Otimização de Processos de Recolha de Dados para Análise e Tratamento de Dados de Faturação

Gabriel Corrêa Cordeiro - a49207

Relatório de estágio apresentado à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Informática no âmbito da dupla diplomação com a Universidade Tecnológica Federal do Paraná.

Trabalho orientado por:

Prof. Rui Pedro Lopes

Prof. Rodrigo Campiolo

Bragança

Outubro de 2025

Otimização de Processos de Recolha de Dados para Análise e Tratamento de Dados de Faturação

Mestrado em Informática

Escola Superior de Tecnologia e Gestão

Gabriel Corrêa Cordeiro - a49207

Trabalho orientado por:

Prof. Rui Pedro Lopes

Prof. Rodrigo Campiolo

Bragança

Outubro de 2025

Agradecimentos

Gostaria, em primeiro lugar, de expressar a minha gratidão à minha família, em especial aos meus irmãos Adriano Corrêa Cordeiro e Izabella Corrêa Cordeiro, pelo apoio incondicional, e à minha mãe, Josenilda Corrêa, que, mesmo sem saber ainda como utilizar um computador, acreditou e apoiou a minha jornada. Por último, mas não menos importante, ao meu tio Valmer Menegat Cordeiro, por promover a busca pelo conhecimento como algo fundamental em minha vida.

À minha esposa, Gabriela Welinski, que esteve incondicionalmente ao meu lado, oferecendo apoio em meio aos desafios impostos durante a minha caminhada, sendo minha maior fonte de apoio.

A todos na empresa TECH X, agradeço pela oportunidade de trabalharmos juntos, pelos valores e humanidade demonstrados por todos que me apoiaram em todos os momentos deste projeto.

Também desejo expressar a minha gratidão ao Instituto Politécnico de Bragança e à Universidade Tecnológica Federal do Paraná por todo o conhecimento que de forma excecional foi administrado durante as aulas e orientações. Também agradeço pelas amizades e oportunidades que me proporcionaram. Em particular, gostaria de agradecer aos meus orientadores, Prof. Rui Pedro Lopes e Prof. Rodrigo Campiolo, pelo apoio, conselhos e presença ao longo da minha jornada académica e profissional, sendo imprescindíveis para que pudesse alcançar meus objetivos.

Por fim, agradeço a todos os amigos que sempre desejaram o meu melhor durante todo meu processo de formação e crescimento, em especial a Felipe Gimenez, Alexandre Colauto Neto, Lucas Guedes Barboza, Helder Valdez, Henrique Pinheiro.

Resumo

O presente trabalho tem como objetivo apresentar os conhecimentos adquiridos e aplicados durante o estágio realizado, focando no desenvolvimento de uma ferramenta para atender às necessidades da empresa TECH X. Fundada em 2020, a TECH X é uma empresa de Tecnologias da Informação (TI) dedicada ao avanço tecnológico e criativo, oferecendo serviços de desenvolvimento *Web*, *Mobile*, *Marketing* e Gestão de recursos.

O projeto surge em resposta à crescente transformação digital que permeia praticamente todos os setores da indústria. Uma das principais soluções desenvolvidas pela empresa é o Gestor Virtual de Energia (GVE), uma ferramenta poderosa para auxiliar na gestão dos processos energéticos de organizações. O GVE organiza os dados de maneira compreensível para os gestores, permitindo que eles tomem decisões de forma mais eficiente e assertiva.

Diante da alta demanda do mercado, a TECH X optou por desenvolver uma ferramenta que oferecesse maior agilidade, automatizar o processo de digitalização documental, desde a importação até a estruturação dos dados. Essa ferramenta recebeu o nome de XFile, promovendo a renovação do GVE, para o Gestor Virtual de Energia Online (GVE Online), uma nova perspetivava de abordagem na gestão energética, a utilizar o XFile como seu gestor documental padrão. Portanto, o XFile adiciona maior flexibilidade no desenvolvimento de novas abordagens, velocidade e escalabilidade no processamento de documentos, segurança e rastreabilidade.

Com base nos conhecimentos adquiridos ao longo do mestrado, foi possível desenvolver uma ferramenta que atende a todos os requisitos estabelecidos pela empresa, impactando positivamente a experiência dos clientes e possibilitando a prospecção de novas soluções

diante da complexidade dos desafios enfrentados.

Palavras-chave: Gestor Virtual de Energia, digitalização documental, processamento de documentos.

Abstract

The present work aims to present the knowledge acquired and applied during the internship, focusing on the development of a tool to meet the needs of the company TECH X. Founded in 2020, TECH X is an Information Technology (IT) company dedicated to technological and creative advancement, offering services in Web development, Mobile, Marketing, and Resource Management.

The project arises in response to the growing digital transformation that permeates practically all sectors of industry. One of the main solutions developed by the company is the Gestor Virtual de Energia (GVE), a powerful tool to assist in the management of energy processes in organizations. The GVE organizes data in a comprehensible way for managers, allowing them to make decisions more efficiently and assertively.

Given the high market demand, TECH X chose to develop a tool that would offer greater agility by automating the document digitization process, from importation to data structuring. This tool was named XFile, driving the renewal of the Gestor Virtual de Energia (GVE) into Gestor Virtual de Energia Online (GVE Online), a new perspective in energy management that uses XFile as its standard document manager. Therefore, XFile adds greater flexibility in developing new approaches, speed and scalability in document processing, as well as security and traceability.

Based on the knowledge acquired throughout the master's degree, it was possible to develop a tool that meets all the requirements established by the company, positively impacting the customer experience and enabling the prospecting of new solutions in the face of the complexity of the challenges faced.

Keywords: Gestor Virtual de Energia, document digitization, document processing.

Conteúdo

Lista de Figuras

Siglas

AMQP Advanced Message Queuing Protocol. 48

API Application Programming Interface. 12, 20, 48, 50

APIs *Application Programming Interfaces*. 2, 6, 7, 26

APIs REST Application Programming Interfaces with Representational State Transfer. 24, 58, 59, 60

CD Continuous Delivery. 13, 14

CI Continuous Integration. 13, 15

CI/CD Continuous Integration/Continuous Delivery. 12

DDD Domain-Driven Design. 48, 51

DIP Dependency Inversion Principle. 52

eSQ Evolved Search Queries. 20

GDPR General Data Protection Regulation. 13

GVE Gestor Virtual de Energia. vi, viii, 4, 6, 7, 46, 50

GVE Online Gestor Virtual de Energia Online. vi, viii, 6, 7, 8, 19, 20, 24, 26, 27, 44, 46, 49, 55, 56, 78, 90

HDFS Hadoop Distributed File System. 12

HIPAA Health Insurance Portability and Accountability Act. 13

HMAC Hash-Based Message Authentication Code. 46

HTML HyperText Markup Language. 43

HTTP Hypertext Transfer Protocol. 48, 58

IAM Identity and Access Management. 13

IDE Integrated Development Environment. 41

ISP Interface Segregation Principle. 52

IT Information Technology. viii

JAR Java Archive. 10

JIT Just-In-Time. 9

JSON JavaScript Object Notation. 4, 6, 11, 19, 23, 41, 54, 59, 60, 78, 91

JVM Java Virtual Machine. 9, 15

JWT Java Web Token. 29, 31, 32, 33, 35, 37, 38, 46, 59, 70

KMS Key Management Service. 13

kNN k-Nearest Neighbors. 20, 21

KNN k-nearest neighbors. 7, 21, 22, 57, 59, 81, 91

LDAP Lightweight Directory Access Protocol. 42

LGPD Lei Geral de Proteção de Dados. 13

LLMs Large Language Models. 20

LSP Liskov Substitution Principle. 52

MTTR Mean Time to Resolution. 20

NoSQL Not only SQL. 11

NVMe Non-Volatile Memory Express. 12

OCP Open/Closed Principle. 52

PDF Portable Document Format. 4, 6, 19, 23, 41, 60, 78, 91

PDF/A PDF for Archiving. 60

POMs Project Object Model. 10

QA Quality Assurance. 15

RAG Retrieval-Augmented Generation. 21

RDP Remote Desktop Protocol. 43

RF Requisitos Funcionais. 22

RNF Requisitos Não Funcionais. 22

SEO Search Engine Optimization. 4

SRE Site Reliability Engineering. 20

SRP Single Responsibility Principle. 51

SSE Server-Side Encryption. 13

SSH Secure Shell. 43

SSPL Server Side Public License. 11

TCO Total Cost of Ownership. 11, 12

TI Tecnologias da Informação. vi, 3, 4

TLS Transport Layer Security. 13

UML Unified Modeling Language. 27

VNC Virtual Network Computing. 43

WAR Web Application Archive. 10

XML Extensible Markup Language. 4, 5, 6, 19, 23, 41, 78, 91

Capítulo 1

Introdução

No presente capítulo, é apresentado o contexto geral do trabalho. Para tanto, o capítulo foi dividido em três secções, a saber: enquadramento, objetivos e estrutura do documento. No enquadramento, é apresentado o contexto em que o trabalho se insere. Nos objetivos, é apresentada a proposta original do projeto. Por fim, na parte de estrutura do documento, é apresentada de forma sucinta como estão organizados os capítulos do presente trabalho.

1.1 Enquadramento

O presente trabalho enquadra-se no contexto da disciplina Estágio pertencente ao Mestrado em Informática, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança. O projeto foi elaborado para se integrar aos serviços fornecidos pela empresa de tecnologia TECH X de modo a melhorar o método de processamento de dados atualmente utilizado. O projeto surge como uma resposta as dificuldades na recolha e interpretação de documentos de faturação energética.

1.2 Objetivos

O objetivo do presente trabalho é abordar os conhecimentos que foram adquiridos e colocados em prática durante a realização do estágio. Com o objetivo de criar uma ferramenta

capaz de melhorar os processos que envolvem a recolha e a interpretação automatizada dos dados, foi definido o seguinte plano de trabalho:

- Familiarizar com as *Application Programming Interfaces* (APIs) existentes na empresa;
- Definir a arquitetura da aplicação;
- Implementar componentes de pesquisas;
- Implementar componentes de extração de dados;
- Realizar testes e validação;
- Escrita do relatório final.

1.3 Estrutura do Documento

O presente trabalho está disposto na seguinte forma: No Capítulo 2 é apresentada a fundamentação teórica sobre todo ecossistema em que o presente projeto se insere. O Capítulo 3 apresenta uma análise do mercado na qual a solução proposta se insere, juntamente com as métricas utilizadas e seus requisitos.

Já no Capítulo 4 é apresentado todo contexto de desenvolvimento, a passar pelo ambiente de trabalho, arquitetura até os resultados do projeto. Por fim, no Capítulo 5 são apresentadas as considerações finais do projeto.

Capítulo 2

Contexto e Tecnologias

Para uma boa compreensão deste documento é necessário entender o contexto em que o problema está inserido juntamente com os conceitos e ferramentas utilizadas. Portanto, neste capítulo será apresentada uma breve descrição da empresa na qual o estágio foi realizado (secção 2.1), em seguida será exposto o problema ao qual o software desenvolvido visa resolver (secção 2.2), o contexto (secção 2.3) e as tecnologias utilizadas para tal (secção 2.4).

2.1 Empresa

A TECH X [1] opera no setor de TI e serviços digitais, um mercado em constante crescimento e evolução. Este setor é caracterizado pela rápida inovação tecnológica e pela crescente demanda por soluções digitais que otimizem processos empresariais e melhorem a experiência do utilizador. A TECH X posiciona-se como uma empresa de vanguarda, focada em desenvolver soluções que atendam às necessidades emergentes do mercado, com um forte compromisso com a sustentabilidade e a redução das emissões de gases de efeito estufa.

A TECH X oferece uma ampla gama de serviços, de modo a incluir:

- **Desenvolvimento de Software**: Criação de aplicações web, móveis (Android e

iOS) e desktop. Desenvolvimento de websites e lojas online. Otimização de websites com Search Engine Optimization (SEO).

- **Marketing Digital**: Campanhas de marketing digital para aumentar a visibilidade e o alcance das empresas. Gestão de redes sociais e criação de conteúdo digital.

- **Formação e Consultoria**: Formação em tecnologias emergentes e boas práticas de TI. Consultoria para a implementação de soluções tecnológicas personalizadas.

2.2 Problema

A aceleração da transformação digital que permeia praticamente todos os setores industriais tem impulsionado uma crescente demanda por soluções inteligentes que melhorem a gestão de dados e processos empresariais. Nesse cenário, destaca-se o *Gestor Virtual de Energia* (GVE), desenvolvido pela empresa TECH X, como uma ferramenta robusta e estratégica para apoiar a administração eficiente dos processos energéticos e financeiros nas organizações.

Em resposta a esse desafio, a TECH X identificou a necessidade de criar o serviço *XFile*, uma solução inovadora voltada à digitalização e tratamento de documentos, desde a importação até a estruturação completa das informações contidas em ficheiros Extensible Markup Language (XML), Portable Document Format (PDF) e JavaScript Object Notation (JSON).

O XFile tem como principal objetivo eliminar a dependência de adaptações manuais no código sempre que surgem novos formatos de documentos, proporcionando assim maior flexibilidade, rapidez, segurança, precisão e qualidade na integração de dados. Com essa abordagem, o GVE e demais soluções desenvolvidas pela TECH X passam a dispor de uma base mais sólida e escalável para o tratamento documental.

Apesar da existência de padrões como o CIUS-PT, que padronizam campos contabilísticos (como totais, preços e impostos), esses formatos não atendem plenamente às

necessidades da TECH X. O desafio reside nos campos relacionados ao consumo, contratos e leituras, que não são padronizados e frequentemente aparecem como texto livre em tags genéricas como `<NOTE>` em um ficheiro XML por exemplo. Essa falta de uniformidade é agravada pela diversidade linguística dos fornecedores, alguns escrevem em português, outros em espanhol, dificultando a extração automatizada e confiável dessas informações.

Atualmente, o processo envolve a conversão do *XML do fornecedor* (em formatos como CIUS-PT, EDI, UBL, etc.) para um *XML intermediário*, estruturado conforme o formato esperado pelo GVE. Esse fluxo funciona como uma espécie de conversão manual documental. Essa abordagem gera dependência de um ficheiro intermediário que precisa ser mantido e adaptado constantemente, gerando interrupções repentinas no processamento, exceções não tratadas e então sendo necessária a correção/adaptação do código para então realizar o reprocessamento. Na figura 2.1 é apresentado um fluxo simplificado de como esse processo é realizado.

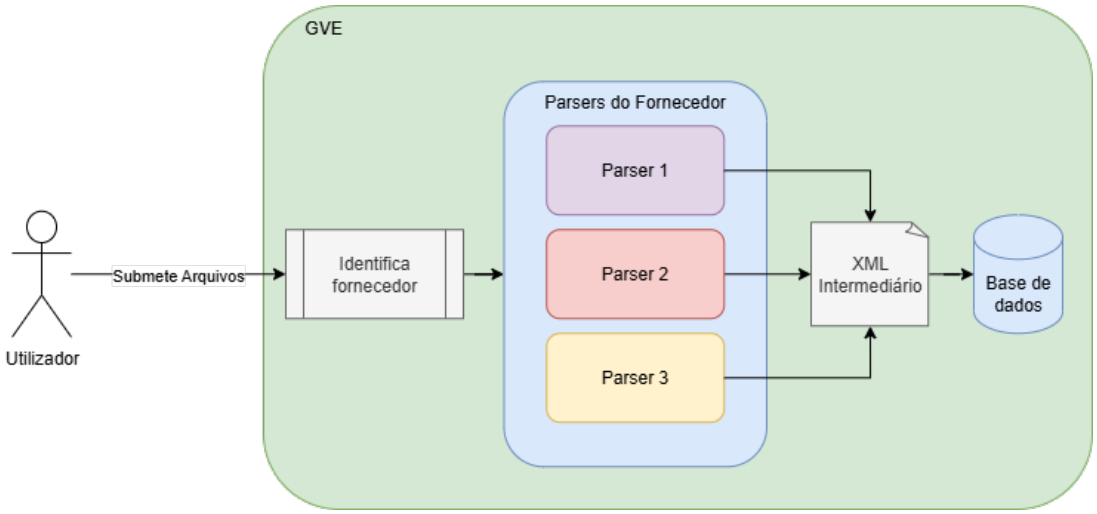

Figura 2.1: Fluxo de conversão atual do GVE.

O XFile surge como resposta a essa limitação, propondo uma arquitetura que elimina a necessidade do XML intermediário e permite que o sistema reconheça e estruture os dados diretamente a partir dos documentos originais, mesmo em cenários de baixa padronização. Além disso, o XFile possibilita que os utilizadores personalizem o mapeamento dos documentos, reconheçam e classifiquem novos campos com facilidade, e façam ajustes

diretamente na interface do serviço. Essa autonomia transforma o processo documental em algo mais transparente e eficaz, alinhando-se às exigências atuais do mercado por soluções dinâmicas, acessíveis e centradas na experiência do utilizador.

2.3 Âmbito do projeto

O setor energético enfrenta hoje desafios crescentes de complexidade operacional, volume de dados e exigências regulatórias [2]. Nesse cenário, a transformação digital torna-se imperativa para otimizar processos, garantir a qualidade da informação e extrair métricas que fundamentem decisões estratégicas de eficiência e sustentabilidade.

O XFile, oferece maior automação na importação de documentos, padronização e gestão documental de diversos formatos (XML, PDF, JSON), eliminando gargalos manuais de tratamento de dados. Integrado ao GVE Online, a plataforma evolui para um ecossistema digital completo, capaz de alimentar dashboards em tempo real, relatórios regulatórios e APIs de consulta com informações homogêneas e confiáveis.

Nesta secção, delineamos o âmbito do XFile na TECH X a partir de cinco vertentes. Primeiro, analisamos a reestruturação digital que deu origem ao GVE Online. Em seguida, exploramos a contribuição do XFile para a eficiência energética e extração de métricas, bem como o seu papel como coletor e pré-processador de dados para o GVE Online. Por fim, descrevemos os mecanismos inteligentes de validação de nomes e valores de campo e a consolidação de uma base de conhecimento destinada a apoiar futuras integrações de inteligência artificial.

2.3.1 Transformação digital na TECH X

Com o início do desenvolvimento do XFile, a TECH X embarcou numa jornada de transformação digital que se concretizou no desenvolvimento do GVE Online, uma evolução integral do sistema GVE. Essa mudança envolveu a reestruturação arquitetural, a adoção de tecnologias modernas e a migração para uma plataforma baseada em microsserviços.

Na figura 2.2 é apresentado um fluxo simplificado com o desacoplamento da gestão documental do GVE para o XFile, sendo responsável por prover essa gestão para o GVE, GVE Online e utilizadores/parceiros estratégicos da TECH X.

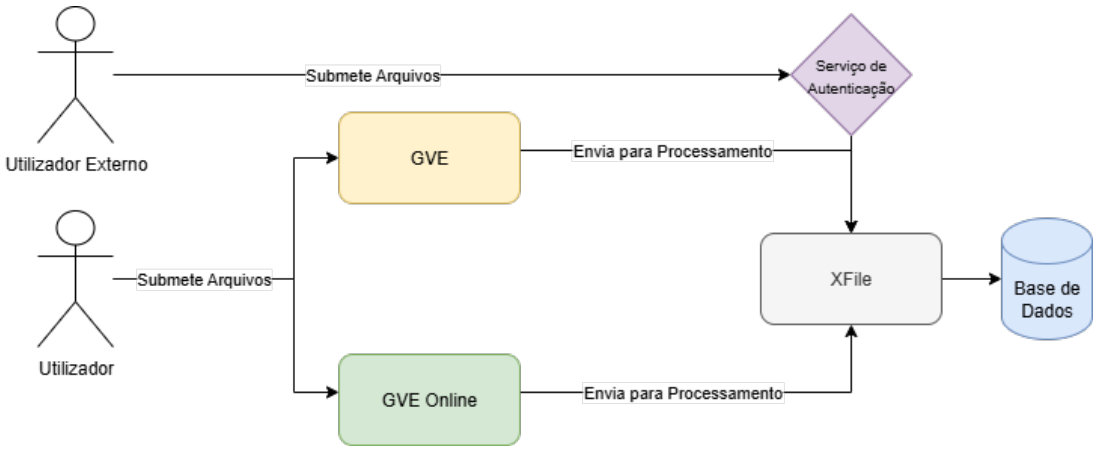

Figura 2.2: Desacoplamento da gestão documental para XFile.

2.3.2 Enquadramento do XFile no GVE Online

No fluxo operacional do GVE Online, o XFile atua como coletor e processador de dados. Todos os documentos digitalizados e indexados pelo XFile são armazenados num repositório central e disponibilizados ao GVE Online por meio de APIs seguras. Essa integração assegura que *dashboards*, relatórios e APIs de consulta sempre se baseiem em informações atualizadas e uniformes, fortalecendo a confiabilidade do sistema.

2.3.3 Validação inteligente de campos e valores

Para lidar com campos desconhecidos ou nomenclaturas variantes, o XFile emprega um mecanismo híbrido de pesquisa por similaridade (Lucene) e k-nearest neighbors (KNN). Quando um campo ou valor não corresponde à base conhecida ou não atinge o nível de confiança exigido, ele é encaminhado para uma lista de validação para o utilizador. Após a confirmação, o XFile adiciona esses valores na entidade descrita *DocumentReference* na secção 4.3). Esse fluxo garante que avisos repetidos não ocorram para elementos já

validados, reduzindo o atrito do utilizador e mantendo a consistência dos metadados ao longo do tempo.

2.3.4 Base de conhecimento e integrações futuras de IA

A indexação inversa dos documentos e a extração de dados armazenados no banco de dados formam uma base de conhecimento estruturada. Essa base servirá como alicerce para módulos de inteligência artificial no GVE Online, possibilitando no futuro a extração semântica de informações e recomendações automatizadas de eficiência energética.

2.4 Tecnologias

No desenvolvimento do projeto XFile, a escolha das tecnologias adequadas desempenha um papel crucial para garantir a eficiência, segurança e escalabilidade da solução. A TECH X, comprometida com a inovação e a excelência, selecionou um conjunto de tecnologias de ponta que suportam as funcionalidades essenciais do XFile.

A seguir, serão apresentadas as tecnologias que compõem a base do XFile, desde as linguagens de programação e frameworks até as ferramentas de gestão de dados e segurança. Cada tecnologia foi escolhida com o propósito de otimizar o desempenho da aplicação, facilitar a integração com outras soluções da TECH X e proporcionar uma experiência de utilizador robusta e intuitiva.

2.4.1 Linguagem de Programação Java

A linguagem de programação Java é amplamente reconhecida por suas características resiliente e versáteis, que a tornam uma escolha popular entre desenvolvedores [3].

O Java é uma linguagem orientada a objetos, o que significa que os programas são organizados em torno de objetos e classes, facilitando a reutilização de código e a manutenção [4].

O lema “Write Once, Run Anywhere” (Escreva uma vez, execute em qualquer lugar)

reflete a capacidade do Java de ser executado em qualquer plataforma que suporte a Java Virtual Machine (JVM), sem a necessidade de recompilação.

O Java foi projetado com a segurança em mente, incluindo recursos como a verificação de bytecode e a gestão de memória automática, que ajudam a prevenir vulnerabilidades comuns.

Suporta a execução de múltiplas threads simultaneamente, permitindo o desenvolvimento de aplicações que podem realizar várias tarefas ao mesmo tempo.

Embora seja uma linguagem interpretada, o uso de técnicas como Just-In-Time (JIT) permite que Java ofereça um desempenho competitivo [5].

Vantagens no Processamento de Dados

O Java oferece várias vantagens significativas no contexto do processamento de dados, tornando-se uma escolha ideal para aplicações que exigem manipulação eficiente e segura de grandes volumes de dados.

Garantindo escalabilidade com a capacidade lidar com múltiplas threads e sua arquitetura robusta permitem que aplicações de processamento de dados escalem eficientemente para lidar com grandes volumes de dados [6].

2.4.2 Apache Lucene

O Apache Lucene [7] é uma biblioteca de software livre para indexação e pesquisa de texto, escrita completamente em Java. O Apache Lucene é conhecido por sua capacidade de realizar indexações e buscas de alta performance, suporta indexação incremental tão rápida quanto a indexação em lote, oferece algoritmos de busca poderosos, permite a busca em múltiplos índices com resultados mesclados, além de oferecer facetas flexíveis, realce, junções e agrupamento de resultados. Por fim, está disponível como software de código aberto sob a licença Apache, permitindo seu uso em programas comerciais.

Vantagens no Processamento de Dados

O Apache Lucene oferece várias vantagens significativas no contexto do processamento de dados, tornando-se uma escolha ideal para aplicações que exigem manipulação eficiente e precisa de grandes volumes de texto.

O Lucene é altamente eficiente na busca de informações relevantes em grandes conjuntos de dados, o que é essencial no mundo dos dados de hoje [8]. Suporta diversos tipos de consulta que atendem a diferentes casos de uso de busca, incluindo buscas por sinónimos e recomendações de documentos semelhantes [7]. A comunidade ativa e a documentação abrangente garantem que os desenvolvedores tenham acesso a recursos e suporte contínuo [9].

2.4.3 Spring Boot

Spring Boot é um framework que simplifica o desenvolvimento de aplicações Java, especialmente no contexto de microsserviços.

O Spring Boot detecta automaticamente as dependências no projeto e configura os componentes necessários, reduzindo o tempo e esforço em configurações manuais. Permite a criação de aplicações que podem ser executadas diretamente como ficheiros Java Archive (JAR) ou Web Application Archive (WAR), sem a necessidade de um servidor de aplicações externo. Inclui servidores como Tomcat, Jetty ou Undertow, simplificando o processo de implantação. Com os Spring Boot Starter Project Object Model (POMs), é possível adicionar dependências comuns com uma única linha no ficheiro pom.xml. Oferece o módulo Spring Boot Actuator, que fornece informações detalhadas sobre o estado da aplicação [10].

Vantagens no Processamento de Dados

Spring Boot oferece várias vantagens significativas no contexto do processamento de dados, tornando-se uma escolha ideal para aplicações que exigem manipulação eficiente e

segura de grandes volumes de dados. Promovendo automação embutida e o amplo conjunto de ferramentas contribuem para uma produtividade excecional durante o ciclo de desenvolvimento. Integra-se perfeitamente com o ecossistema Spring, proporcionando recursos poderosos para desenvolvimento, teste e implantação. A arquitetura orientada a microserviços é facilitada pelo Spring Boot, permitindo o desenvolvimento e escalabilidade eficientes [10].

Considerando as exigências de desempenho, flexibilidade, suporte e escalabilidade do XFile para o processamento de dados, o Spring Boot foi adotado como framework principal para suportar seu desenvolvimento.

2.4.4 MongoDB

O MongoDB é um banco de dados orientado a documentos de plataforma cruzada. Classificado como um banco de dados Not only SQL (NoSQL), o MongoDB usa documentos do tipo JSON com esquemas opcionais. O MongoDB é desenvolvido pela MongoDB Inc. e licenciado sob a Server Side Public License (SSPL) [11].

