

## COORDENAÇÃO DE SISTEMAS DE INFORMAÇÃO

Sistema web para implantação do programa de gerenciamento de riscos ocupacionais

**YULLO COSTA BOMFIM** 

Vitória da Conquista-BA Junho / 2023

#### YULLO COSTA BOMFIM

# Sistema web para implantação do programa de gerenciamento de riscos ocupacionais

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, campus Vitória da Conquista, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof.Dr. Djan Almeida Santos

Vitória da Conquista-BA Junho / 2023

# Sistema web para implantação do programa de gerenciamento de riscos ocupacionais

### **YULLO COSTA BOMFIM**

A presente Monografia, apresentada em sessão realizada em 30 de Junho de 2023, foi avaliada como adequada para a obtenção do Grau de Bacharel em Sistemas de Informação, julgada **aprovada** em sua forma final pela Coordenação do Curso de Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, *campus* Vitória da Conquista.

Vitória da Conquista/BA, 30 de Junho de 2023.

Prof. Me. Alexandro dos Santos Silva (Coordenador do Curso - IFBA campus Vitória da Conquista)

#### **BANCA EXAMINADORA:**

Prof. DSc. Djan Almeida Santos (Orientador) IFBA campus Vitória da Conquista

Prof. Me. Amanda Ferraz de Oliveira Passos IFBA campus Vitória da Conquista

Prof. Me. Alexandro dos Santos Silva IFBA campus Vitória da Conquista

### **RESUMO**

O uso de sistemas de gestão especializados pode ser uma ferramenta útil para auxiliar as empresas a cumprir com suas obrigações legais e promover uma cultura de segurança. Através dele, é possível que empresas monitorem informações sobre ações preventivas, exposição aos agentes nocivos em diferentes funções e locais de trabalho, feito por meio da coleta e análise de dados do inventário de riscos e execução das medidas propostas no plano de ação. Essas informações são essenciais para a tomada de decisões relacionadas à segurança do trabalho. Este trabalho tem como objetivo desenvolver um sistema web que possa apoiar as empresas na gestão do programa de gerenciamento de riscos ocupacionais. O sistema proposto pretende monitorar o inventário de riscos e o plano de ações relacionados à segurança do trabalho. Os principais resultados obtidos com o uso do sistema aconteceu devido ao uso facilidade de acesso à legislações, materiais educativos e manuais de boas práticas sobre segurança do trabalho. Isso contribui para o devido atendimento legal, conscientização e capacitação dos trabalhadores, promovendo a cultura de segurança no ambiente laboral. Com base nos resultados e na literatura especializada, conclui-se que o sistema desenvolvido pode contribuir na gestão de segurança do trabalho nas empresas, proporcionando maior controle, eficiência e segurança aos processos relacionados ao programa de gerenciamento de riscos. Por meio do monitoramento deste programa e da disseminação de informações de segurança, as empresas podem promover a saúde e o bem-estar de seus funcionários, ao mesmo tempo em que cumprem suas responsabilidades legais em relação à segurança do trabalho.

**Palavras-chave:** Sistema Web, Gerenciamento de Riscos Ocupacionais, Segurança do Trabalho

### **ABSTRACT**

The use of specialized management systems can be a useful tool to help companies comply with their legal obligations and promote a safety culture. Through it, it is possible for companies to monitor information on preventive actions, exposure to harmful agents in different functions and workplaces, done through the collection and analysis of inventory data of risks and execution of the measures proposed in the action plan. This information is essential for making decisions related to work safety. This work aims to present a web system for the implementation of the occupational risk management program. The proposed system intends to monitor the risk inventory and the action plan related to work safety. A relevant feature of the system is the ease of access to legislation, educational materials and good practice manuals on work safety. This contributes to due legal service, awareness and training of workers, promoting the culture of safety in the work environment. Based on the results and the literature specialist, it is concluded that the developed system can contribute to the management of work safety in companies, providing greater control, efficiency and security to the processes related to the risk management program. Per through the monitoring of this program and the dissemination of information on safety, companies can promote the health and well-being of their employees, while fulfilling their legal responsibilities in relation to workplace safety.

**Keywords:** Web System, Ocupational Risk Management, Occupational Safety

# Lista de Figuras

| 2.1  | Arquivo de estilo pgr.css                                    | 15 |
|------|--|----|
| 2.2  | Usando Jquery e AJax dentro do código HTML                   | 18 |
| 2.3  | HTML com importação CDN do bootstrap                         | 20 |
| 2.4  | Arquitetura Django   | 22 |
| 2.5  | Estrutura HTML com tags do django entre chaves               | 24 |
| 2.6  | Abstração de queries no Django                               | 26 |
| 2.7  | Arquivo db.sqlite3 com as tabelas utilizadas no projeto      | 29 |
| 2.8  | Conteinerização de aplicações com Docker                     | 33 |
| 3.1  | Esquema de sprints utilizando o modelo de gráfico de Gantt   | 35 |
| 3.2  | Repositório no Github  | 37 |
| 3.3  | Relação dos Commits no Github                                | 37 |
| 3.4  | Levantamento de Requisitos                                   | 40 |
| 3.5  | Modelo com 3 instâncias de objeto                            | 42 |
| 3.6  | Diagrama de Classes para o PGR                               | 43 |
| 3.7  | Configuração do ambiente de desenvolvimento                  | 45 |
| 3.8  | Docker-compose.yml   | 46 |
| 3.9  | Criação do diretório raíz                                    | 47 |
| 3.10 | Padrão utilizado para definir as rotas para diferentes views | 50 |
| 3.11 | Criando as views   | 52 |
| 3.12 | Configurando o banco de dados                                | 53 |
| 3.13 | Criando os templates   | 54 |
| 3.14 | Exemplo de utilização do ModelForm                           | 55 |
| 5.1  | Identificação dos perigos                                    | 62 |
| 5.2  | Avaliação dos riscos   | 63 |
| 5.3  | Plano de Ação  | 64 |
| A 1  | Diagrama de Classes  | 71 |

# Lista de Tabelas

| 3.1 | Links referentes ao software desenvolvido                           | 56 |
|-----|---|----|
| 4.1 | Resumo das principais funcionalidades e diferenças entre os softwa- |    |
|     | res de gestão de segurança do trabalho mencionados:                 | 60 |

