH_Escritorios where
ID_Cidade = @ID_Cidade);
    Set @ID_Cargo = (select ID_Cargo from RH_Cargos where Descricao =
@role);
    Set @ID_Perfil = (select ID_Perfil from RH_Perfis where perfil =
@perfil);
    Set @ID_CentroCusto = (select ID_CentroCusto from RH_CentroCusto
where CentroCusto = @centroCusto
        And ID_Empresa = @ID_Empresa And YEAR(@Dinicio) = Ano);
        print @ID_CentroCusto;
        print @ID_Empresa;
        print @departamento;
        print YEAR(@Dinicio);
```

```
    Set @ID_Colaborador = (select ID_Colaborador from RH_Colaboradores
where Username = @username);
    Set @email = @username+'@armis.pt';
    Set @NomeCompleto = (select Nome_completo from RH_Colaboradores where
ID_Colaborador = @ID_Colaborador);

    if (@admissao is not null)
    begin
        Set @ID_Contrato = (select ID_Contrato from RH_Contrato where
CHARINDEX(@NomeCompleto,Link) > 0
            And CHARINDEX(CONVERT(varchar,@admissao), Admissao)>0);
    end
    else
    begin
        Set @ID_Contrato = (select ID_Contrato from RH_Contrato where
CHARINDEX(@NomeCompleto,Link) > 0
            And Admissao is null);
    end

    print @NomeCompleto;
    print @admissao;

    insert into RH_Historico_Colaboradores( ID_Colaborador, ID_Cargo,
ID_Perfil,
        ID_CentroCusto, ID_EstadoCivil, ID_Contrato, ID_Escritorio,
Telemovel, Morada, Email, Sexo,
        Estrutura, ValorEstrutura, Documento_Identificacao,
Validade_Documento_Identificacao, Contrato,
        Ativo, Horas,DataInicioVigencia, DataFimVigencia,
ID_ModalidadeLaboral)
    values(@ID_Colaborador, @ID_Cargo, @ID_Perfil, @ID_CentroCusto,
1,@ID_Contrato,
        @ID_Escritorio, 'N/A', 'N/A', @email, 0, @estrutura, @vEstrutura,
'N/A', convert(datetime2, '9999-12-31 23:59:59.9999999'),
        @Contrato, 1, @horas, @Dinicio, @Dfim, 1);

    fetch next from c_historico into @departamento, @ID_Empresa,
@ID_Cidade, @centroCusto, @username, @perfil, @role, @Dinicio, @Dfim,
        @horas, @estrutura, @vEstrutura, @Contrato, @admissao, @demissao;
end
close c_historico;
deallocate c_historico;
```

### 9.2.2 Procedimento para carregar tabela dos Centros de Custo

```
USE [RH]
GO
/****** Object:  StoredProcedure [dbo].[insereHistorico] ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO


CREATE   procedure [dbo].[insereHistorico] as
declare @ID_Colaborador int;
declare @ID_Dept int;
declare @ID_Empresa int;
declare @ID_Cargo int;
declare @ID_Perfil int;
declare @ID_CentroCusto varchar(15);
declare @ID_Escritorio int;

declare @ID_Contrato int;
declare @NomeCompleto varchar(150);

declare @Contrato int;
declare @email varchar(100);
declare @estrutura bit;
declare @vEstrutura numeric(18,2);
declare @horas numeric(4,2);
declare @Dinicio datetime;
declare @Dfim datetime;
declare @ID_Cidade int;
declare @role varchar(50);
declare @perfil varchar(50);
declare @username varchar(50);
declare @centroCusto varchar(50);
declare @departamento varchar(100);
declare @ano int;
declare @admissao date;
declare @demissao date;
declare @IdDepartamento int;

declare c_historico cursor for
    select Departamento, ID_Empresa, ID_Cidade, CentroCusto, Username,
Perfil,
        Role, DataInicio, DataFim, Horas, Estrutura, ValorEstrutura,
Contrato, Admissao, Demissao
        from TBL_HISTORICO_COLABORADORES;

open c_historico;
fetch next from c_historico into @departamento, @ID_Empresa, @ID_Cidade,
@centroCusto, @username, @perfil, @role, @Dinicio, @Dfim,
@horas, @estrutura, @vEstrutura, @Contrato, @admissao, @demissao;

while @@FETCH_STATUS = 0
begin
```

```
    if(@ID_Empresa = 7)
    begin
        Set @ID_Empresa = 6;
    end
    if(@ID_Empresa = 9)
    begin
        Set @ID_Empresa = 5;
    end

