



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus “José Santilli Sobrinho”**

ANTONIO CARLOS THEODORO JUNIOR

**EPI FÁCIL: SOFTWARE PARA GESTÃO DE MATERIAIS DE
EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI)**

**Assis/SP
2024**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus “José Santilli Sobrinho”**

ANTONIO CARLOS THEODORO JUNIOR

**EPI FÁCIL: SOFTWARE PARA GESTÃO DE MATERIAIS DE
EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI)**

Projeto de pesquisa apresentado ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e à Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Antonio Carlos Theodoro Junior
Orientador(a): Me. Guilherme de Cleva Farto

**Assis/SP
2024**

FICHA CATALOGRÁFICA

Theodoro Junior, Antonio Carlos

T388e EPI fácil: software para gestão de materiais de equipamento de proteção individual (EPI) /Antonio Carlos Theodoro Junior. --Assis, 2024. -- 36p.

Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas) -- Fundação Educacional do Município de Assis (FEMA), Instituto Municipal de Ensino Superior de Assis (IMESA), 2024.

Orientador: Prof. Me. Guilherme de Cleva Farto.

1. Software. 2. Eficiência organizacional. I Farto, Guilherme de Cleva. II Título.

CDD: 005

EPI FÁCIL: SOFTWARE PARA GESTÃO DE MATERIAIS DE EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI)

ANTONIO CARLOS THEODORO JUNIOR

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Prof. Me. Guilherme de Cleve Farto

Examinador: _____
Prof. Esp. Domingos de Carvalho Villela Junior

Assis/SP

2024

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me sustenta, me protege e ilumina meus passos em cada momento desta jornada.

Quero agradecer de coração à minha esposa, que sempre esteve ao meu lado, me dando apoio e amor durante toda essa caminhada. Agradeço também aos meus pais e à minha irmã, por serem minha base em todos os momentos, e às minhas sobrinhas, que, com sua alegria, deixaram meus dias mais leves.

Aos meus amigos e colegas de faculdade, obrigado por estarem comigo nos momentos difíceis e por me ajudarem a resolver os problemas que surgiram durante a implementação deste projeto.

Por fim, agradeço ao meu orientador, que sempre me apoiou, deu ótimas orientações e esteve disponível para tirar minhas dúvidas ao longo do caminho.

*"A tecnologia move o mundo. Mas, o que realmente importa é como
escolhemos usá-la para transformar vidas."*

— Steve Jobs

RESUMO

Este trabalho apresenta o desenvolvimento do software EPI Fácil, criado para gerenciar a entrega de Equipamentos de Proteção Individual (EPI) em empresas de diferentes portes. O principal objetivo foi implementar uma solução tecnológica de baixo custo e fácil manutenção, capaz de automatizar o processo de gestão de EPIs, contribuindo para a redução de passivos trabalhistas, aumento da lucratividade e melhoria da satisfação dos colaboradores. O software permite a visualização detalhada dos EPIs entregues a cada colaborador, incluindo data e hora da entrega. Além de atender às exigências das Normas Regulamentadoras (NR 06 e NR 01), o sistema visa facilitar a administração eficiente de EPIs, desde a aquisição até o uso pelos colaboradores.

Palavras-chave: Gestão de EPIs, Redução de passivos trabalhistas, Lucro empresarial, Eficiência operacional.

ABSTRACT

This work presents the development of the EPI Fácil software, designed to manage the delivery of Personal Protective Equipment (PPE) in companies of various sizes. The main objective was to implement a low-cost, easy-to-maintain technological solution capable of automating the PPE management process, contributing to the reduction of labor liabilities, increased profitability, and improved employee satisfaction. The software allows for detailed visualization of the PPE delivered to each employee, including the date and time of delivery. In addition to meeting the requirements of Regulatory Standards (NR 06 and NR 01), the system aims to facilitate the efficient management of PPE, from acquisition to use by employees.

Keywords: PPE Management, Reduction of Labor Liabilities, Business Profitability, Operational Efficiency.

LISTA DE ILUSTRAÇÕES

Figura 1 - Mapa mental - Software EPI Fácil;	20
Figura 2 – Diagrama de classes;	21
Figura 3 – Diagrama de entidade e relacionamento;	22
Figura 4 - Diagrama de caso de uso geral - Software EPI Fácil;	22
Figura 5 – Diagrama de caso de uso – Manter colaborador;	23
Figura 6 – Diagrama der caso de uso – Entregar EPI por colaborador;	25
Figura 7 – Diagrama de atividades. Fonte: Aatoria própria;	27
Figura 8 - Tela de cadastro de um novo Colaborador;	28
Figura 9 - Tela de Registro de Entrega de EPI a um Colaborador;	29
Figura 10 - Função para criar um novo Colaborador;	30
Figura 11 - Função realizar a entrega de EPI ao Colaborador.	31

LISTA DE TABELAS

Tabela 01 – Narrativa manter colaborador;	23
Tabela 02 – Narrativa entrega de EPI por colaborador.	25

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CSS	Cascading Style Sheets
EPI	Equipamento de Proteção Individual
HTML	Linguagem de Marcação de Hipertexto
IDE	Integrated Development Environment
JPA	Java Persistence API
JVM	Java Virtual Machine
NR	Norma Regulamentadora
UML	Unified Modeling Language
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	13
1.1.1 OBJETIVO GERAL	13
1.1.2 OBJETIVOS ESPECÍFICOS	13
1.2 PÚBLICO ALVO	14
1.3 JUSTIFICATIVAS	14
1.4 ESTRUTURA DO TRABALHO	14
2 TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO	15
2.1 QUANT-UX	15
2.2 JAVA	16
2.2.1 MÁQUINA VIRTUAL JAVA (JVM)	16
2.3 SPRING BOOT	17
2.4 ANGULAR	18
2.5 BANCO DE DADOS MySQL	18
3 ANÁLISE E ESPECIFICAÇÃO DO SISTEMA	19
3.1 LISTA DE REQUISITOS	19
3.2 MAPA MENTAL	19
3.3 DIAGRAMA DE CLASSE	19
3.4 DIAGRAMA DE ENTIDADE RELACIONAMENTO	20
3.5 DIAGRAMA DE CASO DE USO (GERAL)	21
3.6 – DIAGRAMA E NARRATIVAS DE CASOS DE USO	22
UC01 – MANTER COLABORADOR	22
UC02 – ENTREGAR EPI POR COLABORADOR	24
3.7 DIAGRAMA DE ATIVIDADES	26
4 DESENVOLVIMENTO DO PROJETO	27
4.1 CADASTRO DE NOVO COLABORADOR	27
4.2 REGISTRO DE ENTREGA DE EPI	28
4.3 EXPLICAÇÃO DE CÓDIGOS	28
4.3.1 CRIAR UM NOVO COLABORADOR	28
4.3.2 REALIZAR A ENTREGA DE UM EPI AO COLABORADOR	30
5 CONCLUSÕES	32
5.1 TRABALHOS FUTUROS	32
REFERÊNCIAS	34

1 INTRODUÇÃO

Conforme o decreto de lei número 5.452 publicado em 1 de maio de 1943, que aprovou a Consolidação das Leis do Trabalho, em seu Capítulo V que dispõe sobre Segurança e Medicina do Trabalho, especificamente na seção IV Equipamento de Proteção Individual, diz que todas as empresas são obrigadas a fornecer, de forma gratuita, equipamento de proteção individual adequado ao risco e em perfeito estado de conservação e funcionamento, sempre que as medidas de ordem geral não protejam completamente seus colaboradores contra os riscos de acidentes e doenças ocupacionais (BRASIL, 1943).