# Glossário: Símbolos e Siglas

| Notação | Descrição  | Páginas                            |
|---------|--|------------------------------------|
| API     | Programming Interface- Interface de Programação de Aplicações              | 16, 35                             |
| CBVs    | Visões baseada em classes  | 66                                 |
| CDN     | Content Delivery Network   | 20                                 |
| CLT     | Consolidação das Leis do Trabalho  | 8                                  |
| CRUD    | Create, Read, Update, Delete- Criação,<br>Consulta, Atualização e Exclusão | 38, 53                             |
| CSRF    | Cross-Site Request Forgery   | 24                                 |
| CSS     | Cascading Style Sheets- Folhas de Estilo em Cascata                        | 15, 16,<br>20, 54                  |
| DTL     | Django Template Language- Linguagem<br>de Template do Django               | 23                                 |
| EPI     | Equipamento de Proteção Individual   | 9, 58,<br>60, 67                   |
| GSST    | Gestão de Saúde e Segurança do Trabalho                                    | 2                                  |
| HTML    | HyperText Markup Language- Linguagem<br>de Marcação de Hipertexto          | 14–17,<br>20, 23,<br>25, 27,<br>54 |

| Notação | Descrição   | Páginas                |
|---------|---|------------------------|
| JSON    | JavaScript Object Notation- Notação de<br>Objetos JavaScript                                  | 25                     |
| NR      | Norma Regulamentadora   | 1, 8, 9,<br>38, 72     |
| OHSAS   | Occupational Health Safety Assessment<br>Series- Gestão de Segurança e Saúde Ocu-<br>pacional | 8                      |
| OIT     | Organização Internacional do Trabalho   | 7                      |
| ORM     | Object Relational Mapper- Mapeamento<br>Objeto-Relacional                                     | 41                     |
| PCMSO   | Programa de Controle Médico de Saúde<br>Ocupacional   | 8, 58-<br>60           |
| PGR     | Programa de Gerenciamento de Riscos   | 8, 38,<br>59,<br>62–64 |
| PPRA    | Programa de Prevenção de Riscos Ambientais  | 8, 58–<br>60           |
| SAD     | Sistema de Apoio à Decisão  | 11, 12                 |
| SESMT   | Serviço Especializado em Engenharia de<br>Segurança e Medicina do Trabalho                    | 1                      |
| SIG     | Sistema de Informação Gerencial   | 11, 12                 |
| SST     | Saúde e Segurança do Trabalho   | 9, 10                  |
| TI      | Tecnologia da Informação  | 11                     |
| UML     | Unified Modeling Language   | 41, 42                 |
| URLs    | Uniform Resource Locator  | 41                     |

| Notação | Descrição                 | Páginas |
|---------|---------------------------|---------|
| W3C     | World Wide Web Consortium | 15      |

# Sumário

| Fo | olha d | e Rosto                                    |
|----|--------|--|
| Fi | cha C  | italográfica                               |
| Fc | olha d | e Aprovação                                |
| Re | esum   |  |
| Αl | bstra  | t  |
| Li | sta d  | Figuras                                    |
| Li | sta d  | Tabelas                                    |
| GI | lossá  | o: Símbolos e Siglas i                     |
| 1  | Intr   | dução                                      |
|    | 1.1    | ustificativa                               |
|    | 1.2    | Problema                                   |
|    | 1.3    | Objetivos                                  |
|    |        | 1.3.1 Objetivo Geral                       |
|    |        | 1.3.2 Objetivos Específicos                |
|    | 1.4    | Metodologia                                |
|    | 1.5    | Organização do Trabalho                    |
| 2  | FUN    | DAMENTAÇÃO TEÓRICA                         |
|    | 2.1    | Gestão de Segurança do Trabalho            |
|    |        | 2.1.1 Conceitos Básicos                    |
|    |        | 2.1.2 Legislação e Normas Regulamentadoras |
|    |        | 2.1.3 Programa de Gerenciamento de Riscos  |
|    | 2.2    | Sistemas de Informação                     |
|    | 2.3    | Tecnologias Web                            |
|    |        | 2.3.1 Frontend                             |

|   |     |       | 2.3.1.1 HTML                       | 14  |
|---|-----|-------|------------------------------------|-----|
|   |     |       | 2.3.1.2 CSS                        | 15  |
|   |     |       | 2.3.1.3 JavaScript                 | 16  |
|   |     |       | 2.3.1.4 JQuery                     | 17  |
|   |     |       | 2.3.1.5 AJAX                       | 18  |
|   |     |       | 2.3.1.6 Bootstrap                  | 20  |
|   |     | 2.3.2 | Backend                            | 21  |
|   |     |       | 2.3.2.1 Python                     | 21  |
|   |     |       |                                    | 22  |
|   |     |       | 2.3.2.3 SQLite                     | 28  |
|   |     | 2.3.3 | Ferramentas de desenvolvimento     | 30  |
|   |     |       | 2.3.3.1 ClickUp                    | 30  |
|   |     |       | 2.3.3.2 GitHub                     | 30  |
|   |     |       | 2.3.3.3 Visual Studio Code         | 31  |
|   |     |       | 2.3.3.4 Docker                     | 32  |
| 3 | DES | ENVOI | _VIMENTO                           | 3⊿  |
|   | 3.1 |       | jamento e versionamento do projeto |     |
|   |     | 3.1.1 |                                    |     |
|   |     | 3.1.2 | Diagrama de tarefas                | 36  |
|   |     | 3.1.3 | Versionamento de código            | 37  |
|   | 3.2 | Levan | tamento de requisitos              | 38  |
|   |     | 3.2.1 | Necessidades dos clientes          | 38  |
|   |     | 3.2.2 | Prototipagem do sistema            | 40  |
|   | 3.3 | Mode  | lagem do sistema                   | 41  |
|   |     | 3.3.1 | Metodologia de modelagem           | 41  |
|   |     | 3.3.2 | Diagrama de Classes                | 42  |
|   | 3.4 | Imple | mentação                           | 44  |
|   |     | 3.4.1 | Ambiente de desenvolvimento        | 45  |
|   |     | 3.4.2 | Criando Models                     | 49  |
|   |     | 3.4.3 | Definindo URLs                     | 50  |
|   |     | 3.4.4 | Criando Views                      | 52  |
|   |     | 3.4.5 | Criando Templates                  | 54  |
|   |     | 3.4.6 | Criando Formulários                | 55  |
| 4 | TRA | BALH  | OS CORRELATOS                      | 57  |
|   | 4.1 |       | care                               |     |
|   | 42  |       |                                    | 5.8 |

|    | 4.3 On Safety  | 58 |
|----|--|----|
|    | 4.4 Software SGG   | 58 |
|    | 4.5 Comparativo dos Sistemas   | 59 |
| 5  | RESULTADOS   | 61 |
| 6  | CONCLUSÕES   | 65 |
| 7  | TRABALHOS FUTUROS  | 67 |
| RE | FERÊNCIAS  | 68 |
| A  | Modelagem do banco de dados  | 70 |
| I  | Norma Regulamentadora n.º 01 - Disposições Gerais e Gerenciamento de Riscos Ocupacionais | 72 |

# Capítulo 1

## Introdução

A segurança do trabalho trata de um conjunto de normas, ações, atividades e medidas preventivas destinadas à melhoria do ambiente de trabalho, prevenção de ocorrência de acidentes de trabalho e doenças ocupacionais. Ela visa criar um ambiente organizacional que propicie o bem estar e a segurança dos colaboradores.

No Brasil, esse conceito é tratado como Serviço Especializado em Engenharia de Segurança e Medicina do Trabalho (SESMT), que é obrigatório a todas as empresas, e foi regulamentado pelo Ministério do Trabalho e Emprego por meio da Norma Regulamentadora NR-04, que trata de descrever todos os serviços prestados por essas equipes de segurança do trabalho.