    Set @ID_Escritorio = (select ID_Escritorio from RH_Escritorios where
ID_Cidade = @ID_Cidade);
    Set @ID_Cargo = (select ID_Cargo from RH_Cargos where Descricao =
@role);
    Set @ID_Perfil = (select ID_Perfil from RH_Perfis where perfil =
@perfil);
    Set @ID_CentroCusto = (select ID_CentroCusto from RH_CentroCusto
where CentroCusto = @centroCusto
        And ID_Empresa = @ID_Empresa And YEAR(@Dinicio) = Ano);
    Set @ID_Colaborador = (select ID_Colaborador from RH_Colaboradores
where Username = @username);
    Set @email = @username+'@armis.pt';
    Set @NomeCompleto = (select Nome_completo from RH_Colaboradores where
ID_Colaborador = @ID_Colaborador);

    if (@admissao is not null)
    begin
        Set @ID_Contrato = (select ID_Contrato from RH_Contrato where
            CHARINDEX(@NomeCompleto,Link) > 0
            And CHARINDEX(CONVERT(varchar,@admissao), Admissao)>0);
    end
    else
    begin
        Set @ID_Contrato = (select ID_Contrato from RH_Contrato where
            CHARINDEX(@NomeCompleto,Link) > 0
            And Admissao is null);
    end

    print @NomeCompleto;
    print @admissao;

    insert into RH_Historico_Colaboradores( ID_Colaborador, ID_Cargo,
ID_Perfil,
        ID_CentroCusto, ID_EstadoCivil, ID_Contrato, ID_Escritorio,
Telemovel, Morada, Email, Sexo,
        Estrutura, ValorEstrutura, Documento_Identificacao,
Validade_Documento_Identificacao, Contrato,
        Ativo, Horas,DataInicioVigencia, DataFimVigencia,
ID_ModalidadeLaboral)
    values(@ID_Colaborador, @ID_Cargo, @ID_Perfil, @ID_CentroCusto,
1,@ID_Contrato,
        @ID_Escritorio, 'N/A', 'N/A', @email, 0, @estrutura, @vEstrutura,
'N/A', convert(datetime2, '9999-12-31 23:59:59.9999999'),
        @Contrato, 1, @horas, @Dinicio, @Dfim, 1);

    fetch next from c_historico into @departamento, @ID_Empresa,
@ID_Cidade, @centroCusto, @username, @perfil, @role, @Dinicio, @Dfim,
        @horas, @estrutura, @vEstrutura, @Contrato, @admissao, @demissao;
```