Desta forma, a Norma Regulamentadora 06 - Equipamentos de Proteção Individual, elucida que, além do fornecimento deste equipamento de proteção individual (EPI) adequado ao risco e em perfeito estado de funcionamento e conservação, o empregador deve adquirir somente o aprovado pelo órgão nacional competente em matéria de Segurança e Saúde no Trabalho. Também deve registrar todo o fornecimento destes equipamentos a seus colaboradores, onde podem ser adotados meios de registro como livros, fichas ou sistemas eletrônicos e sistemas biométricos (BRASIL, 1978).

Com o advento da tecnologia e o crescimento de sua utilização no cotidiano das empresas, livros e fichas de registros manuais se tornaram formas obsoletas de manter o histórico de entrega de EPI dos colaboradores, haja vista que antes este procedimento poderia estar suscetível a despadronização de seu preenchimento e ter grande facilidade de extravio desta documentação.

Um sistema de gestão e controle de entrega de EPI padroniza toda a documentação gerada na plataforma e integra outras funções pertinentes ao controle dos EPIs da empresa, gerencia as entradas e saídas de materiais, controla o estoque e demanda de compras.

Neste contexto, o mercado disponibiliza diversos softwares para esta gestão, o software SENIOR em seu módulo de Segurança do Trabalho contempla funcionalidades como manter o cadastro de EPIs, indicar quais equipamentos distribuir conforme o risco, faz a manutenção do estoque, gera comprovantes de entrega dos EPIs e emite relatórios.

O DUAPI.EPI, um outro software de gestão de EPI disponível no mercado, possibilita fazer o cadastro de EPIs, realizar consultas e acompanhar as entregas de materiais. Para a entrega do EPI ele oferece o recurso de assinatura do colaborador por biometria e o acesso pode acontecer de qualquer tipo de dispositivo móvel em tempo real (DUAPI.EPI, 2023).

Os softwares pesquisados possuem em comum o cadastro e consulta de EPIs na base de dados, possibilitam realizar a entrega do equipamento aos colaboradores e emitir uma ficha de comprovação de entrega. Um ponto a ser destacado e que difere do software aqui proposto é que o EPI FÁCIL, além das funcionalidades que o SENIOR e o DUAPI.EPI tem em comum, este poderá manter e controlar todo o estoque de EPI da empresa, realizar a distribuição de EPI por setor ou grupo homogêneo de risco, realizar a conferência do Certificado de Aprovação de cada EPI direto na base de dados do Ministério do Trabalho e Emprego e na base de dados do site ConsultaCA.com, emitindo um e-mail de aviso aos usuários do setor de Segurança do Trabalho com a vigência do Certificado de Aprovação de cada equipamento.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O presente trabalho teve como objetivo implementar um software capaz de realizar a gestão de entregas de Equipamentos de Proteção Individual (EPI) de empresas de pequeno ou grande porte, com tecnologias de baixo custo e manutenção simplificada. A partir do uso do software proposto, foi possível resolver problemas com passivos trabalhistas, aumentar a lucratividade da empresa e melhorar a satisfação dos colaboradores nela registrados.

1.1.2 OBJETIVOS ESPECÍFICOS

Como objetivo específico, a presente proposta pretende implementar o seguinte requisito funcional para o software: visualização dos materiais listados nas entregas realizadas para cada colaborador com detalhes de data e hora de entrega.

1.2 PÚBLICO ALVO

O software EPI Fácil, que foi proposto para ser desenvolvido neste trabalho de conclusão de curso visa atender à necessidade de diversas empresas, de variados nichos e tamanhos que desejam automatizar e aprimorar no que diz respeito ao processo de gestão de Equipamentos de Proteção Individual.

Com isso, o software proposto poderá ser utilizado por diversos níveis de usuários, partindo desde os colaboradores comuns, gestores das áreas, Recursos Humanos e Segurança do Trabalho, possibilitando que seja informatizado todo o processo de gestão, solucionando problemas com passivos trabalhistas que são gerados a partir da má gestão destes.

1.3 JUSTIFICATIVAS

De acordo com Fagundes e Bento (2019), as empresas devem buscar reduzir os riscos nos ambientes de trabalho empregando a utilização de EPI e implementar um sistema de gestão que seja capaz auxiliar na redução de custos em tarefas de compras, armazenamento, reposição, disponibilidade e registro dos EPIs desde a entrada até a saída para os colaboradores. O sistema de gestão de entrega de EPI surge como um diferencial ao atendimento às exigências preconizadas na NR 06 e NR 01, em todos os tipos de empresas.

1.4 ESTRUTURA DO TRABALHO

Este trabalho de conclusão de curso foi desenvolvido com as seguintes partes:

- 1. Capítulo 1 – Introdução**
- 2. Capítulo 2 – Tecnologia e Ferramentas de Desenvolvimento**
- 3. Capítulo 3 – Análise e Especificação do Sistema**
- 4. Capítulo 4 – Desenvolvimento do Sistema**
- 5. Capítulo 5 – Conclusões**
- 6. Referências**

2 TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO

Este capítulo tem como objetivo apresentar as tecnologias empregadas no desenvolvimento do software EPI Fácil. Por se tratar de uma tecnologia de desenvolvimento de software orientada a objetos de alto nível a linguagem de programação *Java* fora a escolhida para o desenvolvimento do *back-end* desta aplicação, isto em conjunto com o ambiente integrado de desenvolvimento, traduzido do inglês *Integrated Development Environment* (IDE), será empregado o IntelliJ IDEA Community Edition, que é totalmente gratuito e permite a integração de componentes, acoplamento de bibliotecas, frameworks e inúmeras *Application Programming Interface* (API).

Assim, utilizar da tecnologia *Java* para o desenvolvimento do software, empregar o banco de dados MySQL, que se trata de um banco de dados relacional muito consumido por diversas grandes empresas em todo o mundo. Para a persistência de dados e manipulação será realizada por meio do framework *Java Persistence API* (JPA), que realiza o mapeamento do objeto-relacional entre as tabelas do banco de dados e as classes de entidades do sistema (LOCKOW; MELO, 2015).

