Revista Perquirere, vol. 23, n. 2: 141-157, 2026
© Centro Universitário de Patos de Minas
https://revistas.unipam.edu.br/index.php/perquirere

Sistema de gerenciamento de atividades cotidianas de profissionais da área de segurança do trabalho

System for managing daily activities of occupational safety professionals

JÚLIA VASCONCELOS CAIXETA RIBEIRO
Discente de Sistemas de Informação (UNIPAM)
juliavasconceloscr@unipam.edu.br

JULIANA LILIS DA SILVA
Professora orientadora (UNIPAM)
juliana@unipam.edu.br

Resumo: O artigo descreve o desenvolvimento de um sistema para otimizar a gestão da segurança do trabalho, com foco na centralização de informações e na análise de riscos ocupacionais. A proposta busca modernizar os processos da área, oferecendo maior precisão, organização e agilidade na gestão de dados relacionados à saúde e segurança dos trabalhadores. A solução apoia profissionais na identificação, monitoramento e mitigação de riscos, auxiliando na tomada de decisão. Para isso, adotou-se uma abordagem exploratória, descritiva e aplicada, orientando o levantamento de requisitos e o desenvolvimento do sistema. A aplicação foi construída com arquitetura de microsserviços, garantindo escalabilidade, flexibilidade e independência entre módulos, utilizando tecnologias como Spring Boot e Angular. Os resultados demonstram que o sistema atende às necessidades identificadas, promovendo integração entre Recursos Humanos e Segurança do Trabalho e contribuindo para uma gestão mais eficiente e preventiva, alinhada às normas regulamentadoras.
Palavras-chave: segurança do trabalho; microsserviços; gestão de riscos; arquitetura de *software*.

Abstract: This article describes the development of a system designed to optimize occupational safety management, focusing on the centralization of information and the analysis of occupational risks. The proposal seeks to modernize processes in the field by providing greater accuracy, organization, and agility in the management of data related to workers' health and safety. The solution supports professionals in identifying, monitoring, and mitigating risks, assisting in decision-making. To this end, an exploratory, descriptive, and applied approach was adopted, guiding requirements elicitation and system development. The application was built using a microservices architecture, ensuring scalability, flexibility, and independence among modules, and employing technologies such as Spring Boot and Angular. The results demonstrate that the system meets the identified needs, promoting integration between Human Resources and Occupational Safety and contributing to more efficient and preventive management aligned with regulatory standards.
Keywords: occupational safety; microservices; risk management; software architecture.

1 INTRODUÇÃO

A tecnologia tem se tornado cada vez mais presente no ambiente profissional, impulsionando a modernização e a otimização de diversos setores. Entre essas inovações, os sistemas de gerenciamento de atividades têm sido amplamente adotados, permitindo a centralização e a organização eficiente de dados que, de outra forma, estariam dispersos em documentos físicos ou planilhas manuais. Esse processo de digitalização facilita o cadastro, a análise de informações e a tomada de decisões, tornando os processos mais ágeis e precisos. Nesse contexto, diferentes setores vêm se adaptando ao avanço tecnológico, especialmente aqueles que lidam com grandes volumes de dados e exigem precisão no gerenciamento de informações.

Entre as áreas que mais se beneficiam da digitalização, a segurança do trabalho se destaca pela necessidade de controle rigoroso e organização no gerenciamento de informações. A tecnologia não apenas otimiza processos administrativos, mas também fortalece a prevenção de acidentes e a conformidade com normas regulamentadoras. Os profissionais desse setor realizam inspeções periódicas, identificam potenciais riscos e implementam medidas corretivas, além de promoverem treinamentos e palestras educativas para conscientizar os funcionários sobre práticas seguras no ambiente de trabalho.

Amparados pelas Normas Regulamentadoras do Ministério do Trabalho, esses profissionais desempenham um papel fundamental na elaboração e execução de programas de prevenção de acidentes. Diante da complexidade e do volume de informações envolvidas, o uso de um sistema de gerenciamento torna-se essencial para proporcionar maior controle, organização e eficiência na execução dessas atividades (Brasil, 2021).

Dessa forma, esta pesquisa responde à seguinte questão norteadora: como a tecnologia pode otimizar a gestão da segurança do trabalho, facilitando a análise de riscos e a tomada de decisões?

Observa-se, neste contexto, que é de grande relevância o desenvolvimento de novas abordagens e aprimoramento de soluções tecnológicas voltadas para a gestão de riscos ocupacionais. Soluções tecnológicas, como o uso de software, podem auxiliar as empresas a terem um maior controle sobre suas atividades diárias, proporcionando uma gestão mais eficiente da segurança no ambiente de trabalho, reduzindo riscos e fortalecendo a conformidade com as normativas vigentes.

Considerando esse cenário, este trabalho teve como objetivo desenvolver uma plataforma integrada para otimizar a gestão da segurança do trabalho, centralizando informações essenciais e disponibilizando ferramentas para uma administração eficiente dos dados ocupacionais. A solução proposta permitiu a organização de informações relacionadas às condições do ambiente de trabalho, fontes geradoras de riscos e funções desempenhadas pelos trabalhadores. A interconexão desses dados possibilitou uma análise mais precisa na avaliação de riscos, facilitando a identificação de padrões, a implementação de medidas preventivas e a tomada de decisões estratégicas para mitigação de acidentes.

Para atingir o objetivo proposto, foram definidos os seguintes objetivos específicos: identificar, junto a profissionais da área de segurança do trabalho, os

requisitos funcionais e não funcionais necessários para o desenvolvimento da plataforma; compreender o processo de análise de riscos na segurança do trabalho; desenvolver um protótipo ilustrativo das telas do sistema; construir uma aplicação capaz de auxiliar na gestão das atividades diárias dos profissionais de segurança do trabalho.

Quanto à metodologia, este trabalho adotou uma abordagem exploratória, descritiva e aplicada, buscando compreender e descrever as necessidades dos profissionais de segurança do trabalho e os desafios enfrentados na gestão de riscos ocupacionais. Com base nesse conhecimento, foi desenvolvida a plataforma proposta.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados conceitos e estudos realizados sobre segurança do trabalho e estilos de arquitetura de *software*.

2.1 SEGURANÇA DO TRABALHO

A Segurança do Trabalho refere-se ao conjunto de medidas destinadas a reduzir acidentes laborais e doenças ocupacionais, além de proteger a integridade e a capacidade de trabalho dos colaboradores. Na atualidade, a área de Saúde e Segurança Ocupacional é vista como estratégica para o crescimento e desenvolvimento das empresas. Ela está completamente integrada aos processos e métodos de trabalho, buscando competitividade, qualidade e melhores condições de vida para os trabalhadores, atuando não apenas na adequação de práticas, mas também na promoção de uma cultura de prevenção (Peixoto, 2010).

Os fatores de risco para a saúde e segurança dos trabalhadores podem ser classificados em cinco grandes grupos, conforme a Organização Pan-Americana da Saúde no Brasil. Esses grupos incluem riscos físicos, que envolvem condições ambientais adversas como ruído e radiações; riscos químicos, relacionados a substâncias químicas em diferentes estados físicos; riscos biológicos, envolvendo microrganismos que geralmente afetam profissionais em ambientes hospitalares e agropecuários; riscos ergonômicos e psicossociais, que decorrem da má organização e gestão do trabalho; e riscos de acidentes, que abrangem desde a proteção inadequada de máquinas até a desorganização do ambiente de trabalho (Guimarães *et al.*, 2004).