O gerenciamento de riscos, pode ser definido como um processo formal em que as incertezas presentes são sistematicamente identificadas, analisadas, estimadas, categorizadas e tratadas (RUPPENTHAL, 2013). A importância de sua implementação está relacionada à capacidade de uma empresa antecipar e lidar efetivamente com eventos que possam causar danos significativos aos seus processos, projetos, reputação e resultados financeiros. Ao adotar uma abordagem sistemática para identificar e tratar os riscos, as empresas podem reduzir a probabilidade de ocorrência de eventos negativos, bem como minimizar o impacto caso eles ocorram.

Existem algumas dificuldades associadas ao gerenciamento de riscos. Primeiramente, identificar todos os riscos potenciais pode ser um desafio, pois o ambiente de trabalho está em constante mudança e novos riscos podem surgir a qualquer momento. Além disso, avaliar a probabilidade e o impacto dos ris-

cos de forma precisa também pode ser complicado, uma vez que envolve estimativas e projeções baseadas em informações limitadas. Outra dificuldade é a resistência organizacional à implementação de medidas de gerenciamento de riscos. As empresas podem subestimar a importância do gerenciamento de riscos ou considerá-lo uma atividade secundária, priorizando outras demandas operacionais. No entanto, ignorar os riscos pode levar a consequências graves, incluindo perdas financeiras, danos à reputação e até mesmo a sobrevivência da empresa.

O gerenciamento de riscos requer uma abordagem holística e a colaboração de todas as partes interessadas, o que pode ser um desafio em empresas complexas. Coletar informações relevantes, envolver os colaboradores, implementar medidas de controle e monitorar continuamente os riscos exigem esforços coordenados e uma cultura de conscientização e responsabilidade dentro da empresa.

Apesar das dificuldades, investir em gerenciamento de riscos é fundamental para uma gestão eficaz e sustentável. Ao antecipar e lidar proativamente com os riscos, as empresas podem melhorar sua resiliência, tomar decisões mais informadas e garantir a continuidade de suas operações.

Desta forma, é importante que as empresas adotem estratégias para gerenciar os riscos ocupacionais e cumprir com suas obrigações legais. Nesse contexto, surge que a utilização de um Sistema de Gestão se mostra uma ferramenta promissora para monitorar os riscos e prover um ambiente adequado aos empregados. Exercer uma boa gestão de segurança do trabalho é uma das principais preocupações dos gestores, pois minimiza a ocorrência dos riscos de acidentes.

Este trabalho busca oferecer uma solução para acesso a um sistema web que auxilia no controle das ações do programa de gerenciamento de riscos ocupacionais, contribuindo para a Gestão de Saúde e Segurança do Trabalho (GSST) e o devido atendimento legal nas empresas.

### 1.1 Justificativa

A justificativa deste trabalho baseia-se na necessidade de aprimorar a gestão de segurança do trabalho, considerando o programa de gerenciamento de riscos ocupacionais, e o cumprimento das obrigações trabalhistas e previdenciárias.

Algumas empresas enfrentam desafios na implementação e monitoramento desse programa, como a falta de recursos tecnológicos e a dificuldade de acesso a informações sobre legislações e boas práticas em segurança do trabalho.

Ao utilizar a tecnologia e a automatização oferecidas por um sistema web, será possível monitorar informações sobre a identificação e avaliação dos riscos em diferentes funções, além de facilitar a coleta e análise de dados do inventário de risco e plano de ação dos empregados.

### 1.2 Problema

Ao implementar um programa de gerenciamento de riscos, é importante estar ciente de alguns desafios, que incluem a falta de recursos adequados para o monitoramento e coleta de informações relevantes sobre segurança do trabalho, o que pode levar a uma tomada de decisão inadequada e a uma gestão de risco ineficiente. É preciso investir em ferramentas de monitoramento e coleta de dados mais eficientes e confiáveis, que possam fornecer informações precisas e atualizadas sobre a segurança do trabalho nas empresas. Além disso, é importante envolver todos os empregados no processo de gerenciamento de riscos, para que todos estejam cientes dos riscos existentes e possam contribuir para a redução dessa exposição. A segurança do trabalho é fundamental para garantir a saúde e o bem-estar dos colaboradores, além de ser uma obrigação legal. Portanto, é essencial investir em ferramentas e estratégias eficazes para gerenciar os riscos ocupacionais e garantir um ambiente de trabalho seguro e saudável para todos.

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um sistema web que possa auxiliar as empresas na gestão do programa de gerenciamento de riscos ocupacionais.

### 1.3.2 Objetivos Específicos

- Desenvolver um sistema que centralize em um único local, informações do inventário de riscos e do plano de ações relacionados à segurança ocupacional.
- Permitir o registro de dados históricos dos empregados, possibilitando avaliar a exposição aos riscos, implementação de medidas preventivas e das ações realizadas ao longo do tempo.
- ▶ Disponibilizar link de acesso às principais legislações trabalhistas, previdenciárias e manuais de boas práticas sobre segurança do trabalho, visando promover a conscientização e capacitação dos trabalhadores.
- Proporcionar maior controle das ações de segurança do trabalho, através do registro detalhado dos riscos identificados de cada função, natureza dos riscos, a probabilidade de ocorrência e seu potencial impacto, auxiliando as empresas no cumprimento de suas responsabilidades legais.

### 1.4 Metodologia

A metodologia utilizada neste trabalho quanto a sua natureza, se classifica em pesquisa aplicada, cujo principal objetivo é a geração de conhecimento para aplicação prática e imediata, dirigidos à solução de problemas específicos envolvendo os interesses locais, territoriais e regionais. Quanto à abordagem, a pesquisa é qualitativa de observação participativa: Neste tipo de estudo, o pesquisador participa do problema ou situação a ser analisada.

Consiste no desenvolvimento de um sistema web para auxiliar na implantação do Programa de Gerenciamento de Riscos Ocupacionais, com as seguintes etapas:

Planejamento e versionamento do Projeto: Nesta etapa foram utilizados um gerenciador de tarefas ClickUp para a gestão de projetos. Com ele, foi possível acessar o projeto utilizando uma interface, que permite a visualização das listas de tarefas em diagramas com metas e prazos definidos. Foi utilizado também o Github para o versionamento e controle de alterações no projeto.

Levantamento de requisitos: Nesta etapa, foram avaliadas as necessidades e requisitos do sistema e, por meio da prototipagem de um sistema web, com informações dos ambientes de trabalho, e dados provenientes do Ministério do Trabalho e Previdência Social, foram moduladas para o gerenciamento dos riscos no ambiente de trabalho. Nesta etapa foi necessário análises de dados técnicos relativos à segurança do trabalho e experiências anteriores do autor do projeto.

Modelagem do sistema: Com base no levantamento de requisitos, foi definido a arquitetura de Model-View-Template, o Sistema de Gerenciamento de Bando de Dados padrão do Django, SQLite, a interface do usuário e as principais funcionalidades. Nesta etapa, foram definidos também os aspectos de segurança e acesso aos dados.

Implementação: Nesta etapa, o sistema foi prototipado com o framework Django onde foram feitas as configurações iniciais do ambiente de desenvolvimento com o docker, instalação das dependências, definição dos modelos e realizado as migrações iniciais, criação das tabelas no banco de dados, o desenvolvimento das views, configuração das rotas, criação dos templates e os formulários do projeto.