```
end
close c_historico;
deallocate c_historico;
```

### 9.2.3 Controlador do Histórico de Colaboradores

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using Collaborators_History.Models;
using Collaborators_History.Models.ViewModels;
using Microsoft.Extensions.Configuration;
using NUglify.Helpers;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Authentication.JwtBearer;

namespace Collaborators_History.Controllers
{
    [Authorize(AuthenticationSchemes =
JwtBearerDefaults.AuthenticationScheme, Roles = "DRH")]
    public class RH_Historico_ColaboradoresController : Controller
    {
        private readonly RH_EntitiesContext _context;
        private readonly IConfiguration conf;
        private readonly int NUMBER_OF_PRODUCTS_PER_PAGE;
        private readonly int NUMBER_OF_PAGES_BEFORE_AND_AFTER;

        public RH_Historico_ColaboradoresController(RH_EntitiesContext
context, IConfiguration config)
        {
            _context = context;
            conf = config;
            NUMBER_OF_PRODUCTS_PER_PAGE =
conf.GetValue<int>("NUMBER_OF_PRODUCTS_PER_PAGE");
            NUMBER_OF_PAGES_BEFORE_AND_AFTER =
conf.GetValue<int>("NUMBER_OF_PAGES_BEFORE_AND_AFTER");
        }

        /// Metodo que retorna pagina com Historico dos colaboradores com
base na pesquisa, filtros e
        /// paginação
        public IActionResult Index(List<string> contratoSelecionada,
List<Boolean> estruturaSelecionada, List<string> cargoSelecionada,
List<string> PerfilSelecionada, List<string> costcenterSelecionada,
List<string> deptSelecionada, List<string> paisSelecionada, List<string>
empresaSelecionada, List<string> cidadeSelecionada, string searchString,
int page = 1)
        {
            decimal number;
```

```
            ViewBag.SearchString = searchString;
            ViewBag.currentPage = page;

            //Criacao e prenchimento de selectLists os dropdowns
            List<SelectListItem> selectListEmpresas = new
List<SelectListItem>();
            List<SelectListItem> selectListCidades = new
List<SelectListItem>();
            List<SelectListItem> selectListPais = new
List<SelectListItem>();
            List<SelectListItem> selectListDept = new
List<SelectListItem>();
            List<SelectListItem> selectListCostCenter = new
List<SelectListItem>();
            List<SelectListItem> selectListPerfil = new
List<SelectListItem>();
            List<SelectListItem> selectListCargo = new
List<SelectListItem>();
            List<SelectListItem> selectListEstrutura = new
List<SelectListItem>();
            List<SelectListItem> selectListContrato = new
List<SelectListItem>();

            _context.RH_Empresa.ForEach(x => selectListEmpresas.Add(new
SelectListItem() { Text = x.Codigo, Value = x.Codigo }));
            _context.RH_Cidades.ForEach(x => selectListCidades.Add(new
SelectListItem() { Text = x.Nome, Value = x.Nome }));
            _context.RH_Pais.ForEach(x => selectListPais.Add(new
SelectListItem() { Text = x.Nome, Value = x.Nome }));
            _context.RH_Departamento.ForEach(x => selectListDept.Add(new
SelectListItem() { Text = x.Departamento, Value = x.Departamento }));
            _context.RH_CentroCusto.ForEach(x =>
selectListCostCenter.Add(new SelectListItem() { Text = x.CentroCusto,
Value = x.CentroCusto }));
            _context.RH_Perfis.ForEach(x => selectListPerfil.Add(new
SelectListItem() { Text = x.descricaoPerfil, Value = x.descricaoPerfil
}));
            _context.RH_Cargos.ForEach(x => selectListCargo.Add(new
SelectListItem() { Text = x.Descricao, Value = x.Descricao }));

            foreach (var x in _context.RH_Historico_Colaboradores)
            {
                if (x.Estrutura.ToString() == "True")
                {
                    selectListEstrutura.Add(new SelectListItem() { Text =
"Yes" , Value = x.Estrutura.ToString() });
                }else if (x.Estrutura.ToString() == "False")
                {
                    selectListEstrutura.Add(new SelectListItem() { Text =
"No" , Value = x.Estrutura.ToString() });
                }

                if(x.Contrato.ToString() == "1") {
                    selectListContrato.Add(new SelectListItem() { Text =
"Has contract", Value = x.Contrato.ToString() });
                }else if (x.Contrato.ToString() == "0")
                {
```

```
                    selectListContrato.Add(new SelectListItem() { Text =
"No contract", Value = x.Contrato.ToString()});
                }
            }

            //Preenchimento do viewModel
            ViewModelHistoricoColaboradores vmhc = new
ViewModelHistoricoColaboradores
            {
                Empresas = _context.RH_Empresa.ToList(),
                Cidades = _context.RH_Cidades.ToList(),
                Paises = _context.RH_Pais.ToList(),
                Departamentos = _context.RH_Departamento.ToList(),
                CurrentPage = page,
                FirstPageShow = Math.Max(2, page -
NUMBER_OF_PAGES_BEFORE_AND_AFTER),
                EmpresasDropDown = selectListEmpresas,
                CidadesDropDown = selectListCidades,
                PaisDropDown = selectListPais,
                DepartamentoDropDown = selectListDept,
                CostCenterDropDown = selectListCostCenter.GroupBy(x=>
x.Text).Select(x => x.First()),