Para o desenvolvimento do *front-end* será utilizado o framework Angular, ele permite a construção de interfaces de aplicações utilizando tecnologias web como HTML, CSS e JavaScript.

2.1 QUANT-UX

QUANT-UX é uma ferramenta de prototipagem de software capaz de criar a interface do usuário rapidamente. Nele é capaz de criar protótipos de telas e elementos da aplicação, inserir operadores lógicos, scripts, assim permitindo que contenha regra de negócio como uma aplicação real, desta forma, quando utilizado seu ambiente de testes, o usuário contribuirá diretamente com o desenvolvimento da aplicação fornecendo dados que são coletados durante suas interações. (QUANT-UX, [s.d.]).

2.2 JAVA

A plataforma de desenvolvimento *Java* foi criada por James Gosling em meados de 1990, seu lançamento ocorreu no ano de 1995 durante seu período de trabalho na Sun Microsystems e originalmente seu nome fora registrado como sendo Oak (SILVEIRA, SILVEIRA, et al., 2020). Originalmente James e sua equipe buscava o desenvolvimento de uma televisão interativa, porém, naquela época foi considerado avançado demais para a tecnologia de cabos digitais disponíveis, desta forma, a equipe de Gosling mudou seu foco para a criação de uma linguagem de programação, originando a conhecida e renomada linguagem de programação *Java*. (MICROSOFT AZURE, [s.d.]

A linguagem de programação *Java* é uma linguagem de alto nível, orientada a objetos e multiplataforma. A Máquina Virtual Java (JVM) garante que o código executado nela seja compatível com diferentes sistemas operacionais disponíveis no mercado como o Linux e o Windows. (SILVEIRA, SILVEIRA, et al., 2020).

No ano de 2004 a linguagem de programação *Java* contava com mais de 3 milhões de desenvolvedores, tornando a linguagem que se difundiu mais rápido na história da computação. Em 2006 a Sun Microsystems concede o código do compilador da Máquina Virtual Java (JVM) sob a anuência *GNU – General Public License*, em 2008 a Oracle adquiriu a Sun Microsystems e durante o processo de aquisição a Sun cria uma nova máquina virtual open source, a OpenJDK, que somente no ano de 2017 foi liberada por completo à comunidade. (KUSZKA, 2019).

2.2.1 MÁQUINA VIRTUAL JAVA (JVM)

Java Virtual Machine (JVM) é o responsável por tornar a linguagem de programação *Java* ser multiplataforma em variados sistemas operacionais. Nela, o código compilado sob extensão `.class` deve conter todas as tarefas que serão executadas pela Máquina Virtual Java durante a execução da aplicação, chamado de bytecode, seu nome tem relação direta com o tamanho de cada instrução deste arquivo que é de um byte cada. (SILVEIRA, SILVEIRA, et al., 2020).

Desta forma a JVM se torna um item imprescindível para a plataforma Java, pois ela é capaz de executar os códigos escritos em Java serem executados em quaisquer tipos de sistemas operacionais. Para a JVM qualquer o que mais importa é o bytecode, sem ele não seria possível a execução em multiplataformas, assim, qualquer linguagem de programação que quando compilada gere um bytecode apto de ser lido por ela, se tornará multiplataforma. (SILVEIRA, SILVEIRA, et al., 2020).

2.3 SPRING BOOT

No ano de 2002 o Spring Framework ocorreu através do livro do Rod Johnson – Expert One-To-One J2EE Design and Development, reunindo conceitos de *design patterns*, inversão de controle e dependências. O SPRING BOOT é um módulo deste framework, que surgiu para suprir uma demanda de configurações e facilitar a criação de novos projetos, permitindo que o programador se mantenha focado na regra de negócio. (CAVALCANTE, 2020).

Inicialmente as configurações do Spring eram realizadas através de arquivos *Extensible Markup Language (XML)*, somente após a sua versão 3.0 que houve a possibilidade de realizar estas configurações através do código Java, porém, ainda geravam bastante problemas, pois, se este código estivesse com erros, durante a compilação estes seriam apresentados fazendo com que o desenvolvedor consuma energia e perdendo o foco principal do projeto que é o desenvolvimento das regras de negócio. (AFONSO, 2017).

De acordo com Cavalcante (2020), as principais anotações do Spring Boot são:

- @Configuration = É uma anotação utilizada para informar ao Spring que determinada classe irá possuir métodos beans;
- @Bean = Indica que o método que está em uma classe anotada com @Configuration, deverá ser gerenciado pelo Spring, ou seja, ele poderá ser injetado em outras classes (injeção de dependências);
- @Autowired = Indica o ponto que ocorrerá a injeção da dependência, ela pode ser colocada sobre atributos ou sobre o construtor;
- @SpringBootApplication = Engloba 3 anotações, @Configuration, @EnableAutoConfiguration e @ComponentScan, é colocada sobre a classe

principal e ativa as configurações do projeto e inicia o servidor embutido para execução da aplicação.

2.4 ANGULAR

ANGULAR é uma plataforma de desenvolvimento de front-end open source de interfaces de aplicações web e Single-Page Applications que utiliza as tecnologias Typescript,, HTML e CSS. (TOTVS, 2020).

Segundo Afonso (2018), desenvolvimento com a plataforma Angular é baseado em componentes, eles que controlam as regras e comportamentos da aplicação, que são de suma importância, pois, são totalmente reaproveitáveis. Estes componentes são subdivididos em 3 elementos chave, sendo eles: um modelo HTML capaz de renderizar o componente, para personalização dele se usa um seletor CSS e para o gerenciamento de comportamentos e propriedades uma classe TypeScript.

2.5 BANCO DE DADOS MySQL

MySQL é um banco de dados relacional amplamente utilizado na rede corporativa pelo fato de poder armazenar e processar dados em tabelas separadas, onde estas estruturas são organizadas para otimizar a velocidade de acesso aos dados. Neste banco de dados é possível configurar regras de relacionamento entre diferentes campos de dados, sendo um para um, um para muitos ou muitos para muitos, exclusivo, obrigatório ou opcional, e ponteiros entre tabelas. Estas regras, quando bem definidas, atuam diretamente para que a aplicação não encontre dados inconsistentes, duplicados, órfãos, desatualizados ou ausentes. (MySQL, 2010).

3 ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

Para um melhor entendimento do software de gestão de equipamentos de proteção individual – EPIFÁCIL, serão apresentados neste capítulo as modelagens de diagramas, ferramentas de análise e especificação de requisitos funcionais.