Também, o MongoDB por ser um banco de dados NoSQL orientado a documentos é usado para armazenamento de dados de alto volume. Em vez de usar tabelas e linhas como nos bancos de dados relacionais tradicionais, o MongoDB faz uso de coleções e documentos. Os documentos consistem em pares de valores-chave que são a unidade básica de dados no MongoDB [12].

As coleções contêm conjuntos de documentos e funções que são equivalentes às tabelas de banco de dados relacional. O MongoDB se tornou um dos bancos de dados NoSQL mais populares, sendo usado para armazenar dados no *back-end* para muitos sites importantes, incluindo eBay [13], Craigslist [14], SourceForge [15] e The New York Times [16].

2.4.5 MinIO

A consolidação de um *data lake* eficiente depende de três pilares: escalabilidade, desempenho e baixo Total Cost of Ownership (TCO). O MinIO alinha-se a esses requisitos ao

oferecer um armazenamento de objetos compatível com a API S3, executável em qualquer ambiente de *bare-metal* a nuvem pública, mantendo simplicidade operacional. A seguir, enumeram-se os principais aspetos positivos corroborados pela literatura.

1. **Escalabilidade horizontal elástica** A arquitetura *shared-nothing* do MinIO permite adicionar novos nós sem interrupção, distribuindo dados por meio de *hashing* e *erasure coding*, o que garante crescimento linear de capacidade e I/O [17], [18].

2. **Desempenho elevado em leitura/escrita** Benchmarks publicados pela própria comunidade reportam picos de 325 GiB/s em operações `GET` e 165 GiB/s em `PUT` em clusters de 32 nós Non-Volatile Memory Express (NVMe), superando soluções baseadas em Hadoop Distributed File System (HDFS). Esse desempenho sustenta workloads analíticos (Spark, Lucene, Trino) e pipelines de *machine learning* [19].

3. **Custo-efetividade** Ao adotar discos genéricos e dispensar licenciamento proprietário, o TCO torna-se competitivo frente a `object stores` tradicionais. O uso de *erasure coding* reduz a sobrecarga de replicação (~1.33×) em relação ao espelhamento 3× do HDFS [17].

4. **Nativo em nuvem (*cloud-native*) e leve** O binário único (<100 MB) segue princípios de conteinerização e infraestrutura como código, facilitando Continuous Integration/Continuous Delivery (CI/CD), automação Kubernetes e implantações *edge* [18].

5. **Ampla integração com o ecossistema analítico** A compatibilidade S3 garante interoperabilidade instantânea com motores como Spark, Flink, Kafka, Presto/-Trino, Iceberg, Hudi e Delta Lake, consolidando o *lakehouse pattern* sem *vendor lock-in* [20].

6. **Resiliência e durabilidade** O esquema de *erasure coding* oferece durabilidade de 99.99999999% (*11 nines*) mesmo diante da perda simultânea de múltiplos discos ou nós, sem degradação significativa de throughput [21].

7. **Segurança corporativa** Criptografia em repouso (Server-Side Encryption (SSE)/Key Management Service (KMS)), Transport Layer Security (TLS) end-to-end, controle de acesso baseado em políticas (Identity and Access Management (IAM)) e trilhas de auditoria atendem requisitos de General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA) e Lei Geral de Proteção de Dados (LGPD) [22].

Coletivamente, esses fatores posicionam o MinIO como uma fundação robusta para *data lakes* modernos, permitindo que a camada de armazenamento acompanhe a evolução exponencial da ingestão de dados sem comprometer desempenho ou orçamento.

2.4.6 GitHub Actions

A integração de *GitHub Actions* no ciclo de vida de desenvolvimento de aplicações Java representa uma solução moderna e eficiente para automatizar processos de Continuous Integration (CI) e Continuous Delivery (CD), promovendo maior confiabilidade, agilidade e rastreabilidade no desenvolvimento de software [23].

Automatização de Workflows

GitHub Actions permite definir workflows em ficheiros `YAML`, armazenados no diretório `.github/workflows/` do repositório. Cada workflow pode ser disparado por eventos como `push`, `pull request` ou `merge`, garantindo que testes e *builds* sejam executados automaticamente a cada alteração no código-fonte [24].

CI

A aplicação Java pode ser testada automaticamente com ferramentas como `JUnit` e `Maven`, logo após cada commit [25]. O processo inclui etapas como:

- Configuração do ambiente Java com `actions/setup-java`
- Execução de testes com `mvn test`

- Validação do build com `mvn package`

CD

Após a validação do código, o workflow pode gerar uma imagem Docker da aplicação e publicá-la em repositórios como o Docker Hub. Em seguida, pode realizar o *deploy* automático em ambientes como Render, Heroku ou servidores privados, dependendo da configuração do pipeline [25].

Segurança e Controlo

O *GitHub Actions* permite definir permissões específicas para cada ambiente, garantindo que apenas utilizadores autorizados possam aprovar *deploys* em produção. Os logs de execução são armazenados e acessíveis diretamente no repositório, promovendo transparência e rastreabilidade [26].

Escalabilidade e Reutilização

Os workflows podem ser reutilizados entre projetos, promovendo padronização e economia de tempo. É possível configurar múltiplos jobs que correm em paralelo ou em sequência, conforme a complexidade da aplicação [23].

2.4.7 Docker

A utilização de Docker para a conteinerização de aplicações Java representa uma abordagem moderna e eficaz na gestão e implementação de software, proporcionando inúmeras vantagens tanto na fase de desenvolvimento como na de produção.

Portabilidade e Consistência

O Docker permite encapsular a aplicação Java e todas as suas dependências num único contêiner, garantindo que funcione de forma idêntica em diferentes ambientes (desenvolvimento, teste, produção), reduzindo os erros relacionados com discrepâncias de configuração [27].

Simplificação do Processo de Deploy

A criação de imagens Docker torna o processo de *deploy* mais previsível, automatizado e repetível. O uso de ferramentas como Docker Compose facilita a orquestração de serviços complementares, como bases de dados e servidores de *cache*.

Isolamento de Ambiente

Cada contêiner executa em ambiente isolado, evitando conflitos entre bibliotecas, versões da JVM e outras dependências. Este isolamento permite executar múltiplas versões de uma mesma aplicação Java lado a lado, sem interferência.

Escalabilidade e Integração com Orquestradores

Os contêineres Docker são compatíveis com orquestradores como Kubernetes, permitindo escalar aplicações Java de forma dinâmica conforme a carga. Esta compatibilidade facilita a adoção de arquiteturas baseadas em microserviços [28].

O Docker permite criar ambientes controlados para testes automatizados e *pipelines* de CI. As imagens podem ser versionadas e utilizadas em ambientes de Quality Assurance (QA) idênticos ao de produção.

A gestão centralizada de imagens permite validar e auditar componentes de forma mais rigorosa, reduzindo o risco de vulnerabilidades ao evitar dependências instaladas de forma manual e não controlada.

Evita configurações complexas de ambientes Java em servidores físicos ou máquinas virtuais. A velocidade de *boot* dos contêineres é significativamente superior, promovendo

maior eficiência operacional [29].

2.4.8 Discord

A comunicação eficiente é um dos pilares do desenvolvimento ágil e colaborativo. No contexto da TECH X, o Discord foi adotado como a principal plataforma de comunicação interna, substituindo soluções tradicionais como e-mail e mensageiros corporativos para a comunicação diária. Originalmente voltado para comunidades de jogos, o Discord evoluiu para atender demandas empresariais, oferecendo canais organizados, integração com bots e alta disponibilidade.

A equipa dispõe de canais de voz e texto para reuniões rápidas, chamadas espontâneas e mensagens persistentes. Isso permite que membros em diferentes fusos horários colaborem sem barreiras temporais.

Os servidores são estruturados em canais temáticos, como `daily-meet`, `lembretes`, `problemas-e-suporte-geral`, `dicas-e-sites-de-interesse`. Também canais de áudio por contextos de projetos, como `XFile` e `GVE-Online`. Esses canais dedicados ao acompanhamento diário de tarefas promovem ciclos curtos de feedback, alinhando expectativas e reduzindo retrabalho. A informalidade da plataforma contribui para um ambiente mais recetivo à crítica construtiva.

O Discord também fornece o suporte a Webhooks e bots personalizados enviam notificações automáticas dos pipelines (GitHub Actions, Azure DevOps) — sucesso ou falha de builds, resultados de testes, além de alertas de erros em ambientes de produção, teste e desenvolvimento, assegurando resposta rápida a incidentes.

A presença constante dos colaboradores nos canais, aliada à possibilidade de personalização (emojis, bots, avatares), fortalece o senso de comunidade e pertencimento, mesmo em contextos de trabalho remoto.

A escolha pelo Discord reflete uma estratégia de comunicação centrada na agilidade, acessibilidade e informalidade produtiva, alinhando-se às práticas modernas de desenvolvimento distribuído. No capítulo 3 será exposto como o XFile utilizou as ferramentas

previamente descritas e as contextualizou em sua abordagem.

Capítulo 3

Análise e Abordagem

Neste capítulo é apresentado uma análise do contexto, mercado em que a solução desse projeto se insere. Também são apresentadas as ferramentas utilizadas no projeto.

3.1 Processo de Digitalização e Inovação de Energia

A digitalização tem-se afirmado como essencial da transição energética, permitindo otimizar a eficiência operacional, gerir a crescente complexidade do sistema de energia e acelerar a descarbonização. Segundo o artigo publicado no Electric Summit [2], tecnologias digitais como inteligência artificial, processamento avançado de dados e Internet das Coisas são determinantes para apoiar o planeamento, a operação e o consumo energético de forma integrada e segura.

No âmbito da TECH X, o XFile assume um papel central neste processo ao oferecer uma plataforma de gestão documental inteligente. Capaz de importar, processar e estruturar automaticamente informações de ficheiros XML, PDF e JSON, o XFile elimina a necessidade de adaptações manuais sempre que surgem novos formatos. Isso garante que o GVE Online receba dados padronizados e de alta qualidade, prontos para análises preditivas, monitorização em tempo real e geração de relatórios regulatórios.

A integração fluida entre XFile e GVE Online promoverá a criação de um espaço de dados compartilhado e seguro no setor energético. Os documentos digitalizados pelo

XFile alimentam diretamente o GVE Online, que por sua vez disponibiliza dashboards, API e relatórios customizados, alinhados às diretrizes de cibersegurança e gestão de dados definidos pela União Europeia.

Dessa forma, o XFile potencializa a inovação digital na gestão de energia, oferecendo aos clientes da TECH X uma solução escalável, transparente e preparada para evoluir com as demandas de um setor em constante transformação.

3.2 Trabalhos Relacionados

Nos últimos anos, técnicas que combinam indexação textual com Apache Lucene e busca por similaridade via k-Nearest Neighbors (kNN) têm ganhado destaque na organização e recuperação de dados não estruturados. Avanços como a incorporação de métodos de busca aproximada em vetores densos no Lucene, descritos por Teofili e Lin [30], e melhorias recentes na eficiência da busca kNN em grandes coleções, investigadas por Ma, Teofili e Lin [31], demonstram a maturidade e a aplicabilidade crescente dessas abordagens em diversos domínios.

Hirsch, Hirsch e Ogunleye [32] apresentam o Evolved Search Queries (eSQ), uma nova abordagem para o agrupamento de documentos baseada na evolução de consultas de múltiplas palavras. O método utiliza algoritmos genéticos para gerar consultas que funcionam como rótulos de *clusters*, oferecendo vantagens como interpretabilidade, possibilidade de modificação manual e explicação causal sobre a formação dos agrupamentos. Após a formação inicial dos *clusters*, documentos não classificados são atribuídos por meio do algoritmo kNN, garantindo cobertura total do conjunto de dados. Em experimentos comparativos, o eSQ apresentou desempenho superior a técnicas consagradas como k-means++ e *agglomerative clustering* em diversos conjuntos de dados heterogêneos, mesmo quando o número de *clusters* não era previamente conhecidos.

De forma complementar, Rawat [33] propõem uma solução para reduzir o Mean Time to Resolution (MTTR) em incidentes de Site Reliability Engineering (SRE) por meio da integração de Elasticsearch, busca por vizinhos mais próximos kNN e Large Language

Models (LLMs) em um fluxo de Retrieval-Augmented Generation (RAG). O sistema indexa dados históricos de alarmes e resoluções, incluindo registros de comunicação, como vetores de *embeddings* semânticos no Elasticsearch com suporte a kNN, possibilitando a recuperação rápida e precisa de incidentes similares. Embora o domínio seja distinto do presente projeto, focado na gestão documental e energética, a solução reforça a relevância técnica da indexação vetorial e da busca aproximada para pesquisa semântica em dados não estruturados, evidenciando o potencial dessa abordagem para acelerar a tomada de decisão e ampliar a eficácia da recuperação de conhecimento em contextos diversos.

3.3 Indexação Vetorial de dados

Conforme exposto na secção 3.1, a transformação digital no contexto energético é fundamental para a contínua otimização no fornecimento de serviços, nas metas da Comissão Europeia, onde os dados estão intrinsecamente ligados, na transição digital da economia e para a utilização de energias limpas no âmbito do pacto ecológico europeu. Sendo assim, a criação de soluções capazes de processamento avançado de dados, contribuem na melhoria da eficiência e gestão dos dados do sistema de energia em todas esferas de abastecimento, desde planeamento, operação, manutenção da infraestrutura, transmissão até o consumidor final [2].

Diante do processo de digitalização do setor energético e enquadramento do XFile na gestão documental energética (secção 2.2), a indexação vetorial de dados pode melhorar significativamente a gestão de documentos, otimizando a recuperação, a compactação e o processamento de consultas de dados [34]. Além dessas vantagens, a indexação vetorial de dados desempenha um papel fundamental no aprimoramento da utilização de algoritmos de classificação, como KNN (secção 3.4). A combinação entre a indexação vetorial de dados e KNN acelera o processo de classificação, reduz a complexidade computacional, permite uma consulta de dados eficiente por meio de índices pré-computados que mapeiam os pontos de dados para seus respetivos recursos ou termos, facilitando pesquisas mais rápidas por meio de uma tabela de pesquisa ou dicionário [35].

3.4 k-Nearest Neighbors

A escolha do KNN como algoritmo de classificação do XFile (secção 2.3.3) ocorre pela sua efetividade na aplicação de identificação de campos e valores em documentos. O KNN é um algoritmo de aprendizado supervisionado simples, mas eficiente, que se baseia em medidas de similaridade para classificar pontos de dados, tornando-o adequado para lidar com documentos textuais onde o reconhecimento de campos e valores é necessário [36].

No XFile, o KNN pode ser aplicado através da seleção de um conjunto de recursos que representam efetivamente o conteúdo dos documentos de faturação (*DocumentReference* na secção 4.3), sendo eles indexados para que as vantagens apresentadas na secção 3.3 possam ser aproveitadas. Através dessa base prévia de conhecimento o KNN tem então um conjunto de treinamento, que através dele consegue identificar características relevantes associadas aos campos e valores, então, quando um novo documento é submetido o algoritmo calcula a distância entre esses novos campos e valores com os dados existentes no conjunto de treinamento para classificar e identificar os campos e valores de acordo com sua parametrização, ou seja, se este novo campo pode ou não ser associado aos campos e valores válidos, ou deixa-lo para revisão do utilizador [37].

3.5 Requisitos

Esta secção apresenta os requisitos que orientam o desenvolvimento do *XFile*, definidos com base nos objetivos funcionais e não funcionais do projeto. As funcionalidades a serem implementadas são formalizadas em dois grupos distintos: os Requisitos Funcionais (RF), que representam as ações esperadas do sistema em termos de comportamento e funcionalidades; e os Requisitos Não Funcionais (RNF), que descrevem atributos de qualidade, restrições técnicas e operacionais que devem ser observados ao longo da implementação.

3.5.1 Requisitos Funcionais

Os requisitos funcionais do XFile estão relacionados a seguir:

RF-1. Gestão de utilizadores

O XFile deve permitir a gestão de utilizadores em sua base mediante as permissões concedidas, abrangendo as funcionalidades de criação, edição, listagem e exclusão. Deverão ser preenchidos os campos de utilizador, a empresa ao qual está relacionado, senha e informações pessoais, sendo que cada utilizador é único e necessita de estar associado a uma empresa responsável.

RF-2. Permitir o *login* de um utilizador

O XFile deve permitir que utilizadores registados façam o *login* no sistema a utilizar email e senha. Todas funcionalidades do sistema só poderão ser utilizadas após o *login*, que irá fornecer um *token* de autenticação/sessão que deverá ser fornecido em todas requisições de dados.

RF-3. Gestão de permissões

O XFile deve permitir que seja atribuído ou retirado permissões de acesso aos conteúdos e funcionalidades, seja por cargo, categoria de utilizadores dentro de uma empresa ou até a um funcionário em específico.

RF-4. Submissão de documentos

O XFile deve permitir que utilizadores possam submeter documentos de faturas para seu respetivo processamento, podendo esse ficheiro ser XML, PDF ou JSON. Juntamente, o utilizador poderá categorizar os documentos de acordo com as especificações previamente fornecidas pelo utilizador.

RF-5. Gestão de categorias de documentos

O XFile deve permitir que utilizadores cadastrem diferentes categorias de documentos, como por exemplo: fatura simples e fatura completa.

RF-6. Integração com serviços externos ao GVE Online

O XFile deve expor uma interface de integração padronizada (por exemplo, Application Programming Interfaces with Representational State Transfer (APIs REST) seguras), permitindo que parceiros e sistemas externos utilizem suas funcionalidades de importação, processamento e exportação de documentos de forma autónoma, sem depender do contexto do GVE Online.

RF-7. Categorizar os conteúdos de ficheiros submetidos

O XFile deve categorizar os conteúdos dos ficheiros submetidos entre 2 grupos: campos validados, campos não validados.

RF-8. Permitir a recuperação dos documentos submetidos

O XFile deverá permitir que utilizadores busquem documentos submetidos utilizando por exemplo parte do conteúdo do documento, nome, ou sua seleção em uma lista dos documentos já submetidos.

RF-9. Permitir a edição dos documentos submetidos

O XFile deverá permitir que utilizadores editem documentos recuperados.

RF-10. Permitir a gestão pelas empresas de seus funcionários

O XFile deve permitir que os funcionários de uma empresa possam ser categorizados por suas posições funcionais nas empresas.

RF-11. Permitir a gestão localidades

O XFile deve permitir que os utilizadores possam gerir as localizações cadastradas internamente no sistema.

3.5.2 Requisitos Não Funcionais

Os Requisitos não funcionais do XFile estão relacionados a seguir:

RNF-1. Linguagem de programação

O XFile deverá ser escrito em Java.

RNF-2. Banco de dados

O XFile deverá utilizar o banco de dados MongoDB.

RNF-3. Aberto a novas implementações

O XFile deverá ser flexível para a implementação de novas abordagens de indexação e validação sem a necessidade de alterações nas interfaces de comunicação.

RNF-4. Framework

O XFile deverá ser desenvolvido com o Spring/SpringBoot.

RNF-5. Motor de busca

O XFile deverá utilizar Apache Lucene como motor de busca documental.

RNF-6. Ambiente de execução

O XFile deve ser executado em um ambiente isolado, consistente e facilmente replicável, utilizando containers Docker.

RNF-7. Idioma de documentos

O XFile deverá suportar Português como linguagem principal durante esta fase do desenvolvimento.

3.6 Casos de Uso

Os casos de uso do XFile descrevem os cenários essenciais em que os diferentes atores como administradores, utilizadores finais e sistemas externos interagem com a plataforma para realizar tarefas críticas de gestão, processamento e disponibilização de documentos. Cada caso de uso ilustra o fluxo de eventos principal, as pré-condições necessárias, as validações envolvidas e os resultados esperados, garantindo clareza sobre o comportamento do sistema em situações reais.

Ao mapear esses cenários, asseguramos a rastreabilidade das funcionalidades definidas nos requisitos funcionais e não funcionais, destacando pontos de integração com o GVE Online, a interface de APIs para serviços externos e as etapas de validação pelo utilizador. Esta visão orienta a implementação prática do XFile, servindo como referência para testes, documentação e futuras evoluções da plataforma.

No XFile existem dois atores principais: o GVE Online e o utilizador que pode interagir diretamente com o XFile, fora do escopo direto do GVE Online. O fluxo de acesso às funcionalidades do XFile obedece a um procedimento estrito de cadastro e autenticação:

1. A Empresa Parceira submete-se ao registo inicial no GVE Online e aguarda aprovação administrativa.

2. Após a validação, um utilizador padrão é criado no XFile, com o perfil mínimo necessário para gerir contas internas.

3. Esse utilizador padrão pode então registar colaboradores da própria Empresa Parceira e atribuir-lhes permissões granulares conforme as suas responsabilidades.

4. Para cenários em que a Empresa Parceira requer comunicação direta com o XFile via integração externa, é acordado com a equipa TECH X o estabelecimento de quais endereços/portas serão utilizados pela empresa para aceder (endereço de origem) ao XFile e uma chave de cifragem simétrica. Essa chave garante confidencialidade e integridade dos dados trocados.

5. Com a chave de cifragem provisionada e as credenciais do utilizador padrão, a Empresa Parceira passa a aceder a todo o conjunto de funcionalidades do XFile.

Para simplificar a apresentação dos "Casos de Uso", consideramos este processo de onboarding e configuração inicial pré-concluído, permitindo focar diretamente nos cenários de operação do XFile. Abaixo são expostos em diagramas Unified Modeling Language (UML) os casos de usos do XFile.

3.6.1 Caso de Uso: Gerir Empresas

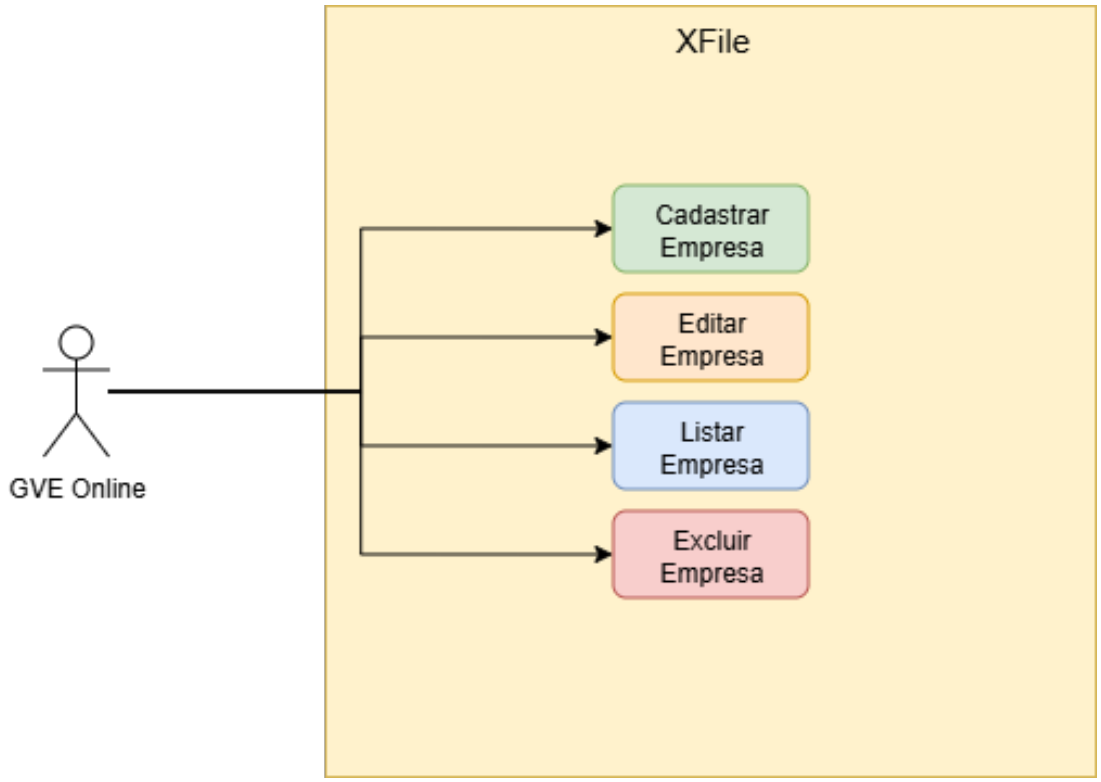

Figura 3.1: Casos de uso Gerir Empresas.

Conforme na figura 3.1 é possível gerir empresas com os seguintes requisitos:

Pré-condição: O GVE Online tem acesso e controle total ao sistema.

Ator: GVE Online.

Objetivo: Cadastrar, editar, listar e excluir empresas.

Fluxo principal:

1. Selecionar ação (cadastrar, editar, listar ou excluir)
2. Preencher dados ou confirmar exclusão
3. Sistema valida e mostra confirmação

Pós-condição: Base de dados de empresas atualizada com a operação realizada.

3.6.2 Caso de Uso: Gerir Planos da Empresa

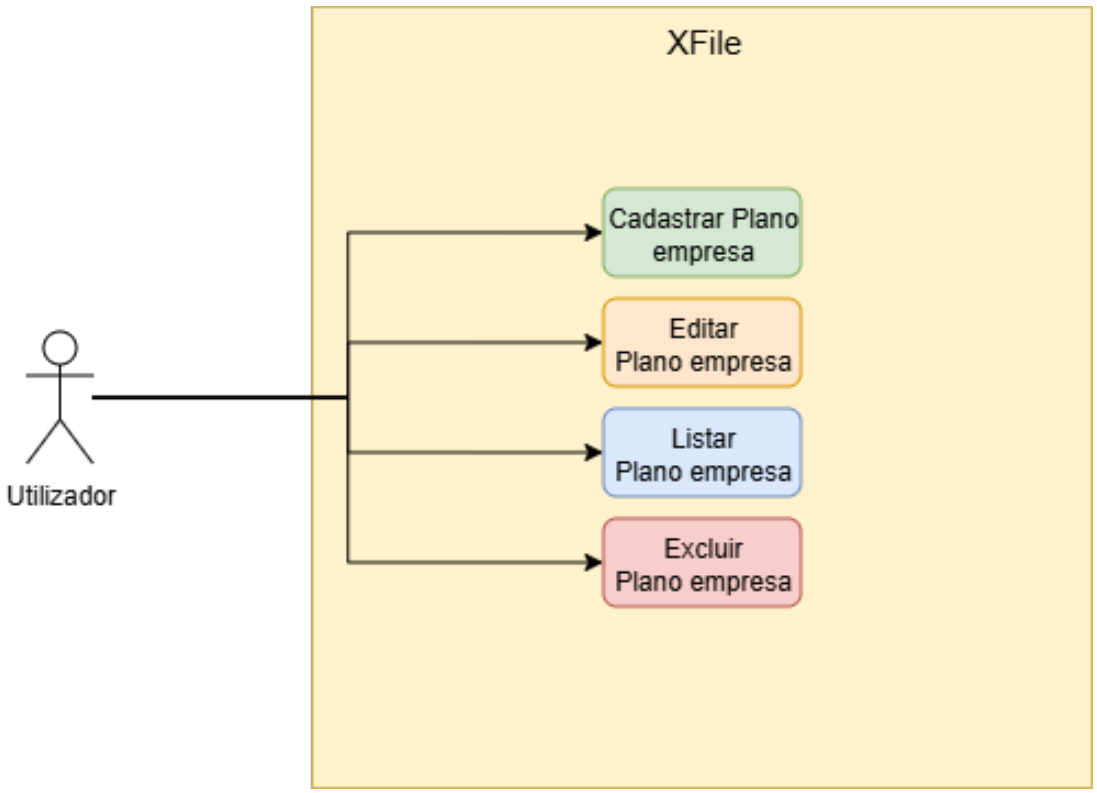

Figura 3.2: Casos de uso gerir planos da empresa.