                PerfilDropDown = selectListPerfil,
                CargoDropDown = selectListCargo,
                EstruturaDropDown = selectListEstrutura.GroupBy(x =>
x.Text).Select(x => x.First()),
                ContratoDropDown = selectListContrato.GroupBy(x =>
x.Text).Select(x => x.First())
            };


            //atualizacao do viewModel com filtros, pesquisa e paginação
            if (!String.IsNullOrEmpty(searchString))
            {
                number = _context.RH_Historico_Colaboradores.Include(r =>
r.Cargo).Include(r => r.CentroCusto).Include(r =>
r.Colaborador).Include(r => r.ContratoFK).Include(r =>
r.Escritorio).Include(r => r.EstadoCivil).Include(r =>
r.ModalidadeLaboral).Include(r => r.Perfil)
                    .Where(p => p.Colaborador.Nome.Contains(searchString)
|| p.Colaborador.Nome_Completo.Contains(searchString) ||
                    p.Colaborador.Username.Contains(searchString) ||
p.Perfil.perfil.Contains(searchString) ||
                    p.Cargo.Cargo.Contains(searchString))
                    .Where(p => contratoSelecionada.Any() ?
contratoSelecionada.Contains(p.Contrato.ToString()) : true)
                    .Where(p => estruturaSelecionada.Any() ?
estruturaSelecionada.Contains(p.Estrutura) : true)
                    .Where(p => PerfilSelecionada.Any() ?
PerfilSelecionada.Contains(p.Perfil.descricaoPerfil) : true)
                    .Where(p => cargoSelecionada.Any() ?
cargoSelecionada.Contains(p.Cargo.Descricao) : true)
                    .Where(p => costcenterSelecionada.Any() ?
costcenterSelecionada.Contains(p.CentroCusto.CentroCusto) : true)
                    .Where(p => deptSelecionada.Any() ?
deptSelecionada.Contains(p.CentroCusto.Departamento.Departamento) : true)
```

```
                    .Where(p => paisSelecionada.Any() ?
paisSelecionada.Contains(p.Escritorio.Cidade.Pais.Nome) : true)
                    .Where(p => cidadeSelecionada.Any() ?
cidadeSelecionada.Contains(p.Escritorio.Cidade.Nome) : true)
                    .Where(p => empresaSelecionada.Any() ?
empresaSelecionada.Contains(p.CentroCusto.Empresa.Codigo) :
true).Count();


                vmhc.historicos =
_context.RH_Historico_Colaboradores.Include(r => r.Cargo).Include(r =>
r.CentroCusto).Include(r => r.Colaborador).Include(r =>
r.ContratoFK).Include(r => r.Escritorio).Include(r =>
r.EstadoCivil).Include(r => r.ModalidadeLaboral).Include(r => r.Perfil)
                    .Where(p => p.Colaborador.Nome.Contains(searchString)
|| p.Colaborador.Nome_Completo.Contains(searchString) ||
                    p.Colaborador.Username.Contains(searchString) ||
p.Perfil.perfil.Contains(searchString) ||
                    p.Cargo.Cargo.Contains(searchString))
                    .Where(p => contratoSelecionada.Any() ?
contratoSelecionada.Contains(p.Contrato.ToString()) : true)
                    .Where(p => estruturaSelecionada.Any() ?
estruturaSelecionada.Contains(p.Estrutura) : true)
                    .Where(p => PerfilSelecionada.Any() ?
PerfilSelecionada.Contains(p.Perfil.descricaoPerfil) : true)
                    .Where(p => cargoSelecionada.Any() ?
cargoSelecionada.Contains(p.Cargo.Descricao) : true)
                    .Where(p => costcenterSelecionada.Any() ?
costcenterSelecionada.Contains(p.CentroCusto.CentroCusto) : true)
                    .Where(p => deptSelecionada.Any() ?
deptSelecionada.Contains(p.CentroCusto.Departamento.Departamento) : true)
                    .Where(p => paisSelecionada.Any() ?
paisSelecionada.Contains(p.Escritorio.Cidade.Pais.Nome) : true)
                    .Where(p => cidadeSelecionada.Any() ?
cidadeSelecionada.Contains(p.Escritorio.Cidade.Nome) : true)
                    .Where(p => empresaSelecionada.Any() ?
empresaSelecionada.Contains(p.CentroCusto.Empresa.Codigo) : true)
                    .Skip((page - 1) * NUMBER_OF_PRODUCTS_PER_PAGE)
                    .Take(NUMBER_OF_PRODUCTS_PER_PAGE);

            }
            else
            {
                number = _context.RH_Historico_Colaboradores.Include(r =>
r.Cargo).Include(r => r.CentroCusto).Include(r =>
r.Colaborador).Include(r => r.ContratoFK).Include(r =>
r.Escritorio).Include(r => r.EstadoCivil).Include(r =>
r.ModalidadeLaboral).Include(r => r.Perfil)
                    .Where(p => contratoSelecionada.Any() ?
contratoSelecionada.Contains(p.Contrato.ToString()) : true)
                    .Where(p => PerfilSelecionada.Any() ?
PerfilSelecionada.Contains(p.Perfil.descricaoPerfil) : true)
                    .Where(p => cargoSelecionada.Any() ?
cargoSelecionada.Contains(p.Cargo.Descricao) : true)
                    .Where(p => costcenterSelecionada.Any() ?
costcenterSelecionada.Contains(p.CentroCusto.CentroCusto) : true)
```

```
                    .Where(p => deptSelecionada.Any() ?
deptSelecionada.Contains(p.CentroCusto.Departamento.Departamento) : true)
                    .Where(p => paisSelecionada.Any() ?
paisSelecionada.Contains(p.Escritorio.Cidade.Pais.Nome) : true)
                    .Where(p => cidadeSelecionada.Any() ?
cidadeSelecionada.Contains(p.Escritorio.Cidade.Nome) : true)
                    .Where(p => empresaSelecionada.Any() ?
empresaSelecionada.Contains(p.CentroCusto.Empresa.Codigo) : true)
                    .Where(p => estruturaSelecionada.Any() ?