De acordo com Guimarães *et al* (2004), esses riscos ocupacionais, frequentemente associados a atividades insalubres e perigosas, podem gerar sérios impactos na saúde dos profissionais, especialmente em ambientes como hospitais, onde o controle sobre agentes biológicos, químicos, físicos e mecânicos é fundamental. Especialistas em Higiene e Segurança do Trabalho destacam que, mais do que conceder adicionais de insalubridade ou periculosidade, é essencial criar condições que tornem o ambiente de trabalho seguro e saudável, prevenindo assim os danos à saúde dos trabalhadores.

2.2 MICROSSERVIÇOS

Arquitetura de microsserviços é uma abordagem na qual uma aplicação é constituída por componentes de *software* que se comunicam e são independentes entre si. Cada serviço pode utilizar sua própria tecnologia e ser construído a partir de uma parte do negócio, realizando, dessa forma, uma função específica e bem determinada (Villaça; Pimenta Junior; Azevedo, 2018).

Esses serviços podem se comunicar de várias formas. De acordo com Villaça, Pimenta Junior e Azevedo (2018), a principal delas é por *web services*. Nesse caso, o serviço fica exposto através de um *endpoint*, recebe as requisições de outros serviços, que agem como clientes, e é enviada a resposta, realizando a comunicação. A web *service* SOAP utiliza tecnologias baseadas em XML (*Extensible Markup Language*), enquanto o *web service* REST (*Representational State Transfer*) é baseado em métodos HTTP. No entanto, REST é geralmente mais fácil de usar, pois permite um estilo de design de aplicações mais coeso e menos acoplado, baseado em recursos nomeados.

Na arquitetura de microsserviços, fatores como confiabilidade e tolerância a falhas precisam de atenção, mas podem ser garantidos utilizando estratégias para lidar com a sobrecarga da rede. Os microsserviços representam um modelo arquitetural mais sustentável, embora sua implementação demande um investimento significativo em termos de construção e manutenção, além de exigir um profundo entendimento do problema em questão. Isso faz com que algumas empresas não estejam prontas para adotar esse estilo arquitetural e, consequentemente, não possam aproveitar todos os benefícios que ele oferece (Mendes, 2021).

2.3 PADRÃO REST

De acordo com Galindo Junior, Rocha e Maciel (2021), REST é uma arquitetura desenvolvida para sistemas distribuídos baseados em redes, que foi criada por Roy Thomas Fielding, descrita em sua dissertação de doutorado na Universidade da Califórnia, em Irvine. Esse padrão é definido por Fielding (2000) como um conjunto de restrições ou princípios que visam otimizar a estruturação de um sistema, a maneira como os componentes se identificam e se comunicam, a comunicação das informações e a evolução independente dos elementos.

REST é um modelo para organizar os recursos de um sistema web, permitindo uma comunicação padronizada e possibilitando que esses recursos operem de maneira independente. Nesse modelo, o sistema é composto por diversos recursos ou componentes que funcionam autonomamente, facilitando a troca de informações por meio de *web services*. Essa abordagem de particionamento oferece vantagens significativas, pois permite que as partes do sistema evoluam sem afetar o funcionamento geral. Por exemplo, um recurso de gestão de ordens de serviço pode ser aprimorado para incluir um algoritmo de agendamento e até mesmo integrar-se com um serviço de e-mail para enviar notificações de ordens de serviço diretamente ao e-mail do cliente (Galindo Junior; Rocha; Maciel, 2021).

No contexto do modelo de API REST, cada método HTTP tem uma semântica específica e bem definida. O método GET deve ser utilizado para recuperar uma

representação do estado de um recurso, enquanto o HEAD é utilizado para recuperar o metadado associado ao estado do recurso. O método PUT deve ser usado para adicionar um novo recurso a um repositório ou atualizar um recurso existente. O DELETE remove um recurso de sua hierarquia. Já o método POST deve ser utilizado para criar um novo recurso a uma coleção e executar *controllers*. Uma mensagem de requisição POST pode incluir tanto cabeçalhos quanto um corpo como entradas para a função de um *controller* (Masse, 2012).

2.4 QUALIDADE DE *software*

Os seis principais atributos de qualidade de *software* são: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade. De acordo com Vasconcelos *et al*. (2007):

- Funcionalidade: refere-se à capacidade do *software* de atender a necessidades explícitas e implícitas durante seu uso.
- Confiabilidade: diz respeito à sua capacidade de manter o desempenho previsto sob condições especificadas.
- Usabilidade: envolve a facilidade de compreensão, aprendizado, uso e valorização pelo usuário.
- Eficiência: está relacionada ao desempenho adequado em relação aos recursos utilizados.
- Manutenibilidade: representa a facilidade de modificação para correções, melhorias ou adaptações diante de mudanças no ambiente ou nos requisitos.
- Portabilidade: diz respeito à capacidade do *software* de ser transferido entre diferentes ambientes.

No contexto da qualidade de *software*, a implementação dos atributos mencionados pode ser respaldada pelo livro Código Limpo, de Robert C. Martin. Martin (2009) destaca o impacto negativo de um código confuso para as equipes de desenvolvimento e ressalta a importância de um código limpo, detalhando ao longo da obra como implementá-lo de forma padronizada e de fácil manutenção.

Outro fator relacionado com qualidade de *software* e essencial para alcançar esse objetivo é a adoção de testes unitários. Sem uma coleção de testes, os desenvolvedores perdem a capacidade de garantir que o sistema continue funcionando corretamente após modificações no código-fonte, além de não conseguirem assegurar que mudanças em uma parte do sistema não afetem outras (Martin, 2009).

Para Roy Osherove (2013), um teste unitário é um trecho de código que invoca uma unidade de trabalho e verifica um resultado específico dessa unidade. Caso as suposições sobre o resultado esperado estejam incorretas, o teste unitário falha. O escopo de um teste unitário pode variar desde um único método até múltiplas classes. Para ele, um teste unitário deve ser automatizado e repetível, fácil de implementar, relevante no futuro, executado com o pressionar de um botão, rápido, consistente em seus resultados, ter controle total sobre a unidade testada, ser totalmente isolado e, em caso de falha, permitir a fácil detecção do que era esperado e a identificação do problema.

3 METODOLOGIA

Quanto à metodologia, este trabalho adotou uma abordagem exploratória, com o objetivo de proporcionar maior familiaridade com o problema e aprofundar o entendimento sobre as demandas do setor de segurança do trabalho. Esse tipo de pesquisa é especialmente útil em contextos onde há pouco conhecimento consolidado, permitindo identificar variáveis relevantes, levantar hipóteses e direcionar futuras decisões de desenvolvimento. A pesquisa também teve caráter descritivo, ao detalhar os requisitos e processos envolvidos, e aplicado, ao buscar soluções práticas por meio da construção de um sistema voltado à melhoria da gestão de riscos ocupacionais.

Foram utilizadas pesquisas em livros e em sites que continham informações relevantes e atualizadas sobre tecnologia e segurança do trabalho. Além disso, realizou-se um levantamento de requisitos por meio de reuniões com uma consultora da área de segurança do trabalho, a fim de compreender profundamente o problema e encontrar uma solução que atenda às necessidades do mercado.