3.1 LISTA DE REQUISITOS

Neste capítulo estão listados os principais requisitos do software proposto:

- Manter Colaborador;
- Manter EPI;
- Manter Cargos;
- Manter Riscos;
- Registrar a entrega de EPI por colaborador;
- Controlar entrega de EPI por colaborador;

3.2 MAPA MENTAL

Como forma de melhor elucidar o software proposto, foi elaborado um Mapa Mental. Este modelo de diagrama tem como objetivo facilitar o entendimento de uma ideia por meio de tópicos em ordem lógica e organizada (SOUZA, 2004).



Figura 1 - MAPA MENTAL - Software EPI Fácil. Fonte: Autoria própria.

3.3 DIAGRAMA DE CLASSE

Diagrama de Classes é um dos diagramas fundamentais para o processo de modelagem de objetos e modelagem da estrutura estática do software. Com ele é possível modelar os

objetos que compõem o sistema, visualizar os relacionamentos entre objetos e os serviços que eles fornecem. (IBM, 2021).

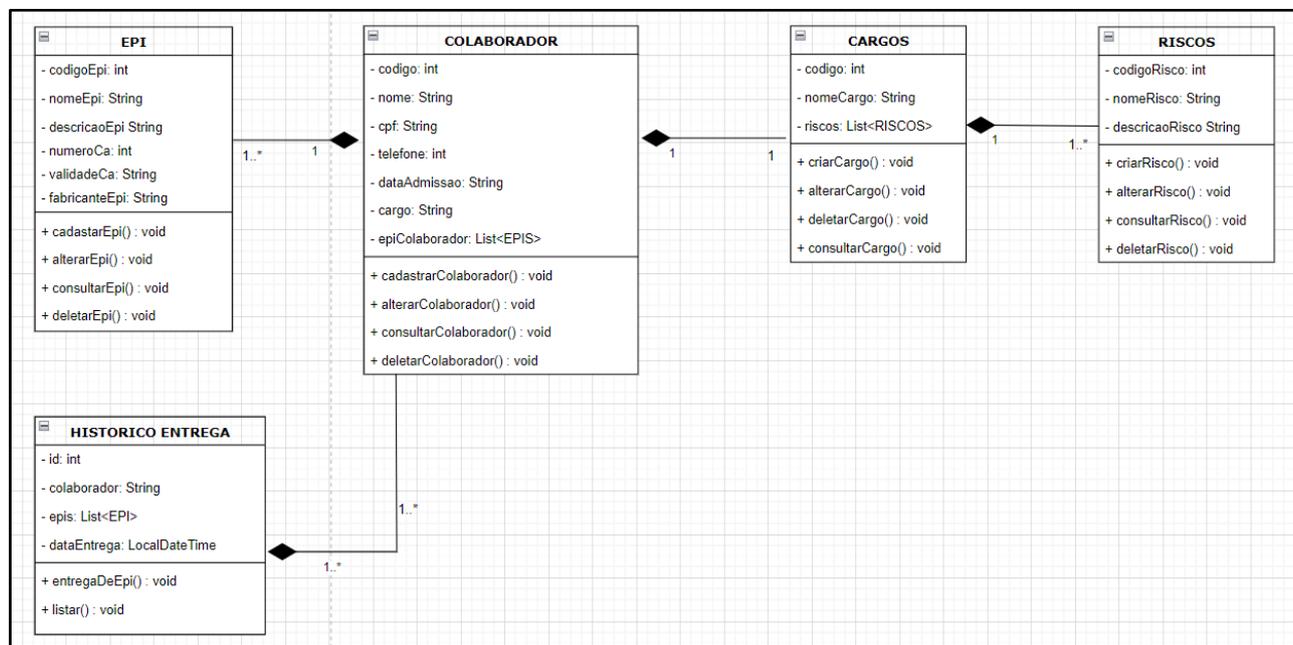


Figura 2 – Diagrama de classes. Fonte: Autoria própria.

3.4 DIAGRAMA DE ENTIDADE RELACIONAMENTO

O Diagrama de Entidade e Relacionamento ilustra como as entidades se relacionam entre si dentro do sistema, também são utilizados no projeto ou depuração do banco de dados relacionais. Neste diagrama os atributos e os relacionamentos entre as entidades são descritos conforme o envolvimento no domínio da regra de negócio, isto quando a base de dados é relacional. (LUCIDCHART, [s.d.]) (DEVMEDIA, 2014).

3.6 – DIAGRAMA E NARRATIVAS DE CASOS DE USO

Os diagramas de caso de uso são ferramentas de análise de software que auxiliam a capturar os requisitos deste sistema. Nele são descritos as funcionalidades e o escopo do software, são identificadas as interações entre os atores e o próprio sistema, identificando e descrevendo as ações dos atores e as reações do sistema. (IBM, 2021).

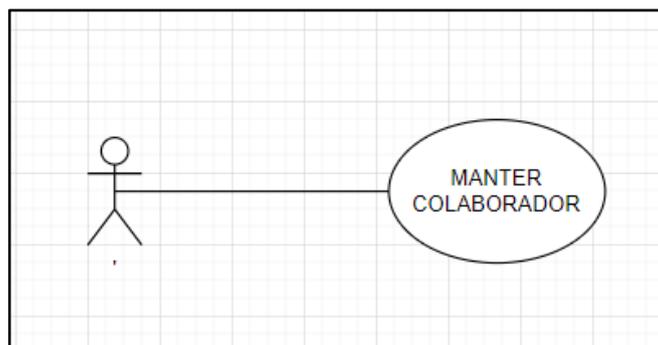


Figura 5 – Diagrama de caso de uso – Manter colaborador. Fonte: Autoria própria.

UC01 – MANTER COLABORADOR

1- Finalidade/ Objetivo	PERMITIR QUE O ATOR FAÇA A INCLUSÃO, ALTERAÇÃO, CONSULTA E EXCLUSÃO DE UM COLABORADOR NO SISTEMA;
2- Atores	DP;
3- Pré-condições	O ATOR PRECISA ESTAR LOGADO NO SISTEMA.
4- Evento Inicial	O ATOR COMEÇA O CASO DE USO CLICANDO NA OPÇÃO CADASTRAR COLABORADORES
5- Fluxo Principal	<ul style="list-style-type: none"> a) O SISTEMA APRESENTA NA TELA UM CAMPO PARA BUSCA POR NOME DE COLABORADOR; b) O ATOR INFORMA O NOME; [A1] c) O SISTEMA APRESENTA TODOS OS DADOS DO COLABORADOR SELECIONADO [E1][A2][A3] d) O ATOR CLICA NO BOTÃO VOLTAR; e) CASO DE USO É ENCERRADO.
6- Fluxo alternativo	<p>A1 – INCLUIR NOVO COLABORADOR</p> <ul style="list-style-type: none"> a) O ATOR CLICA NA OPÇÃO “INCLUIR NOVO COLABORADOR”; b) O SISTEMA APRESENTA OS CAMPOS PARA DIGITAÇÃO DOS DADOS DO COLABORADOR[T1];