estruturaSelecionada.Contains(p.Estrutura) : true).Count();


                vmhc.historicos =
_context.RH_Historico_Colaboradores.Include(r => r.Cargo).Include(r =>
r.CentroCusto).Include(r => r.Colaborador).Include(r =>
r.ContratoFK).Include(r => r.Escritorio).Include(r =>
r.EstadoCivil).Include(r => r.ModalidadeLaboral).Include(r => r.Perfil)
                    .Where(p => contratoSelecionada.Any() ?
contratoSelecionada.Contains(p.Contrato.ToString()) : true)
                    .Where(p => PerfilSelecionada.Any() ?
PerfilSelecionada.Contains(p.Perfil.descricaoPerfil) : true)
                    .Where(p => cargoSelecionada.Any() ?
cargoSelecionada.Contains(p.Cargo.Descricao) : true)
                    .Where(p => costcenterSelecionada.Any() ?
costcenterSelecionada.Contains(p.CentroCusto.CentroCusto) : true)
                    .Where(p => deptSelecionada.Any() ?
deptSelecionada.Contains(p.CentroCusto.Departamento.Departamento) : true)
                    .Where(p => paisSelecionada.Any() ?
paisSelecionada.Contains(p.Escritorio.Cidade.Pais.Nome) : true)
                    .Where(p => cidadeSelecionada.Any() ?
cidadeSelecionada.Contains(p.Escritorio.Cidade.Nome) : true)
                    .Where(p => empresaSelecionada.Any() ?
empresaSelecionada.Contains(p.CentroCusto.Empresa.Codigo) : true)
                    .Where(p => estruturaSelecionada.Any() ?
estruturaSelecionada.Contains(p.Estrutura) : true)
                    .Skip((page - 1) * NUMBER_OF_PRODUCTS_PER_PAGE)
                    .Take(NUMBER_OF_PRODUCTS_PER_PAGE);

            }

            vmhc.TotalPages = (int)Math.Ceiling( number /
NUMBER_OF_PRODUCTS_PER_PAGE);

            vmhc.LastPageShow = Math.Min(vmhc.TotalPages, page +
NUMBER_OF_PAGES_BEFORE_AND_AFTER);
            vmhc.FirstPage = 1;
            vmhc.LastPage = vmhc.TotalPages;
            vmhc.CurrentSearchString = searchString;
            vmhc.CargoSelecionada = cargoSelecionada;
            vmhc.ContratoSelecionada = contratoSelecionada;
            vmhc.EstruturaSelecionada = estruturaSelecionada;
            vmhc.PerfilSelecionada = PerfilSelecionada;
            vmhc.CostCenterSelecionada = costcenterSelecionada;
            vmhc.DeptSelecionada = deptSelecionada;
            vmhc.PaisSelecionada = paisSelecionada;
            vmhc.CidadeSelecionada = cidadeSelecionada;
            vmhc.EmpresaSelecionada = empresaSelecionada;
```

```
            return View(vmhc);
        }

        /// Metodo que retorna uma pagina com todos os detalhes
        public async Task<IActionResult> Details(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }

            var rH_Historico_Colaboradores = await
_context.RH_Historico_Colaboradores
                .Include(r => r.Cargo)
                .Include(r => r.CentroCusto)
                .Include(r => r.Colaborador)
                .Include(r => r.ContratoFK)
                .Include(r => r.Escritorio)
                .Include(r => r.EstadoCivil)
                .Include(r => r.ModalidadeLaboral)
                .Include(r => r.Perfil)
                .FirstOrDefaultAsync(m => m.ID_HistoricoColaborador ==
id);
            if (rH_Historico_Colaboradores == null)
            {
                return NotFound();
            }

            return View(rH_Historico_Colaboradores);
        }

        /// Metodo que retorna pagina com dados a prencher do historico
        public IActionResult Create()
        {
            List<SelectListItem> selectListCostCenter = new
List<SelectListItem>();
            List<RH_Empresa> emp = _context.RH_Empresa.ToList();

            foreach (var x in _context.RH_CentroCusto.OrderByDescending(o
=> o.Ano))
            {
                foreach(var i in emp)
                {
                    if(i.ID_Empresa == x.ID_Empresa)
                    {
                        selectListCostCenter.Add(new SelectListItem() {
Text = i.Codigo + " - "+ x.Ano +" - "+ x.Detalhe , Value =
x.ID_CentroCusto.ToString() });
                    }
                }
            }


            ViewData["ID_Cargo"] = new SelectList(_context.RH_Cargos,
"ID_Cargo", "Descricao");
            ViewData["ID_CentroCusto"] = selectListCostCenter;
```

```
            ViewData["ID_Colaborador"] = new
SelectList(_context.RH_Colaboradores.OrderBy(c => c.Nome),
"ID_Colaborador", "Nome");
            ViewData["ID_Escritorio"] = new
SelectList(_context.RH_Escritorios, "ID_Escritorio", "Morada");
            ViewData["ID_EstadoCivil"] = new
SelectList(_context.Set<RH_EstadoCivil>(), "ID_EstadoCivil",
"EstadoCivil");
            ViewData["ID_ModalidadeLaboral"] = new
SelectList(_context.Set<RH_ModalidadeLaboral>(), "ID_ModalidadeLaboral",
"ModalidadeLaboral");
            ViewData["ID_Perfil"] = new SelectList(_context.RH_Perfis,
"ID_Perfil", "descricaoPerfil");
            ViewData["ID_Contrato"] = new
SelectList(_context.Set<RH_Contrato>(), "ID_Contrato", "Link");

            return View();
        }

        /// Metodo que cria um novo historico
        [HttpPost]
        public async Task<IActionResult>
Create([Bind("ID_HistoricoColaborador,ID_Colaborador,ID_Cargo,ID_Perfil,I
D_CentroCusto,ID_EstadoCivil,ID_Contrato,ID_Escritorio,ID_ModalidadeLabor