Para o desenvolvimento do *software* foram utilizados *frameworks* e tecnologias como a linguagem Java e o *framework* Spring Boot, que fornecem um suporte à implementação de sistemas robustos. O editor de código IntelliJ IDEA Community foi utilizado para construir a parte de *back-end* do sistema.

Já para o *front-end* foram utilizadas tecnologias como o *framework* Angular, com a linguagem TypeScript, e o editor de código VSCODE. Foi feita também uma prototipação utilizando o *software* Figma, para que pudesse ser exibida uma visão das páginas, podendo assim, colher *feedbacks* da consultora da área de segurança do trabalho. Para garantir que as soluções propostas atingissem os níveis esperados de funcionalidade, usabilidade e compatibilidade, foram realizados testes unitários.

Por fim, tendo em vista que é de extrema importância manter-se organizado durante o desenvolvimento, deixando os objetivos claros e próximos passos evidentes, foi utilizada a ferramenta Trello. Esse recurso permite a implementação do *framework* Kanban, que auxilia na organização e divisão de tarefas por tempo. Além disso, adotou-se o método de entrega por iterações, que permitiu receber *feedbacks* constantes de um profissional da área de segurança do trabalho a cada entrega de pequeno pacote de desenvolvimento do trabalho, otimizando-o e diminuindo as possibilidades de modificações e manutenções futuras.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS/ DESENVOLVIMENTO

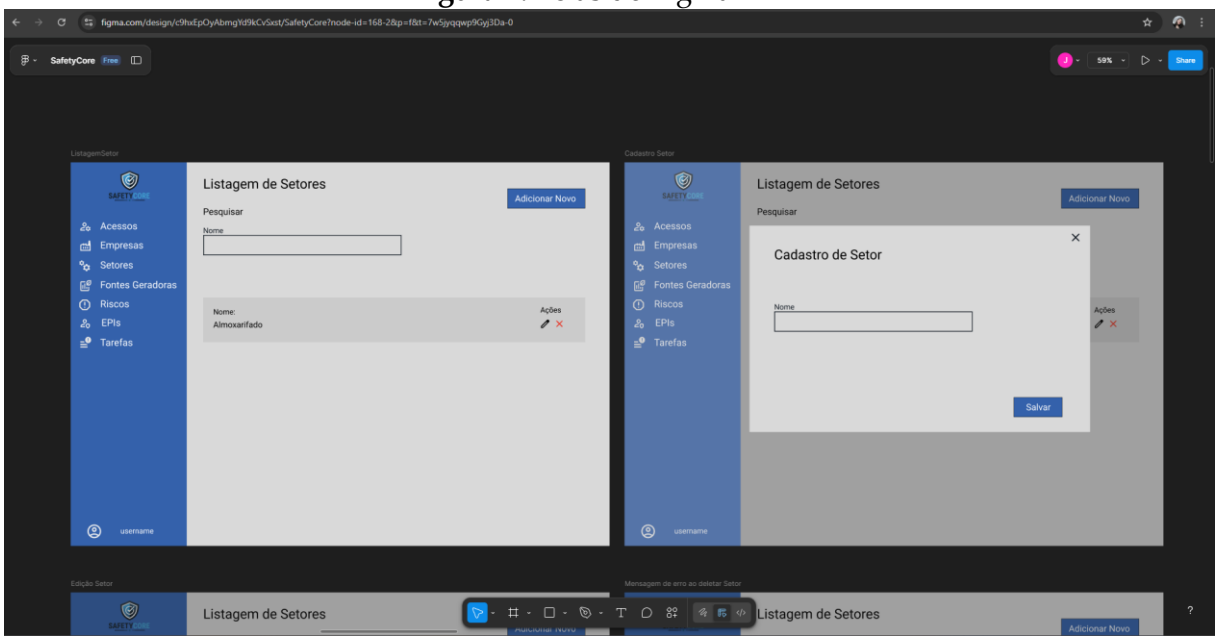
Esta seção tem como objetivo apresentar os resultados obtidos durante o desenvolvimento do sistema. Está organizada em duas partes: a primeira apresenta uma visão geral da aplicação; a segunda aborda a arquitetura adotada e descreve brevemente o processo de construção do sistema.

4.1 VISÃO GERAL DO SISTEMA

O desenvolvimento da aplicação teve início com o levantamento dos principais requisitos do sistema, entre os quais se destacaram a redução de cadastros redundantes, a centralização das informações e a integração entre os módulos de Recursos Humanos e Gestão de Riscos. Além disso, considerou-se essencial que o sistema permitisse o cadastro de várias empresas, atendendo especialmente aos consultores da área de Segurança do Trabalho que necessitam gerenciar múltiplas organizações.

A partir desses requisitos, definiu-se a visão do sistema sob a perspectiva de um consultor de Segurança do Trabalho, capaz de administrar a segurança de empresas, cadastrar diversas organizações e utilizar cadastros básicos, como riscos, funções e fontes geradoras, para vinculá-los a essas empresas. Assim, foram modeladas interfaces no Figma, com o objetivo de atender aos requisitos de forma flexível e intuitiva. Como exemplo, pode-se observar, na Figura 1, a apresentação da funcionalidade de "Listagem de Setores". Nessa figura, está representado o padrão adotado para as telas da aplicação, com a barra de navegação à esquerda, posicionamento da logomarca da empresa na parte superior e acesso às funcionalidades na parte central da tela.

Figura 1: Telas do Figma



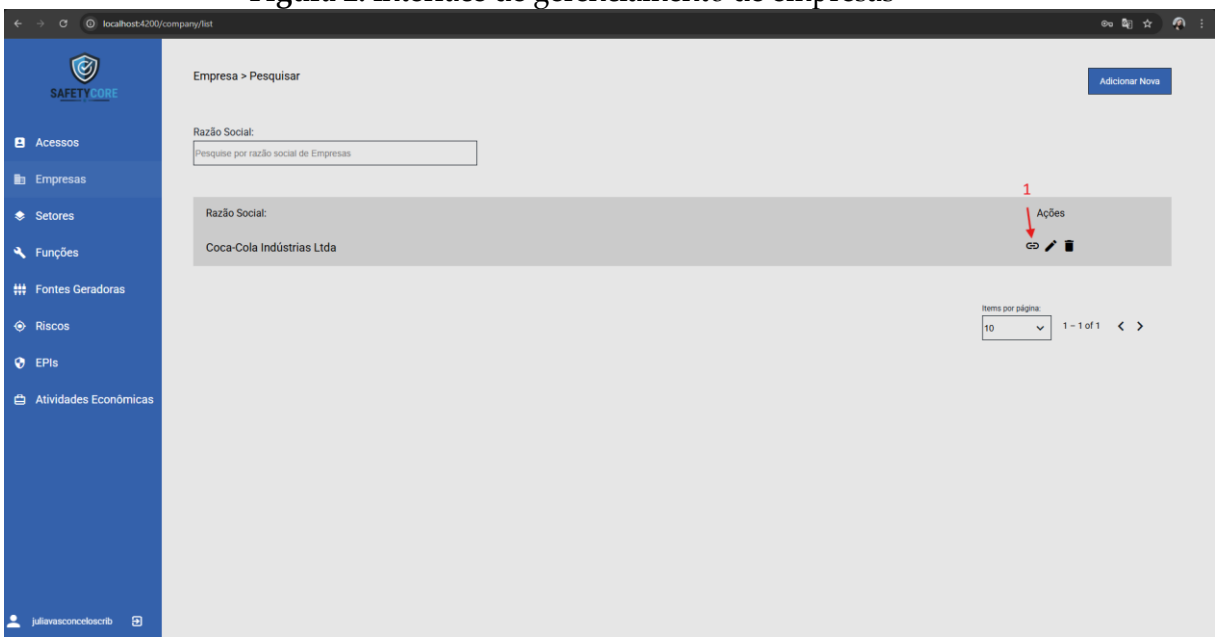
Fonte: elaborado pelas autoras, 2025.