Conforme na figura 3.2 é possível gerir planos da empresa com os seguintes requisitos:

Pré-condição: Utilizador com token Java Web Token (JWT) de sessão válido e com as devidas permissões.
Ator: Utilizador.
Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.
Fluxo principal:

1. Realizar login de sessão
2. Selecionar ação (cadastrar, editar, listar ou excluir)
3. Preencher dados ou confirmar exclusão
4. Sistema valida e mostra confirmação

Pós-condição: Base de dados de planos da empresa atualizada com a operação realizada.

3.6.3 Caso de Uso: Gerir visualizações da empresa

Conforme na figura 3.3 é possível gerir visualizações da empresa com os seguintes requisitos:
Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.
Ator: Utilizador.
Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.
Fluxo principal:

1. Realizar login de sessão
2. Selecionar ação (cadastrar, editar, listar ou excluir)
3. Preencher dados ou confirmar exclusão
4. Sistema valida e mostra confirmação

Pós-condição: Base de dados de visualizações da empresa atualizada com a operação realizada.

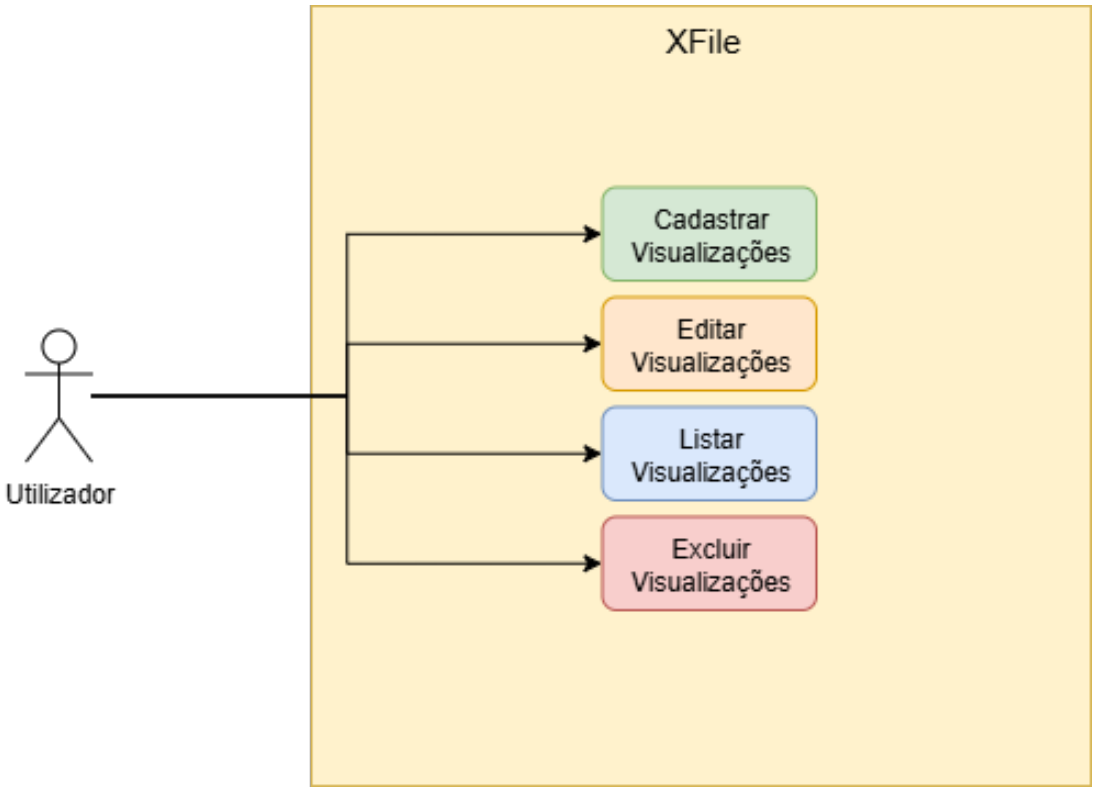

Figura 3.3: Casos de uso gerir visualizações da empresa.

3.6.4 Caso de Uso: Gerir Utilizadores

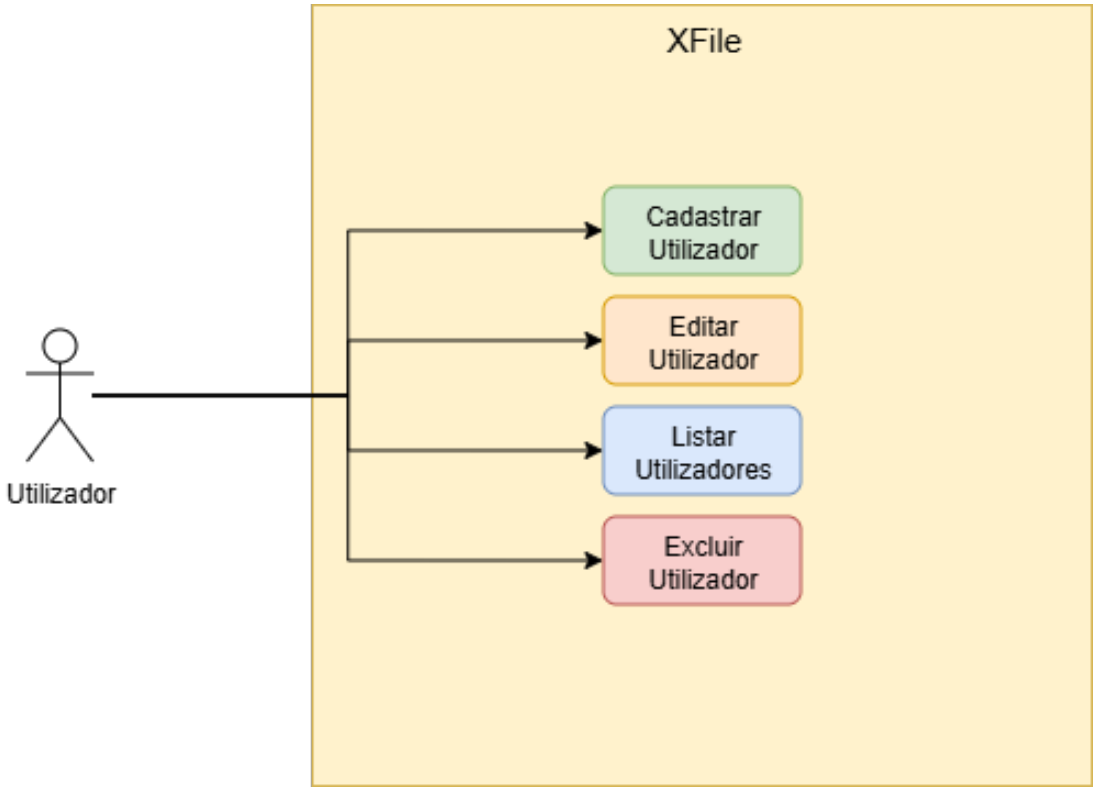

Figura 3.4: Casos de uso Gerir utilizadores.

Conforme na figura 3.4 é possível gerir utilizadores com os seguintes requisitos:

Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.

Ator: Utilizador.

Objetivo: Cadastrar, editar, listar e excluir utilizadores.

Fluxo principal:

1. Realizar login de sessão
2. Selecionar ação (cadastrar, editar, listar ou excluir)
3. Preencher dados ou confirmar exclusão

4. Sistema valida e mostra confirmação

Pós-condição: Base de dados de utilizadores atualizada com a operação realizada.

3.6.5 Caso de Uso: Gerir Cargos de Utilizadores

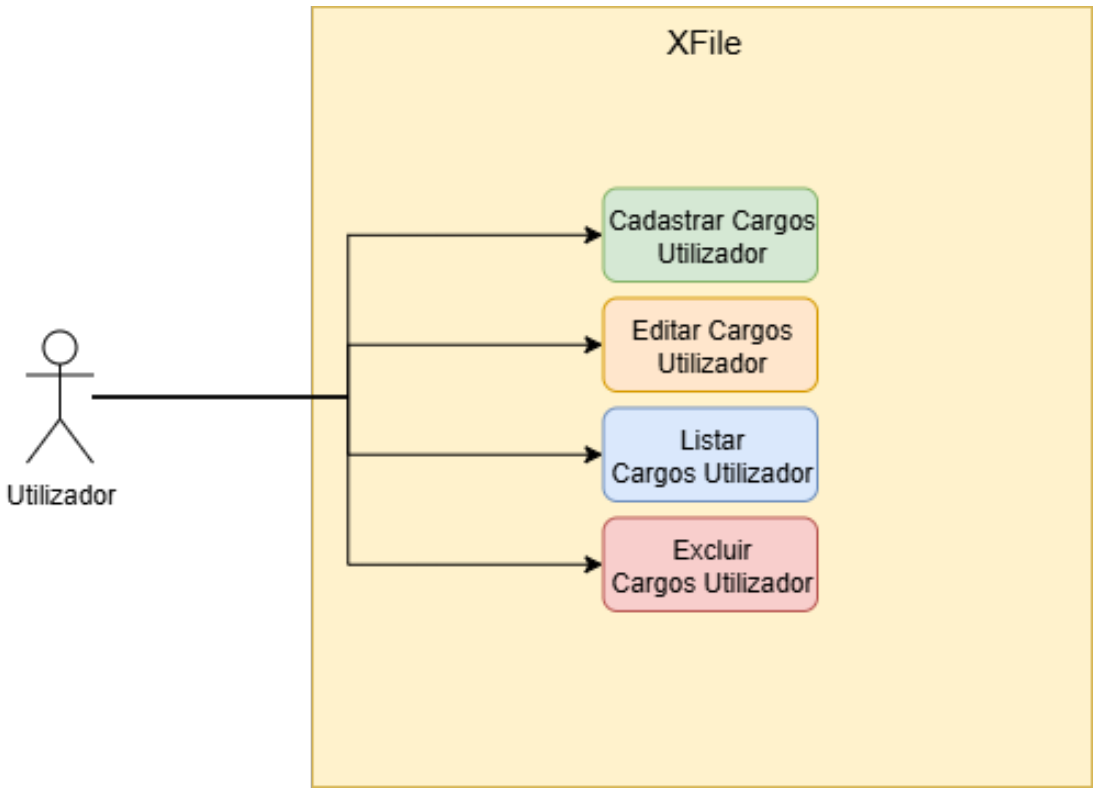

Figura 3.5: Casos de uso Gerir cargos utilizadores.

Conforme na figura 3.5 é possível gerir cargos de utilizadores com os seguintes requisitos:

Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.

Ator: Utilizador.

Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.

Fluxo principal:

1. Realizar login de sessão

2. Selecionar ação (cadastrar, editar, listar ou excluir)

3. Preencher dados ou confirmar exclusão

4. Sistema valida e mostra confirmação

Pós-condição: Base de dados de cargos de utilizadores atualizada com a operação realizada.

3.6.6 Caso de Uso: Gerir Permissões de acesso

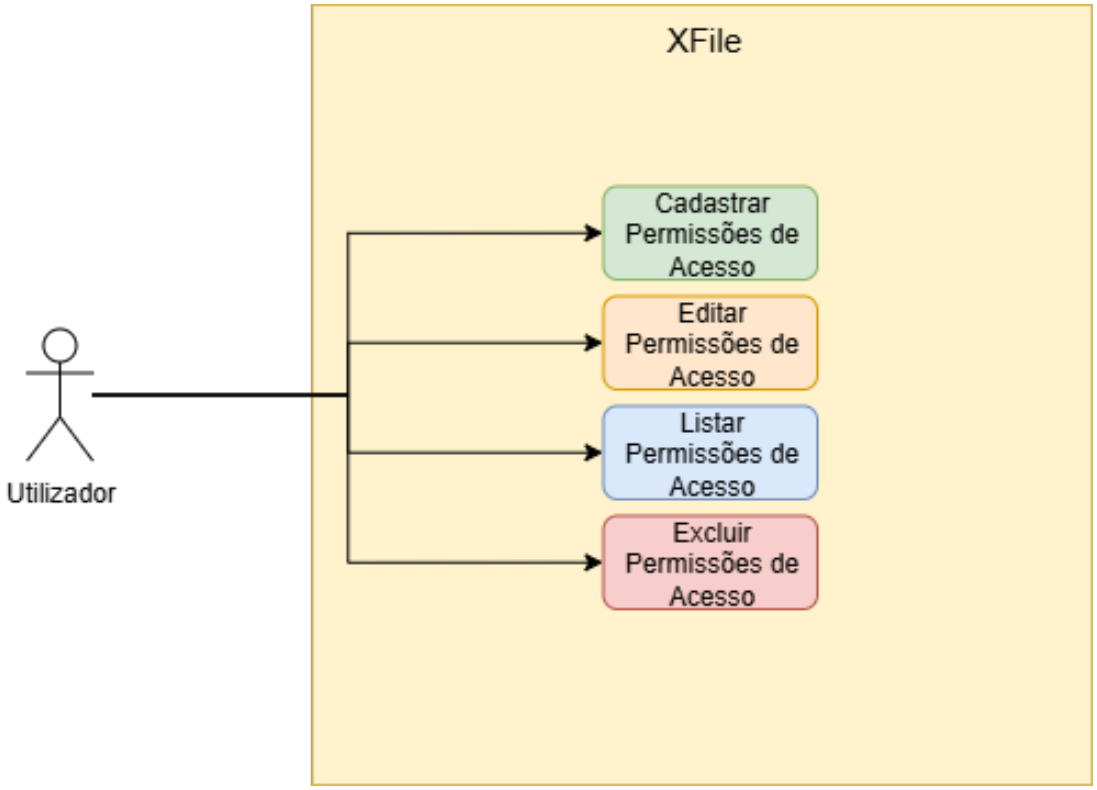

Figura 3.6: Casos de uso Gerir permissões de acesso.

Conforme na figura 3.6 é possível gerir permissões de acesso com os seguintes requisitos:

Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.

Ator: Utilizador.

Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.

Fluxo principal:

1. Realizar login de sessão
2. Selecionar ação (cadastrar, editar, listar ou excluir)
3. Preencher dados ou confirmar exclusão
4. Sistema valida e mostra confirmação

Pós-condição: Base de dados de permissões de acesso atualizada com a operação realizada.

3.6.7 Caso de Uso: Gerir Permissões de serviços

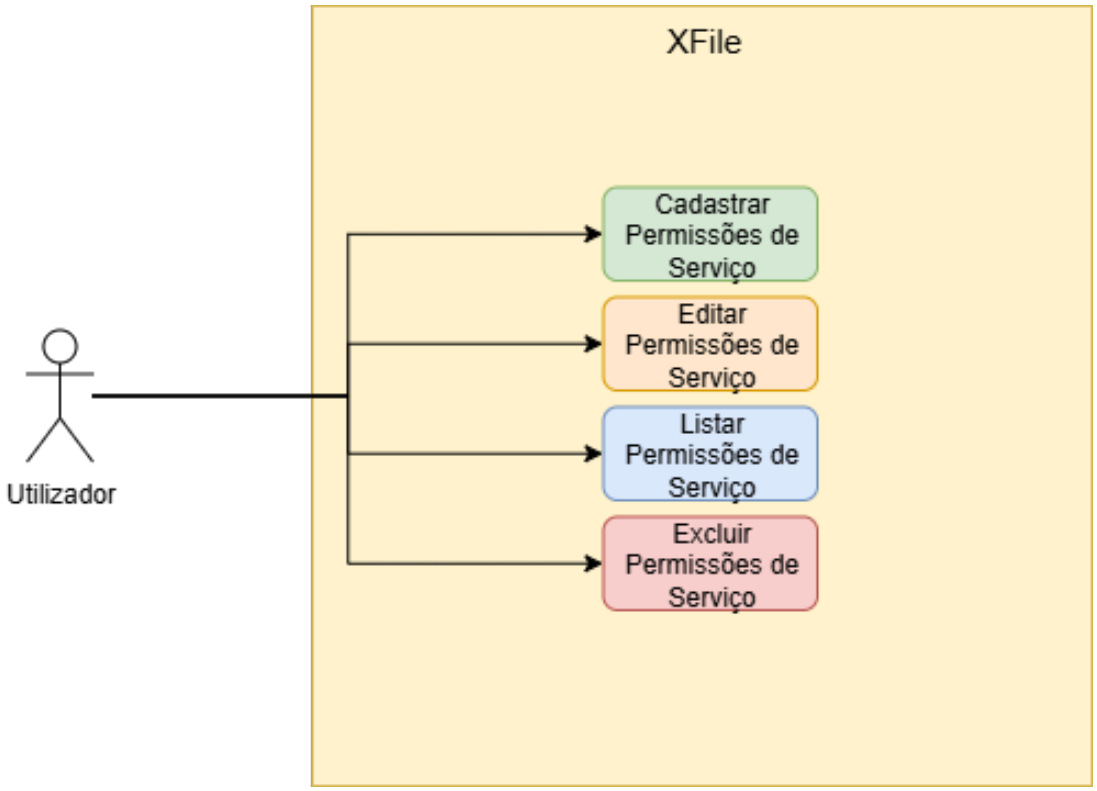

Figura 3.7: Casos de uso gerir permissões de serviços.

Conforme na figura 3.7 é possível gerir permissões de serviços com os seguintes requisitos:

Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.

Ator: Utilizador.

Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.

Fluxo principal:

1. Realizar login de sessão
2. Selecionar ação (cadastrar, editar, listar ou excluir)
3. Preencher dados ou confirmar exclusão
4. Sistema valida e mostra confirmação

Pós-condição: Base de dados de permissões de serviços atualizada com a operação realizada.

3.6.8 Caso de Uso: Gerir localizações da empresa

Conforme na figura 3.8 é possível gerir localizações da empresa com os seguintes requisitos:
Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.
Ator: Utilizador.
Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.
Fluxo principal:

1. Realizar login de sessão
2. Selecionar ação (cadastrar, editar, listar ou excluir)
3. Preencher dados ou confirmar exclusão
4. Sistema valida e mostra confirmação

Pós-condição: Base de dados de localizações da empresa atualizada com a operação realizada.

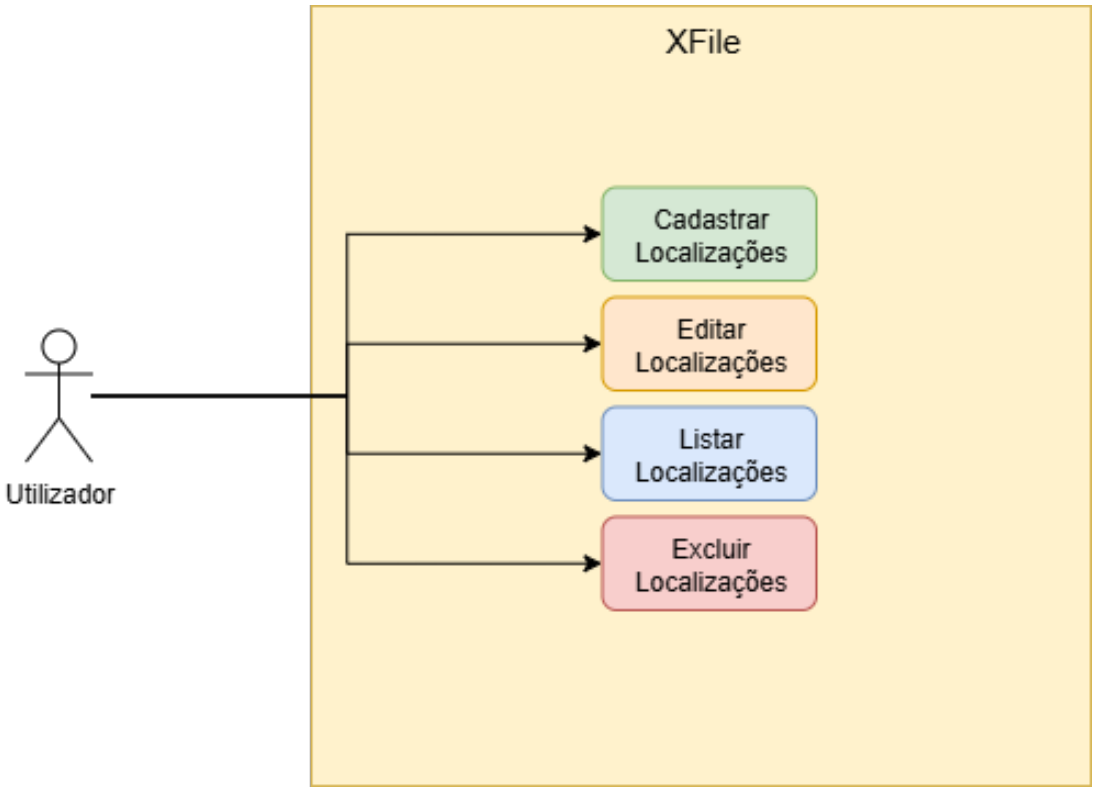

Figura 3.8: Casos de uso gerir localizações da empresa.

3.6.9 Caso de Uso: Gerir documentos da empresa

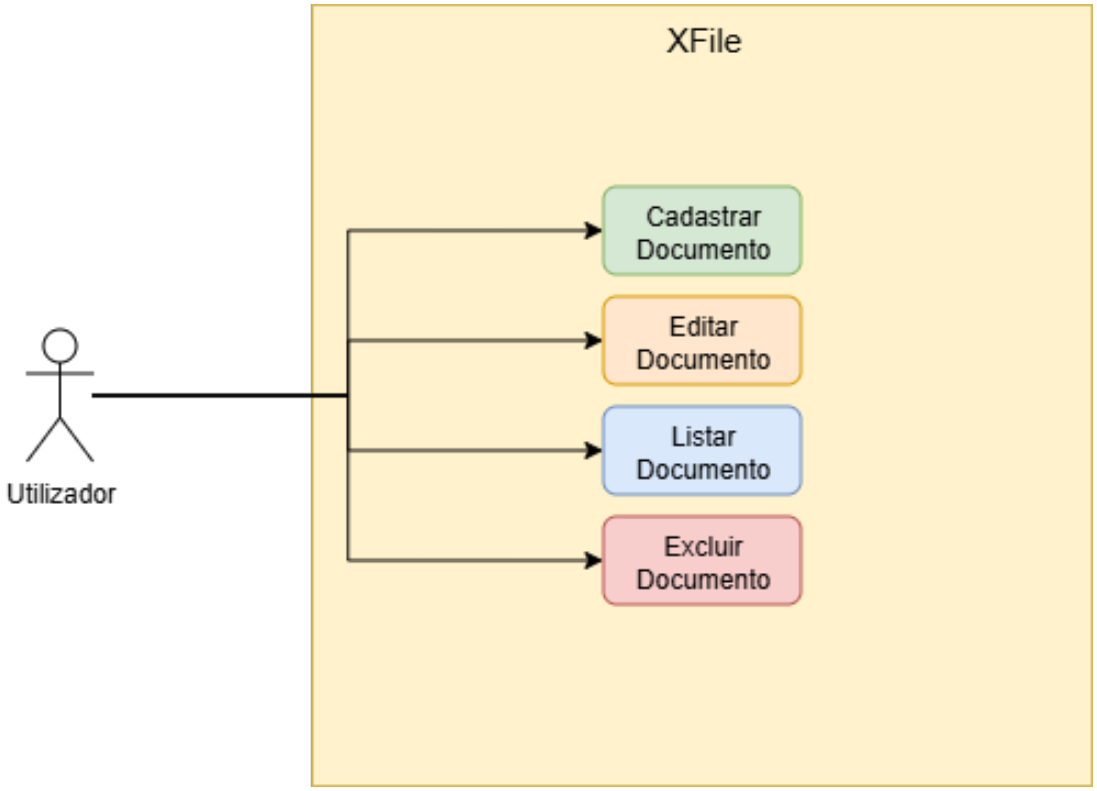

Figura 3.9: Casos de uso gerir documentos da empresa.

Conforme na figura 3.9 é possível gerir documentos da empresa com os seguintes requisitos:

Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.

Ator: Utilizador.

Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.

Fluxo principal:

1. Realizar login de sessão

2. Selecionar ação (cadastrar, editar, listar ou excluir)

3. Preencher dados ou confirmar exclusão

4. Sistema valida/processa e mostra confirmação

Pós-condição: Base de dados de documentos da empresa atualizada com a operação realizada.

3.6.10 Caso de Uso: Gerir tipos de documentos da empresa

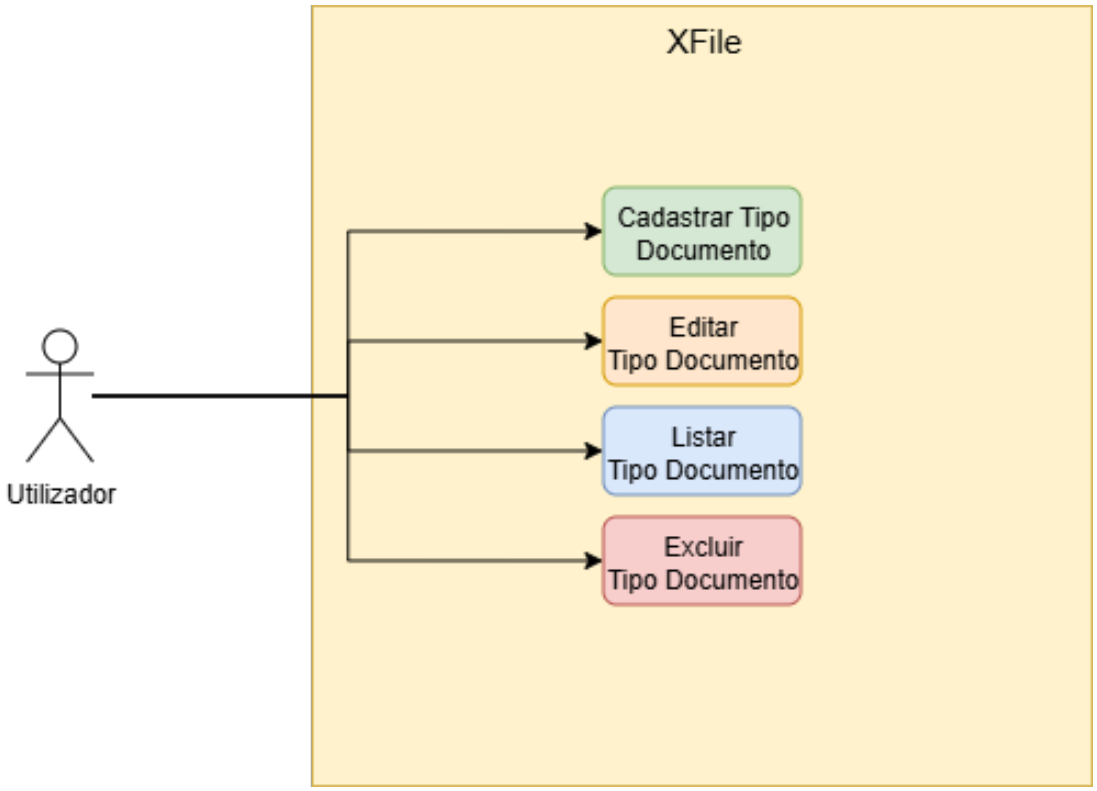

Figura 3.10: Casos de uso gerir tipos de documentos da empresa.