al,Telemovel,Morada,Email,Sexo,Estrutura,ValorEstrutura,Documento_Identif
icacao,Validade_Documento_Identificacao,Contrato,Ativo,Horas,DataInicioVi
gencia,DataFimVigencia,SysStartTime,SysEndTime,Autor,NIF,NSS")]
RH_Historico_Colaboradores rH_Historico_Colaboradores)
        {

            if(teleExist(rH_Historico_Colaboradores.ID_Colaborador,
rH_Historico_Colaboradores.Telemovel)||

docIdentificacaoExist(rH_Historico_Colaboradores.ID_Colaborador,
rH_Historico_Colaboradores.Documento_Identificacao)||
                EmailExist(rH_Historico_Colaboradores.ID_Colaborador,
rH_Historico_Colaboradores.Email)||
                rH_Historico_Colaboradores.DataFimVigencia <
rH_Historico_Colaboradores.DataInicioVigencia)
            {
                return View(rH_Historico_Colaboradores);
            }

            if (!rH_Historico_Colaboradores.Estrutura)
            {
                rH_Historico_Colaboradores.ValorEstrutura = 0;
            }
            if (ModelState.IsValid)
            {
                _context.Add(rH_Historico_Colaboradores);
                await _context.SaveChangesAsync();
                return RedirectToAction("Details", new { id =
rH_Historico_Colaboradores.ID_HistoricoColaborador });
            }
            ViewData["ID_Cargo"] = new SelectList(_context.RH_Cargos,
"ID_Cargo", "Descricao", rH_Historico_Colaboradores.ID_Cargo);
```

```
            ViewData["ID_CentroCusto"] = new
SelectList(_context.RH_CentroCusto, "ID_CentroCusto", "CentroCusto",
rH_Historico_Colaboradores.ID_CentroCusto);
            ViewData["ID_Colaborador"] = new
SelectList(_context.RH_Colaboradores, "ID_Colaborador", "Nome",
rH_Historico_Colaboradores.ID_Colaborador);
            ViewData["ID_Contrato"] = new
SelectList(_context.Set<RH_Contrato>(), "ID_Contrato", "Link"+"Admissao",
rH_Historico_Colaboradores.ID_Contrato);
            ViewData["ID_Escritorio"] = new
SelectList(_context.RH_Escritorios, "ID_Escritorio", "Morada",
rH_Historico_Colaboradores.ID_Escritorio);
            ViewData["ID_EstadoCivil"] = new
SelectList(_context.Set<RH_EstadoCivil>(), "ID_EstadoCivil",
"EstadoCivil", rH_Historico_Colaboradores.ID_EstadoCivil);
            ViewData["ID_ModalidadeLaboral"] = new
SelectList(_context.Set<RH_ModalidadeLaboral>(), "ID_ModalidadeLaboral",
"ModalidadeLaboral", rH_Historico_Colaboradores.ID_ModalidadeLaboral);
            ViewData["ID_Perfil"] = new SelectList(_context.RH_Perfis,
"ID_Perfil", "descricaoPerfil", rH_Historico_Colaboradores.ID_Perfil);
            return View(rH_Historico_Colaboradores);
        }

        /// Metodo que retorna pagina com campos preenchidos
        /// do historico selecionado de forma a ser editado
        public async Task<IActionResult> Edit(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }

            var rH_Historico_Colaboradores = await
_context.RH_Historico_Colaboradores.FindAsync(id);
            if (rH_Historico_Colaboradores == null)
            {
                return NotFound();
            }

            List<SelectListItem> selectListCostCenter = new
List<SelectListItem>();
            List<RH_Empresa> emp = _context.RH_Empresa.ToList();

            foreach (var x in _context.RH_CentroCusto.OrderByDescending(o
=> o.Ano))
            {
                foreach (var i in emp)
                {
                    if (i.ID_Empresa == x.ID_Empresa)
                    {
                        selectListCostCenter.Add(new SelectListItem() {
Text = i.Codigo + " - " + x.Ano + " - " + x.Detalhe, Value =
x.ID_CentroCusto.ToString() });
                    }
                }
            }
```

```
            ViewData["ID_Cargo"] = new SelectList(_context.RH_Cargos,
"ID_Cargo", "Descricao", rH_Historico_Colaboradores.ID_Cargo);
            ViewData["ID_CentroCusto"] = selectListCostCenter;
            ViewData["ID_Colaborador"] = new
SelectList(_context.RH_Colaboradores, "ID_Colaborador", "Nome",
rH_Historico_Colaboradores.ID_Colaborador);
            ViewData["ID_Contrato"] = new
SelectList(_context.Set<RH_Contrato>(), "ID_Contrato", "Link",
rH_Historico_Colaboradores.ID_Contrato);
            ViewData["ID_Escritorio"] = new
SelectList(_context.RH_Escritorios, "ID_Escritorio", "Morada",
rH_Historico_Colaboradores.ID_Escritorio);
            ViewData["ID_EstadoCivil"] = new
SelectList(_context.Set<RH_EstadoCivil>(), "ID_EstadoCivil",
"EstadoCivil", rH_Historico_Colaboradores.ID_EstadoCivil);
            ViewData["ID_ModalidadeLaboral"] = new
SelectList(_context.Set<RH_ModalidadeLaboral>(), "ID_ModalidadeLaboral",
"ModalidadeLaboral", rH_Historico_Colaboradores.ID_ModalidadeLaboral);
            ViewData["ID_Perfil"] = new SelectList(_context.RH_Perfis,
"ID_Perfil", "descricaoPerfil", rH_Historico_Colaboradores.ID_Perfil);
            return View(rH_Historico_Colaboradores);
        }

        /// MEtodo que atualiza os dados do historico enviado
        /// pelo utilizador
        [HttpPost]
        public async Task<IActionResult> Edit(int id,
[Bind("ID_HistoricoColaborador,ID_Colaborador,ID_Cargo,ID_Perfil,ID_Centr
oCusto,ID_EstadoCivil,ID_Contrato,ID_Escritorio,ID_ModalidadeLaboral,Tele
movel,Morada,Email,Sexo,Estrutura,ValorEstrutura,Documento_Identificacao,
Validade_Documento_Identificacao,Contrato,Ativo,Horas,DataInicioVigencia,