## 1.5 Organização do Trabalho

Este trabalho está organizado em 7 Capítulos, divididos da seguinte maneira: No primeiro capítulo, a Introdução trata da motivação deste projeto, a justificativa, problema, objetivos e a sua metodologia. O segundo traz a fundamentação teórica que embasa o desenvolvimento desta aplicação web. O terceiro apresenta o desenvolvimento do software e do processo de planejamento, arquitetura, tecnologias e etapas do desenvolvimento do projeto. O quarto mostra os trabalhos correlatos com um comparativo entre alguns softwares disponíveis no mercado. Já o quinto capítulo apresenta os resultados do sistema desenvolvido. No sexto, as conclusões a partir do trabalho finalizado, e, o sétimo capítulo, traz possíveis sugestões para trabalhos futuros.

# Capítulo 2

# **FUNDAMENTAÇÃO TEÓRICA**

### 2.1 Gestão de Segurança do Trabalho

#### 2.1.1 Conceitos Básicos

Para compreender a gestão de segurança e saúde no trabalho, é fundamental explorar alguns conceitos básicos que norteiam esta matéria. De acordo com a Organização Internacional do Trabalho (OIT), o acidente de trabalho é definido como "um evento súbito e indesejado que ocorre no local de trabalho, resultando em lesão, doença ou morte". Já as doenças ocupacionais são descritas como "doenças adquiridas ou agravadas no ambiente de trabalho, resultantes da exposição a riscos ocupacionais". (OIT, 2011).

Para a Prevenção de Acidentes, há uma definição abrangente sobre seu conceito e pode ser bem referenciada por Gonçalves (2017), que afirma que "Prevenção de acidentes é o conjunto de medidas adotadas com o objetivo de evitar a ocorrência de eventos indesejáveis que possam causar lesões, doenças ou danos materiais nos locais de trabalho".

No que diz respeito à gestão de segurança do trabalho, Junior (2021) apresenta a seguinte definição: "Gestão de segurança do trabalho é o processo pelo qual uma organização identifica, analisa e controla os riscos ocupacionais presentes em suas atividades, visando à prevenção de acidentes e a promoção de um ambiente de trabalho seguro e saudável".

### 2.1.2 Legislação e Normas Regulamentadoras

No âmbito da legislação trabalhista e das normas regulamentadoras, a Consolidação das Leis do Trabalho (CLT) é uma referência essencial. Conforme previsto na (CLT), a segurança do trabalho deve ser assegurada mediante a adoção de medidas que visem à prevenção de acidentes e doenças ocupacionais, bem como à promoção de condições adequadas de trabalho (BRASIL, 2021). Além disso, as Normas Regulamentadoras (NRs), emitidas pelo Ministério do Trabalho e Emprego, estabelecem requisitos específicos para a gestão de segurança e saúde no trabalho.

A Norma Internacional OHSAS 18001 (Occupational Health and Safety Assessment Series) apresenta um conjunto de requisitos para a implementação de um sistema de gestão de saúde e segurança ocupacional eficaz. Esses requisitos incluem, a política de saúde e segurança no trabalho, a identificação de perigos, avaliação e implementação de controles de riscos, atrelados ao monitoramento do desempenho e a melhoria contínua do sistema.

A Norma Regulamentadora NR-1 estabelece as diretrizes básicas para o Gerenciamento de Riscos Ocupacionais em empresas, de forma a garantir a implementação de medidas de prevenção de acidentes e doenças ocupacionais. Por meio do PGR identificam-se os agentes nocivos, avalia-se a intensidade e frequência de exposição e são estabelecidas medidas de controle para prevenir acidentes e doenças ocupacionais.

A NR-7 estabelece a obrigatoriedade da elaboração e implementação do Programa de Controle Médico de Saúde Ocupacional (PCMSO), que visa a preservar e proteger a saúde dos trabalhadores. Esse programa envolve ações como exames médicos periódicos, monitoramento da saúde dos funcionários e medidas de prevenção e controle de doenças ocupacionais.

A NR-9 trata do Programa de Prevenção de Riscos Ambientais (PPRA), que tem como objetivo identificar, avaliar e controlar os riscos presentes no ambiente de trabalho, visando à prevenção de doenças e acidentes. A partir dessas análises, são estabelecidas ações preventivas, como melhorias na infraestrutura, adoção de medidas de controle coletivas e individuais, e treinamentos específicos para os trabalhadores.

Além disso, é essencial que as empresas adotem ações de segurança e prevenção de acidentes. A norma NR-6 estabelece a obrigatoriedade do fornecimento de equipamentos de proteção individual (EPIs) aos trabalhadores, de acordo com os riscos presentes no ambiente de trabalho. A implementação correta dessas medidas de proteção individual, juntamente com treinamentos específicos sobre o uso adequado dos EPIs, contribui para reduzir os riscos de acidentes e lesões.

Ou seja, temos nas normas regulamentadoras, importantes ferramentas para a promoção de um ambiente de trabalho seguro e saudável. Ações, atividades e medidas preventivas baseadas nessas normas garantem a proteção da saúde dos trabalhadores, a prevenção de acidentes e doenças ocupacionais, e contribuem para o bem-estar e a produtividade no local de trabalho.

### 2.1.3 Programa de Gerenciamento de Riscos

No que diz respeito ao programa de gerenciamento de riscos, Nunes (2010) enfatiza a importância de uma abordagem sistemática para identificar, avaliar e controlar os riscos ocupacionais. O autor destaca que a implementação de um programa eficaz de gerenciamento de riscos contribui para a prevenção de acidentes e doenças ocupacionais, proporcionando um ambiente de trabalho mais seguro.

A implementação de um programa de gerenciamento de riscos e a adoção de práticas efetivas de segurança e saúde no trabalho trazem uma série de benefícios para os trabalhadores e as organizações. De acordo com Silva (2019), a gestão efetiva de segurança e saúde no trabalho resulta na redução de acidentes e melhoria da qualidade de vida dos trabalhadores. Além dele, Leigh (2019) ressalta que a gestão adequada de SST também gera benefícios econômicos, como a redução de custos associados a acidentes, afastamentos e processos legais.

Sendo a liderança fundamental para estabelecer uma cultura de segurança, demonstrando comprometimento e fornecendo recursos adequados, com comunicação clara e eficaz entre a gestão e os trabalhadores, capacitação e treinamento, garantem que todos compreendam os riscos, medidas de controle e sua responsabilidade para a promoção de um ambiente de trabalho seguro.

Ter um sistema que auxilie as empresas com a gestão SST é muito importante. Assim é preciso estar sempre atento para não descumprir com as obrigações. Um software de SST é um sistema que ajuda no controle e gestão da segurança do trabalho em uma empresa. Dessa forma, ele faz o gerenciamento de todas as informações dos funcionários exigidas pelo governo.