Conforme na figura 3.10 é possível gerir tipos de documentos da empresa com os seguintes requisitos:

Pré-condição: Utilizador com token JWT de sessão válido e com as devidas permissões.

Ator: Utilizador.

Objetivo: Cadastrar, editar, listar e excluir cargos de utilizadores.

Fluxo principal:

1. Realizar login de sessão
2. Selecionar ação (cadastrar, editar, listar ou excluir)
3. Preencher dados ou confirmar exclusão
4. Sistema valida/processa e mostra confirmação

Pós-condição: Base de dados de tipos de documentos da empresa atualizada com a operação realizada.

No capítulo 4, exploraremos a concretização dos casos de uso, descrevendo a arquitetura adotada, as tecnologias escolhidas, as estratégias de implementação e apresentaremos os resultados obtidos do XFile no atendimento aos objetivos.

Capítulo 4

Desenvolvimento e Resultados

Neste capítulo é descrito o trabalho de implementação de toda a arquitetura que compõe o serviço XFile, desenvolvido pela empresa TECH X, voltado para o tratamento e estruturação de documentos digitais em formato XML, PDF, JSON. Serão apresentadas as principais ferramentas utilizadas no desenvolvimento, bem como os detalhes da arquitetura adotada para garantir escalabilidade, flexibilidade e eficiência no processamento documental.

4.1 Ambiente de Trabalho

Nesta secção serão apresentadas as ferramentas utilizadas no desenvolvimento do projeto.

4.1.1 Intellij IDEA

Para o desenvolvimento do serviço XFile, foi escolhida a ferramenta *IntelliJ IDEA*, uma das Integrated Development Environment (IDE) mais completas e avançadas do mercado, amplamente utilizada para o desenvolvimento de aplicações baseadas em Java e outras linguagens. A IDE dispõe de ferramentas robustas de refatoração inteligente, detecção de erros em tempo real, integração com sistemas de controle de versão com git, suporte a plugins como validação de XML e MongoDB. Alem de compatibilidade com ferramentas

de automação e integração continua como Docker e Kubertes. Com base nas vantagens apresentadas o Intellij IDEA, torna-se uma ferramenta estratégica para garantir o um bom fluxo de desenvolvimento e integração direta com o ecossistema do XFile [38].

4.1.2 Git

O Git é um sistema de controle de versão distribuído desenvolvido por Linus Torvalds em 2005. Esse sistema mantém um histórico completo de modificações e é possível obter uma cópia de qualquer momento deste histórico. Assim, temos um sistema de versionamento altamente disponível e alta capacidade de restauração [39].

Todo software em desenvolvimento necessita de versões, seja ela definida em números ou nomes exclusivos. Assim o software pode oferecer um maior detalhamento de seu estado ao longo do tempo para todos que estão envolvidos em seu desenvolvimento. Por exemplo, é possível ver quando uma nova funcionalidade foi adicionada ou uma biblioteca foi removida [40].

4.1.3 Gitea

A Gitea [41] é uma plataforma leve, rápida e de código aberto para hospedagem de repositórios Git, desenvolvida com foco em simplicidade e eficiência. Escrita em linguagem Go, é compatível com diversos sistemas operativos, como Linux, macOS e Windows, e pode ser facilmente implantada em servidores locais ou ambientes *cloud*.

Originalmente derivado do projeto Gogs, o Gitea evoluiu para oferecer uma solução completa de desenvolvimento colaborativo, com o suporte a criação e gestão de repositórios Git com suporte a ramificações, *tags*, histórico de *commits* e permissões de acesso. Suporte a *pull requests*, comentários em linha e integração com fluxos de trabalho colaborativos. Organização de tarefas, bugs e funcionalidades através de *issues*, etiquetas, quadros Kanban, marcos e dependências. Com o *Gitea Actions* também é possível configurar *pipelines* utilizando sintaxe compatível com *GitHub Actions*. Gestão de permissões, autenticação de utilizadores, integração com Lightweight Directory Access Protocol (LDAP) e suporte

a *webhooks* para automação.

A leveza e facilidade de manutenção tornam o Gitea ideal para equipas pequenas ou médias que desejam autonomia na gestão de seus repositórios, sem depender de plataformas externas. No contexto do desenvolvimento do serviço XFile, o Gitea foi utilizado como ferramenta central para versionamento de código, colaboração entre desenvolvedores da TECH X e integração com sistemas de automação e monitoramento.

4.1.4 Apache Guacamole

Apache Guacamole [42] é uma plataforma de acesso remoto *clientless*, ou seja, não requer a instalação de software adicional no dispositivo do utilizador. Baseado em tecnologias web como HyperText Markup Language (HTML), permite que qualquer máquina com navegador moderno aceda a ambientes remotos através de protocolos como Remote Desktop Protocol (RDP), Virtual Network Computing (VNC) e Secure Shell (SSH).

A arquitetura do Guacamole é composta por dois principais componentes: o *guacd*, responsável por traduzir os protocolos de acesso remoto para o protocolo interno do Guacamole, e a aplicação web, que fornece a interface gráfica ao utilizador. Essa separação permite que os ambientes remotos permaneçam isolados e seguros, enquanto continuam acessíveis via navegador [43].

Contribuição para o Projeto XFile

No contexto do desenvolvimento do serviço XFile, o Apache Guacamole desempenha um papel estratégico ao permitir que os desenvolvedores da TECH X acedam de forma segura e centralizada aos ambientes de desenvolvimento e servidores de testes, independentemente da localização física. Entre os principais benefícios destacam-se:

- **Acesso remoto seguro:** Utilização de autenticação integrada e isolamento de rede para proteger os ambientes de desenvolvimento.
- **Facilidade de manutenção:** Permite que administradores acedam a servidores e containers sem depender de clientes RDP ou SSH locais.

- **Mobilidade e flexibilidade:** Desenvolvedores podem trabalhar remotamente com acesso completo às ferramentas e ambientes necessários para o desenvolvimento do XFile.

- **Integração com autenticação personalizada:** Possibilidade de adaptar o sistema de login do Guacamole às políticas internas da TECH X.

A adoção do Apache Guacamole contribui para a eficiência operacional da equipa de desenvolvimento, reduzindo barreiras técnicas e promovendo um ambiente de trabalho mais ágil e seguro.

4.2 Arquitetura

Durante o processo de criação e desenvolvimento de software, é fundamental estabelecer previamente os padrões de arquitetura do projeto e a arquitetura de código. No caso do XFile, sua proposta central é integrar-se ao ecossistema de serviços oferecidos pela TECH X, tendo como principal referência o GVE Online, conforme discutido no capítulo 3. As subsecções seguintes detalham, respetivamente, a arquitetura de projeto e a arquitetura de código adotadas no desenvolvimento do XFile, considerando os padrões e diretrizes estabelecidos pela TECH X.

4.2.1 Arquitetura de projeto

Esta subsecção descreve a arquitetura de projeto adotada no desenvolvimento do XFile (figura 4.1), evidenciando os principais componentes, suas interações e as decisões estruturais que orientaram a construção da solução.

Como é possível observar na figura 4.1 o XFile é divido em 4 camadas: *Presentation*, *App Services*, *Computing*, *Computing Services*, *External Services*. Abaixo a descrição de cada item contido em seus respetivos módulos.

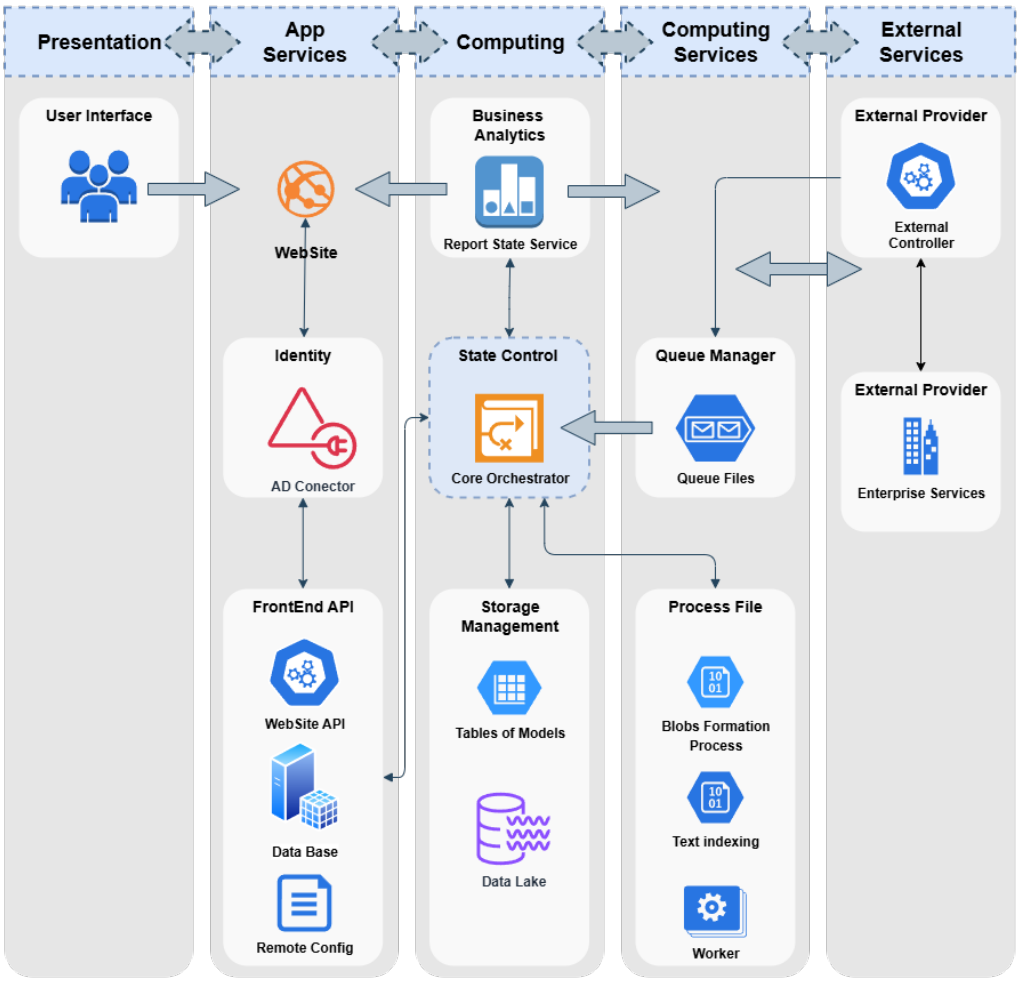

Figura 4.1: Arquitetura de Projeto XFile.

Presentation

A camada *Presentation* é responsável por fornecer acesso aos utilizadores cadastrados, seguindo o modelo de negócio previamente definido. Seu objetivo é atender às necessidades específicas do negócio, oferecendo uma interface amigável e funcional para os utilizadores interagirem com o sistema, podendo ser denominada de GVE Online.

App Services

A Camada *App Services* é responsável pelo fornecimento de recursos, gestão de permissões e estruturação de dados para camada de Apresentação. Considerando o processo de desenvolvimento ainda corrente do GVE Online, alguns serviços como *Active Directory* serão integrados futuramente conforme o processo de desenvolvimento e migração do GVE para o GVE Online.

- **AD Conector:**Desempenha um papel fundamental na arquitetura do XFile para o desenvolvimento e unificação de acessos para o GVE Online, sendo responsável pela autenticação e validação dos utilizadores. Atualmente, o XFile utiliza-se do serviço de autenticação interno, baseado em *Spring Security* para controlo de sessão. A autenticação é realizada por meio de tokens JWT, assinados com o algoritmo *HS256* — uma técnica que combina Hash-Based Message Authentication Code (HMAC) com *SHA-256*, utilizando uma chave secreta para garantir a integridade e autenticidade dos tokens.

- **FrontEnd API:** Responsável por fornecer aos utilizadores acesso a informações, incluindo a submissão de documentos, visualização de processamentos em andamento e já concluídos, bem como outras operações definidas nos requisitos funcionais da secção 3.5.1. Também assume a responsabilidade pelo gerenciamento de dados ao nível do utilizador e pela parametrização do *Orchestrator* na camada de *Computing*.

Computing

A camada *Computing* isola as responsabilidades entre *App Services* e *Computing Services*, gerencia o fluxo de dados entre elas e fornece interfaces para módulos de armazenamento e análise. Isso garante maior controle de recursos, segurança e disponibilidade, sem comprometer a coesão das integrações.

- **Business Analytics:** Responsável por fornecer informações do estado do *Core Orchestrator* (Orquestrador central), para tomadas de decisões de recursos, monitoramento e notificações aos utilizadores, de acordo com as definições fornecidas pela camada de serviços de aplicação.
- **State Control:** É possível observar, na figura 4.1, o destaque intencional dado a este serviço, por se tratar do *Core Orchestrator* (Orquestrador Central). Este componente é responsável pelo controlo de recursos, pela gestão dos ficheiros que podem ou não ser processados em paralelo pela delegação ao serviço *Worker* adequado para o processamento dos ficheiros de entrada dos Parceiros, bem como pelo armazenamento dos ficheiros de entrada e saída no serviço *Store Management* (Gerenciador de Armazenamento), de forma a garantir suporte a auditorias e validações de possíveis erros.
- **Storage Management:** Serviços responsáveis por prover o armazenamento de configurações específicas ao *Core Orchestrator* (Orquestrador central) e o Data Lake para o armazenamento e gerenciamento dos ficheiros recebidos e gerados após o processo de validação do ficheiro para seguimento do fluxo.

Computing Services

A Camada *Computing Services* tem como responsabilidade garantir entrega dos dados à camada de *Computing* e quando solicitada, estruturar os documentos para então processar de acordo com os parâmetros disponibilizados pelo *Core Orchestrator* e por fim, devolver seus resultados para o *Core Orchestrator*.

- **Queue Manager:** Responsável por garantir a entrega, rastreabilidade e confiabilidade no processamento dos ficheiros recebidos e enviados, mesmo que sob grande carga de requisições. A adoção desse componente tem como justificativa a necessidade garantir a disponibilidade dos serviços mesmo sob grande fluxo de envio de documentos. .

- **Process Files:** Responsável por realizar a sanitização e mapeamento dos ficheiros de acordo com os modelos, para que o *worker* designado pelo orquestrador possa realizar a extração dos dados, devolvendo ao *Core Orchestrator* o resultado.

External Services

Responsável por disponibilizar a parceiros estratégicos da TECH X o acesso às funcionalidades do XFile, mediante o fornecimento prévio de credenciais de autenticação.

- **External Controller:** Responsável por disponibilizar as funcionalidades da aplicação e definir as especificações das interfaces de comunicação externas, com base em princípios RESTful. Atua de forma semelhante à *FrontEnd API*, mas com foco específico no fornecimento de serviços a parceiros estratégicos, por meio de contratos de API bem definidos e aderentes aos padrões de integração estabelecidos pela TECH X.

- **External Provider:** Responsável por habilitar a comunicação independente com os parceiros, por meio de interfaces que podem operar de forma síncrona (ex.: requisições Hypertext Transfer Protocol (HTTP)) ou assíncrona (ex.: filas de mensagens via WebSocket ou Advanced Message Queuing Protocol (AMQP)), conforme as especificações definidas nos contratos de integração.

4.2.2 Arquitetura de Código

No contexto de arquitetura de código, esta subsecção aborda princípios e padrões que orientam a construção do XFile, como *Clean Architecture*, Domain-Driven Design (DDD),

princípios **SOLID** e os *Design Patterns* utilizados. Tais abordagens promovem a separação clara de responsabilidades, facilitam a testabilidade e reduzem o acoplamento entre módulos, contribuindo para uma base sólida que sustenta a evolução contínua do XFile.

Clean Architecture

Assim como em outras abordagens arquiteturais, a principal proposta da *Clean Architecture* é promover a separação clara de responsabilidades dentro do sistema. Essa arquitetura organiza o código em camadas concêntricas, cada uma com um nível distinto de abstração, de modo que as regras de negócio — representadas pelo domínio — ocupem o núcleo da aplicação. O projeto é orientado ao domínio, evitando dependências diretas com elementos externos como *frameworks*, bancos de dados ou interfaces de utilizadores. Essa estrutura favorece a independência, testabilidade e flexibilidade, permitindo que mudanças em tecnologias externas não impactem o coração da aplicação.

Contextualizando com o XFile, e corroborando com a arquitetura de projeto subsecção 4.2.1, a aplicação de *Clean Architecture* com as tecnologias utilizadas (Secção 2.4), facilita a integração do XFile na infraestrutura atual da TECH X, posteriormente a fácil integração com o GVE Online em uma infraestrutura *cloud*. No XFile os módulos estão dispostos da seguinte forma:

- **Controller:** Responsável por receber requisições externas de acordo com os requisitos funcionais definidos no projeto (Subsecção 3.5.1), validar dados de requisição, como por exemplo, se os atributos ou campos estão de acordo com o esperado pelo *controller* utilizado, e então delegar ao caso de uso correspondente e por fim ao obter a resposta do caso de uso, adequar para uma resposta coerente ao utilizador.

- **Configurations:** Responsável por configurar e inicializar os componentes externos que a aplicação precisa para funcionar. Ele não contém lógica de negócio, mas obtém dados necessários definidos nas propriedades do projeto para que a aplicação possa operar corretamente, como portas, endereços, chaves privadas e limite de

processos paralelos. Como por exemplo: Segurança, Banco de Dados, Mensageria, Armazenamentos, Serialização e Monitoramento.

- **Core:** Responsável pela coordenação central das operações documentais, incluindo o controlo de estado de processamento, a delegação de tarefas aos módulos de processamento (*Workers*) e a execução dos casos de uso relacionados ao acesso e persistência de dados. Atua como elo entre o domínio da aplicação e os serviços externos, garantindo que as regras de negócio sejam aplicadas de forma consistente e independente das tecnologias subjacentes.

- **Entities:** Representa os conceitos fundamentais e duradouros do domínio da aplicação. Contém as entidades de negócio, *enums* semânticos e objetos de valor que expressam regras invariantes e estruturas essenciais. Esta camada é totalmente independente de tecnologias externas e serve como base para os casos de uso e orquestrações definidos no módulo Core.

- **Repository:** Responsável por adaptar os modelos e operações de persistência da aplicação para o mecanismo de armazenamento utilizado (ex.: MongoDB). Contém interfaces de repositório, conversores entre entidades e modelos de banco, e os próprios modelos persistentes. Atua como camada intermediária entre os casos de uso e a infraestrutura, garantindo que o domínio permaneça desacoplado da tecnologia de persistência.

- **Services:** Responsável por encapsular integrações e funcionalidades externas à aplicação, como chamadas a API (Integração com o GVE), envio e recuperação de dados ao MinIO, processamento assíncrono da extração inicial dos dados de documentos enviados ao XFile. Contém interfaces e implementações de serviços que podem ser utilizados pelos casos de uso do Core, mantendo o domínio desacoplado das tecnologias externas e *Workers*.

- **UseCases:** Responsável por implementar os casos de uso da aplicação, coordenando a execução de operações de negócio com base nas regras definidas no domínio. Neste

módulo, são aplicados padrões como *Factory* e *Strategy* para selecionar e instanciar dinamicamente os *Workers* apropriados para o processamento de documentos de acordo com suas características. Atua como elo entre o Core e os serviços técnicos, garantindo que a lógica de aplicação seja executada de forma flexível e extensível, possibilitando ao XFile não limitar-se a implementação atual que se propõe, mas sendo possível implementar novas abordagens, garantindo assim a sua evolução contínua.

Domain-Driven Design

O DDD é uma abordagem arquitetural centrada no domínio da aplicação, ou seja, nas regras de negócio e nos conceitos que representam a realidade que o sistema pretende modelar. No XFile, o DDD se tornou essencial para organizar e dar sentido aos módulos, pois permitiu que cada parte da arquitetura estivesse alinhada à linguagem do domínio, desde as entidades até os casos de uso e orquestradores.

Essa abordagem facilitou a evolução do sistema, permitindo que mudanças no domínio fossem incorporadas sem reestruturações invasivas na infraestrutura, facilitando a perceção da TECH X de que o XFile esta a evoluir de acordo com o contexto de negócio a que se propõe.

Princípios SOLID

Os princípios *SOLID* foram definidos por Martin [44], com o objetivo de tornar sistemas orientados a objetos mais compreensíveis, flexíveis e robustos. No projeto XFile, estes princípios ajudam a estruturar corretamente módulos como `UseCases`, `Core`, `Entities` e `Repositories`, promovendo desacoplamento e clareza no fluxo de responsabilidades.

- **S — Single Responsibility Principle (SRP)** Cada classe ou módulo deve ter uma única responsabilidade bem definida. No XFile, os *Workers* são responsáveis apenas pelo processamento específico de documentos, enquanto a orquestração reside no `Core`.

- **O — Open/Closed Principle (OCP)** Os componentes devem estar abertos para extensão, mas fechados para modificação. No XFile, por exemplo, o padrão `Strategy` permite adicionar novas abordagens de classificação sem alterar os *Use-Cases* existentes.

- **L — Liskov Substitution Principle (LSP)** Subtipos devem poder substituir os tipos base sem alterar o comportamento do programa. No XFile, diferentes implementações de `StorageService` podem ser trocadas por injeção sem quebrar a lógica de armazenamento, como por exemplo MinIO que utiliza-se da mesma interface de comunicação que o Amazon S3.

- **I — Interface Segregation Principle (ISP)** Interfaces devem ser específicas para seus clientes, evitando contratos grandes e genéricos. No XFile, serviços externos podem ser acessados através de interfaces dedicadas como `IExternalService`.

- **D — Dependency Inversion Principle (DIP)** Módulos de alto nível não devem depender de módulos de baixo nível, mas sim de abstrações. O XFile aplica este princípio por meio de interfaces e da inversão de dependência via Spring, como detalhado a seguir.

Inversão de Dependência na arquitetura

A inversão de dependência é um dos princípios fundamentais do SOLID, que propõe que módulos de alto nível (que contêm regras de negócio) não devem depender de módulos de baixo nível (como banco de dados ou bibliotecas externas), mas sim ambos dependerem de abstrações. Esse princípio visa criar sistemas desacoplados, flexíveis e mais fáceis de testar e manter.

No projeto *XFile*, esse conceito foi amplamente aplicado, com apoio direto do Spring Framework, que fornece ferramentas robustas de *Injeção de Dependência* e configuração baseada em anotações. A arquitetura foi desenhada de forma que os componentes centrais do domínio não têm conhecimento direto sobre implementações técnicas.

- As interfaces dos `Repositories` foram definidas no núcleo do domínio, enquanto suas implementações específicas (como acesso ao MongoDB) foram fornecidas externamente e injetadas pelos mecanismos do Spring.

- Os `UseCases` dependem de serviços declarados por abstrações (interfaces) sem conhecer detalhes da infraestrutura.

- O Spring é responsável pelo ciclo de vida dos componentes e realiza a injeção automática das dependências, permitindo a substituição fácil de implementações sem alterar o fluxo da aplicação.

Essa estratégia promoveu uma separação clara entre domínio e infraestrutura, possibilitando maior modularidade e testes unitários eficazes, além de permitir que novas tecnologias ou frameworks possam ser integrados sem impactar as regras de negócio.

Design Patterns e extensibilidade de *workers* no XFile

Os Design Patterns são soluções recorrentes e testadas para problemas comuns na engenharia de software. No projeto *XFile*, esses padrões foram aplicados com foco em extensibilidade e organização modular, permitindo que novas funcionalidades fossem integradas sem comprometer a arquitetura existente [45].

Em especial, os padrões *Strategy* e *Factory* desempenham papel central na construção dos *workers* responsáveis por etapas como extração, classificação e indexação de documentos.

- O padrão *Strategy* permite que diferentes abordagens de classificação e indexação sejam encapsuladas como estratégias independentes, todas derivadas de uma mesma interface. Isso facilita a introdução de novos algoritmos, como classificadores baseados em IA ou métodos heurísticos, sem afetar o código existente.

- O padrão *Factory* centraliza a lógica de criação dos *workers*, instanciando a estratégia adequada conforme o contexto do documento, tipo de processamento ou

regras do negócio. Essa fábrica pode ser expandida com facilidade para acomodar novos tipos de *workers*, inclusive aqueles que utilizam tecnologias externas ou que demandam passos adicionais de pré-processamento.

- A união desses padrões promove a inversão de controle e o princípio aberto-fechado, garantindo que o sistema esteja preparado para evolução constante sem reescrita de fluxos principais.

A adoção consciente de Design Patterns no XFile contribui para um sistema adaptável e sustentável, permitindo que novas abordagens de indexação e classificação sejam facilmente integradas por meio de novos *workers* registados na fábrica, mantendo a coesão com o domínio e alinhando-se aos requisitos do negócio.

4.3 Base de Dados

A persistência de dados no projeto XFile é realizada através do MongoDB, uma base de dados orientada a documentos que oferece elevada flexibilidade na modelação de dados. Esta escolha está alinhada com os princípios da arquitetura do sistema, que privilegia a modularidade, a escalabilidade e a independência tecnológica entre as camadas.

O MongoDB permite armazenar documentos em formato JSON, o que facilita a representação de estruturas complexas e heterogéneas, como os diferentes tipos de documentos processados pelo sistema. Essa abordagem é especialmente vantajosa no contexto do XFile, onde os dados podem variar conforme o tipo de parceiro, o fluxo de processamento e os metadados associados.

A arquitetura do XFile garante que o acesso à base de dados seja feito de forma desacoplada, através de interfaces definidas no módulo *Repository*, respeitando o princípio da inversão de dependência. As entidades do domínio não têm conhecimento da estrutura física da base de dados, e os modelos persistentes são convertidos por adaptadores específicos, mantendo a integridade da lógica de negócio.

Embora as coleções do MongoDB apresentem uma estrutura abrangente envolvendo

empresas, funcionários e documentos, sua finalidade no contexto do XFile é exclusivamente voltada à rastreabilidade e ao controle interno. Essas entidades permitem identificar os utilizadores que interagem com o sistema e acompanhar o fluxo de processamento dos documentos. O resultado final, após o processamento, é persistido e disponibilizado para consumo externo através do GVE Online.

Esta secção apresenta a estrutura de coleções, os modelos de dados utilizados, com foco na eficiência, consistência e escalabilidade da solução. A figura 4.2 representa as relações entre esses modelos e uma breve descrição de suas atribuições no XFile.