DataFimVigencia,SysStartTime,SysEndTime,Autor,NIF,NSS")]
RH_Historico_Colaboradores rH_Historico_Colaboradores)
        {
            if (id != rH_Historico_Colaboradores.ID_HistoricoColaborador)
            {
                return NotFound();
            }
            if (rH_Historico_Colaboradores.DataFimVigencia == null &&
_context.RH_Historico_Colaboradores.OrderBy(o =>
o.DataInicioVigencia).Any(p => p.DataInicioVigencia >=
rH_Historico_Colaboradores.DataInicioVigencia &&
p.ID_HistoricoColaborador !=
rH_Historico_Colaboradores.ID_HistoricoColaborador && p.ID_Colaborador ==
rH_Historico_Colaboradores.ID_Colaborador))
            {
                return View("ErrorDBOperation", "Shared");
            }


           if (ModelState.IsValid)
            {
                try
                {
                    _context.Update(rH_Historico_Colaboradores);
                    await _context.SaveChangesAsync();
                }
```

```
                catch (DbUpdateConcurrencyException)
                {
                    if
(!RH_Historico_ColaboradoresExists(rH_Historico_Colaboradores.ID_Historic
oColaborador))
                    {
                        return NotFound();
                    }
                    else
                    {
                        throw;
                    }
                }
                return RedirectToAction(nameof(Index));
            }
            ViewData["ID_Cargo"] = new SelectList(_context.RH_Cargos,
"ID_Cargo", "Descricao", rH_Historico_Colaboradores.ID_Cargo);
            ViewData["ID_CentroCusto"] = new
SelectList(_context.RH_CentroCusto, "ID_CentroCusto", "ID_CentroCusto",
rH_Historico_Colaboradores.ID_CentroCusto);
            ViewData["ID_Colaborador"] = new
SelectList(_context.RH_Colaboradores, "ID_Colaborador", "ID_Colaborador",
rH_Historico_Colaboradores.ID_Colaborador);
            ViewData["ID_Contrato"] = new
SelectList(_context.Set<RH_Contrato>(), "ID_Contrato", "ID_Contrato",
rH_Historico_Colaboradores.ID_Contrato);
            ViewData["ID_Escritorio"] = new
SelectList(_context.RH_Escritorios, "ID_Escritorio", "ID_Escritorio",
rH_Historico_Colaboradores.ID_Escritorio);
            ViewData["ID_EstadoCivil"] = new
SelectList(_context.Set<RH_EstadoCivil>(), "ID_EstadoCivil",
"ID_EstadoCivil", rH_Historico_Colaboradores.ID_EstadoCivil);
            ViewData["ID_ModalidadeLaboral"] = new
SelectList(_context.Set<RH_ModalidadeLaboral>(), "ID_ModalidadeLaboral",
"ID_ModalidadeLaboral", rH_Historico_Colaboradores.ID_ModalidadeLaboral);
            ViewData["ID_Perfil"] = new SelectList(_context.RH_Perfis,
"ID_Perfil", "ID_Perfil", rH_Historico_Colaboradores.ID_Perfil);
            return View(rH_Historico_Colaboradores);
        }

        private bool RH_Historico_ColaboradoresExists(int id)
        {
            return _context.RH_Historico_Colaboradores.Any(e =>
e.ID_HistoricoColaborador == id);
        }

        public RH_Historico_Colaboradores getById(int id)
        {

            RH_Historico_Colaboradores historico = (from h in
_context.RH_Historico_Colaboradores
                                                    where
h.ID_Colaborador == id
                                                    orderby
h.DataInicioVigencia descending
                                                    select
h).FirstOrDefault();
```

```
            return historico;
        }

        public List<RH_Contrato> getContratosColaborador(int id)
        {
            RH_Colaboradores colaborador = (from col in
_context.RH_Colaboradores
                                            where col.ID_Colaborador ==
id
                                            select col).FirstOrDefault();

             List<RH_Contrato> contratos = (from c in
_context.RH_Contrato
                                            where
c.Link.Contains(colaborador.Nome_Completo)
                                            select c).ToList();

            return contratos;
        }

        /// verificar se a data de fim e sempre maior que a do inicio
        public IActionResult VerifyDates(DateTime DataInicioVigencia,
DateTime DataFimVigencia, int ID_Colaborador, int
ID_HistoricoColaborador)
        {
            bool isEdit = _context.RH_Historico_Colaboradores.Any(p =>
p.ID_HistoricoColaborador == ID_HistoricoColaborador);

            var registoRecente = _context.RH_Historico_Colaboradores
                .OrderBy(o => o.DataInicioVigencia)
                .Where(p => p.ID_Colaborador == ID_Colaborador &&
p.ID_HistoricoColaborador != ID_HistoricoColaborador &&
p.DataInicioVigencia > DataInicioVigencia)
                .FirstOrDefault();

            if (isEdit)
            {
                if (registoRecente != null)
                {

                    if (DataFimVigencia >
registoRecente.DataInicioVigencia && DataFimVigencia != null)
                    {
                        return Json($"A data de fim deste historico tem
de ser menor que a do historico mais recente");
                    }
                }
            }

            if (DataInicioVigencia > DataFimVigencia)
            {
                return Json($"Initial Date must be before End Date");
            }
            else
            {
                return Json(true);