Conforme demonstra a Figura 1, na barra de navegação foram disponibilizados os cadastros básicos. Esses cadastros permitem a inserção de dados que poderão ser utilizados por todas as outras empresas cadastradas. Por exemplo, no caso dos setores, nomes idênticos podem existir em diferentes empresas, como "Almoxarifado". Para simplificar o processo, o usuário pode acessar a funcionalidade "Setores" e realizar o cadastro uma única vez, reutilizando-o em várias empresas. O mesmo princípio aplica-se aos cadastros de funções, fontes geradoras e Equipamentos de Proteção Individual (EPIs).

Após o registro dos cadastros básicos, é necessário realizar os vínculos, etapa fundamental do sistema. Nela, são criadas as associações entre empresas, setores, funções e os riscos correspondentes. Posteriormente, essas informações podem ser relacionadas aos funcionários de cada empresa, permitindo associar de forma simples os trabalhadores aos riscos aos quais estão expostos.

A partir da Figura 2, é possível observar as telas implementadas no sistema. Nesse exemplo, que apresenta o "Gerenciamento de Empresas", ao selecionar o item 1, o usuário tem acesso à funcionalidade de associações da empresa.

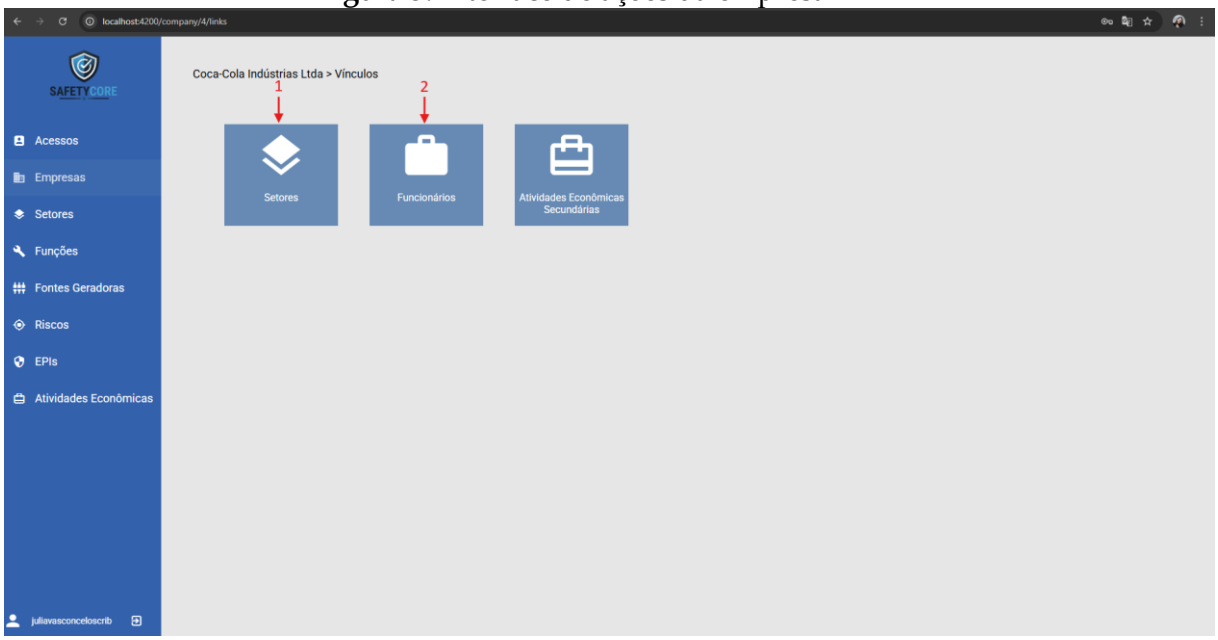
Figura 2: Interface de gerenciamento de empresas



Fonte: elaborado pela autora, 2025.

Conforme apresentado na Figura 3, a aplicação permite cadastrar funcionários, declarar atividades econômicas secundárias e gerenciar os vínculos de setor, função e risco, especificamente daquela empresa.

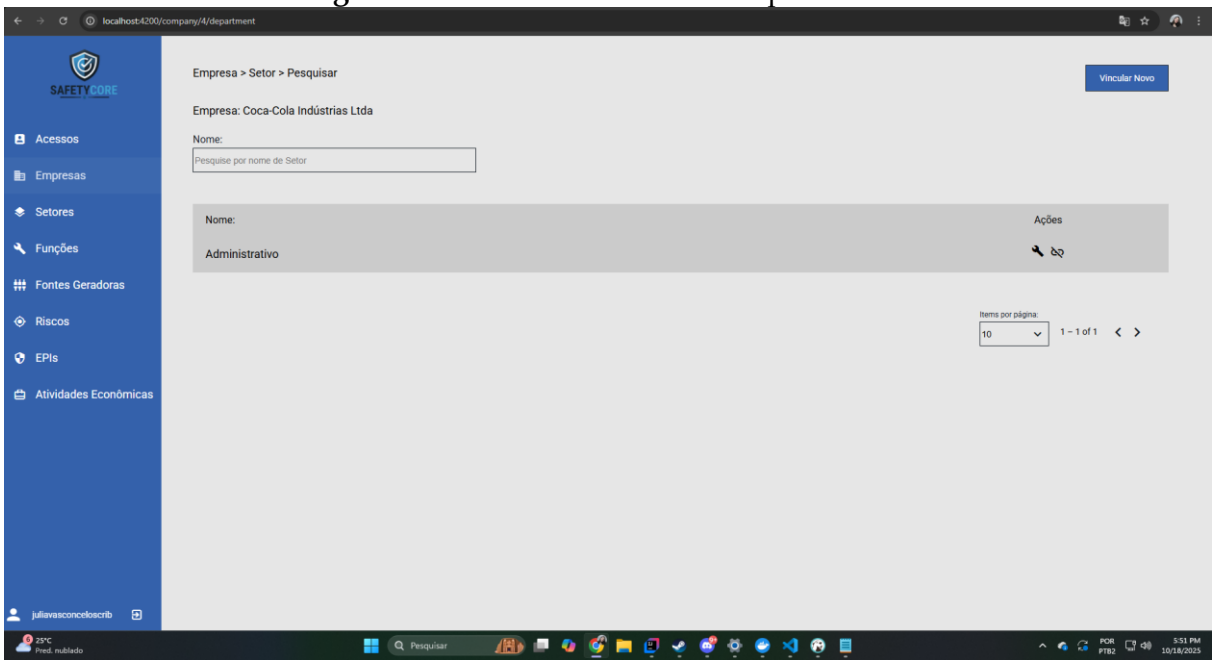
Figura 3: Interface de ações da empresa



Fonte: elaborado pelas autoras, 2025.