	<p>c) O ATOR INFORMA OS DADOS E CLICA NO BOTÃO SALVAR;</p> <p>d) OS DADOS SÃO SALVOS;</p> <p>e) VOLTA AO PASSO A DO FLUXO PRINCIPAL;</p> <p>A2 – ALTERAR COLABORADOR</p> <p>a) O ATOR CLICA NA OPÇÃO “ALTERAR COLABORADOR”;</p> <p>b) O ATOR BUSCA O COLABORADOR POR NOME;</p> <p>c) O SISTEMA APRESENTA OS DADOS JÁ EXISTENTES DO COLABORADOR E OS CAMPOS PARA DIGITAÇÃO. [T1];</p> <p>d) O ATOR ALTERA OS DADOS DO COLABORADOR DESEJADO E CLICA NO BOTÃO “SALVAR”;</p> <p>e) OS DADOS SÃO SALVOS;</p> <p>f) O CASO DE USO É ENCERRADO.;</p> <p>A3 – EXCLUIR COLABORADOR</p> <p>a) O ATOR CLICA NA OPÇÃO DE “EXCLUIR COLABORADOR”</p> <p>b) O ATOR CLICA NO BOTÃO DE “EXCLUIR”</p> <p>c) O SISTEMA IMPRIME UM AVISO PARA QUE O ATOR TENHA CERTEZA QUE QUER EXCLUIR O COLABORADOR</p> <p>d) O COLABORADOR É EXCLUÍDO DO SISTEMA</p> <p>e) O CASO DE USO É ENCERRADO.</p>
7- Fluxo de Exceção	<p>E1 – COLABORADOR NÃO ENCONTRADO</p> <p>a) O CLIENTE NÃO EXISTE E O SISTEMA EMITE A MENSAGEM NA TELA “COLABORADOR NÃO ENCONTRADO”;</p> <p>b) LIMPA O CAMPO DE BUSCA E VOLTA AO PASSO “C” DO FLUXO PRINCIPAL;</p>
8- Casos de Teste	<p>T1 – TESTAR CAMPOS</p> <p>TESTAR OS CAMPOS DATAS;</p> <p>TESTAR O CAMPO CPF, VERIFICAR SE ESTÁ CORRETO;</p>

Tabela 1 – Narrativa Manter Colaborador. Fonte: Autoria própria.

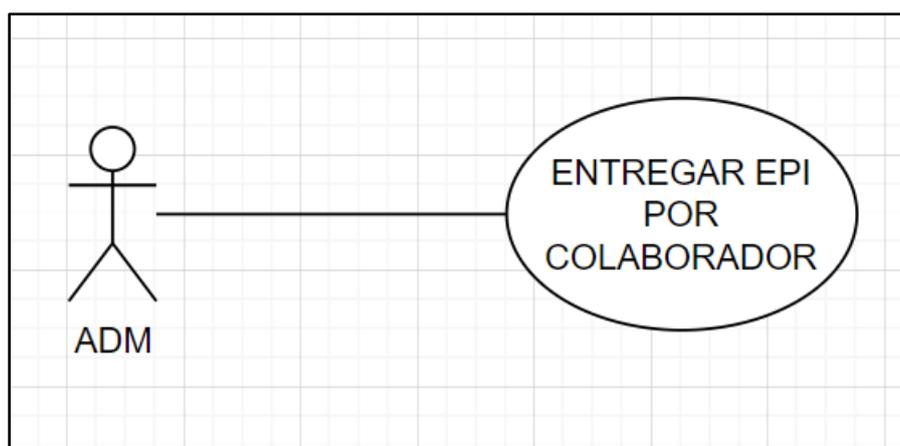


Figura 6 – Diagrama de caso de uso – Entregar EPI por colaborador. Fonte: Autoria própria.

UC02 – ENTREGAR EPI POR COLABORADOR

1 -Finalidade/ Objetivo	PERMITIR QUE O ATOR ENTREGUE O EPI AO COLABORADOR VIA SISTEMA;
2- Atores	ADM
3- Pré-condições	O ATOR PRECISA ESTAR LOGADO NO SISTEMA.
4- Evento Inicial	O ATOR COMEÇA O CASO DE USO CLICANDO NA OPÇÃO ENTREGAR EPI
5- Fluxo Principal	<ul style="list-style-type: none"> a) O SISTEMA APRESENTA NA TELA TODOS OS EPIS QUE ESTÃO PENDENTES DE ENTREGA; b) O ATOR CLICA NA OPÇÃO ENTREGAR EPI; [A1] c) O ATOR CLICA NA OPÇÃO EMITIR COMPROVANTE DE ENTREGA DE EPI [A2] d) O ATOR CLICA NO BOTÃO VOLTAR; e) CASO DE USO É ENCERRADO.
6- Fluxo alternativo	<p>A1 – ENTREGAR EPI</p> <ul style="list-style-type: none"> f) O ATOR CLICA NA OPÇÃO “ENTREGAR EPI”; g) O SISTEMA APRESENTA OS DADOS DO COLABORADOR E DO EPI SOLICITADO; h) O ATOR CLICA NO BOTÃO ENTREGAR; i) VOLTA AO PASSO A DO FLUXO PRINCIPAL; <p>A2 – Emitir comprovante de ENTREGA</p>

	<ul style="list-style-type: none"> j) O ATOR CLICA NA OPÇÃO “EMITIR COMPROVANTE DE ENTREGA DE EPI”; k) O ATOR BUSCA O COLABORADOR PELO NOME[E1]; l) O SISTEMA APRESENTA OS DADOS DO COMPROVANTE DE ENTREGA DE EPI; m) O ATOR EMITE O COMPROVANTE; n) O CASO DE USO É ENCERRADO;
7- Fluxo de Exceção	<p>E1 – SOLICITAÇÃO NÃO REALIZADA</p> <ul style="list-style-type: none"> o) O COLABORADOR NÃO EFETUOU O SOLICITAÇÃO DE EPI E É EXIBIDO UMA MENSAGEM; p) LIMPA O CAMPO DE BUSCA E VOLTA AO PASSO “B” DO FLUXO [A2];
8- Casos de Teste	

Tabela 2 – Narrativa Entregar EPI por Colaborador. Fonte: Autoria própria.

3.7 DIAGRAMA DE ATIVIDADES

Diagrama de Atividades tem um papel fundamental para a visualização de um sistema, ele é capaz de descrever a sequência de ações de um processo, bem semelhantes a um fluxograma comum, porém o Diagrama de atividades é capaz de mostrar fluxos paralelos ou simultâneos e fluxos alternativos. (IBM, 2021).