Esses programas também têm a função de automatizar vários processos do ambiente de trabalho. Por exemplo, emissão de relatórios, coleta de dados, monitoramentos diversos e outras atividades do tipo. SST Por meio de um sistema de gestão SST, se facilita várias ações que levariam mais tempo para serem realizadas. Dessa maneira, pode-se fazer o controle de várias questões da empresa como: Controle de Funções x Riscos da atividade; Dados dos funcionários; Identificação, Avaliação e Gerenciamento de Riscos; Geração de documentos e planejamento de ações preventivas; disponibilidade dos dados via web a qualquer momento.

Portanto, compreender os conceitos básicos de SST, conhecer a legislação e as normas regulamentadoras, implementar um programa de gerenciamento de riscos e usufruir dos benefícios da gestão de SST são elementos essenciais para promover um ambiente de trabalho seguro e saudável, tanto para os trabalhadores quanto para as organizações.

### 2.2 Sistemas de Informação

Laudon (2004) define sistema de informação e mostra como as informações servem como base para a tomada de decisão.

"Um sistema de informação pode ser definido como um conjunto de componentes inter-relacionados que coleta, processa, armazena e distribui informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização. Além de dar suporte à tomada de decisões, à coordenação e ao controle, esses sistemas também auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos."(LAUDON, 2004).

Esses sistemas são responsáveis por fornecer suporte aos processos de tomada de decisão e controle da organização, tanto pelo setor de Tecnologia da Informação (TI) quanto pelos donos ou responsáveis pelo negócio. Existem duas categorias principais de sistemas de informação:

- ► Sistemas de Informação Gerencial (SIG), que têm como objetivo coletar, processar, armazenar e entregar informações em forma de relatórios para os gestores e tomadores de decisão. Eles fornecem informações para auxiliar no planejamento, controle e operação da organização. O SIG é responsável por coletar dados brutos, transformá-los em informações significativas e entregá-las aos usuários finais na forma de relatórios estruturados.
- Sistemas de Apoio à Decisão (SAD), que são projetados para auxiliar na resolução de problemas ou situações específicas. Eles utilizam as informações coletadas pelo SIG e aplicam técnicas e modelos analíticos para identificar tendências, padrões e ocorrências relevantes. O SAD fornece suporte aos tomadores de decisão em termos de análise de dados, simulações, projeções e identificação de alternativas, ajudando-os a tomar decisões informadas e estratégicas.

O Sistema de Apoio à Decisão (SAD) e o Sistema de Informações Gerenciais (SIG) desempenham um papel crucial no Programa de Gerenciamento de Riscos,

conforme estabelecido na Norma Regulamentadora NR01. O SAD, quando aplicado ao Programa de Gerenciamento de Riscos, oferece suporte aos tomadores de decisão, como gestores e responsáveis pela segurança no trabalho, fornecendo informações e análises relevantes. Ele ajuda a avaliar os riscos presentes nos ambientes de trabalho, identificar suas causas e propor medidas preventivas e corretivas adequadas. O SAD permite a análise de dados, como estatísticas de acidentes, registros de incidentes, inspeções de segurança e resultados de avaliações de riscos.

Por sua vez, o SIG é responsável por coletar, armazenar, organizar e fornecer acesso às informações relacionadas ao gerenciamento de riscos. Ele centraliza os dados pertinentes, como relatórios de avaliação de riscos, planos de ação, indicadores de desempenho e histórico de acidentes. Com o SIG, é possível acompanhar a evolução dos riscos, monitorar a eficácia das medidas adotadas e gerar relatórios que auxiliem na tomada de decisões estratégicas.

Ambos os sistemas são fundamentais para atender às exigências da norma regulamentadora, que incluem a identificação dos riscos, a avaliação dos mesmos, a implementação de medidas de controle, a comunicação e o treinamento dos colaboradores, além do monitoramento contínuo e da revisão periódica do programa de gerenciamento de riscos.

Ao utilizar o SAD e o SIG no Programa de Gerenciamento de Riscos Ocupacionais, as empresas podem obter benefícios significativos, como uma visão abrangente dos riscos existentes, maior eficiência nas ações preventivas, monitoramento mais preciso dos indicadores de segurança e aprimoramento contínuo do sistema de gerenciamento de riscos para garantir um ambiente de trabalho mais seguro e saudável.

Ambos podem ser utilizados em conjunto para obter uma visão abrangente e embasada na análise dos resultados do inventário de riscos e do plano de ação no Gerenciamento de Riscos Ocupacionais.

### 2.3 Tecnologias Web

As tecnologias web englobam um conjunto de ferramentas, protocolos, linguagens de programação e metodologias utilizadas para desenvolver e implementar aplicações e serviços na internet. Elas desempenham um papel essencial na criação de websites, aplicações web e outros recursos online. Essas tecnologias podem ser divididas em duas partes distintas: a parte do *backend* e a parte do *frontend*. Serão apresentadas nas próximas seções seus conceitos e que elas têm responsabilidades diferentes mas trabalham em conjunto para oferecer uma experiência completa e satisfatória aos usuários

O backend é responsável por tudo o que acontece nos bastidores de um sistema, incluindo a interação com bancos de dados, a execução de cálculos complexos e a manipulação de dados. Ele também é responsável por fornecer os dados e as funcionalidades necessárias para a interface do usuário.

O *frontend* é a parte visível e interativa de um sistema, por meio da qual os usuários podem interagir e consumir informações.

#### 2.3.1 Frontend

O frontend está relacionado à apresentação e à interação do usuário com a aplicação web. Envolve a criação da interface visual, o design responsivo, a usabilidade e a experiência do usuário. As tecnologias fundamentais do frontend incluem HTML (Hypertext Markup Language) para a estruturação do conteúdo, CSS (Cascading Style Sheets) para a estilização e layout, e JavaScript para a interatividade e dinamismo. Além disso, existem diversos frameworks e bibliotecas, como React, Angular e Vue.js, que auxiliam no desenvolvimento de interfaces mais complexas e funcionais. O *frontend* lida com o design, a aparência e a usabilidade de uma aplicação, focado em apresentar informações de forma clara, organizada e atraente para os usuários. Isso envolve o desenvolvimento da interface gráfica, a implementação de elementos interativos, a disposição dos elementos na página e a otimização da experiência do usuário.

#### 2.3.1.1 HTML

HTML (HyperText Markup Language ou Linguagem de Marcação de Hipertexto) é um linguagem utilizada para estruturação semântica de conteúdos de páginas web e permite criar a estrutura básica de um site na internet. Embora ele não seja uma linguagem de programação, usando marcações de códigos chamados "Tags", possibilita que informações sejam organizadas e formatadas para dar estrutura e forma à página. Quando visitamos uma página na web, percebemos que existem hierarquias entre textos, menus, imagens e vídeos. É exatamente essa distribuição das informações que é estabelecida usando HTML. Dessa forma, é graças à linguagem de HTML e aos navegadores que conseguimos acessar e interagir de forma apropriada com diversos conteúdos. Por ser uma linguagem baseada em marcações, na programação em HTML são usadas tags para indicar como a informação marcada deve ser apresentada, desse modo, as tags vão dando orientações ao navegador sobre como as informações devem ser distribuídas e exibidas. Isso diferencia títulos de parágrafos, de links, de menus, etc.