            }
```

```
        }

        /// verificar se o telemovel não existe noutro colaborador
      public IActionResult VerifyTelemovel(string Telemovel, int
ID_Colaborador)
        {
            var teleExiste = teleExist(ID_Colaborador, Telemovel);

            if (teleExiste)
            {
                return Json($"Another employee has this phone number
already!");
            }
            else
            {
                return Json(true);
            }
        }

        /// verificar se o documento de identificacao não existe noutro
colaborador
        public IActionResult VerifyDocIdent(string
Documento_Identificacao, int ID_Colaborador)
        {
            var docExiste = docIdentificacaoExist(ID_Colaborador,
Documento_Identificacao);
            if (docExiste)
            {
                return Json($"Another employee has this document number
already!");
            }
            else
            {
                return Json(true);
            }
        }

        /// verificar se o nif e unico
        public IActionResult VerifyNIF(string NIF, int ID_Colaborador)
        {
            var nifExiste = NifExist(ID_Colaborador, NIF);

            if (nifExiste)
            {
                return Json($"Another employee has this NIF already!");
            }
            else
            {
                return Json(true);
            }
        }

        /// verificar se o nss e unico
        public IActionResult VerifyNSS(string NSS, int ID_Colaborador)
        {
            var nssExiste = NSSExist(ID_Colaborador, NSS);
```

```
            if (nssExiste)
            {
                return Json($"Another employee has this NSS already!");
            }
            else
            {
                return Json(true);
            }
        }

        /// verificar se o email nao e usado por outro colaborador
        public IActionResult VerifyEmail(string Email, int
ID_Colaborador)
        {
            var emailExiste = EmailExist(ID_Colaborador, Email);

            if (emailExiste)
            {
                return Json($"Another employee has e-mail number
already!");
            }
            else
            {
                return Json(true);
            }
        }

        /// validacao das datas no historico
        public IActionResult VerifyStartDate(DateTime DataInicioVigencia,
int ID_HistoricoColaborador, int ID_Colaborador, DateTime
DataFimVigencia)
        {
            bool isEdit = _context.RH_Historico_Colaboradores.Any(p =>
p.ID_HistoricoColaborador == ID_HistoricoColaborador);
            var ultimoRegisto =
_context.RH_Historico_Colaboradores.OrderByDescending(o =>
o.DataInicioVigencia).Where(p => p.ID_Colaborador ==
ID_Colaborador).FirstOrDefault();
            var ultimoRegistoEdit =
_context.RH_Historico_Colaboradores.OrderByDescending(o =>
o.DataFimVigencia).Where(p => p.ID_Colaborador == ID_Colaborador &&
p.ID_HistoricoColaborador != ID_HistoricoColaborador &&
p.DataInicioVigencia < DataInicioVigencia).FirstOrDefault();


            if (isEdit)
            {

                if(ultimoRegistoEdit == null)
                {
                    return Json(true);
                }
                if (ultimoRegistoEdit.DataFimVigencia >
DataInicioVigencia)
                {
                    return Json($"Start Date must be after last history's
End Date");
```

```
                }
                else
                {
                    return Json(true);
                }
            }
            if(ultimoRegisto != null)
            {
                if (ultimoRegisto.DataFimVigencia > DataInicioVigencia)
                {
                    return Json($"Start Date must be after last history's
End Date");
                }
                else
                {
                    return Json(true);
                }
            }
            return Json(true);
        }

        private bool teleExist(int id, string numero)
        {
            return _context.RH_Historico_Colaboradores.Any(p =>
p.Telemovel == numero && p.ID_Colaborador != id);
        }

        private bool docIdentificacaoExist(int id, string docIdent)
        {
            return _context.RH_Historico_Colaboradores.Any(p =>
p.Documento_Identificacao == docIdent && p.ID_Colaborador != id);
        }
        private bool NifExist(int id, string nif)
        {
            return _context.RH_Historico_Colaboradores.Any(p => p.NIF ==
nif && p.ID_Colaborador != id);
        }
        private bool NSSExist(int id, string nss)
        {
            return _context.RH_Historico_Colaboradores.Any(p => p.NSS ==
nss && p.ID_Colaborador != id);
        }
        private bool EmailExist(int id, string email)
        {
            return _context.RH_Historico_Colaboradores.Any(p => p.Email
== email && p.ID_Colaborador != id);
        }

        [HttpPost]
        public void updateEndDate([FromBody] UpdateEndDate updateObj)
        {
            var ultimoHistorico =
_context.RH_Historico_Colaboradores.OrderByDescending(o =>
o.DataInicioVigencia).Where(p => p.ID_Colaborador ==
updateObj.id).First();
            ultimoHistorico.DataFimVigencia = updateObj.endDate;
            _context.SaveChanges();
```

```
            }
        }
}
```