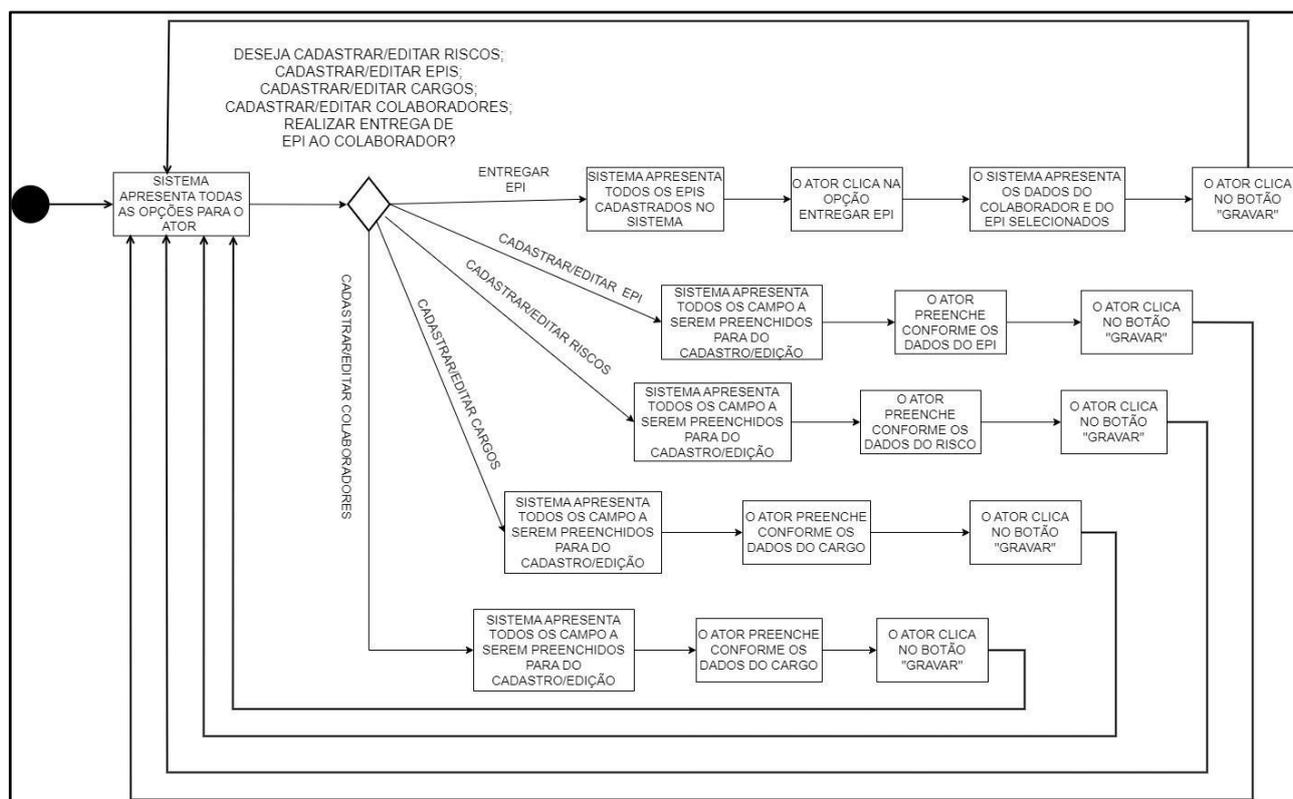


Figura 7 – Diagrama de atividades. Fonte: Autoria própria.

4 DESENVOLVIMENTO DO PROJETO

Neste capítulo fica detalhado o desenvolvimento do sistema EPI FÁCIL, onde o foco principal é o fluxo de entrega de EPI para um colaborador. Esta funcionalidade é de suma importância no sistema pois assim serão criados os registros de entregas com data e horário.

4.1 CADASTRO DE NOVO COLABORADOR

Após realizar a carga dos Riscos, Cargos e EPIs no sistema, é necessário realizar o cadastramento de um novo colaborador na base de dados. Nesta funcionalidade é possível criar um registro de um novo colaborador apontando um Cargo a ele, haja vista, que para este Cargo existem Riscos pertinentes no desempenho da função

The screenshot displays the 'COLABORADORES' registration page in the EPI FÁCIL system. The page title is 'COLABORADORES' and the section is 'CADASTRO'. The form contains the following fields and values:

Field	Value
NOME DO COLABORADOR: *	ANTONIO CARLOS THEODORO
CPF DO COLABORADOR: *	01537495801
TELEFONE: *	(18)997600641
ADMISSÃO: *	18/09/2012
CARGO: *	2 - REPOSITOR DE MERCADORIAS 3 - OPERADOR DE CAIXA 4 - AUXILIAR DE LIMPEZA 5 - MOTORISTA DE VEICULOS LEVES 6 - PADEIRO

Additional details from the screenshot include a sidebar menu with options like HOME, RISCOS, EPIS, CARGOS, COLABORADORES, TRANSAÇÕES, and ENTREGAS. The user is logged in as 'Start Bootstrap'. A footer link for 'Privacy Policy - Terms & Conditions' is visible at the bottom right.

Figura 8 – Tela de cadastro de um novo colaborador. Fonte: Autoria própria.

4.2 REGISTRO DE ENTREGA DE EPI

Após realizar o cadastramento do colaborador no Sistema EPI FÁCIL, resta a etapa de registrar a entrega do EPI a ele. Nesta tela é possível selecionar o colaborador que deseja realizar a entrega bem como os EPIS que serão registrados no ato da entrega e por fim clicar no botão “GRAVAR” para salvar as informações no banco de dados.



The screenshot displays the 'ENTREGAS DE EPI' (EPI Deliveries) page in the 'EPI FÁCIL' system. The page has a dark sidebar on the left with navigation options: SISTEMA, HOME, CADASTROS, RISCOS, EPIS, CARGOS, COLABORADORES, TRANSAÇÕES, and ENTREGAS. The main content area is titled 'ENTREGAS DE EPI' and contains a 'CADASTRO' section. It features a dropdown menu for 'SELECIONE O COLABORADOR: *' with the selected option '11 - ANTONIO CARLOS THEODORO JUNIOR - MOTORISTA DE VEICUL'. Below this is a list of EPIs under 'SELECIONE OS EPIS: *', including '16 - BOTINA DE SEGURANÇA', '17 - ÓCULO SUMMER', '18 - PROTETOR FACIAL', and '19 - LUVA TATIL'. At the bottom of the form are two buttons: a green 'GRAVAR' button and a red 'VOLTAR' button. The footer of the page includes 'Copyright © EPI FÁCIL 2024' and a link for 'Privacy Policy - Terms & Conditions'. The browser address bar shows 'localhost:4200/bsi/entrega-lista'.

Figura 9 – Tela de Registro de Entrega de EPI a um colaborador. Fonte: Autoria própria.