Um website pode conter várias páginas HTML, pois cada subpágina do site pode ter sua própria construção em HTML. Contudo o HTML por si só não define a estilização do conteúdo mostrado, como analogia podemos indicar que o HTML seria o esqueleto de uma página web (RESENDE, 2021).

#### 2.3.1.2 CSS

O CSS é uma sigla para Cascading Style Sheets (Folhas de Estilo em Cascata) é a linguagem padrão da W3C (World Wide Web Consortium) para definir os estilos visuais de páginas na Web, fazendo com que o HTML cuide somente da estrutura dos elementos da página.

Uma folha de estilos CSS é composta por instruções que definirão o visual do documento, chamadas de *style rules* (regras de estilo), Figura 2.1, que descrevem como um elemento ou grupo deve ser mostrado. Cada regra seleciona um elemento e declara como ele deve se parecer. As regras possuem duas seções principais: o seletor e a declaração. O seletor identifica qual elemento será afetado. A declaração fornece instruções de renderização e é composta pela propriedade e seu valor, separados por dois pontos e espaço e quando mais declarações são feitas para o mesmo seletor, devem ser separadas por ponto e vírgula.

```
static > css > 😎 pgr.css > ધ .header
        .header {
            border-bottom: 1px solid ■white;
            padding: 20px;
          .body-form {
            padding: 20px;
          .font-logo {
   11
            margin-left: 10px;
   12
            font-size: 40px;
          .btn-custom-primary {
            background-color: ■#3D7B80;
            border: none;
  18
            box-shadow: 2px 2px 5px 1px □black;
  19
            color: ■white;
            font-size: 20px;
  21
            padding: 5px 20px;
FONTE: Elaborado pelo autor
            border-radius: 10px;
  23
            margin-top: 20px;
  24
          .link-float {
            text-decoration: none;
             float: right:
```

**Figura 2.1** – Arquivo de estilo pgr.css

### 2.3.1.3 JavaScript

O JavaScript é uma linguagem de programação de scripts que adiciona interatividade e comportamentos personalizados a uma página web. Ela ocorre na máquina do usuário e não no servidor, ou seja, no lado do cliente (client-side). Ela é uma linguagem dinâmica, e não precisa de um compilador para interpretar o código antes de ser executado, o navegador lê e interpreta o código à medida que ele for necessário. Ela também é fracamente tipada e não precisa que o desenvolvedor defina o tipo de variável ao declará-la, ou seja, se uma variável for igual a um número, o JavaScript saberá que é um número automaticamente (ROBBINS, 2018).

O JavaScript é suportado por todos os principais navegadores da web e é amplamente utilizado para criar efeitos de animação, validar formulários e muito mais. Com ele, é possível criar uma experiência de usuário mais interativa e envolvente. Outra característica importante é sua capacidade de trabalhar com APIs para acessar dados e recursos de outros sistemas. Por exemplo, uma aplicação pode usar a API do Google Maps para exibir mapas e direções em sua página web.

Considerado uma linguagem de programação versátil, permite que os desenvolvedores usem diferentes estilos de programação, como programação procedural, orientada a objetos e funcional. Isso torna a linguagem adequada para uma ampla gama de projetos e aplicações.

Com o uso de bibliotecas e frameworks, como o React e o Angular, é possível criar interfaces de usuário complexas e escaláveis. Além disso, o JavaScript é usado em conjunto com outras tecnologias web, sendo que o principal uso neste trabalho se dará com o HTML5 e o CSS, para criar animações e transições suaves e responsivas.

### 2.3.1.4 **JQuery**

*jQuery* é uma biblioteca JavaScript que simplifica a manipulação de documentos HTML, manipulação de eventos, animação e interações com o servidor. Ela fornece uma sintaxe fácil de usar que abstrai muitos detalhes complicados do JavaScript puro.

O *jQuery* é amplamente utilizado para desenvolver páginas web interativas e dinâmicas, permitindo aos desenvolvedores criar efeitos visuais interessantes e melhorar a experiência do usuário em seus sites. Ele também oferece uma ampla gama de *plugins e widgets* que podem ser facilmente integrados em suas aplicações web para adicionar recursos como carrosséis de imagens, menus *drop-down*, calendários e muito mais.

Uma das principais vantagens do *jQuery* é sua compatibilidade com todos os principais navegadores da web, o que significa que você pode ter certeza de que sua aplicação funcionará corretamente em todos os dispositivos e plataformas. Além disso, sua sintaxe simplificada facilita o aprendizado e a utilização do jQuery, mesmo para aqueles que não têm muita experiência em programação.

Outra grande vantagem do jQuery é sua capacidade de facilitar a interação com o servidor usando Ajax, permitindo que os desenvolvedores atualizem partes específicas de uma página web sem precisar recarregá-la completamente. Isso pode melhorar significativamente a velocidade e a eficiência de sua aplicação, proporcionando uma experiência mais fluida e agradável para os usuários. Com ele é possível acessar elementos, conteúdos e atributos de um HTML, além de manipulálos se necessário.

### 2.3.1.5 AJAX

Ajax é uma técnica de programação que permite atualizar partes de uma página da web sem precisar recarregar toda a página. Com Ajax, é possível enviar e receber dados do servidor em segundo plano, sem interromper a experiência do usuário. Holdener (2008) explora os conceitos fundamentais do Ajax, incluindo sua arquitetura, técnicas de implementação e melhores práticas. O autor destaca que o Ajax tem um impacto significativo no desenvolvimento web, permitindo uma experiência mais fluida e responsiva para os usuários.

O JavaScript é utilizado normalmente para a criação de pequenos efeitos visuais na página como elementos com movimento, que mudam de cor e tamanho conforme a posição do mouse ou da tela, como podemos ver nos menus deslizantes apresentados na figura 2.2.

O conteúdo dinâmico é algo muito comum em aplicações e sites na web. Atualmente, ele permite que páginas tenham seu conteúdo modificado sem a necessidade de recarregá-la a cada modificação, tornando a experiência do usuário muito mais suave e rápida. Para isso foram utilizados o JavaScript em conjunto com o AJAX e sua biblioteca JQuery no projeto.

Figura 2.2 – Usando Jauery e Alax dentro do código HTML

Para uma aplicação Web dinâmica o uso do JavaScript e seus recursos é de extrema necessidade, visto que para manter a aplicação reativa e entregar uma melhor experiência do usuário se torna fundamental e imprescindível a utilização da linguagem no *frontend*.

Com o Ajax, os desenvolvedores podem criar aplicações web mais responsivas, fornecendo atualizações em tempo real para o conteúdo da página e melhorando a experiência do usuário. Além disso, o Ajax permite que as aplicações web sejam mais eficientes, já que apenas as partes da página que precisam ser atualizadas são atualizadas, reduzindo a necessidade de recarregar toda a página a cada interação do usuário.O uso de Ajax é especialmente útil para aplicações web que possuem atualizações frequentes de dados, como redes sociais e aplicações de mensagens instantâneas. Usando o Ajax, essas aplicações podem atualizar automaticamente as informações na página sem que o usuário precise atualizar a página manualmente. Outra vantagem do Ajax é que ele pode ser facilmente integrado com outras tecnologias web, como o jQuery, para fornecer um desenvolvimento mais rápido e eficiente. Além disso, o Ajax é amplamente suportado pelos principais navegadores da web, tornando-o uma opção viável para a criação de aplicações web em larga escala.