Ao acionar o card de "Setores", item 1 da Figura 3, o usuário é direcionado para a interface ilustrada na Figura 4, na qual são exibidos todos os setores vinculados à empresa selecionada anteriormente. Nela, é possível adicionar novos setores, remover vínculos existentes ou acessar a funcionalidade de "Funções".

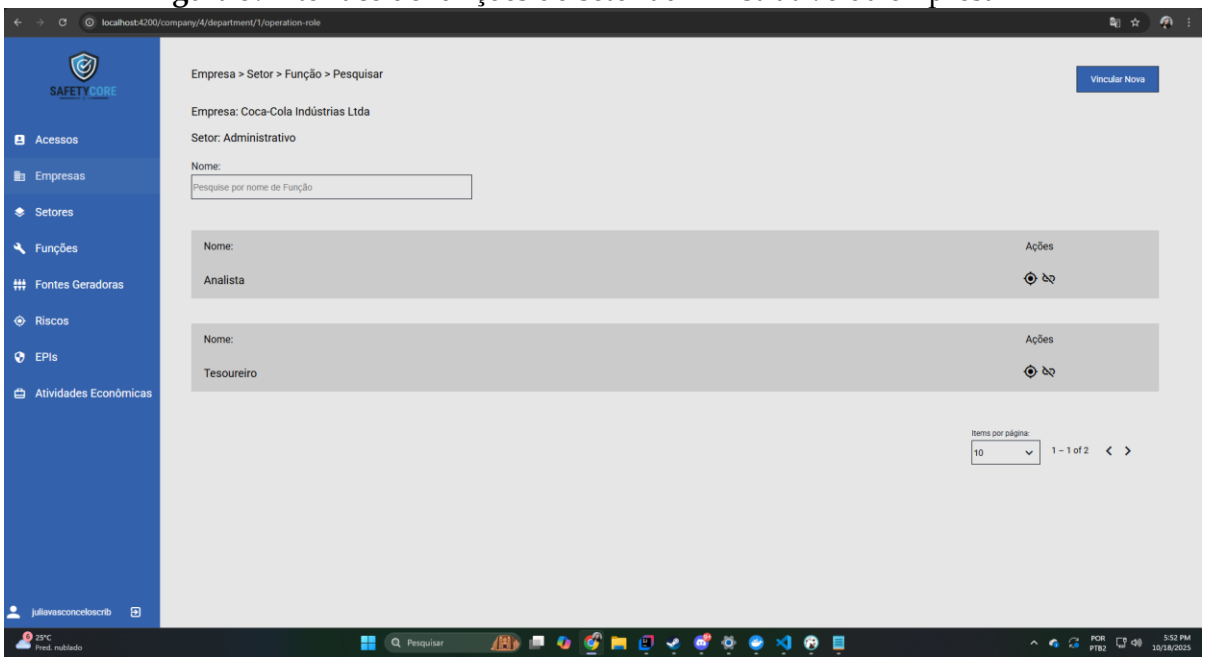
Figura 4: Interface de setores da empresa



Fonte: elaborado pelas autoras, 2025.

Já ao selecionar o ícone de “Funções”, o usuário pode visualizar, conforme mostra a Figura 5, todas as funções exercidas no setor escolhido. Nessa interface, é possível realizar novos vínculos, excluir associações existentes ou acessar os riscos relacionados àquela função e setor.

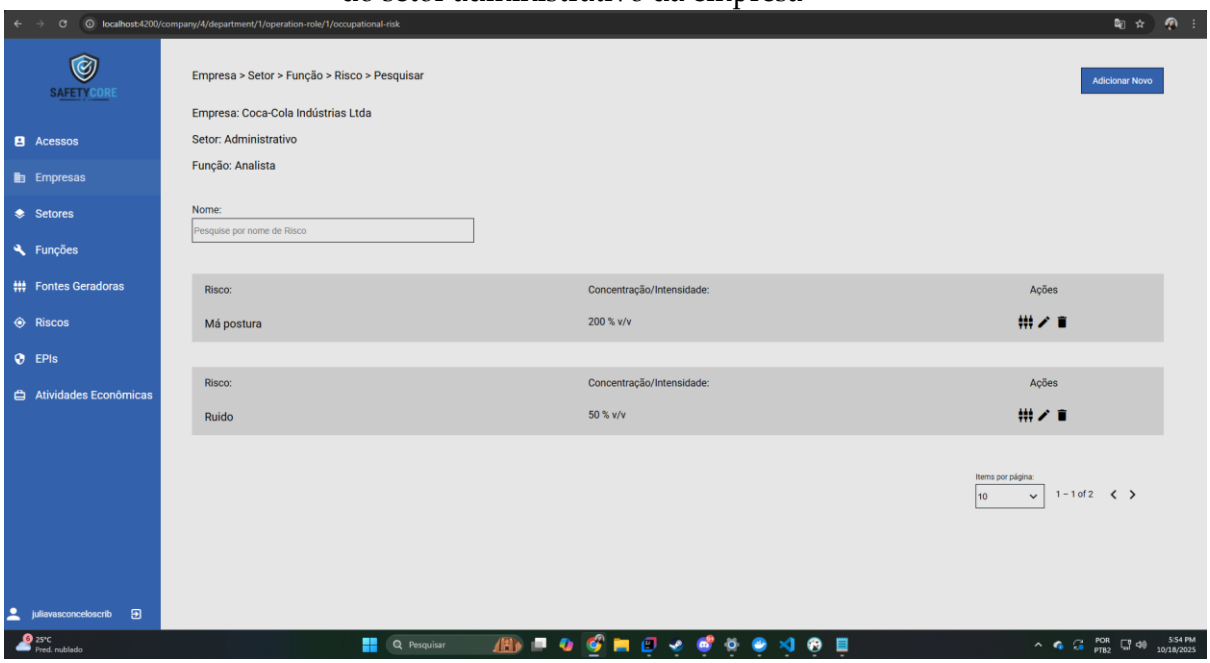
Figura 5: Interface de funções do setor administrativo da empresa



Fonte: elaborado pelas autoras, 2025.

Como mostra a Figura 6, o sistema possibilita associar riscos às funções exercidas em determinados setores, definindo concentração ou intensidade. Essas ações são executadas de forma simples e sem duplicidade de cadastros, uma vez que as informações são aproveitadas dos registros previamente realizados nas funcionalidades acessíveis pela barra de navegação. Dessa forma, obtém-se uma significativa redução no número de cadastros e maior eficiência no gerenciamento das informações.