4.3 EXPLICAÇÃO DE CÓDIGOS

4.3.1 CRIAR UM NOVO COLABORADOR

A funcionalidade em destaque “salvar()” é usada para criar e salvar um novo colaborador no sistema. Ela recebe os dados do colaborador via requisição “POST”, verifica se o cargo fornecido existe, cria um novo objeto “ColaboradorModel” com os dados recebidos, e finalmente salva esse objeto no banco de dados, garantindo que a operação seja feita de forma transacional.

```

@RestController
@RequestMapping("/colaboradores")
@CrossOrigin("http://localhost:4200")
public class ColaboradorController {

    @Autowired
    private ColaboradorRepository colaboradorRepository;
    @Autowired
    private CargosRepository cargosRepository;

    public ColaboradorController(ColaboradorRepository colaboradorRepository,
                                CargosRepository cargosRepository){
        this.colaboradorRepository = colaboradorRepository;
        this.cargosRepository = cargosRepository;
    }

    @PostMapping
    @Transactional
    public ColaboradorModel salvar(@RequestBody DadosColaboradoresDTO dadosColaboradoresDTO){
        CargosModel cargo = cargosRepository.findById(dadosColaboradoresDTO.getCargo().getCodigoCargo())
            .orElseThrow(() -> new EntityNotFoundException("Cargo não encontrado"));

        ColaboradorModel colaborador = new ColaboradorModel();
        colaborador.setNomeColaborador(dadosColaboradoresDTO.getNomeColaborador());
        colaborador.setCpf(dadosColaboradoresDTO.getCpf());
        colaborador.setTelefone(dadosColaboradoresDTO.getTelefone());
        colaborador.setDataAdmissao(dadosColaboradoresDTO.getDataAdmissao());
        colaborador.setCargo(cargo);
        colaborador.setAtivo(true);
        return colaboradorRepository.save(colaborador);
    }
}

```

Figura 10 – Função para criar um novo colaborador. Fonte: Autoria própria.

Este código inicia sua operação realizando uma busca no banco de dados do cargo associado ao colaborador utilizando o código do cargo que foi passado. Após isso a funcionalidade *“findById()”* da interface Repository retorna um *“Optional<CargosModel>”*, que é então processado com o método *“orElseThrow”*. Se o cargo não for encontrado na base de dados, uma exceção será lançada através do *‘EntityNotFoundException’*, indicando que o cargo não existe.

O passo seguinte do código é a criação de um novo objeto do tipo *“ColaboradorModel”*. Os atributos deste objeto são definidos através dos valores recebidos no *“DadosColaboradoresDTO”*, menos o atributo cargo, este é definido através do objeto Cargo que fora previamente buscado.

E por fim, é realizado o salvamento dos dados do colaborador no banco de dados com a utilização do método “save” da interface Repository, este método então retorna o objeto “ColaboradorModel” criado juntamente com seu respectivo código gerado.

4.3.2 REALIZAR A ENTREGA DE UM EPI AO COLABORADOR

A funcionalidade “entregaDeEpi” gerencia o processo de registro de entrega de EPIs a um colaborador. Ela cria um registro no histórico de entregas, atualiza a lista de EPIs do colaborador, e salva essas informações no banco de dados. Em caso de sucesso, retorna a confirmação da operação.

```
@PostMapping
@Transactional
public ResponseEntity<HistoricoEntregaModel> entregaDeEpi(@RequestBody DadosListHistEntregaDTO dadosListHistEntregaDTO) {
    HistoricoEntregaModel historicoEntregaModel = new HistoricoEntregaModel(dadosListHistEntregaDTO);
    historicoEntregaRepository.save(historicoEntregaModel);
    Optional<ColaboradorModel> colaboradorOptional = colaboradorRepository.findById(dadosListHistEntregaDTO.getColaboradorId());
    if (colaboradorOptional.isPresent()) {
        ColaboradorModel colaborador = colaboradorOptional.get();
        if (colaborador.getEpis() == null) {
            colaborador.setListaEpi(new ArrayList<>());
        }
        for (EpiModel epi : dadosListHistEntregaDTO.getEpis()) {
            colaborador.getEpi().add(epi);
        }
        colaboradorRepository.save(colaborador);
    } else {
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body(null);
    }

    return ResponseEntity.ok(historicoEntregaModel);
}
```

Figura 11 – Função realizar a entrega de EPI ao colaborador. Fonte: Autoria própria.

O código inicia sua operação criando um novo objeto “HistoricoDeEntrega” a partir dos dados recebidos em “dadosListHistEntregaDTO”, após a criação deste objeto que representa uma entrega de EPI a interface Repository salva o registro no banco de dados. Logo após é realizada a busca do colaborador no banco de dados através da interface Repository pelo método “findById()” encapsula o colaborador encontrado, seguindo com os passos do código, é feita a validação se a lista de EPIs do colaborador encontrado tem valor nulo, caso positivo para nulo uma nova lista é criada e atribuída a este colaborador.

Em seguida para cada EPI passado como parâmetro em *“dadosListHistEntregaDTO”*, o código realiza a adição deste item à lista de EPIs do colaborador, e por fim o colaborador é salvo no banco de dados com a lista de EPIS atualizada.

5 CONCLUSÕES

O presente trabalho teve como seu objetivo geral criar um software de gestão de entregas de EPI voltado para empresas de diferentes portes. O desenvolvimento da solução se deu através de tecnologias de baixo custo e manutenção simplificada que se mostrou eficaz na mitigação de passivos trabalhistas, contribuindo para o aumento da lucratividade das empresas e a melhoria da satisfação dos colaboradores nela registrados.

Além disso, a funcionalidade do objetivo específico foi plenamente alcançada, onde é possível visualizar todos os materiais entregues a um colaborador com detalhes precisos de data e hora. Essa funcionalidade agregou valor ao processo de gestão de entrega de EPIs como também promoveu uma maior transparência e controle no acompanhamento das entregas, assegurando que todas as informações relevantes estejam acessíveis de maneira clara e organizada.

Em resumo, o software desenvolvido atendeu aos requisitos estabelecidos, demonstrando a viabilidade técnica e sua eficácia prática no ambiente corporativo. A implementação da funcionalidade de rastreamento de entregas de EPIs com registro detalhado de data e horários, facilita a auditoria e o monitoramento das entregas. Além disso, ao reduzir os passivos trabalhistas e melhorar a satisfação dos colaboradores, o software mostrou ser uma ferramenta estratégica para as empresas que buscam otimizar seus processos operacionais e sua lucratividade.