Em resumo, o Ajax é uma técnica de programação importante para o desenvolvimento de aplicações web dinâmicas e interativas. Com sua capacidade de atualizar partes específicas da página, sem a necessidade de recarregar toda a página, o Ajax pode melhorar significativamente a experiência do usuário e tornar as aplicações web mais eficientes.

### 2.3.1.6 Bootstrap

Bootstrap é um framework *frontend* que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples. Além disso, pode lidar com sites de desktop e páginas de dispositivos móveis da mesma forma.

O Bootstrap é distribuído usando o site oficial. Pode-se também criar sua própria distribuição usando o código-fonte, ou por meio da própria rede de distribuição de conteúdo conhecida como CDN do Bootstrap. Um CDN permite que um site da web recupere com frequência arquivos públicos usados de servidores distribuídos globalmente, apresentados na figura 2.3. Para aplicar esses estilos, é

**Figura 2.3** – HTML com importação CDN do bootstrap

necessário usar as classes definidas pelo framework nos elementos desejados, e importar os arquivos CSS e Javascript em seus respectivos documentos HTML.

Isso ajuda a criar um formulário visualmente atraente e responsivo, com os estilos e layout consistentes do Bootstrap.

#### 2.3.2 Backend

O backend refere-se à infraestrutura e às ferramentas utilizadas para desenvolver a lógica e a funcionalidade das aplicações web. Isso envolve a criação e o gerenciamento de servidores, bancos de dados, sistemas de autenticação, APIs (Interfaces de Programação de Aplicativos) e outras tecnologias que processam os dados e fornecem respostas às solicitações dos usuários. Linguagens de programação como Python, Java, Ruby e PHP são comumente usadas no backend, juntamente com frameworks e bibliotecas específicas para facilitar o desenvolvimento.

### 2.3.2.1 Python

Python é uma linguagem de programação de alto nível, interpretada e de propósito geral, amplamente utilizada no desenvolvimento de aplicações web. Atualmente em sua versão 3, o código Python é projetado para ser legível e fácil de manter, e inclui mecanismos como módulos e programação orientada a objetos, que promovem a reutilização de código. Uma das principais vantagens do Python para o desenvolvimento web é a sua vasta comunidade e a disponibilidade de recursos. Existem muitos tutoriais, documentações e fóruns online onde os desenvolvedores podem encontrar ajuda e compartilhar conhecimentos. É uma linguagem usada para completar tarefas mais rapidamente que linguagens estaticamente compiladas.

Outras vantagens dela são a tipagem dinâmica, o gerenciamento de memória automático, o suporte para programação em grande escala, os tipos de objetos incorporados, as ferramentas incorporadas, as bibliotecas e os utilitários de outros fornecedores.

### 2.3.2.2 Django

O Django é um framework web em Python que utiliza a arquitetura Model-View-Template (MVT) para o desenvolvimento de aplicações. Ele fornece uma estrutura sólida e flexível para o desenvolvimento de aplicações web, sendo possível criar aplicações web robustas, escaláveis e de fácil manutenção. Nessa arquitetura, as responsabilidades são divididas em três componentes principais: Model (modelo), View (visão) e Template (modelo de apresentação), conforme Figura 2.4.

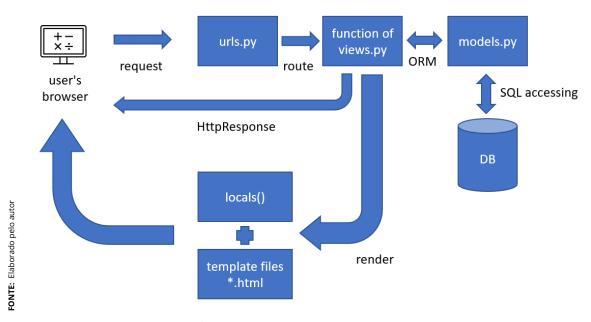


Figura 2.4 – Arquitetura Django

O componente Model é responsável por definir a estrutura e a lógica de negócios dos dados da aplicação. Ele representa a camada de acesso aos dados, sendo responsável pela criação, leitura, atualização e exclusão dos objetos no banco de dados. Os modelos do Django são geralmente implementados utilizando classes que herdam da classe django.db.models.Model.

O componente View é responsável por processar as requisições HTTP e retornar as respostas adequadas. Ele recebe as requisições do cliente e interage com o componente Model para obter os dados necessários. Em seguida, ele formata esses dados e os passa para o Template para renderização. As views do Django são implementadas como funções ou métodos de classe decorados com o @django.views.decorators.http ou @django.views.generic.base para definir o comportamento esperado para cada tipo de requisição.

O componente Template é responsável pela apresentação dos dados ao usuário final. Ele define a estrutura e o estilo da página que será enviada ao cliente. O Django utiliza uma linguagem de template própria, que permite a inserção de variáveis, expressões condicionais, iterações e outros recursos para manipulação e exibição dos dados. Os templates do Django são geralmente escritos em arquivos HTML com tags e filtros especiais do Django.

Ao utilizarmos o Django, os templates que definem as páginas do projeto são escritos usando HTML, com algumas tags especiais do próprio framework que ajudam a inserir lógica de programação mais facilmente em suas páginas. Essas tags fazem parte da Django Template Language (DTL, linguagem de template do Django), como podemos ver na Figura 2.5. A DTL possui uma sintaxe diferente do HTML normal, seus elementos são colocados entre caracteres de chave. Por exemplo: para variáveis passadas como contexto aos templates são usadas chaves duplas para chamá-las, como em: "variável"; e para indicar que uma template é extensão de outra é usado chave e a caractere de porcentagem com a tag *extends*.

```
{% load static %}
           <!doctype html>
           <html lang="pt-br">
             <head>
               <meta charset="utf-8">
               <meta name="viewport" content="width=device-width, initial-scale=1">
               <title> {% block title %} Programa de Gerenciamento de Riscos {% endblock %}</title>
               klink href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" re
               <link href='{% static 'css/pgr.css' %}' rel='stylesheet'>
             <div class="container-fluid d-flex flex-column h-90 p-0">
               {% include 'includes/topo.html' %}
{% include 'includes/navbar.html' %}
               <div class="body-card">
                  {% block content %}
                  {% endblock content %}
                </div>
                  {% include 'includes/footer.html' %}
                <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.mi</pre>
                 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.j</pre>
                  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script</pre>
             </div>
             </body>
FONTE: Elaborado pelo autor
             <script type="text/javascript">
                $(document).ready(function () {
                    $('.navbar-light .dmenu').hover(function () {
                         $(this).find('.sm-menu').first().stop(true, true).slideDown(150);
                         $(this).find('.sm-menu').first().stop(true, true).slideUp(105)
```

**Figura 2.5** – Estrutura HTML com tags do django entre chaves

O conceito de herança de templates permite reutilizar partes comuns do layout em vários arquivos de template. Ao estender um template, é possível incluir o conteúdo do template pai para o filho, evitando duplicação de código e facilitando a manutenção. "{ % extends 'base.html' % }": Indica que o arquivo de template atual estende do template chamado "base.html".