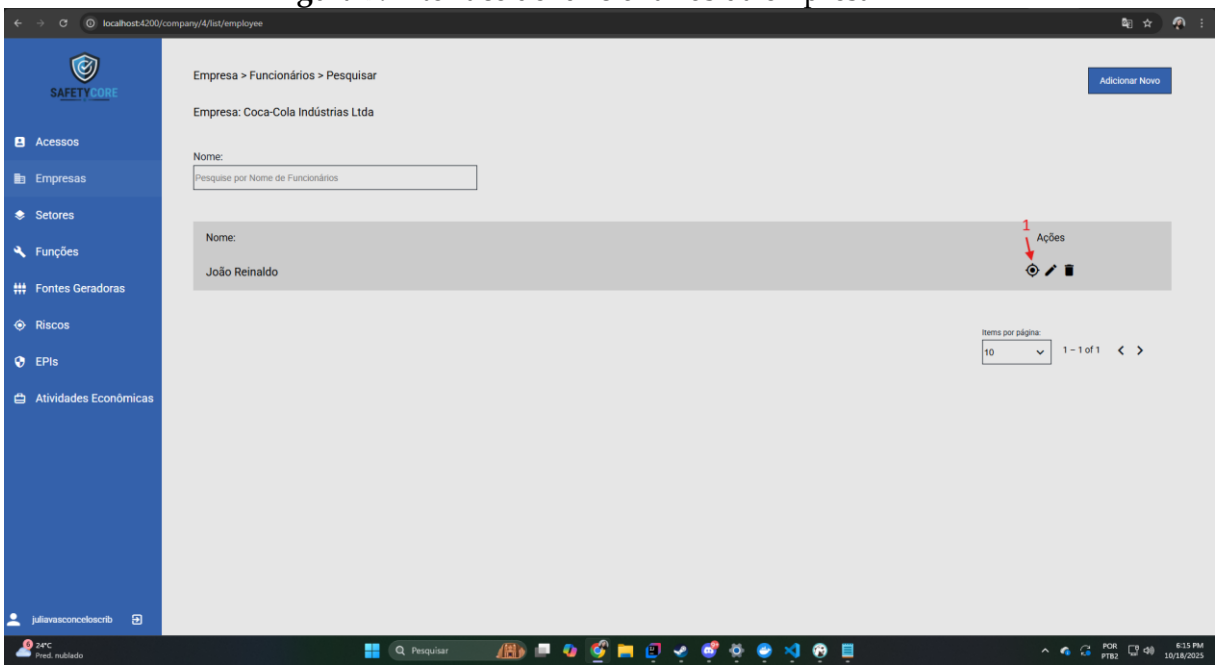
Figura 6: Interface de riscos associados à função analista do setor administrativo da empresa



Fonte: elaborado pelas autoras, 2025.

Retornando à Figura 3 e acessando a aba de funcionários, identificada pelo item 2, o usuário é direcionado para a interface representada na Figura 7, que evidencia a integração entre os módulos de Recursos Humanos e Segurança do Trabalho. Nela, é possível verificar a lista de funcionários de uma empresa, gerenciar seus cadastros e visualizar os riscos aos quais cada colaborador está exposto.

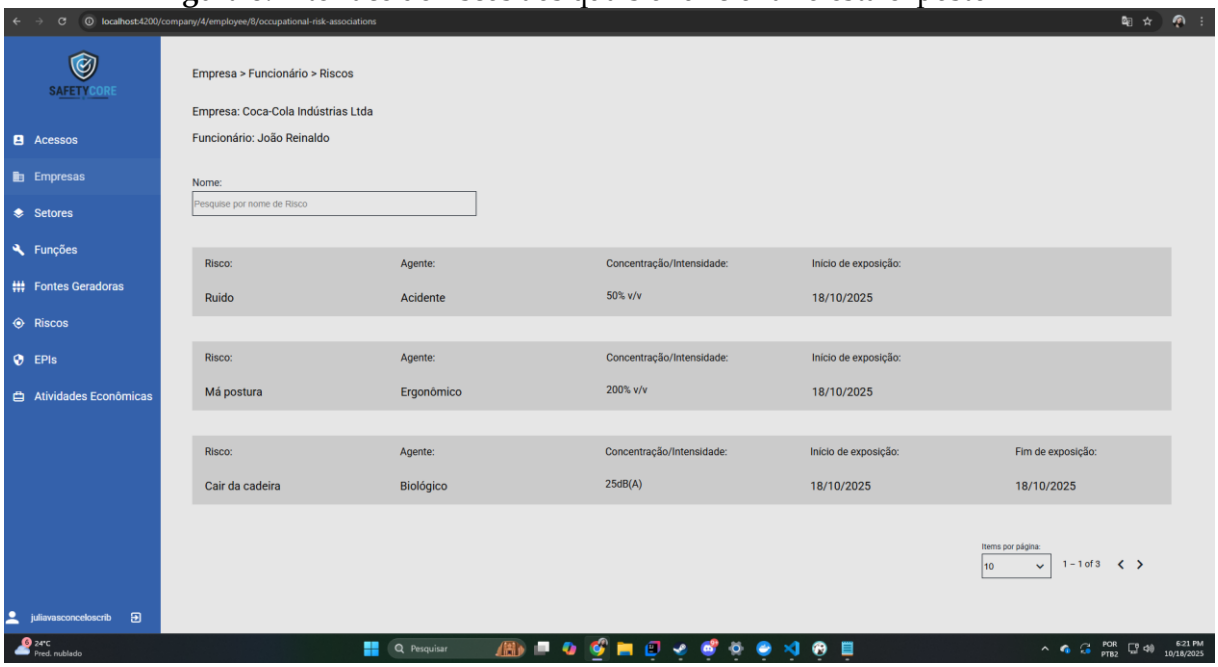
Figura 7: Interface de funcionários da empresa



Fonte: elaborado pelas autoras, 2025.

Além disso, não há repetição de cadastros, pois, ao criar um funcionário e selecionar seu setor e função, o sistema associa automaticamente os riscos correspondentes, previamente definidos para aquele setor e função, incluindo a data de início e, quando aplicável, a data de término da exposição. Essa funcionalidade pode ser visualizada na Figura 8, ao acionar o ícone de riscos na listagem de funcionários, item 1 da Figura 7.

Figura 8: Interface de riscos aos quais o funcionário está exposto



Fonte: elaborado pelas autoras, 2025.

Essa implementação resulta em um gerenciamento mais eficiente dos riscos ocupacionais, oferecendo maior segurança e confiabilidade ao trabalho dos profissionais responsáveis pela área. Além disso, o sistema reduz significativamente o esforço necessário para a realização dos cadastros, tornando o processo mais ágil, intuitivo e menos suscetível a erros.