5.1 TRABALHOS FUTUROS

Apesar da implementação bem sucedida do software, foram identificados aprimoramentos com funcionalidades que o tornarão ainda mais robusto como sendo:

- Controle de Estoque por EPI - onde será possível realizar, de maneira detalhada, o controle de entradas e saídas dos materiais, garantindo que a empresa tenha uma visão clara das quantidades disponíveis, alertas de reposição e gerenciamento eficaz do estoque;
- Relatórios de Entrega de EPI - capacidade de gerar relatórios para cada EPI, onde será possível detalhar as entregas realizadas, as quantidades distribuídas bem como os colaboradores que receberam cada um dos itens;

- Relatório de Entrega por Colaborador - criação de relatório personalizados para cada colaborador, permitindo que a empresa acompanhe de forma precisa o histórico de entrega de EPIs, facilitando auditorias e o cumprimento das normas de Segurança do Trabalho;
- Assinatura Digital da Ficha de Entrega de EPI - este requisito visa aumentar a segurança e a conformidade legal da entrega dos EPIs aos colaboradores, onde, no ato da entrega, será implementada a possibilidade de assinatura digital da ficha de entrega de EPIs, que proporcionará um registro seguro e em conformidade com as normas trabalhistas.

REFERÊNCIAS

- AFONSO, A. O que é Angular? **Algaworks**, 2018. Disponível em: <<https://blog.algaworks.com/o-que-e-angular/>>. Acesso em: 25 Mar. 2024.
- AFONSO, A. **Produtividade no Desenvolvimento de Aplicações Web com Spring Boot**. 3ª. ed. [S.l.]: Algaworks, 2017.
- ALURA, **SITE DA ALURA, 2023**. Java: o que é, linguagem e um Guia para iniciar na tecnologia. Disponível em: <https://www.alura.com.br/artigos/java>. Acesso em: 10 mar. 2024.
- Angular, **SITE DO ANGULAR [s.d.]**. What is Angular?. Disponível em: <https://angular.io/guide/what-is-angular>. Acesso em: 10 mar. 2024.
- AZURE, **SITE DA AZURE, [s.d.]**. O que é o Java? Um guia para iniciantes em Java | Microsoft Azure. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-java-programming-language#:~:text=tudo%20%7C%20Recolher%20tudo-,O%20que%20%C3%A9%20o%20Java%3F,dos%20computadores%20e%20dispositivos%20m%C3%B3veis> . Acesso em: 10 mar. 2024.
- AZURE, **SITE DA AZURE, [s.d.]**. O que é o Java Spring Boot? Uma introdução ao Spring Boot, a popular ferramenta baseada em Java para desenvolver aplicativos Web e microsserviços. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-java-spring-boot>. Acesso em: 10 mar. 2024.
- BOOCH, Grady; JACOBSON, Ivar; RUMBAUCH, James. **UML – Guia do Usuário**. 2ª Edição. Tradução Fábio Freitas da Silva e Cristiana de Amorim Machado. Rio de Janeiro: Elsevier, 2005.
- BRASIL. Ministério do Trabalho. Gabinete do Ministro. Portaria n.º 3214, de 8 de junho de 1978. Norma Regulamentadora n.º 6: Equipamento de proteção individual – EPI, **Diário Oficial da União**, Brasília, DF, 8 jul. 1978a. Acesso em 10 nov. 2023.
- BRASIL. Presidência da República. Casa Civil. Subchefia para Assuntos Jurídicos. **Decreto-Lei N.º 5.452. Aprova a Consolidação das Leis do Trabalho**. Brasília: DF; 1943.
- CAVALCANTE, P. H. A. Spring Framework: o que é, seus módulos e exemplos! **Geek Hunter**, 2020. Disponível em: <<https://blog.geekhunter.com.br/spring-framework/>>. Acesso em: 25 Mar. 2024.
- DEVMEDIA. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). **DEVMEDIA**, 2014. Disponível em: <<https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>. Acesso em: 25 Mar. 2024.
- DUAPI.EPI. **SITE DO DUAPI.EPI**. Funcionalidades. Disponível em: <https://www.duapiepi.com.br/funcionalidades/>. Acesso em: 20 nov. 2023.
- FAGUNDES, G.BENTO, A. R.. **A IMPORTÂNCIA DA GESTÃO DE EPI PARA PREVENÇÃO DE ACIDENTES E REDUÇÃO DE CUSTOS NO SETOR AUTOMOTIVO**.

In: 74 Congresso Anual da ABM, 2019, São Paulo. 74 Congresso Anual da ABM. São Paulo: ABM, 2019. v. 74. p. 1877-1885.

IBM, **SITE DA IBM, 2021**. Diagramas de Atividades. Disponível em: <https://www.ibm.com/docs/pt-br/rational-soft-arch/9.7.0?topic=diagrams-activity>. Acesso em: 10 mar. 2024

IBM, **SITE DA IBM, 2021**. Diagramas de Caso de Uso. Disponível em: <https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case>. Acesso em: 10 mar. 2024.

IBM, **SITE DA IBM, 2021**. Diagramas de Classes. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em: 10 mar. 2024.

KUSZKA, B. A Evolução do Java até a Era da Nuvem. **Canaltech**, 2019. Disponível em: <https://canaltech.com.br/computacao-na-nuvem/a-evolucao-do-java-ate-a-era-da-nuvem/>. Acesso em 26 mar. 2024.

LOCKOW, Décio; MELO, Alexandre. **Programação Java para Web – 2ª Edição**. São Paulo: Novatec, 2015.

LUCIDCHART, **SITE DO LUCIDCHART, [s.d.]**. O que é um diagrama entidade relacionamento?. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento>. Acesso em 10 mar. 2024.

MENDES, L. **O QUE É, E PARA QUE SERVE O CRONOGRAMA**. 2015. Disponível em: <https://docplayer.com.br/5464728-O-que-e-e-para-que-serve-o-cronograma.html>. Acesso em 18 nov. 2023.

MySQL, **SITE DO MySQL, [s.d.]**. 1.2.1 What is MySQL? Disponível em: <https://dev.mysql.com/doc/refman/8.3/en/what-is-mysql.html>. Acesso em: 10 mar. 2024.

QUANT-UX, **SITE DO QUANT-UX, [s.d.]**. Prototype, Test and Learn. Disponível em: <https://quant-ux.com/>. Acesso em: 10 mar. 2024.

SILVEIRA, P. et al. **Introdução à Arquitetura e Design de Software - Uma visão sobre a plataforma Java**. São Paulo: Elsevier, 2020.

SOUSA, Lucilene. **Redesign da Informação no Processamento de Imagem**. Tese (Doutorado) – Universidade Federal de Santa Catarina, Florianópolis, 2004.

SPRING, **SITE DO SPRING BOOT, [s.d.]**. Spring Boot Reference Documentation. Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#using>. Acesso em: 10 mar. 2024.

TOTVS. Por que Angular é um framework tão poderoso? **TOTVS**, 2020. Disponível em: <https://www.totvs.com/blog/developers/angular/>. Acesso em: 25 Mar. 2024.