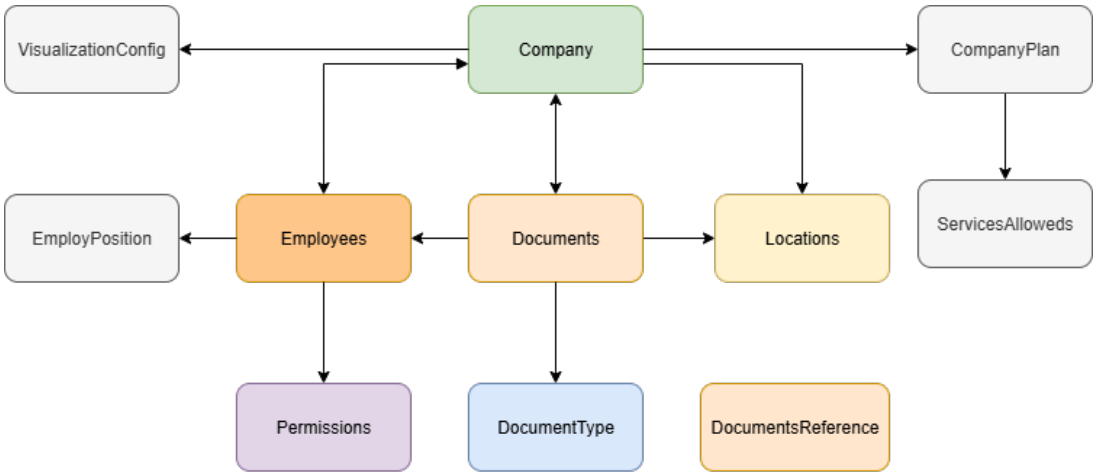

Figura 4.2: Modelagem de dados no MongoDB

- **Company:** Representa uma organização cadastrada no sistema, contendo informações essenciais para rastreabilidade e controle interno no XFile. No MongoDB, é armazenada como um documento que inclui atributos como identificador único, nome da empresa, descrição da empresa, sua identificação fiscal, endereço da empresa (definido pela entidade Locations), referência ao bucket utilizando no MinIO, lista de funcionários associados, lista de documentos já submetidos, Plano associado e uma lista de configurações de visualização. Essa entidade não participa diretamente do processamento dos documentos, mas serve como referência para identificar o contexto institucional de cada operação realizada.

- **CompanyPlan:** Representa o plano de serviços contratado por uma organização

no sistema. Cada documento inclui um identificador único, descrição detalhada do plano, valor financeiro associado, data de faturamento acordada, e uma lista de funcionalidades disponíveis.

- **ServicesAlloweds:** Representa os serviços computacionais disponíveis no XFile. Cada documento inclui um identificador único, que garante o acesso a recursos computacionais como diferentes *Workers*, que podem ser utilizados conforme a demanda da organização.

- **VisualizationConfig:** Define as preferências de visualização de documentos para uma determinada organização, aplicáveis aos serviços oferecidos pela TECH X, como o GVE Online. Cada documento inclui um identificador único e um *map* de configurações, onde a chave representa o tipo de visualização desejado (por exemplo, modo tabela, gráfico ou leitura contínua) e o valor contém os parâmetros específicos associados a essa visualização. Essas definições podem ser interpretadas pelo GVE Online ou por ferramentas externas, conforme a escolha e integração da organização.

- **Employees:** Representa um utilizador do sistema que atua como empregado de uma organização. Cada registo contém um identificador único, nome completo, número de identificação fiscal, endereço de e-mail, senha armazenada de forma criptografada, cargo ocupado (referenciado pela entidade EmployPosition), vínculo hierárquico com seu superior direto, conjunto de permissões atribuídas (referenciado pela entidade Permissions), e uma referência à organização à qual pertence.

- **EmployPosition:** Define o cargo ou função ocupada por um empregado dentro da organização. Cada registo inclui um identificador único e uma descrição textual que caracteriza as responsabilidades ou o título do cargo.

- **Permissions:** Especifica uma permissão que pode ser atribuída a um empregado dentro da organização. Cada registo contém um identificador único e uma descrição textual que define a ação autorizada, como editar documentos, acessar informações confidenciais da empresa ou gerenciar utilizadores.

- **Documents:** Representa um documento reconhecido pelo sistema. Cada registo inclui um identificador único, nome do documento, data de submissão, e a localização relativa do documento dentro do diretório da empresa no *Data Lake*(MinIO). Contém dois mapas: um para os campos e atributos que requerem revisão após o processamento, e outro para os dados extraídos com sucesso durante esse processamento. Além disso, o documento referencia o empregado responsável pela submissão, a companhia à qual pertence, o tipo de documento (definido pela entidade DocumentType), e uma lista de localidades associadas (considerando, por exemplo, uma fatura pode estar vinculada a múltiplas localidades).

- **Locations:** Representa uma localização geográfica reconhecida pelo sistema. Cada registo inclui um identificador único, país, cidade, rua e um campo de texto livre destinado a informações complementares, como referências internas, detalhes logísticos ou observações específicas.

- **DocumentsReference:** Contém referências a campos conhecidos pelo XFile e previamente validados pelos utilizadores. Seu objetivo é servir como base de conhecimento para o processamento de documentos, contribuindo para a identificação mais precisa de campos e atributos relevantes. Os valores dos campos inseridos são alterados antes, mas seguem o mesmo padrão dos valores reais, permitindo uma aproximação eficiente por algoritmos como KNN e facilitando a indexação via Apache Lucene. Essa abordagem garante a proteção dos dados reais da empresa, preservando a privacidade sem comprometer a eficácia do sistema. Essa entidade é única na base de dados e é acessada exclusivamente pelo *worker*, que a utiliza como fonte confiável de padrões e nomenclaturas reconhecidas.

4.4 Bibliotecas

Ao longo do desenvolvimento deste projeto, foram incorporadas várias bibliotecas de código aberto, todas organizadas por meio do *Maven*, ferramenta oficial para gerenciamento

de dependências em projetos Java. A utilização desses recursos possibilita ao desenvolvedor focar nas funcionalidades centrais da aplicação, ao mesmo tempo em que aproveita componentes consolidados e amplamente validados pela comunidade técnica.

Este projeto utiliza diversas bibliotecas para garantir funcionalidades modernas, seguras e escaláveis. A seguir, descrevem-se brevemente as principais dependências utilizadas.

4.4.1 spring-boot-starter-web

Responsável por habilitar funcionalidades de desenvolvimento web com Spring MVC, incluindo suporte a APIs REST, servidor embutido (Tomcat) e roteamento de requisições HTTP.

4.4.2 spring-boot-starter

Dependência central do Spring Boot que ativa configurações automáticas, logging e suporte básico à aplicação.

4.4.3 spring-boot-starter-test

Inclui bibliotecas como JUnit, Mockito e Hamcrest para testes unitários e de integração, facilitando a validação de funcionalidades durante o desenvolvimento.

4.4.4 spring-boot-starter-data-mongodb

Permite a integração com o MongoDB utilizando Spring Data, simplificando operações de persistência e consultas reativas.

4.4.5 spring-boot-starter-security

Adiciona suporte à autenticação e autorização, protegendo endpoints da aplicação com configurações flexíveis e integração com padrões como OAuth2.

4.4.6 spring-boot-starter-validation

Habilita validações de dados com base na especificação Bean Validation (JSR 380), utilizando anotações como *@NotNull*, *@Email* e *@Valid* para garantir integridade dos dados recebidos.

4.4.7 jjwt-api, jjwt-impl, jjwt-jackson

Bibliotecas da JJWT para criação, assinatura e verificação de JWT. O módulo *jjwt-api* define a interface pública, *jjwt-impl* contém a implementação interna, e *jjwt-jackson* permite serialização JSON com Jackson.

4.4.8 lombok

Facilita a escrita testde código Java ao gerar automaticamente métodos como getters, setters, construtores e *toString*, reduzindo boilerplate e aumentando a legibilidade.

4.4.9 springdoc-openapi-starter-webmvc-ui

Gera documentação interativa da APIs REST utilizando Swagger UI, baseada na especificação OpenAPI. Permite visualização e testes dos endpoints diretamente via navegador.

4.4.10 lucene-core, lucene-analysis-common, lucene-queryparser, lucene-queries

Conjunto de bibliotecas do Apache Lucene para indexação e busca textual eficiente. Inclui analisadores linguísticos, parsers de consulta, suporte a queries avançadas como fuzzy search e integração com KNN.

4.4.11 minio

Cliente Java para integração com o sistema de armazenamento de objetos MinIO, compatível com a API do Amazon S3. Utilizado para upload, download e gerenciamento de ficheiros em buckets.

4.4.12 pdfbox

Biblioteca da Apache para leitura, criação e manipulação de documentos PDF. Permite extração de texto, geração de ficheiros e validação de conformidade com padrões como PDF for Archiving (PDF/A).

4.4.13 jackson-databind

Componente central da biblioteca Jackson para serialização e desserialização de objetos Java em JSON. Utilizado amplamente em APIs REST para conversão automática de dados.

4.5 Entregas e Testes

Tendo por base os requisitos e os casos de uso descritos no Capítulo 3, esta secção apresenta os resultados obtidos a partir da implementação das soluções propostas, enquadradas no contexto tecnológico previamente delineado. Os testes realizados visam validar a eficácia das abordagens adotadas, bem como demonstrar a conformidade com os objetivos definidos ao longo do processo de desenvolvimento.

4.5.1 MongoDB

Antes de apresentar a estrutura detalhada e os testes realizados, é necessário contextualizar os dados já existentes na base de dados, em conformidade com os pré requisitos descritos na secção 3.6, que servirão de suporte para a validação das funcionalidades descritas ao longo desta secção.

Toda a estrutura que será apresentada está armazenada em um ambiente de teste, ou seja, com uma base simples para demonstrar todas as funcionalidades que serão descritas.

A empresa e o Utilizador são previamente cadastrados pela equipa da TECH X, e é disponibilizado ao parceiro um acesso principal para que possa gerir seus recursos, sendo assim, o utilizador pode ser identificado na figura 4.3.

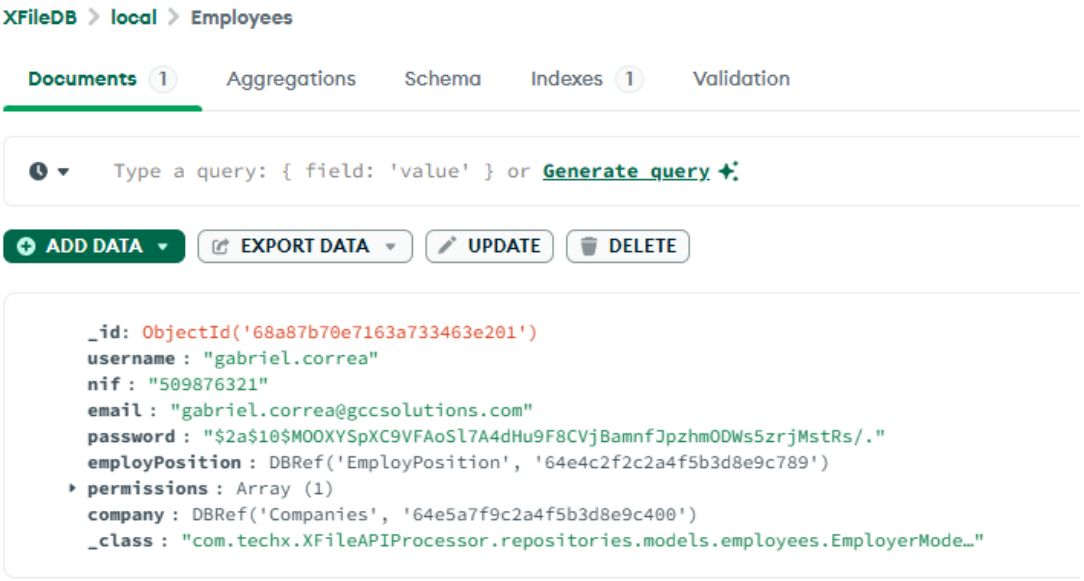

Figura 4.3: Utilizador padrão XFile.

Como é possível observar, a aplicação armazena a palavra-passe do utilizador de forma encriptada, utilizando uma chave que permanece inacessível a qualquer outro utilizador, mesmo nas propriedades internas do ficheiro. Para garantir a segurança das credenciais, foi implementado um mecanismo de encriptação baseado no algoritmo de hash criptográfico *SHA-256*.

O *SHA-256* pertence à família *SHA-2*, desenvolvida pela *National Security Agency (NSA)*, e é amplamente utilizado em sistemas modernos devido à sua robustez e fiabilidade. Este algoritmo transforma a palavra-passe original numa sequência de 256 bits, de forma irreversível, assegurando que o valor original não pode ser recuperado a partir do hash gerado.

Durante o processo de autenticação, a palavra-passe fornecida pelo utilizador é novamente convertida com o mesmo algoritmo, e o sistema compara o novo hash com o valor armazenado. Se ambos coincidirem, o acesso é concedido. Este método garante a integridade e confidencialidade dos dados, mesmo em cenários de violação da base de dados.

Ainda no contexto do utilizador, a figura 4.4 apresenta a permissão atribuída ao

mesmo, que corresponde a um acesso irrestrito à plataforma XFile. Esta configuração permite ao utilizador aceder a todas as funcionalidades disponíveis no sistema, sem restrições, refletindo o perfil de um utilizador com privilégios administrativos ou de gestão plena dos recursos da aplicação.

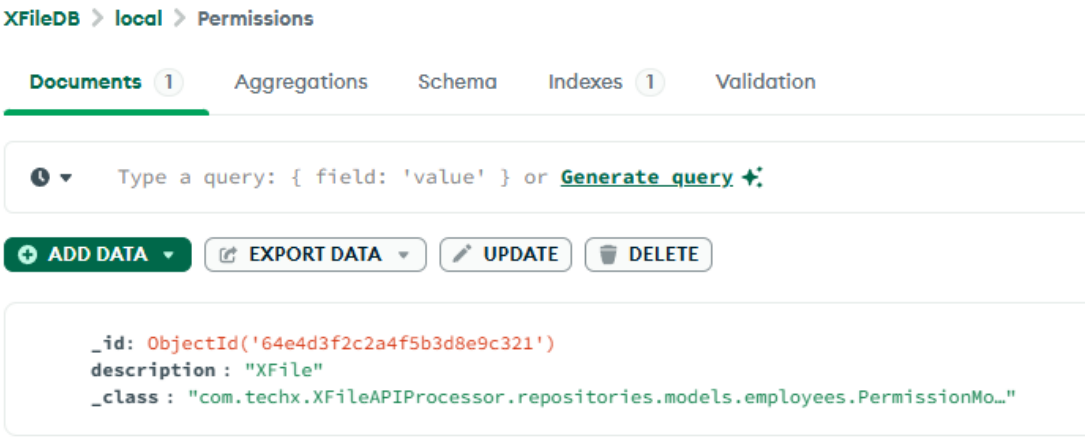

Figura 4.4: Permissão padrão XFile.

A figura 4.5 apresenta o cargo atribuído ao utilizador em questão, o qual pode ser utilizado pela empresa como mecanismo de controlo interno. Esta atribuição permite identificar a responsabilidade dentro da aplicação, contribuindo para uma gestão mais segura e estruturada.

Após a apresentação dos dados relativos ao utilizador com acesso total às funcionalidades da plataforma *XFile*, a figura 4.6 ilustra a estrutura da empresa à qual este utilizador está associado, bem como os seus principais atributos. Esta representação permite compreender o contexto organizacional em que o utilizador opera, evidenciando a relação entre os perfis de acesso e a empresa correspondente.

Como parte dos atributos associados à empresa, a figura 4.7 apresenta a configuração de visualizações que podem ser definidas pela empresa para suportar a geração de representações gráficas de dados. Estas configurações podem ser utilizadas na camada de

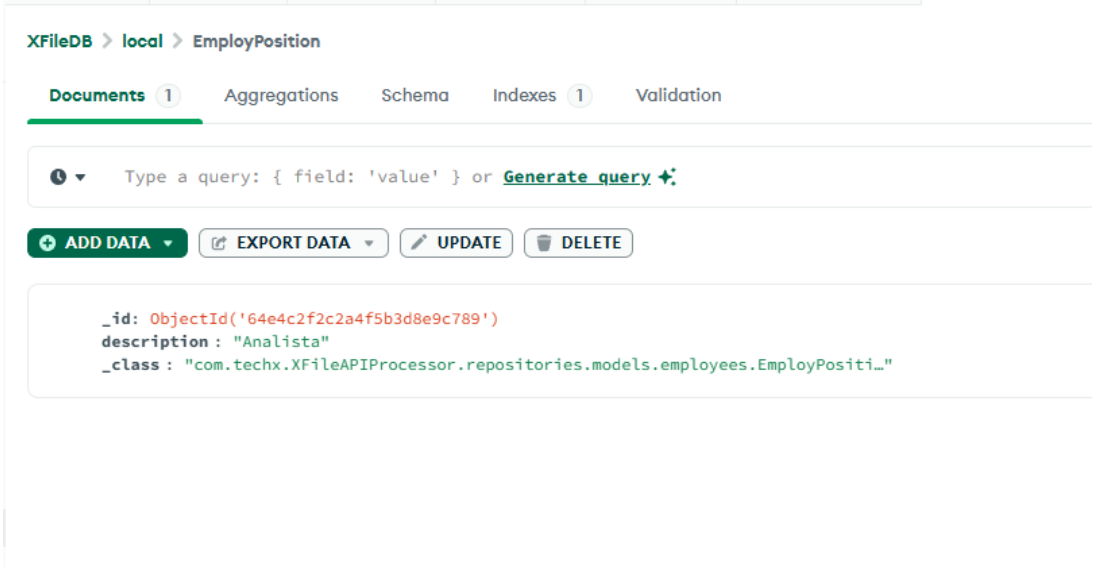

Figura 4.5: Permissão padrão XFile.

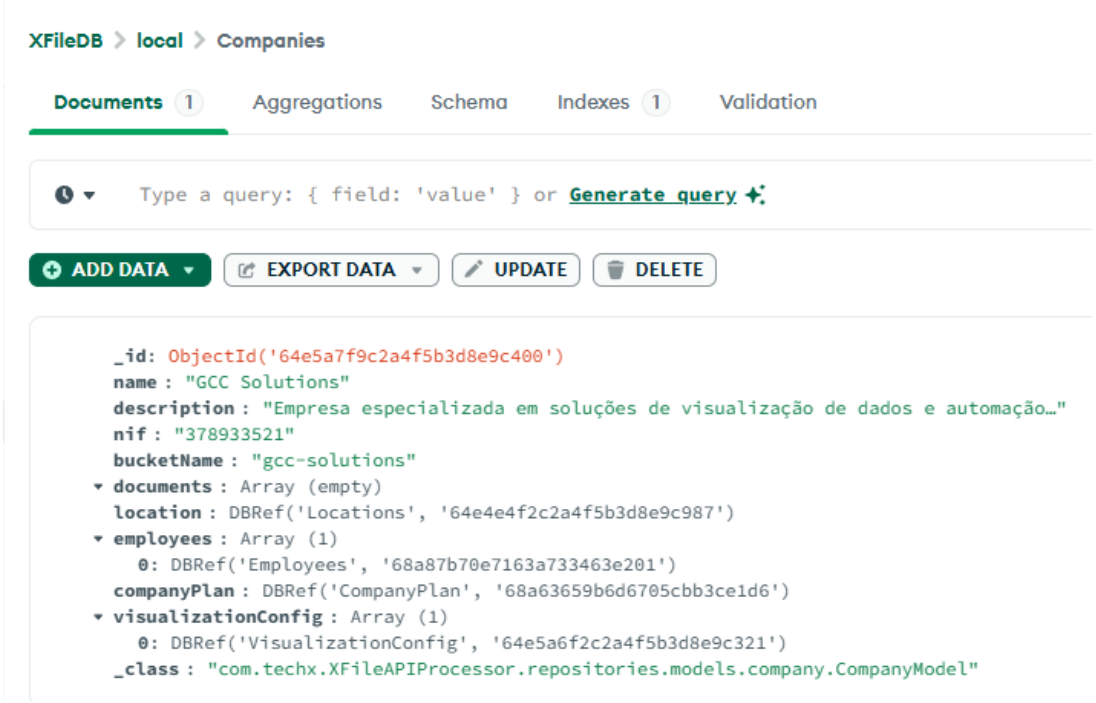

Figura 4.6: Empresa cadastrada no XFile.

Business Analytics, conforme descrito na secção 4.2, permitindo que os dados armazenados sejam interpretados de forma visual e dinâmica, de acordo com os critérios definidos pelo utilizador.

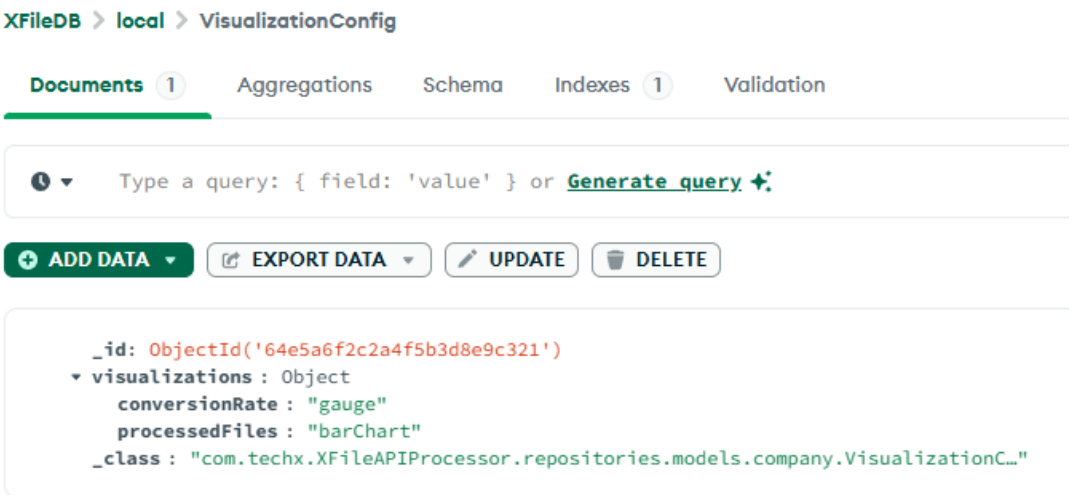

Figura 4.7: Visualizações da Empresa no XFile.

A figura 4.8 apresenta o plano associado à empresa registada, contendo as informações relativas ao tipo de subscrição e aos serviços disponibilizados à empresa. Esta estrutura permite identificar os recursos contratados, bem como as funcionalidades que podem ser acedidas, de acordo com o plano atribuído à organização.

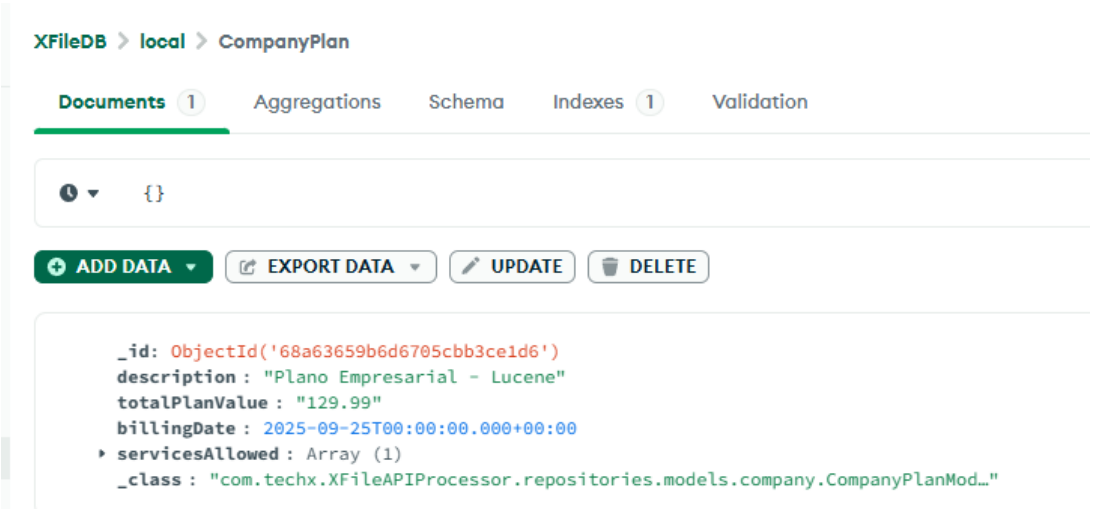

Figura 4.8: Plano atribuído à Empresa no XFile.

A figura 4.9 apresenta o serviço associado ao plano cadastrado, sendo este um elemento fundamental para a identificação automática do tipo de processamento a ser aplicado aos documentos enviados pela empresa. Esta configuração permite que a aplicação reconheça, de forma dinâmica, qual serviço deverá ser acionado para o tratamento adequado dos dados, garantindo consistência e eficiência na gestão documental.

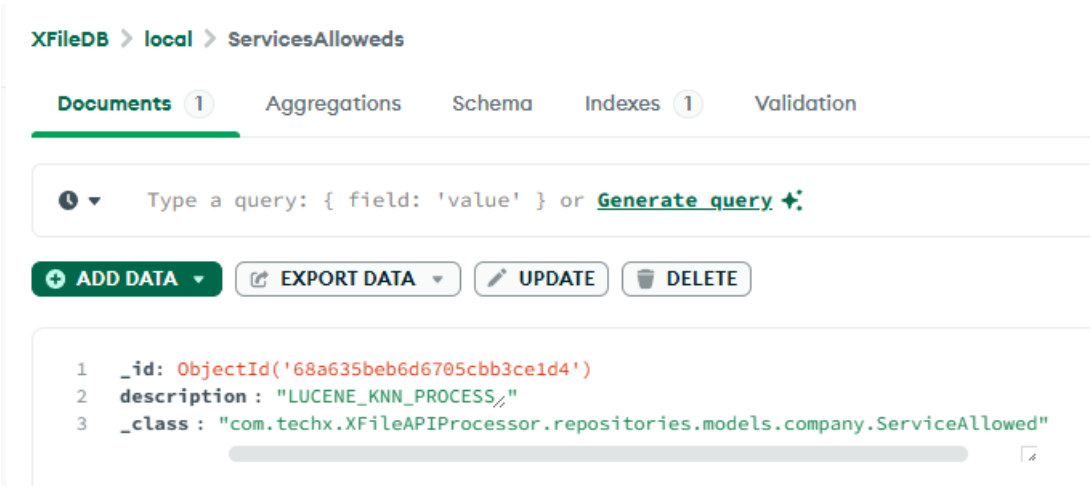

Figura 4.9: Serviço cadastrado no plano da empresa no XFile.

A figura 4.10 apresenta o endereço registado para a empresa, o qual integra o conjunto de informações essenciais para o controlo e gestão da organização. Esta informação contribui para a identificação geográfica da empresa, sendo relevante tanto para fins administrativos como para possíveis funcionalidades específicas da aplicação que dependem da localização da entidade.

A figura 4.11 apresenta o registo de um documento submetido na plataforma XFile, contendo as informações que foram automaticamente extraídas durante o processo de ingestão. Esta estrutura evidencia a capacidade do sistema em identificar e armazenar os dados relevantes de forma organizada, permitindo a posterior edição e busca através de seu conteúdo.