4.2 ARQUITETURA

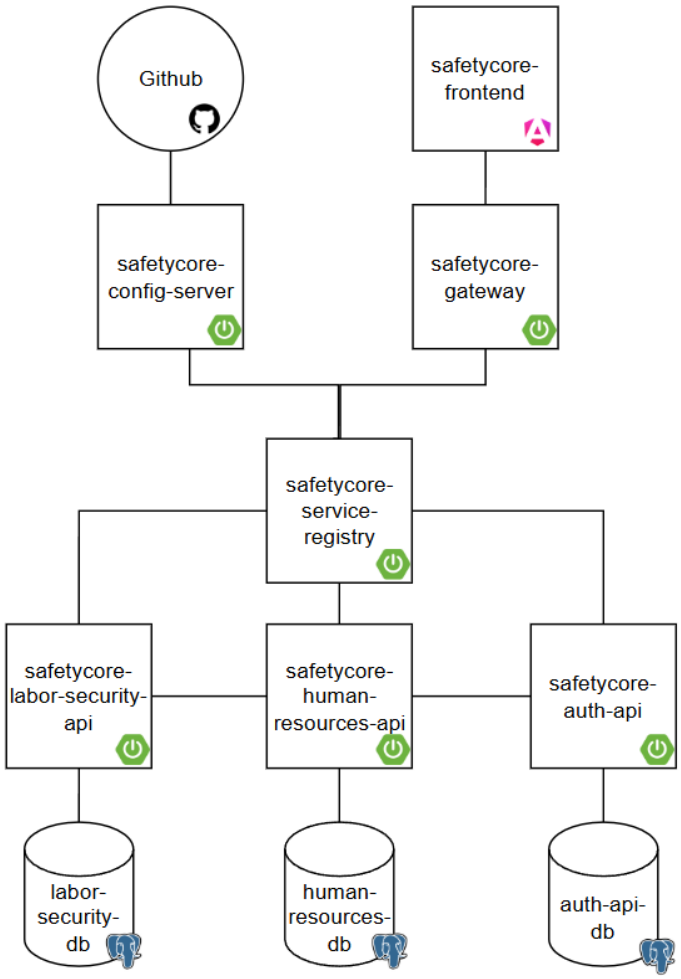
Para o desenvolvimento da aplicação, adotou-se a arquitetura de microsserviços, permitindo distribuir responsabilidades entre serviços distintos. Cada serviço gerencia suas próprias informações e possui *deploys* independentes, o que facilita a escalabilidade, aumenta a disponibilidade e reduz o acoplamento entre componentes. Essa abordagem simplifica o desenvolvimento por tornar as aplicações menores e menos complexas, embora exija maior especialização para manter as integrações e compreender a comunicação entre os serviços.

Foram implementados três serviços principais: o *safetycore-human-resources-api*, responsável pelos dados de funcionários e vínculos com setores, funções e riscos ocupacionais; o *safetycore-labor-security-api*, que gerencia empresas, setores, funções,

fontes geradoras e relacionamentos; e o *safetycore-auth-api*, encarregado da autenticação e gerenciamento de usuários.

Além dos serviços principais, como ilustra a Figura 9, há componentes de suporte essenciais para o funcionamento da arquitetura. O *safetycore-service-registry* implementa o *Service Registry Pattern*, permitindo registro e descoberta dinâmica de serviços. O *safetycore-gateway* aplica o *API Gateway Pattern*, funcionando como ponto único de entrada, realizando roteamento de requisições e controle de acesso. O *safetycore-config-server* centraliza arquivos de configuração obtidos do repositório GitHub, promovendo consistência na configuração dos serviços (Microservices.IO, 2025). Por fim, o *safetycore-frontend* concentra a interface de interação com o usuário.

Figura 9: Modelo dos serviços



Fonte: elaborado pelas autoras, 2025.

Por se tratar de uma arquitetura distribuída, cada microsserviço mantém seu próprio banco de dados (*Database per Service Pattern*), garantindo isolamento e autonomia entre os componentes. Entretanto, essa abordagem traz desafios relacionados à consistência e à sincronização de dados, conforme o Teorema CAP, que estabelece que um sistema distribuído pode garantir simultaneamente apenas duas das três

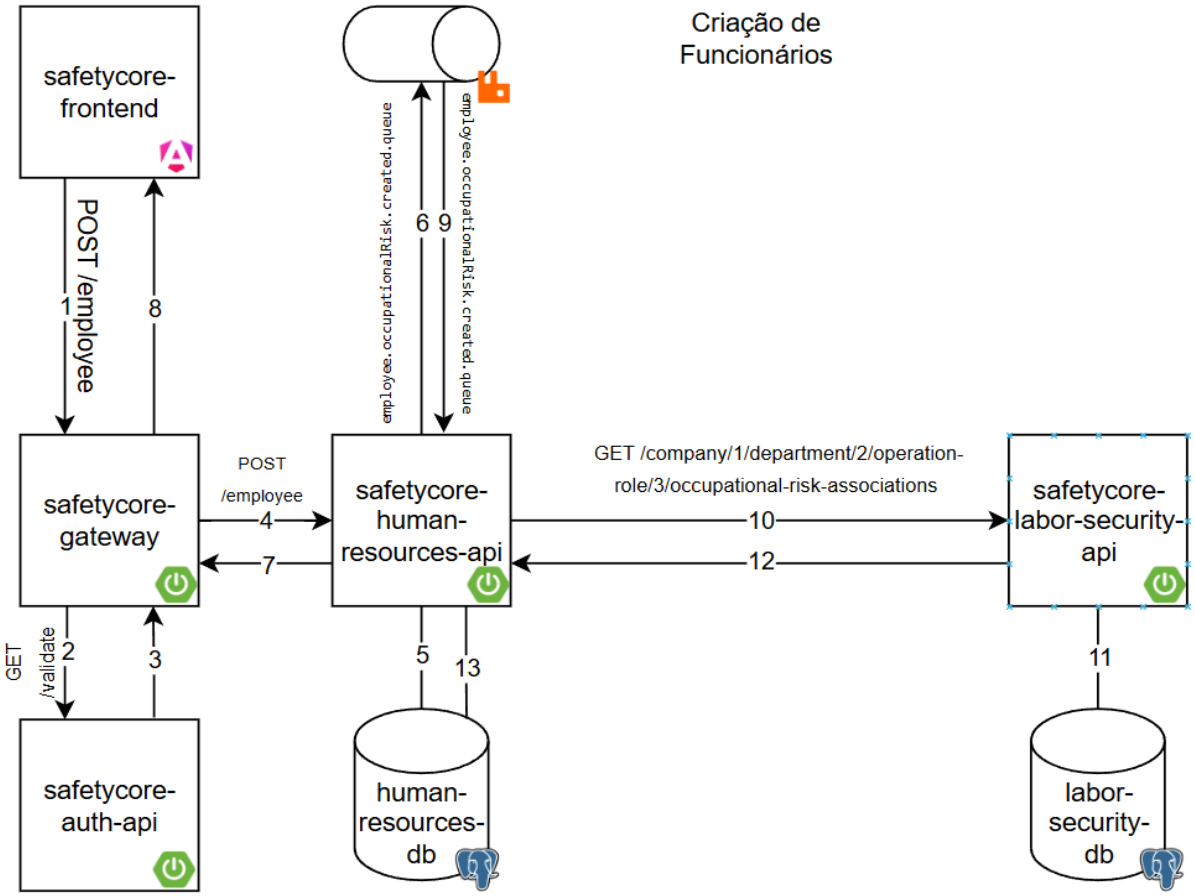
propriedades desejáveis: consistência (C), disponibilidade (A) e tolerância à partição (P) (IBM, 2025).

Como a tolerância a partições é inerente a esse tipo de arquitetura, tornou-se necessário optar entre consistência e disponibilidade. Neste trabalho, optou-se por priorizar a disponibilidade, assegurando que o sistema permanecesse acessível mesmo diante de falhas de comunicação entre serviços. Em decorrência dessa escolha, admite-se a consistência eventual, na qual pequenas divergências de dados podem ocorrer temporariamente até que a sincronização seja concluída.

Para mitigar essas limitações, adotou-se uma estratégia híbrida, que combina comunicação assíncrona baseada em eventos e replicação seletiva de dados essenciais. Essa abordagem permite que cada serviço mantenha localmente as informações necessárias às suas operações, equilibrando disponibilidade, desempenho e isolamento entre domínios.