A figura 4.12 apresenta o registo correspondente ao tipo de documento submetido na plataforma XFile, informação esta que desempenha um papel fundamental na categorização e separação dos ficheiros armazenados no *dataLake* (MinIO). Esta classificação

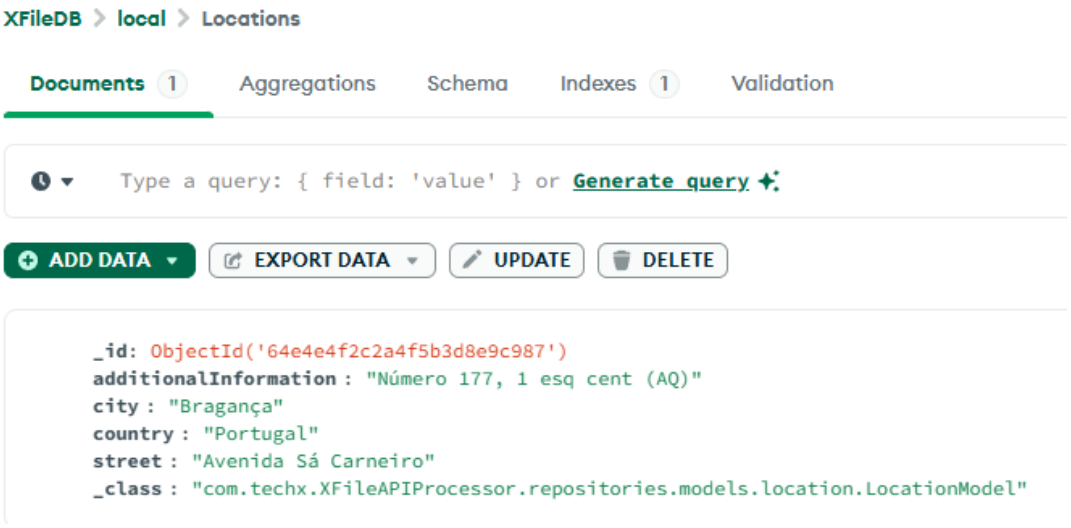

Figura 4.10: Endereço cadastrado pela empresa no XFile.

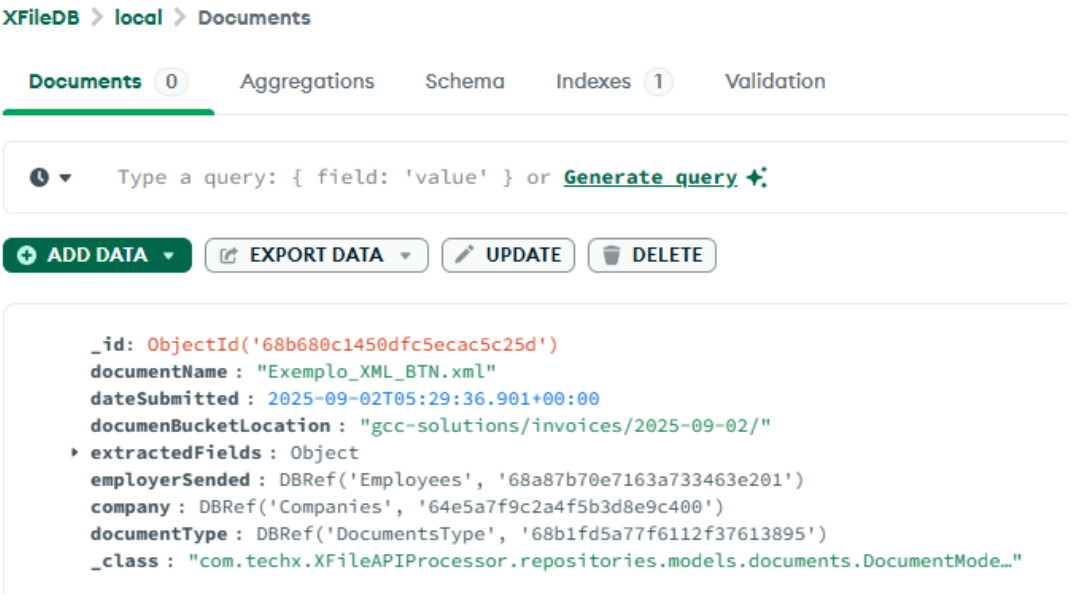

Figura 4.11: Documento submetido pela empresa no XFile.

permite que os documentos sejam organizados de forma eficiente, facilitando o seu tratamento posterior e garantindo que cada tipo de conteúdo seja encaminhado para o diretório adequado.

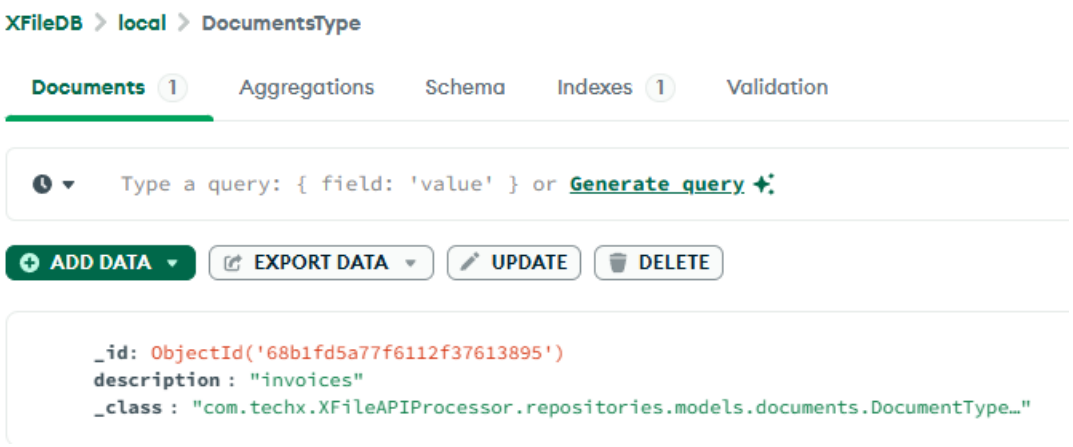

Figura 4.12: Tipo do documento submetido pela empresa no XFile.

4.5.2 MinIO

Quando um documento é submetido na plataforma XFile, além da autenticação do utilizador, são também validadas as informações da empresa à qual este pertence. Este processo garante que a submissão ao *dataLake* decorra de forma eficiente e transparente, assegurando a unicidade de cada documento e o seu armazenamento alinhado com o contexto de negócio.

Cada documento submetido segue uma estrutura padronizada de registo no *dataLake*, sendo o diretório principal definido pelo *bucketName* associado à empresa, conforme ilustrado na figura 4.6. Em seguida, é incorporado o tipo de documento, conforme apresentado na figura 4.12, seguido pela data exata da submissão e, por fim, o nome atribuído ao ficheiro, conforme evidenciado na figura 4.11.

Desta forma, e seguindo o exemplo previamente apresentado no MongoDB, o caminho gerado no MinIO para o armazenamento do documento deve ser: `xfile-archives/`

`gcc-solutions/invoices/2025-09-02/Exemplo_XML_BTN.xml`. A figura 4.13 demonstra visualmente o resultado desta submissão estruturada.

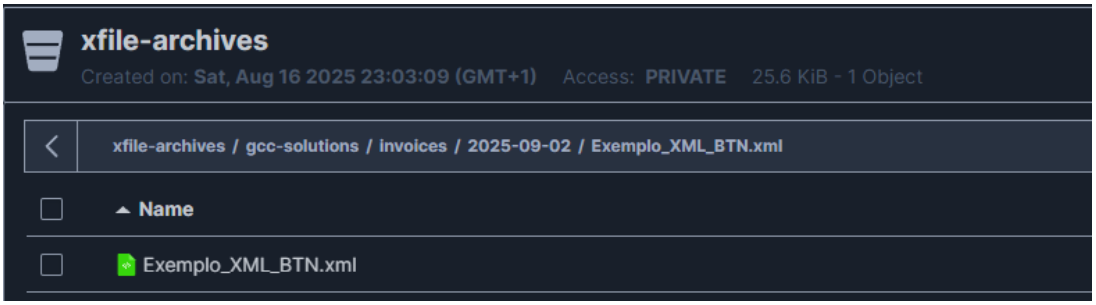

Figura 4.13: documento submetido pela empresa no XFile.

4.5.3 Swagger

Após o registo inicial da empresa e do respetivo utilizador principal, este passa a ter acesso à interface do Swagger disponibilizada pela plataforma XFile. Através desta interface, é possível visualizar todas as funcionalidades expostas pela aplicação, permitindo ao utilizador compreender e explorar os recursos disponíveis de forma estruturada.

A figura 4.14 apresenta uma visão geral das funcionalidades documentadas no Swagger, que serve como referência técnica para os *endpoints* da API, incluindo os contratos de comunicação, os formatos esperados de requisição (*requests*) e os modelos de resposta (*responses*). Esta documentação é essencial para garantir uma integração eficiente com o produto, facilitando o desenvolvimento de soluções que aproveitem plenamente os serviços oferecidos pela plataforma.

Para cada *controller* apresentada na figura 4.14, existe um conjunto de *endpoints* que fornece ao utilizador os meios necessários para realizar alterações nas entidades relacionadas, desde que possua as permissões adequadas dentro da empresa à qual está associado. Esta estrutura garante que as operações realizadas respeitam os níveis de acesso definidos, promovendo segurança e controlo sobre os dados manipulados.

A figura 4.15 ilustra um dos dois recursos disponíveis na *auth-controller*, sendo este especificamente reservado a utilizadores internos da TECH X. Este *endpoint* permite o

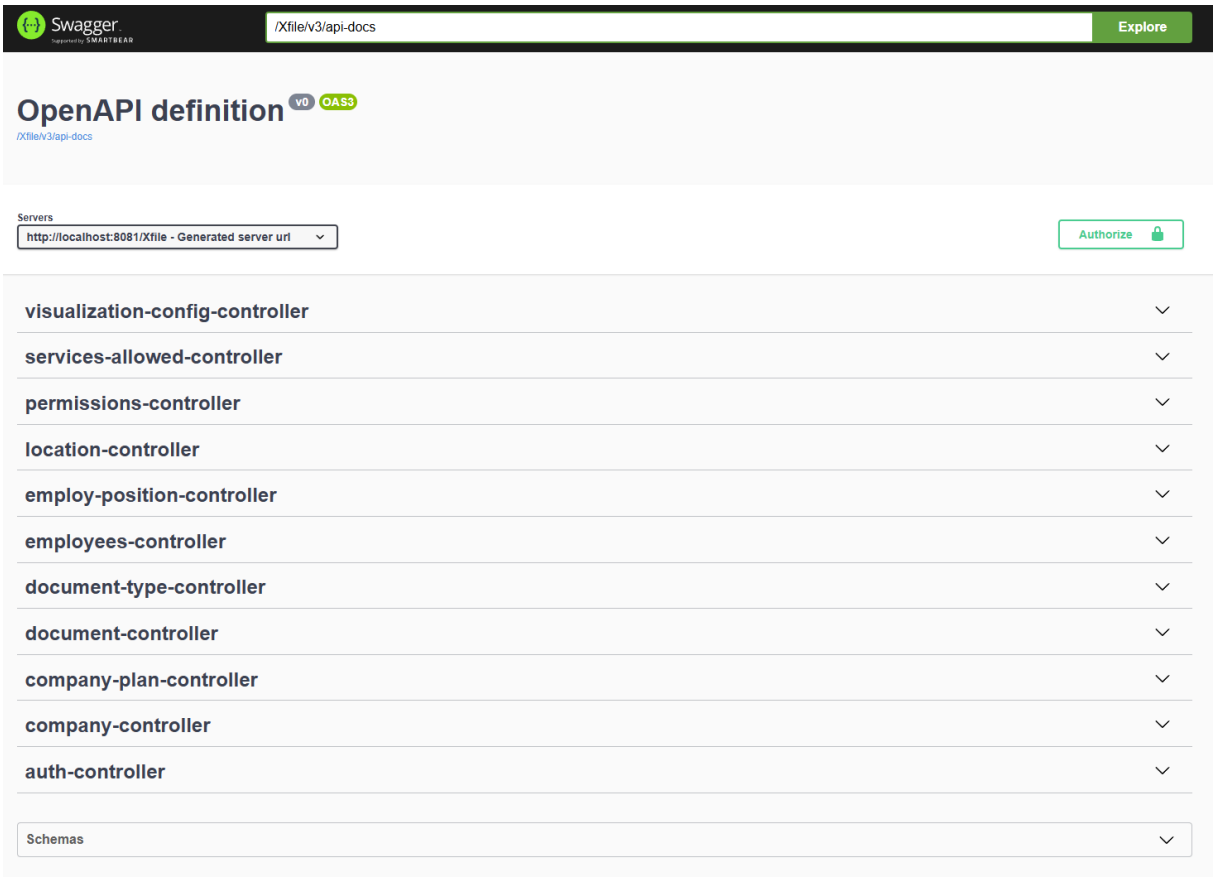

Figura 4.14: Visão geral do Swagger XFile.

registo de novos utilizadores, com a possibilidade de os associar a qualquer empresa previamente cadastrada na plataforma, reforçando o papel administrativo da equipa interna na gestão de acessos e perfis.

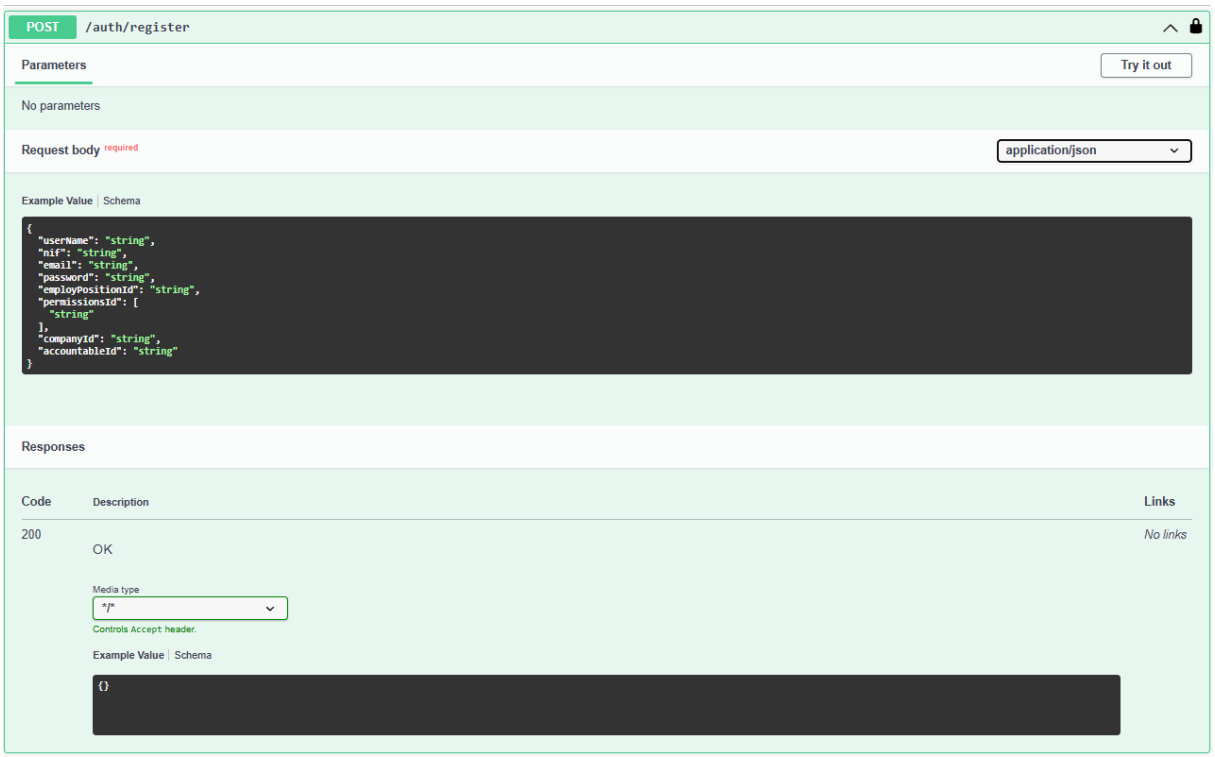

Figura 4.15: Post auth/register

O segundo recurso relacionado ao *auth-controller*, demonstrado na figura 4.16 que pode ser utilizados por todos utilizadores registrados, para autenticação no XFile e por fim obter o token JWT que poderá ser utilizado para acesso aos outros recursos do XFile.

Com o token JWT, conforme mencionado anteriormente, o utilizador passa a ter acesso aos recursos que lhe foram atribuídos, de acordo com as permissões definidas no seu perfil. Neste ambiente de teste, o utilizador em questão possui privilégios administrativos no XFile, o que lhe permite executar qualquer operação disponível em cada contexto funcional da aplicação.

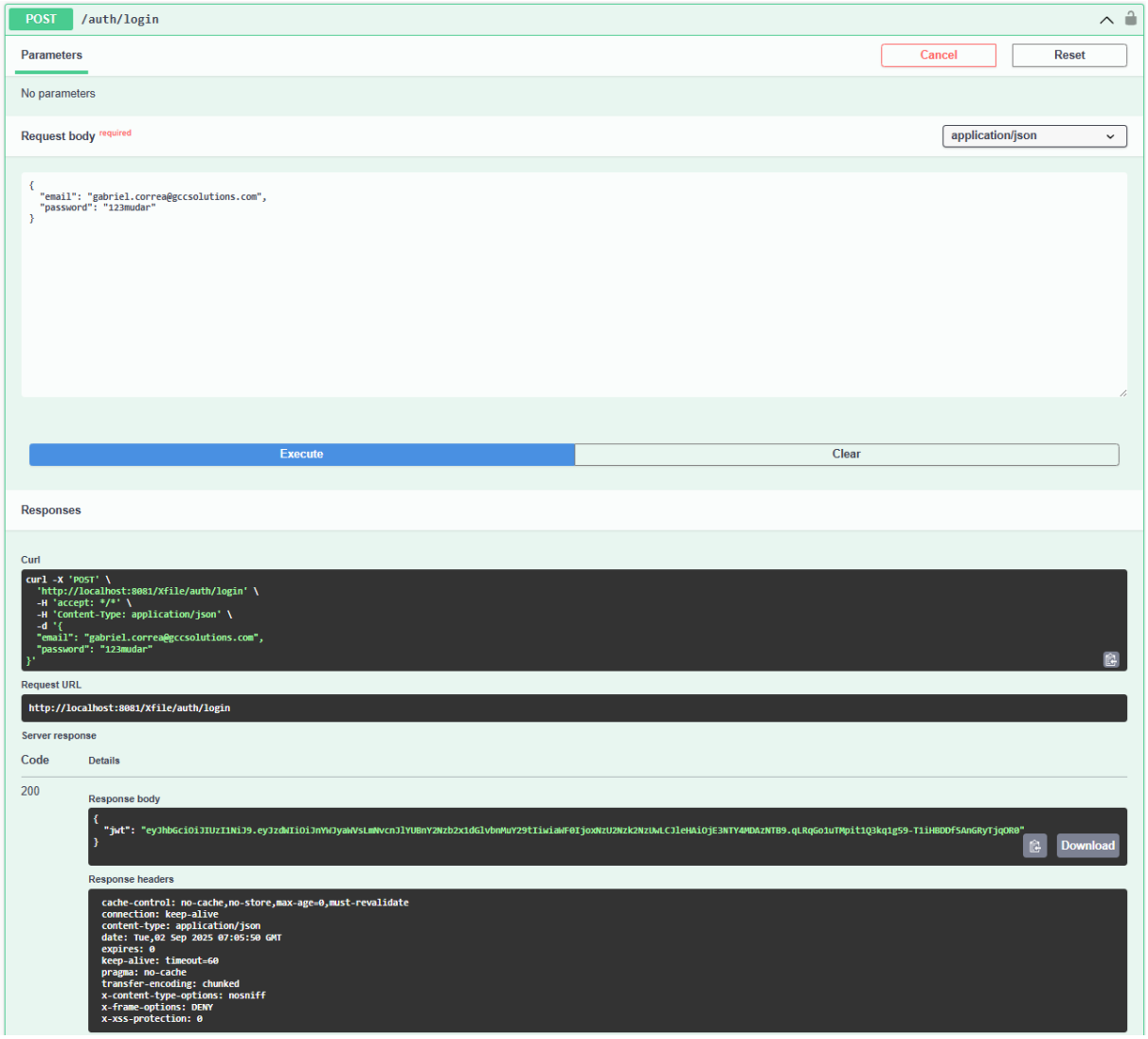

Figura 4.16: Post auth/login

A figura 4.17 apresenta o conjunto completo de recursos disponibilizados pela *visualization-config-controller*, evidenciando as operações que podem ser realizadas para configurar visualizações de dados, conforme os parâmetros definidos pela empresa.

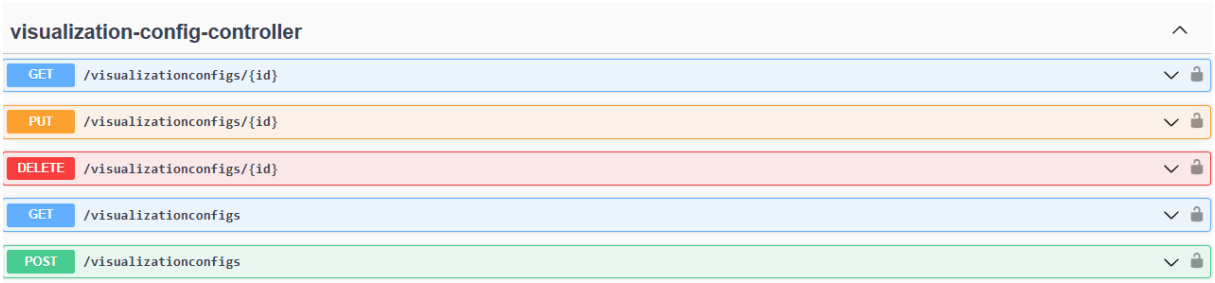

Figura 4.17: Recursos da visualization-config-controller.

Seguindo a ordem dos recursos apresentados na figura 4.14, a figura 4.18 apresenta o conjunto completo de funcionalidades disponibilizadas pela *services-allowed-controller*. Esta interface permite realizar operações relacionadas com a gestão dos serviços atribuídos a cada plano, os quais estão diretamente associados a empresas previamente registadas na plataforma.

A correta configuração desses serviços é essencial para determinar, de forma automatizada, qual será o mecanismo de tratamento aplicado aos dados de um documento submetido. Assim, este recurso desempenha um papel central na lógica de processamento da aplicação, garantindo que cada submissão seja encaminhada para o serviço adequado, conforme definido no plano da empresa.

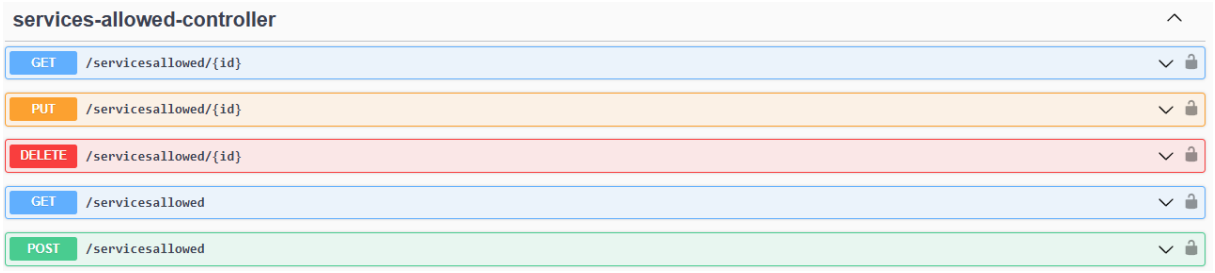

Figura 4.18: Recursos da services-allowed-controller.

A figura 4.19 apresenta o conjunto completo de funcionalidades disponibilizadas pela

permissions-controller, responsável por centralizar a gestão das permissões atribuídas aos utilizadores no XFile. Este controlador permite a criação, edição e remoção de registos de permissões, operações que estão restritas exclusivamente aos utilizadores internos da TECH X. Para os demais utilizadores, o acesso é limitado às funcionalidades de consulta, como listagem geral e pesquisa por identificadores específicos, garantindo assim um controlo rigoroso e seguro sobre os níveis de acesso definidos no sistema.

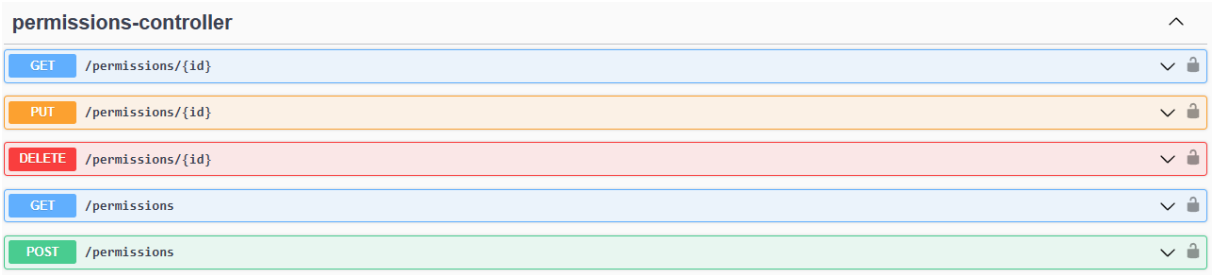

Figura 4.19: Recursos da permissions-controller.

A figura 4.20 apresenta as funcionalidades relacionadas à gestão de localizações disponibilizadas pela *location-controller* no XFile, abrangendo tanto os contextos documentais como empresariais. No caso de documentos, especialmente faturas (*invoices*), é possível associar múltiplas localizações de cobrança, refletindo cenários complexos de distribuição geográfica. Já no contexto empresarial, conforme ilustrado na figura 4.6, cada empresa possui uma localização previamente atribuída, essencial para a sua identificação e organização no sistema.

A listagem e visualização de localizações são estritamente condicionadas ao contexto da empresa autenticada, garantindo que entidades distintas não tenham acesso às informações geográficas de outras organizações. Esta abordagem reforça a segurança e a segmentação dos dados, assegurando que cada utilizador opere exclusivamente dentro dos limites da sua estrutura empresarial.

A figura 4.21 apresenta as funcionalidades relacionadas à gestão de posições profissionais dos utilizadores disponibilizadas pela *employ-position-controller* no XFile. Este

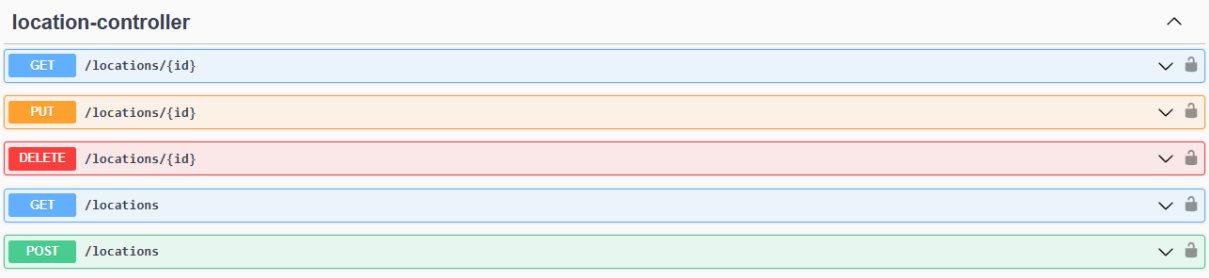

Figura 4.20: Recursos da location-controller.

recurso pode ser utilizado pelas empresas para atribuir e descrever cargos internos, permitindo uma organização mais clara das funções desempenhadas por cada colaborador. No entanto, no contexto geral da aplicação, essa informação possui caráter meramente descritivo, não influenciando diretamente os fluxos operacionais ou os níveis de permissão. Trata-se, portanto, de uma funcionalidade voltada à estruturação interna das empresas, com foco na gestão organizacional.