Para exemplificar o funcionamento da comunicação entre os serviços, a Figura 10 detalha o processo de criação de um funcionário, desde a requisição inicial até a propagação dos dados entre os domínios envolvidos. O fluxo inicia-se quando o *safetycore-frontend* envia uma requisição HTTP REST ao *safetycore-gateway*, que se autentica através do *safetycore-auth-api*, identifica o serviço de destino através do *safetycore-service-registry* e encaminha a solicitação ao *safetycore-human-resources-api*. O *gateway* atua como ponto único de entrada do sistema, assegurando segurança, roteamento dinâmico e isolamento dos serviços internos.

Figura 10: Modelo do processo de criação de funcionários



Fonte: elaborado pelas autoras, 2025.

Em sequência ao processo ilustrado no parágrafo anterior, ao receber a solicitação, o *safetycore-human-resources-api* realiza a persistência dos dados do novo funcionário em seu banco de dados, aplicando o princípio de independência entre domínios de negócio. Após a gravação da entidade, o serviço publica um evento de domínio indicando a criação do novo funcionário por meio do *message broker*. O próprio serviço atua como consumidor desse evento, processando-o de forma assíncrona. Durante esse processamento, ele realiza uma consulta REST ao *safetycore-labor-security-api* para obter as informações detalhadas sobre os riscos ocupacionais associados à função, departamento e empresa do funcionário recém-criado.

As informações retornadas pelo *safetycore-labor-security-api* são então armazenadas localmente no banco de dados do *safetycore-human-resources-api*, garantindo autonomia e melhor desempenho em operações futuras relacionadas ao funcionário, mesmo em casos de indisponibilidade temporária do serviço de segurança do trabalho.

Esses dados obtidos tornam-se parte fundamental do processo de integração entre os domínios, uma vez que permitem ao serviço de Recursos Humanos manter informações consistentes e atualizadas sobre o funcionário e seus vínculos. Assim, a arquitetura combina comunicação assíncrona baseada em eventos com invocações síncronas via REST e descoberta dinâmica de serviços, adotando uma abordagem

híbrida que equilibra consistência eventual, resiliência e integração entre os domínios de Recursos Humanos e Segurança do Trabalho, assegurando escalabilidade e isolamento.

5 CONCLUSÃO

A adoção da arquitetura de microsserviços mostrou-se uma decisão acertada para o desenvolvimento da aplicação, permitindo alcançar de forma consistente os objetivos específicos propostos. Destaca-se, principalmente, a criação de uma solução funcional capaz de apoiar a gestão das atividades diárias dos profissionais de segurança do trabalho e de promover a integração eficaz entre os domínios de Recursos Humanos e Segurança do Trabalho. A independência operacional de cada serviço e a capacidade de gerenciar seus próprios dados contribuíram para a redução do acoplamento e para uma comunicação eficiente entre os componentes, garantindo maior escalabilidade, modularidade e isolamento entre os domínios.

Nesse contexto, a aplicação desenvolvida atendeu plenamente às expectativas estabelecidas, resultando em uma plataforma coerente com o processo de análise de riscos ocupacionais. O sistema demonstrou capacidade de representar de maneira fiel as necessidades práticas identificadas junto aos profissionais da área, oferecendo recursos que possibilitam identificar, monitorar e gerenciar riscos por setor e por colaborador, o que contribui significativamente para decisões mais assertivas no contexto da segurança do trabalho.

De modo geral, o trabalho evidenciou o potencial da arquitetura de microsserviços na construção de sistemas corporativos mais robustos e alinhados às demandas reais das organizações. Como continuidade deste trabalho, propõe-se o desenvolvimento de novas funcionalidades, como o módulo de controle e distribuição de Equipamentos de Proteção Individual (EPIs) e a automatização da inserção de funcionários por meio de arquivos XML. Tais aprimoramentos poderão ampliar o escopo do sistema, tornando-o ainda mais eficiente e consolidando-o como uma ferramenta estratégica para o aprimoramento das práticas de segurança e saúde ocupacional.

REFERÊNCIAS

BRASIL. Ministério do Trabalho e Previdência. Portaria nº 671, de 8 de novembro de 2021. **Diário Oficial da União**, seção 1, Brasília, DF, n. 212, p. 217, 11 nov. 2021. Disponível em: https://in.gov.br/en/web/dou/-/portaria-359094139.

FIELDING, Roy Thomas. **Architectural styles and the design of network-based software architectures**. 2000. Disponível em: https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.

GALINDO JUNIOR, Edemilton Alcides; ROCHA, Romeu Dias; MACIEL, Ronierison de Souza. **Desenvolvimento de api rest com spring boot**. 2021. Disponível em: https://www.publicacoes.unirios.edu.br/index.php/revistarios/article/view/102/102.

GUIMARÃES, Raphael Mendonça; MAURO, Maria Yvone Chaves; MUZI, Camila Drumond; MAURO, Carla Christina Chaves. Riscos ocupacionais em saúde. **Rev enferm UERJ**, v. 12, n. 3, p. 338-45, 2004. Disponível em: https://www.researchgate.net/profile/Raphael-Guimaraes/publication/342211512_Riscos_ocupacionais_em_saude/links/5ee8dd03299bf1faac59f87c/Riscos-ocupacionais-em-saude.pdf.

IBM. **O que é o teorema CAP?** Disponível em: https://www.ibm.com/br-pt/think/topics/cap-theorem.

MARTIN, Robert C. **Código limpo**: habilidades práticas do Agile software craftsmanship. São Paulo: Alta Books, 2009.

MASSE, Mark. **REST API design rulebook**. O'Reilly Media, Inc., 2012.

MENDES, Iasmin Santos. **Arquitetura monolítica vs microsserviços**: uma análise comparativa. 2021. Disponível em: https://bdm.unb.br/bitstream/10483/30715/1/2021_IasminSantosMendes_tcc.pdf.

OSHEROVE, Roy. **The art of unit testing**: with examples in C#. 2nd ed. Greenwich: Manning Publications, 2013.

PEIXOTO, Neverton Hofstadler. **Curso técnico em automação industrial**: segurança do trabalho. 3. ed. Santa Maria: Universidade Federal de Santa Maria: Colégio Técnico Industrial de Santa Maria, 2010. Disponível em: https://appcursosdegraca.s3.amazonaws.com/apostilas-br/profissionalizantes/seguranca-do-trabalho/seguranca-do-trabalho-por-etec.pdf.

RICHARDSON, Chris. **Microservices Patterns**. Disponível em: https://microservices.io.

VASCONCELOS, Alexandre Marcos Lins de; ROUILLER, Ana Cristina; MACHADO, Cristina Ângela Filipak. **Introdução à engenharia de software e à qualidade de software**. Lavras: Universidade Federal de Lavras, 2006. Disponível em: http://nti.facape.br/jocelio/es/apostilas/Mod.01.MPS_Engenharia&QualidadeSoftware_V.28.09.06.pdf.

VILLAÇA, Luis. Henrique; PIMENTA JUNIOR, Antônio Francisco; AZEVEDO, Leonardo Guerreiro. **Construindo aplicações distribuídas com microsserviços**. Tópicos em Sistemas de Informação: Minicursos XV Simpósio Brasileiro de Sistemas de Informação. SBC, 2018.