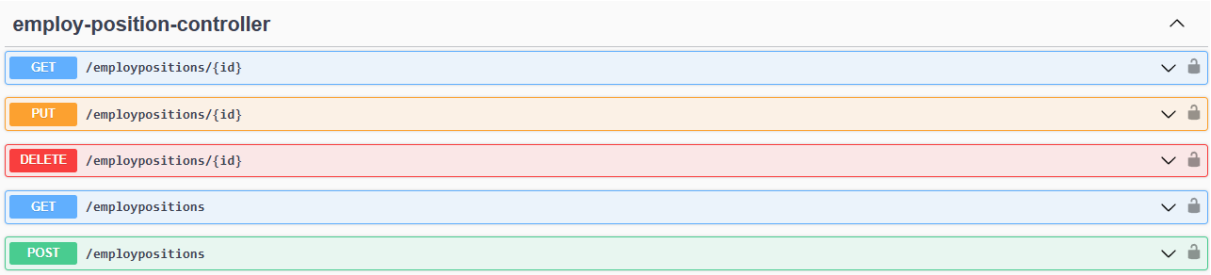

Figura 4.21: Recursos da employ-position-controller.

A figura 4.22 apresenta as funcionalidades relacionadas à gestão de utilizadores disponibilizadas pela *employees-controller* no XFile. Este conjunto de recursos é amplamente acessível às empresas, permitindo-lhes executar todas as operações implementadas no contexto da sua própria organização, tais como adicionar, editar, listar, pesquisar por identificador único, registar novos utilizadores e também removê-los. Todas essas ações são rigorosamente condicionadas pelos requisitos de segurança, garantindo que cada empresa apenas aceda aos utilizadores pertencentes ao seu próprio domínio.

É por meio desta controladora que a empresa consegue gerir os seus utilizadores de acordo com as suas necessidades operacionais. O processo de registo e encriptação de senhas segue o mesmo padrão de segurança definido na *auth-controller* (figura 4.15), utilizado pela equipa interna da TECH X para a criação do utilizador principal da empresa.

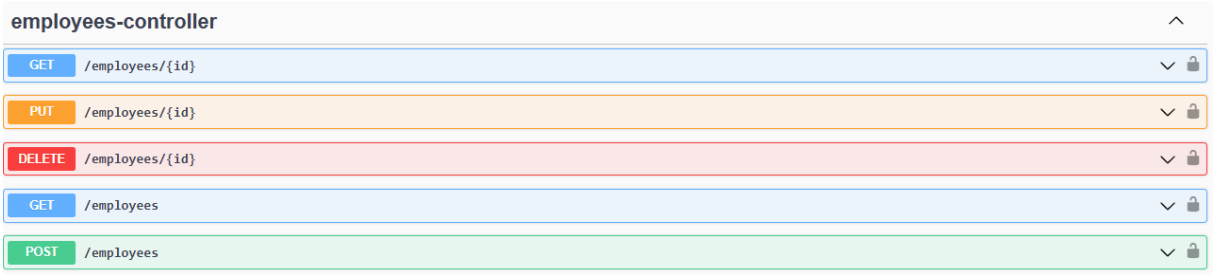

Figura 4.22: Recursos da employees-controller.

A figura 4.23 apresenta as funcionalidades relacionadas à gestão documental disponibilizadas pela *document-controller* no XFile. Este conjunto de operações representa o núcleo funcional da aplicação, cuja principal finalidade é centralizar e facilitar a gestão de documentos empresariais de forma eficiente e segura.

Entre as funcionalidades tradicionais disponíveis nesta controladora, destacam-se, a criação de entidades documentais sem necessidade de upload imediato, a edição de atributos, a pesquisa por identificador único e a exclusão de documentos. Todas essas operações são executadas dentro do contexto operacional da empresa autenticada, respeitando os limites de acesso definidos.

Além dessas operações básicas, a *document-controller* também oferece recursos avançados, como a pesquisa textual em documentos já submetidos permitindo localizar qualquer menção específica no conteúdo armazenado, o download dos ficheiros originais e o acesso aos mecanismos de processamento documental, ativados no momento do upload. Estas funcionalidades ampliam significativamente o potencial da plataforma, tornando-a uma solução robusta para a gestão integrada de documentos empresariais.

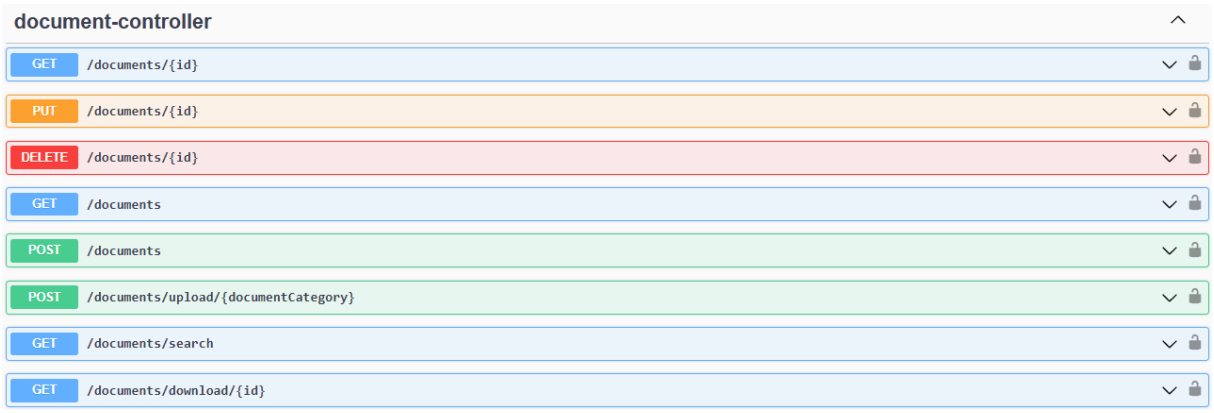

Figura 4.23: Recursos da document-controller.

A figura 4.24 apresenta as funcionalidades relacionadas à gestão de planos empresariais disponibilizadas pela *company-plan-controller* no XFile. Por meio desta interface, a empresa pode consultar o plano atualmente atribuído, bem como visualizar os planos disponíveis que incorporam novas funcionalidades implementadas na aplicação.

As permissões concedidas às empresas nesta controladora são exclusivamente de leitura, permitindo apenas a visualização dos planos próprios e dos planos disponíveis. Todas as operações relacionadas à criação, edição e remoção de planos estão reservadas à equipa interna da TECH X, uma vez que envolvem diretamente a definição dos serviços e funcionalidades que compõem a estrutura operacional do *XFile*. Esta separação de responsabilidades garante maior segurança e consistência na gestão dos recursos empresariais.

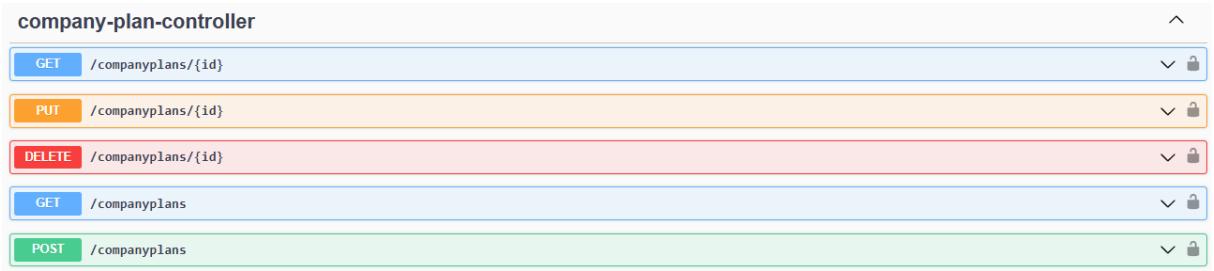

Figura 4.24: Recursos da company-plan-controller.

A figura 4.25 apresenta as funcionalidades relacionadas à gestão de empresas disponibilizadas pela *company-controller* no XFile. A maior parte dessas operações — como criação, listagem e exclusão de registos empresariais — está restrita à equipa interna da TECH X, garantindo que apenas utilizadores autorizados possam realizar alterações estruturais no sistema.

Para as empresas já registadas, está disponível a possibilidade de editar informações que não comprometam dados sensíveis ou pessoais, sendo permitido apenas visualizar os próprios registos e os ativos diretamente associados. A edição desses dados é igualmente limitada ao utilizador principal da empresa, cuja criação é realizada exclusivamente pela equipa da TECH X no momento da configuração inicial.

Adicionalmente, o controlo de acesso às informações empresariais pode ser gerido por meio das permissões atribuídas aos utilizadores da própria organização, utilizando os recursos da *permissions-controller*, conforme ilustrado na figura 4.19. Esta abordagem assegura uma gestão segura e segmentada, alinhada com as políticas de privacidade e gestão da plataforma.

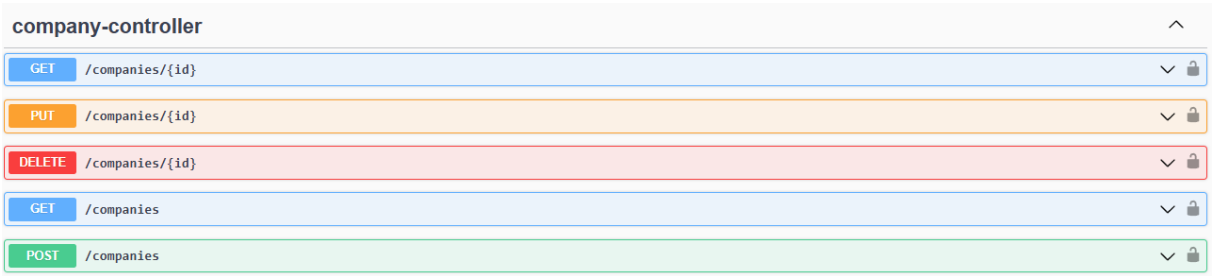

Figura 4.25: Recursos da company-controller.

4.5.4 Implementação Técnica e Recursos Computacionais

Durante o planeamento e desenvolvimento deste trabalho, foram adotadas estratégias que visam não apenas a resolução eficiente do problema proposto, mas também a otimização dos recursos computacionais, a garantia da integridade dos dados e a proteção das informações empresariais e dos utilizadores envolvidos. A implementação técnica do XFile foi

concebida com foco na robustez, escalabilidade e segurança, alinhando-se aos requisitos definidos na arquitetura do projeto (subsecção 4.2.1).

Nesta subsecção, serão apresentadas as decisões técnicas que sustentam a aplicação, incluindo os componentes computacionais utilizados, os mecanismos de persistência e comunicação, bem como os recursos que asseguram o desempenho e a fiabilidade da plataforma em ambientes reais de operação.

4.5.5 Componentes Computacionais Utilizados

Considerando os recursos disponibilizados pela *document-controller*, conforme apresentado na subsecção 4.5.3, o desenvolvimento deste projeto tem como objetivo oferecer um serviço robusto de gestão documental capaz de sustentar o GVE Online e empresas parceiras que queiram utilizar os serviços do XFile. A proposta visa introduzir uma abordagem inovadora na provisão desses recursos, promovendo maior escalabilidade na execução e evolução contínua da aplicação, bem como a implementação de novos métodos de classificação documental.

No que diz respeito aos componentes computacionais, o *XFile* fundamenta-se no processamento assíncrono de documentos submetidos ao sistema, independentemente do seu formato, XML, PDF ou JSON. A aplicação foi desenvolvida em Java, utilizando o ecossistema Spring, que permite uma arquitetura modular e flexível, beneficiando-se da inversão de dependências para promover uma gestão inteligente dos recursos computacionais.

Integridade e privacidade

Cada requisição enviada ao *XFile* é submetida a uma validação rigorosa do contexto empresarial do utilizador, garantindo que as ações realizadas estejam estritamente vinculadas à empresa à qual o utilizador pertence. Esta abordagem impede a execução de operações acidentais ou mal-intencionadas fora do seu domínio autorizado, incluindo restrições aplicadas aos próprios utilizadores internos da TECH X, que não podem aceder a dados sensíveis de empresas sem autorização explícita e inclusão formal no respetivo contexto.

Os dados extraídos e indexados a partir dos documentos submetidos são considerados propriedade exclusiva da empresa responsável pela submissão, assegurando que a privacidade e a segurança sejam princípios fundamentais em todas as etapas de execução.

Submissão de um ficheiro

No momento da submissão de um ficheiro, o utilizador precisa apenas fornecer dois elementos: o próprio ficheiro e a categoria documental à qual este pertence.

Uma vez recebido, o XFile inicia um processo de identificação e validação, no qual os dados do utilizador e da empresa são incorporados num objeto que acompanha todo o ciclo de processamento até à persistência final. Esta estrutura garante rastreabilidade, atomicidade e isolamento de cada operação, tornando o processamento documental único e independente dentro do sistema.

Após a criação do objeto que associa o documento ao utilizador e à respetiva empresa, é considerado como validado que ambos possuem as permissões necessárias para o processamento. Em seguida, o documento é encaminhado para uma fila de processamento dedicada, isolada por empresa. Ou seja, cada empresa possui uma fila exclusiva, e para cada uma é alocada uma *Thread* de processamento independente.

O XFile pode ser configurado para operar com um número finito de *Threads* disponíveis para processamento simultâneo entre empresas. Caso esse limite seja atingido, novas requisições de submissão deverão aguardar a disponibilidade de uma *Thread* para que o processamento possa ser iniciado. Esta abordagem garante isolamento entre contextos empresariais, evita concorrência indevida entre documentos de diferentes organizações e assegura que o processamento ocorra de forma controlada, rastreável e escalável.

Quando uma empresa adquire uma *Thread* de processamento, passa a ter o direito de executar todas as operações sobre os documentos presentes na sua fila dedicada. Sempre que um novo documento é submetido, ele é automaticamente incluído nessa fila por meio do orquestrador de estados, conforme descrito na secção 4.2, garantindo que o fluxo de processamento seja contínuo e isolado por contexto empresarial. Caso não ocorra a submissão de qualquer novo documento durante um pequeno período de ociosidade da

Thread, ela é liberada para que outra empresa possa utilizar o recurso.

Esse tempo também é configurável assim como a quantidade total de *Threads* através das propriedades do XFile, garantindo que possa ser atualizado facilmente pela equipa TECH X, sem interrupções no processamento de documentos.

O processamento de cada documento segue etapas fundamentais, sendo a primeira delas o armazenamento imediato do ficheiro e da estrutura inicial do objeto documental. Esta etapa assegura o registo da submissão e marca o início formal do ciclo de processamento, permitindo rastreabilidade desde o primeiro momento.

Todo o processo respeita rigorosamente o atributo empresarial denominado *bucketName*, conforme apresentado na subsecção 4.5.1. Este atributo, definido no momento da criação da empresa no *XFile*, é imutável e garante que todos os documentos sejam armazenados no mesmo caminho lógico dentro do MinIO. Além disso, os ficheiros são indexados no mesmo diretório pelo Apache Lucene, assegurando consistência na organização dos dados e impedindo qualquer acesso fora do contexto empresarial validado. Esta abordagem reforça os princípios de segurança, isolamento e integridade dos dados ao longo de todo o ciclo de vida documental.

A segunda etapa do processamento documental consiste na identificação da extensão do ficheiro submetido, etapa essencial para a execução da extração inicial dos dados. Esta identificação é realizada de forma dinâmica, por meio da aplicação dos padrões de projeto *Strategy* e *Factory*, que permitem encapsular o comportamento específico de cada tipo de documento e instanciar o componente adequado conforme o formato identificado.

Caso seja necessário suportar um novo tipo de ficheiro, basta implementar a respetiva estratégia de extração e integrá-la ao mecanismo existente. A arquitetura do sistema garante que, no momento da submissão, o novo formato será automaticamente reconhecido e processado pela estratégia correspondente, sem necessidade de alterações na lógica central da aplicação. Esta abordagem assegura flexibilidade, extensibilidade e baixo acoplamento entre os componentes, permitindo a evolução contínua da plataforma.

Após a extração inicial e a geração dos *Blobs*, o sistema identifica, com base no plano empresarial, qual serviço de processamento está disponível para utilização. Essa definição

determina o *worker* responsável por executar a próxima etapa do fluxo. Assim como ocorre na fase inicial de extração, a seleção do serviço é realizada de forma dinâmica, por meio da aplicação dos padrões de projeto *Strategy* e *Factory*.

Essa abordagem facilita a introdução de novas estratégias de processamento, que podem ser integradas ao sistema com mínima interferência na estrutura existente. Sempre que uma empresa possuir determinado serviço habilitado em seu plano, o sistema o reconhece automaticamente e o aplica ao documento em questão, garantindo escalabilidade, modularidade e aderência às regras de negócio definidas.

Uma vez identificado o serviço de processamento disponível para a empresa, o documento é encaminhado para execução. No contexto atual do XFile, existe um único serviço ativo, responsável pela indexação vetorial dos documentos por meio do framework Apache Lucene.

O resultado desse processo de indexação é armazenado em um diretório definido nas propriedades de configuração do XFile, concatenado com o *bucketName* da empresa, conforme previamente estabelecido. Cada documento submetido contribui para a expansão da base de índices mantida pelo Lucene, permitindo que o sistema realize buscas textuais eficientes e escaláveis sobre os conteúdos armazenados. Esta estrutura garante que os dados indexados permaneçam organizados e isolados por contexto empresarial.

Após a etapa de indexação, os atributos relevantes do documento são identificados por meio da aplicação do algoritmo KNN, integrado ao mecanismo de busca do Apache Lucene. Este processo permite a extração inteligente de campos com base na similaridade vetorial entre documentos previamente indexados, promovendo uma classificação contextual e precisa dos dados submetidos.

Uma vez identificados, os campos extraídos são armazenados na coleção correspondente ao documento, garantindo que todas as informações estejam devidamente persistidas e associadas ao seu registo original. Com isso, o ciclo de processamento é concluído, assegurando rastreabilidade, integridade e conformidade com os requisitos funcionais da plataforma. A figura 4.26 apresenta o resultado da execução de uma requisição de submissão de ficheiro à plataforma. Como resposta, o sistema retorna um *HTTP Status 200*

OK, acompanhado da confirmação de que o documento foi encaminhado com sucesso para a respetiva fila de processamento.

Este retorno indica que todas as validações iniciais foram concluídas corretamente e que o ficheiro encontra-se em estado de espera para ser processado de acordo com o contexto empresarial e os serviços disponíveis no plano da empresa.

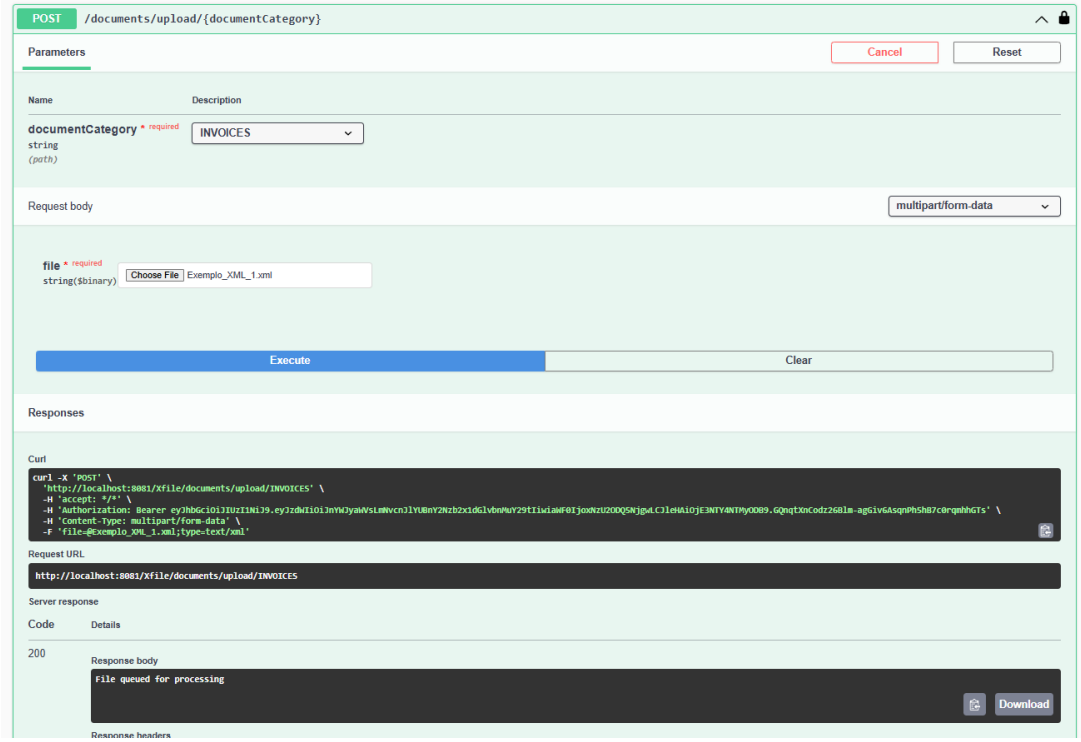

Figura 4.26: Submissão de um novo ficheiro.

Durante a etapa de identificação dos campos extraídos, é preservada a hierarquia estrutural de cada atributo, permitindo que a profundidade e o posicionamento relativo no documento sejam corretamente armazenados. Esta abordagem facilita a compreensão do contexto em que cada atributo se encontra, refletindo com precisão a sua relação semântica e estrutural dentro do conteúdo submetido.

Ao manter essa hierarquia, o sistema assegura que os dados extraídos possam ser interpretados de forma contextualizada, o que é especialmente relevante para documentos complexos, como faturas ou formulários estruturados, nos quais a posição de um campo pode alterar significativamente o seu significado. A figura 4.27 apresenta o resultado

da indexação e classificação de um ficheiro submetido à plataforma. Para preservar a hierarquia dos atributos extraídos sem comprometer a integridade do armazenamento, foi adotado o identificador "_dot_" como substituto do caractere ".". Esta substituição é necessária porque, no sistema de armazenamento do MinIO, o ponto é interpretado como delimitador de atributos, o que poderia causar inconsistências na estrutura dos dados armazenados. Para evitar esse problema, foi implementada uma lógica de substituição automática integrada à configuração do MinIO, executada pelo motor responsável pela busca e persistência das coleções. Esta abordagem garante que a hierarquia dos dados seja mantida corretamente, sem interferir na organização dos diretórios ou na integridade dos índices gerados pelo Apache Lucene.

A figura 4.28 apresenta o retorno de uma requisição realizada ao *document-controller*, buscando o documento pelo seu respetivo identificador único, evidenciando a substituição reversa do identificador "_dot_" pelo caractere original ".". Esta conversão é aplicada automaticamente pelo sistema, garantindo que os dados, ao serem submetidos a qualquer processo interno do *XFile* ou recuperados para fins de consulta, sejam apresentados em sua forma original e semanticamente adequada. Essa abordagem assegura a integridade da estrutura hierárquica dos atributos, preservando a legibilidade e o contexto dos dados, sem comprometer a compatibilidade com os mecanismos de armazenamento utilizados, como o MinIO.

Download de Ficheiros e Busca Textual Indexada

Através da submissão de um novo ficheiro, conforme realizado anteriormente, é possível verificar que a empresa possui agora dois documentos associados, conforme ilustrado na figura 4.29. Esta visualização evidencia a correta associação dos ficheiros ao contexto empresarial.

A figura 4.30 ilustra a execução da funcionalidade de download de ficheiro, aplicada ao documento localizado na posição 1 da lista de documentos associados à empresa. Esta operação permite ao utilizador recuperar o ficheiro original anteriormente submetido, respeitando o contexto empresarial e os mecanismos de segurança definidos pelo XFile.

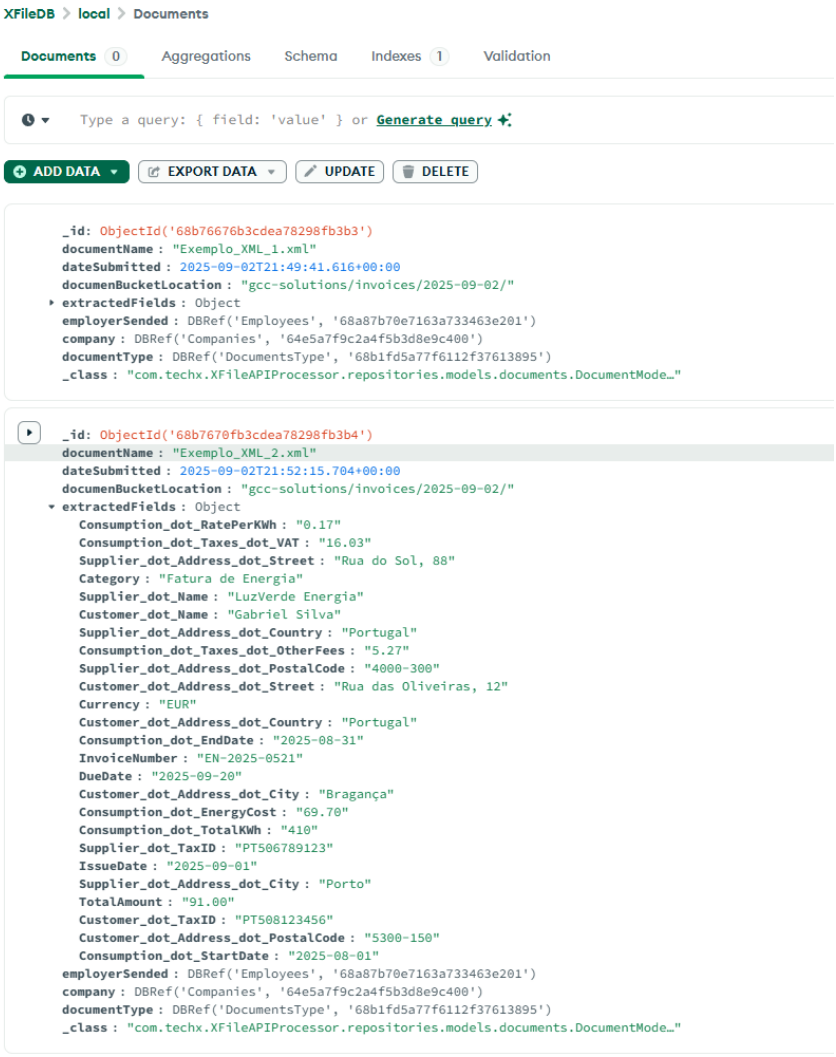

Figura 4.27: Atributos extraídos de um ficheiro de teste.

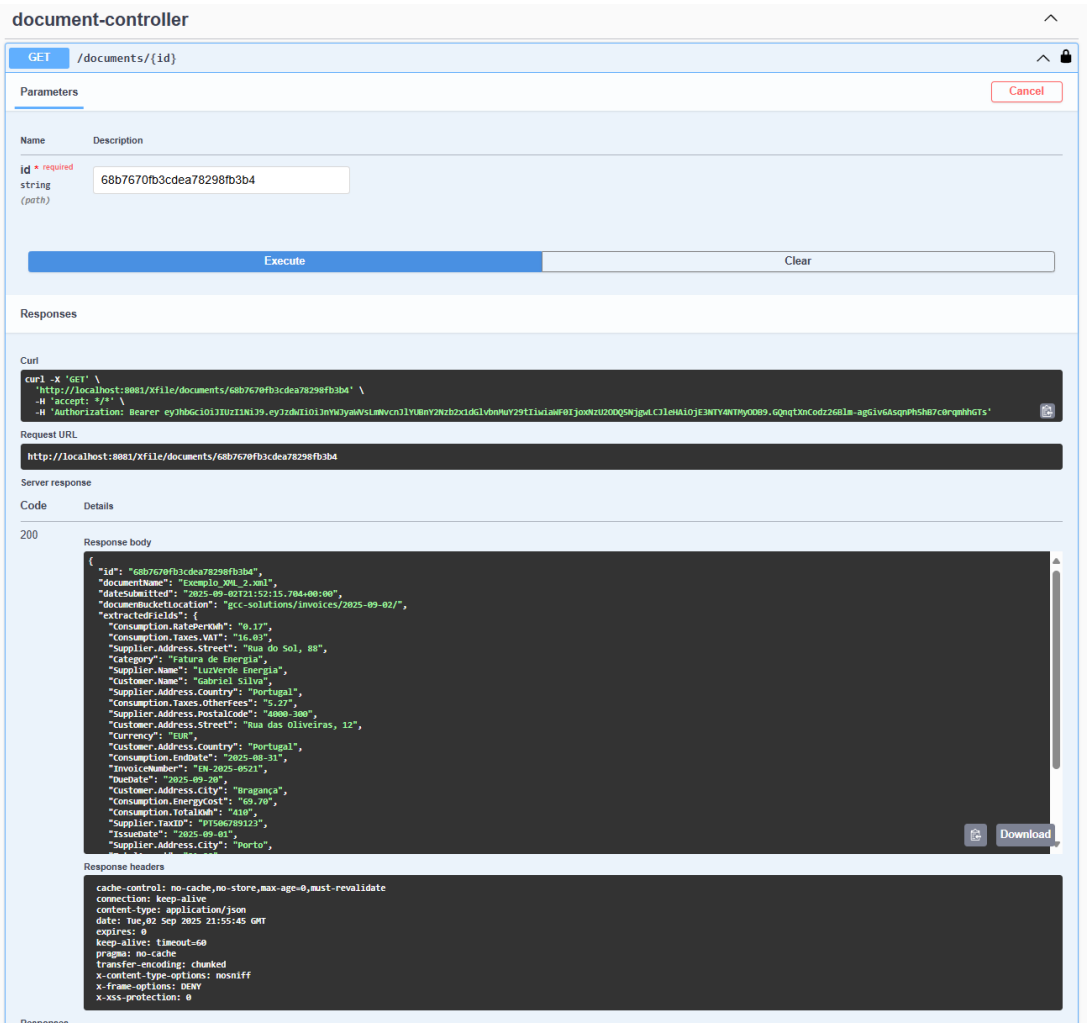

Figura 4.28: Atributos extraídos de um ficheiro de teste.

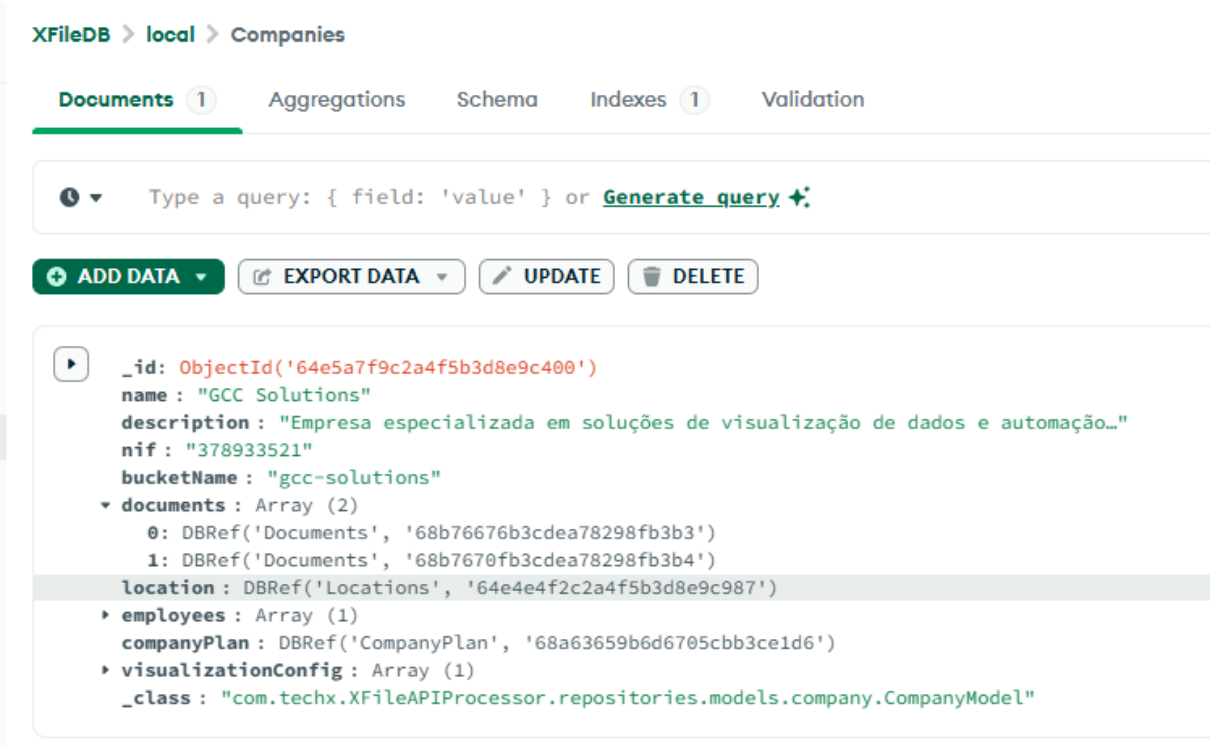

Figura 4.29: Empresa com 2 ficheiros registados.

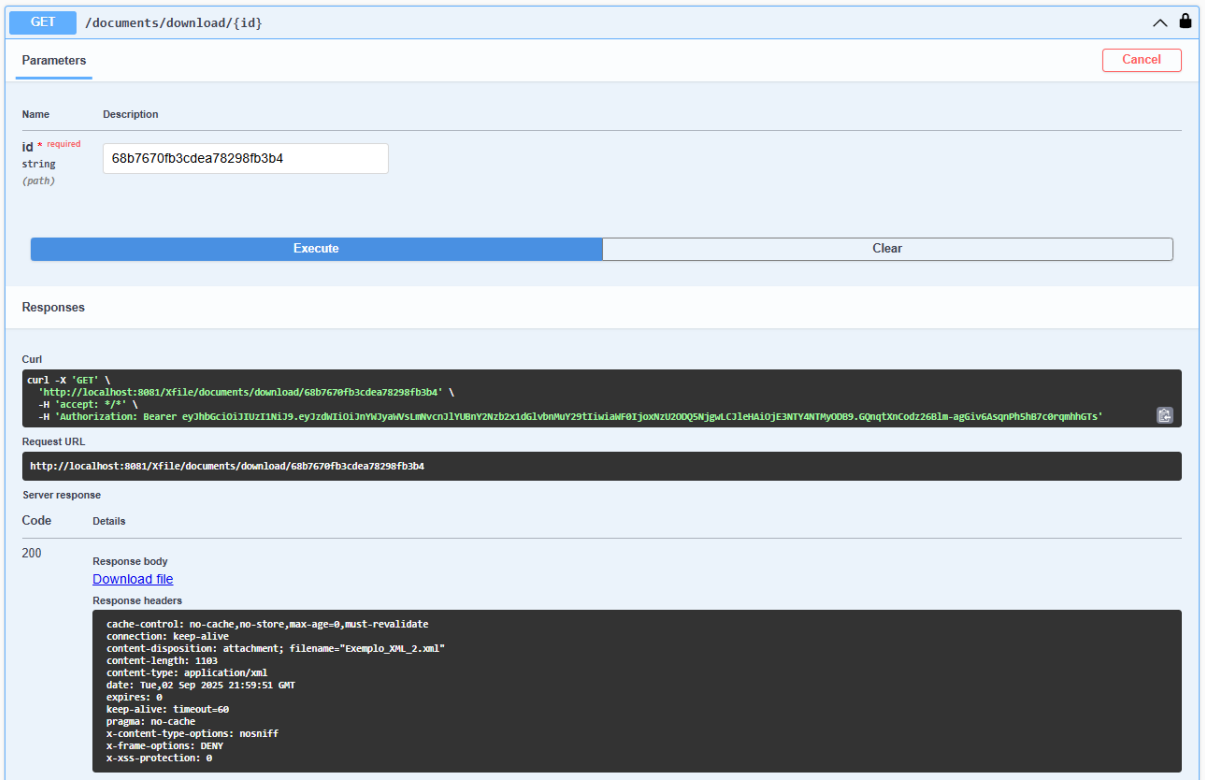

Figura 4.30: Download do ficheiro original.

Após a demonstração dos casos de uso anteriores, foi realizada a extração e indexação dos dados referentes ao segundo ficheiro submetido, conforme ilustrado na figura 4.28, cujo identificador único é “68b7670fb3cdea78298fb3b4”. A figura 4.31 apresenta o resultado de uma busca textual pela palavra “Oliveiras”, que corresponde a parte do endereço presente na fatura submetida.

O *XFile*, por meio da indexação vetorial implementada com Apache Lucene, é capaz de localizar documentos relacionados com base em termos individuais, expressões completas ou até mesmo nomes de campos. Esta funcionalidade permite realizar consultas semânticas sobre o conteúdo dos documentos, independentemente da sua estrutura original, garantindo precisão e relevância nos resultados apresentados.

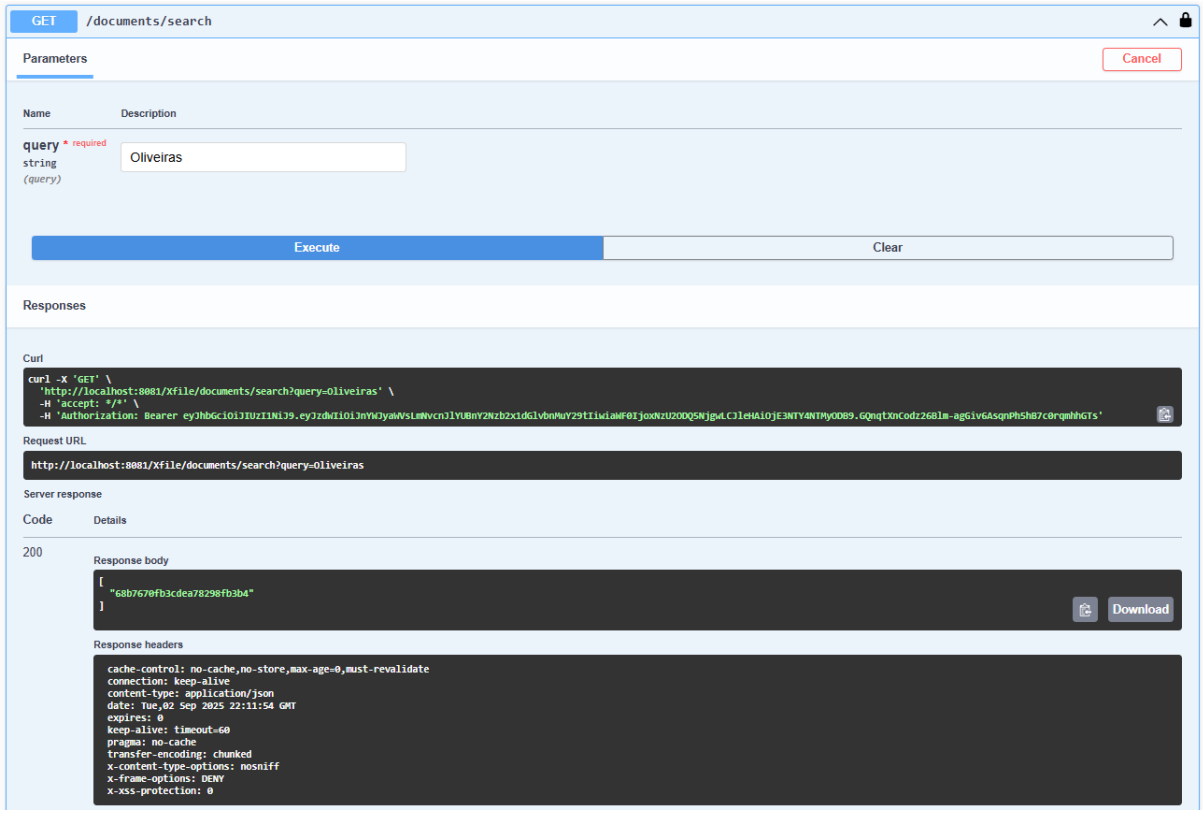

Figura 4.31: Busca por parte de conteúdo de um atributo do ficheiro.

O *XFile* realiza buscas utilizando três abordagens complementares internamente, otimizadas pela indexação vetorial com Apache Lucene. A primeira consiste na pesquisa por

conteúdo textual completo, permitindo localizar documentos que contenham exatamente a expressão ou frase consultada. A segunda abordagem realiza a busca por ocorrência nos nomes dos atributos extraídos, possibilitando identificar documentos com campos que correspondam ao termo pesquisado. Por fim, o sistema também executa buscas por fragmentos ou partes do conteúdo, permitindo localizar documentos que contenham variações, termos parciais ou menções indiretas ao texto fornecido.

Essa estratégia combinada garante maior flexibilidade e precisão na recuperação de documentos, independentemente da forma como o conteúdo foi estruturado ou submetido.

Capítulo 5

Conclusões e Trabalhos Futuros

O avanço da digitalização e da inovação no setor energético tem-se revelado essencial frente às crescentes exigências do mercado, seja no que diz respeito à eficiência energética, à otimização de processos operacionais ou à gestão da complexidade dos sistemas de energia. Neste cenário, soluções tecnológicas que promovam automação, rastreabilidade e segurança tornam-se cada vez mais indispensáveis.

O *XFile* posiciona-se como uma plataforma de gestão documental não apenas robusta e funcional, mas também altamente parametrizável e extensível. Sua arquitetura foi concebida para permitir a integração contínua de novas abordagens e serviços, assegurando que o sistema se mantenha estável, eficiente e escalável diante das transformações do setor.

Com o propósito de consolidar-se como o gestor documental central do GVE Online, o *XFile* integra um conjunto de tecnologias modernas por meio de uma arquitetura fiável e coerente com os requisitos funcionais e operacionais da plataforma. A sua implementação representa um passo significativo na direção de uma gestão documental inteligente, segura e adaptável às necessidades futuras do ecossistema energético.

Embora o *XFile* tenha sido concebido como uma solução robusta para gestão documental no contexto energético, diversas oportunidades de evolução podem ser exploradas em trabalhos futuros. A seguir, são destacadas algumas direções promissoras:

- **Aprimoramento da classificação com KNN**: Atualmente, a base de conhecimento utilizada pelo algoritmo KNN é limitada pela quantidade reduzida de documentos previamente categorizados, o que impacta diretamente na qualidade dos resultados obtidos. O enriquecimento contínuo dessa base, com maior diversidade de atributos e exemplos representativos, pode melhorar significativamente a capacidade de generalização e precisão da classificação. Além disso, a adoção de analisadores personalizados no Apache Lucene — como *tokenizers*, *filters* e *stemmers* ajustados ao domínio energético — pode contribuir para uma indexação mais refinada e contextual, aumentando a assertividade na identificação de padrões e na recuperação de documentos relevantes.

- **Expansão dos formatos suportados**: Atualmente, o sistema realiza extração e indexação de documentos nos formatos PDF, XML e JSON. A inclusão de novos formatos, como imagens escaneadas com OCR ou documentos em *DOCX*, pode ampliar significativamente o alcance da plataforma.

- **Integração com sistemas externos**: A criação de conectores para ERPs, plataformas de faturação eletrónica ou sistemas de gestão energética permitiria uma interoperabilidade mais fluida e uma automação mais abrangente dos processos.

- **Aprimoramento da busca semântica**: A utilização de modelos de linguagem mais avançados, como embeddings contextuais ou redes neurais para classificação documental, pode tornar a busca textual ainda mais precisa e inteligente.

- **Implementação do módulo de Business Analytics**: Embora previsto na arquitetura do XFile, o módulo de *Business Analytics* ainda não foi implementado. Sua futura integração representa uma oportunidade estratégica para o desenvolvimento de dashboards interativos voltados à visualização dos documentos indexados, acompanhamento do histórico de submissões e análise de trilhas de auditoria. Este módulo poderá oferecer recursos analíticos avançados, permitindo que gestores e

analistas extraiam informações relevantes a partir dos dados documentais, promovendo uma gestão mais informada e orientada por dados.

- **Mecanismos de versionamento e controle de alterações**: A implementação de um sistema de versionamento documental permitiria acompanhar modificações ao longo do tempo, garantindo rastreabilidade e conformidade regulatória.
- **Avaliação de desempenho em larga escala**: Estudos futuros podem investigar o comportamento do *XFile* em ambientes com alta carga de documentos, múltiplas empresas e diferentes perfis de utilização, visando otimizações de escalabilidade e tempo de resposta.

Essas propostas visam não apenas ampliar a funcionalidade do *XFile*, mas também consolidá-lo como uma solução adaptável e estratégica para o setor energético e outros domínios que demandem gestão documental inteligente.

Bibliografia

[1] T. X, *TECH X - Home*, Acessado em 5 de março de 2025. URL: https://techx.pt/.

[2] Electric Summit. «A urgência da digitalização da energia.» Acessado em: 10 out. 2025. (2022), URL: https://electricsummit.negocios.pt/transicao-energetica/a-urgencia-da-digitalizacao-da-energia/.

[3] S. Karus e H. Gall, «A Study of Language Usage Evolution in Open Source Software,» em *Proceedings of the 8th Working Conference on Mining Software Repositories (MSR)*, mai. de 2011, pp. 13–22. DOI: 10.1145/1985441.1985447.

[4] A. E. Fleury, «Programming in Java,» *ACM SIGCSE Bulletin*, vol. 32, n.º 1, pp. 197–201, mar. de 2000. DOI: 10.1145/331795.331854.

[5] J. Kwon, A. Wellings e S. King, «Ravenscar-Java: A High-Integrity Profile for Real-Time Java,» *Concurrency and Computation: Practice and Experience*, vol. 17, n.º 5–6, pp. 681–713, fev. de 2005. DOI: 10.1002/cpe.843.

[6] C. Praschl, A. Pointner, D. Baumgartner e G. A. Zwettler, «Imaging Framework: An Interoperable and Extendable Connector for Image-Related Java Frameworks,» *SoftwareX*, vol. 16, p. 100 863, nov. de 2021. DOI: 10.1016/j.softx.2021.100863.

[7] A. Lucene, *Lucene™ Features*, Acessado em 5 de março de 2025. URL: https://lucene.apache.org/core/features.html.

[8] P. Sojka e M. Líška, «The Art of Mathematics Retrieval,» em *Proceedings of the ACM Symposium on Document Engineering*, set. de 2011, pp. 57–60. DOI: 10.1145/2034691.2034703.

[9] A. Lucene, *Apache Lucene™ 10.1.0 Documentation*, Acessado em 5 de março de 2025. URL: https://lucene.apache.org/core/10_1_0/index.html.

[10] C. Walls, *Spring Boot in Action*. Manning Publications, 2016, ISBN: 978-1617292545.

[11] MongoDB, Inc., *MongoDB — Manual de Referência*, Acesso em 10 de agosto de 2025, MongoDB, Inc., 2025. URL: https://docs.mongodb.com/manual/.

[12] K. Banker, D. Garrett, P. Bakkum e S. Verch, *MongoDB in Action: Covers MongoDB version 3.0*. Simon e Schuster, 2016.

[13] «eBay.» Acesso em: 01 ago. 2025, eBay Inc. (2025), URL: https://www.ebay.com.

[14] «Craigslist.» Acesso em: 01 ago. 2025, Craigslist, Inc. (2025), URL: https://www.craigslist.org.

[15] «SourceForge.» Acesso em: 01 ago. 2025, SourceForge Media, LLC. (2025), URL: https://sourceforge.net.

[16] «The New York Times.» Acesso em: 01 ago. 2025, The New York Times Company. (2025), URL: https://www.nytimes.com.

[17] W. A. Rousan. «Building Scalable and Cost-Effective Data Warehouses with MinIO.» Acesso em: 01 ago. 2025. (ago. de 2024), URL: https://datahubanalytics.com/building-scalable-and-cost-effective-data-warehouses-with-minio/.

[18] «Data Lakehouse Solutions.» Acesso em: 01 ago. 2025, MinIO, Inc. (2025), URL: https://min.io/solutions/modern-data-lakes-lakehouses.

[19] «349 GB/s GET Benchmark on 32 NVMe Nodes.» Acesso em: 01 ago. 2025, MinIO, Inc. (2025), URL: https://min.io/blog/325-gibps-benchmark.

[20] N. Tiwari. «Modern Data Lake with MinIO : Part 1.» Acesso em: 01 ago. 2025. (2018), URL: https://blog.min.io/modern-data-lake-with-minio-part-1/.

[21] «Erasure Coding in MinIO.» Acesso em: 01 ago. 2025, MinIO, Inc. (2025), URL: https://min.io/docs/minio/erasure-coding.html.

[22] «Security Overview.» Acesso em: 01 ago. 2025, MinIO, Inc. (2025), URL: https://min.io/docs/minio/security-overview.html.

[23] GitHub, *Continuous Integration with GitHub Actions*, https://docs.github.com/pt/actions/about-github-actions/about-continuous-integration-with-github-actions, Acesso em: 31 jul. 2025, 2023.

[24] M. Reker, *Java CI/CD Pipeline using GitHub Actions*, https://github.com/MathiasReker/Java-CI-CD, Repositório GitHub demonstrando pipeline completo com Maven, Docker e deploy via SSH, 2022.

[25] S. Sivanandan, *Implementing a Robust CI/CD Pipeline with GitHub Actions for Cloud Applications*, https://stonetusker.com/implementing-a-robust-ci-cd-pipeline-with-github-actions-for-cloud-applications-accelerate-your-software-delivery/, Estudo de caso com aplicação Java Spring PetClinic e deploy em AWS, 2024.

[26] R. C. Thota, «CI/CD Pipeline Optimization: Enhancing Deployment Speed and Reliability,» *International Journal of Innovative Research in Management and Political Science*, vol. 2, n.º 2, 2020, Artigo técnico sobre otimização de pipelines CI/CD com GitHub Actions. URL: https://www.ijirmps.org/papers/2020/2/232185.pdf.

[27] DigitalOcean, *O Ecossistema do Docker: Uma Introdução aos Componentes Comuns*, 2015. URL: https://www.digitalocean.com/community/tutorials/o-ecossistema-do-docker-uma-introducao-aos-componentes-comuns-pt.

[28] R. Hat, *O que é orquestração de containers?* 2022. URL: https://www.redhat.com/pt-br/topics/containers/what-is-container-orchestration.

[29] J. P. Falcão, *Análise de Desempenho entre Máquinas Virtuais e Containers Utilizando o Docker*, 2022. URL: https://www.grupounibra.com/repositorio/

REDES/2022/analise-de-desempenho-entre-maquinas-virtuais-e-containers-utilizando-o-docker3.pdf.

[30] T. Teofili e J. Lin, *Lucene for Approximate Nearest-Neighbors Search on Arbitrary Dense Vectors*, 2019. arXiv: 1910.10208 [cs.IR]. URL: https://arxiv.org/abs/1910.10208.

[31] X. Ma, T. Teofili e J. Lin, *Anserini Gets Dense Retrieval: Integration of Lucene's HNSW Indexes*, 2023. arXiv: 2304.12139 [cs.IR]. URL: https://arxiv.org/abs/2304.12139.

[32] L. Hirsch, R. Hirsch e B. Ogunleye, «Document clustering with evolved multi-word search queries,» *Evolutionary Intelligence*, vol. 18, n.º 2, fev. de 2025, ISSN: 1864-5917. DOI: 10.1007/s12065-025-01018-w. URL: http://dx.doi.org/10.1007/s12065-025-01018-w.

[33] G. Rawat, «Reducing Incident Mean Time to Resolution Using Elasticsearch and Large Language Models,» *International Journal of Computer Trends and Technology*, vol. 73, pp. 125–132, mar. de 2025. DOI: 10.14445/22312803/IJCTT-V73I3P116.

[34] G. E. Pibiri e R. Venturini, «Techniques for Inverted Index Compression,» *ACM Computing Surveys*, vol. 53, n.º 6, pp. 1–36, dez. de 2020. DOI: 10.1145/3415148.

[35] N. Ukey, Z. Yang, B. Li, G. Zhang, Y. Hu e W. Zhang, «Survey on Exact kNN Queries over High-Dimensional Data Space,» *Sensors (Basel, Switzerland)*, vol. 23, n.º 2, p. 629, jan. de 2023. DOI: 10.3390/s23020629.

[36] P. Soucy e G. W. Mineau, «A simple KNN algorithm for text categorization,» em *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, nov. de 2001, pp. 647–648. DOI: 10.1109/icdm.2001.989592.

[37] A. Moldagulova e R. B. Sulaiman, «Using KNN algorithm for classification of textual documents,» em *Proceedings of the International Conference on Information*

Technology (ICITech), mai. de 2017, pp. 665–671. DOI: 10.1109/ICITech.2017.8079924.

[38] A. Arcuri, G. Fraser e J. Campos, «Unit Test Generation During Software Development: EvoSuite Plugins for Maven, IntelliJ and Jenkins,» em *Proceedings of the IEEE International Conference on Software Testing, Verification and Validation (ICST)*, abr. de 2016. DOI: 10.1109/ICST.2016.44.

[39] S. Chacon e B. Straub, *Pro Git*, 2, illustrated. Apress, 2014, p. 419, ISBN: 9781484200773.

[40] J. Ingeno, *Software Architect's Handbook: Become a Successful Software Architect by Implementing Effective Architecture Concepts*. Packt Publishing Ltd, 2018, p. 594, ISBN: 9781788627672.

[41] Gitea Contributors, *Gitea Documentation*, https://docs.gitea.com/, Acesso em: 10 de agosto de 2025, 2025.

[42] Wikipedia, *Apache Guacamole - Wikipedia*, https://en.wikipedia.org/wiki/Apache_Guacamole, Acesso em: 31 jul. 2025, 2025.

[43] A. S. Foundation, *Apache Guacamole®*, https://guacamole.apache.org/, Acesso em: 31 jul. 2025, 2025.

[44] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*. Upper Saddle River, NJ: Prentice Hall, 2003, ISBN: 9780135974445.

[45] E. Gamma, R. Helm, R. Johnson e J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995, ISBN: 0-201-63361-2.