"{ % block content % }": Esse bloco define um espaço reservado chamado "content"no template. Nele pode ser preenchido com conteúdo específico ou nos templates que o estendem. O conteúdo dentro desse bloco será renderizado no template pai.

"{ % csrf\_token % }": Essa linha insere um campo de token CSRF (Cross-Site Request Forgery) no formulário. O token CSRF é uma medida de segurança que protege contra ataques CSRF. Ao enviar o formulário, o token é verificado para ga-

rantir que a solicitação seja legítima e originada do próprio site, protegendo contra a execução de ações não autorizadas por terceiros. O Django utiliza um sistema de roteamento para direcionar as requisições HTTP para a view adequada. Esse roteamento é definido no arquivo urls.py do projeto Django. Nele, são mapeados as URLs para as views correspondentes, usando expressões regulares e nomes de namespaces.

Quando uma requisição HTTP chega no servidor Django, o roteador examina a URL solicitada e a compara com as definições presentes no arquivo urls.py. Uma vez que o roteador encontra uma correspondência, ele encaminha a requisição para a view associada. Essa view recebe a requisição encaminhada pelo roteador e processa os dados para gerar uma resposta. A view pode interagir com o componente Model para buscar ou modificar dados no banco de dados, e em seguida, ela passa esses dados para o Template para renderização.

O componente Template é responsável por renderizar a resposta HTML que será enviada de volta ao cliente. A view passa os dados necessários para o template e o template usa esses dados para construir a página final. O resultado é um arquivo HTML que pode conter variáveis, estruturas de controle e outros elementos definidos pela linguagem de template do Django.

Após a renderização do template, a view recebe o arquivo HTML resultante e cria uma resposta HTTP. Essa resposta contém o conteúdo da página gerada e outros detalhes, como cabeçalhos HTTP, status de resposta e outros metadados relevantes. A resposta é então enviada de volta ao cliente que fez a requisição. Além disso, é importante mencionar que o Django também suporta outros tipos de resposta, como JSON, XML, arquivos de download, entre outros. Isso permite que as views retornem diferentes tipos de dados, dependendo das necessidades da aplicação.

O Django possui um sistema de formulários que facilita a validação e o processamento de dados enviados pelo usuário. Os formulários no Django são criados usando classes, chamadas de Forms. Eles fornecem uma maneira conveniente de definir campos, validações e mensagens de erro. Os formulários podem ser renderizados automaticamente em HTML e também podem lidar com a validação de dados enviados pelo usuário.

O Django oferece o conceito de ModelForms, formulários específicos para um modelo Django. Um ModelForm é uma representação de um modelo do Django como um formulário. Ele é criado automaticamente a partir do modelo, gerando os campos correspondentes ao modelo, validações e mensagens de erro. Model-Forms simplificam o processo de criação e manipulação de formulários relacionados a modelos.

O Django fornece um mapeamento objeto-relacional que permite interagir com o banco de dados por meio de objetos Python, sem a necessidade de escrever consultas SQL manualmente. Os modelos do Django são usados como uma camada de abstração sobre as tabelas do banco de dados, permitindo a definição de campos, relacionamentos e métodos de acesso aos dados. O ORM realiza a tradução entre os objetos Python e as operações de banco de dados, vide Figura 2.6.

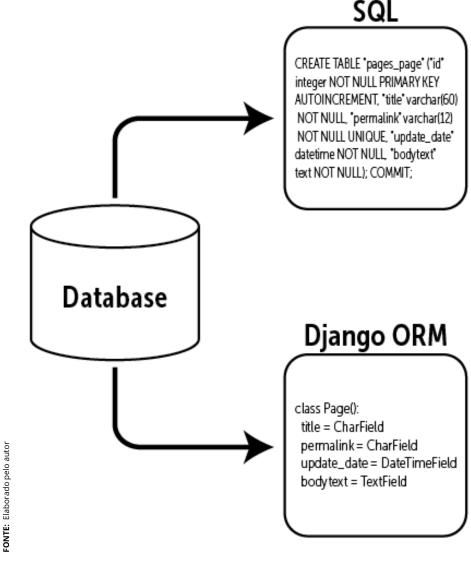


Figura 2.6 – Abstração de queries no Django

No Django, as views podem ser implementadas como funções ou classes. No caso das views baseadas em classes, o Django fornece um conjunto de classes genéricas que podem ser estendidas para simplificar a implementação de funcionalidades comuns. Essas classes genéricas fornecem funcionalidades pré-definidas, como listagem de objetos, criação, edição e exclusão. Ao estender essas classes, é possível personalizar o comportamento das views para atender às necessidades específicas da aplicação.

Algumas das filosofias deste framework são: desenvolvimento rápido, pouco código, design limpo, elementos independentes com acoplação fraca, e a aplicação do conceito Don't Repeat Yourself (DRY, não se repita)(CURTIN, 2016). Ao aderir ao princípio DRY podemos criar um código mais limpo, modular e fácil de manter. Isso resulta em um desenvolvimento mais eficiente e evita erros causados pela duplicação de lógica ou informação.

Os modelos permitem definir a estrutura do banco de dados e mapeá-la para classes Python. Ao usar modelos, você pode definir uma vez a estrutura dos dados e utilizá-la em várias partes do seu aplicativo. As views são responsáveis por processar as requisições HTTP e retornar as respostas adequadas, ou seja, podemos escrever views genéricas e reutilizáveis que podem ser usadas em diferentes partes do aplicativo, evitando a duplicação de código. O sistema de templates permite separar a lógica de apresentação do código Python. Com os templates, é possível definir a aparência das páginas web de forma reutilizável, evitando a repetição de código HTML.

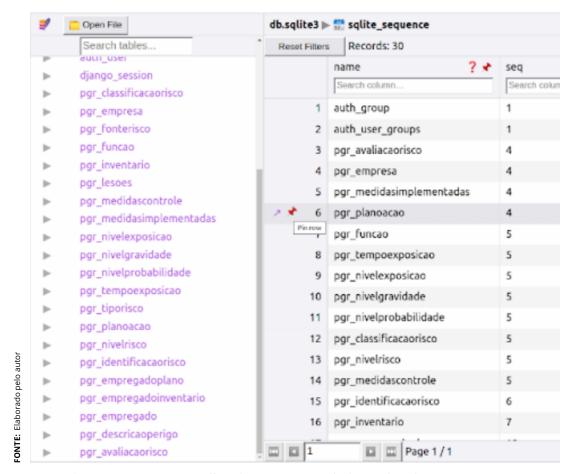
### 2.3.2.3 **SQLite**

O SQLite é um sistema de gerenciamento de banco de dados relacional embutido, ou seja, é um banco de dados que é incorporado diretamente em um aplicativo e não requer a instalação de um servidor separado. Ele foi desenvolvido por D. Richard Hipp em 2000 e é amplamente utilizado em uma variedade de aplicativos, desde dispositivos móveis até aplicativos de desktop.

Ele armazena todo o banco de dados em um único arquivo, tornando-o fácil de transportar e compartilhar. Foi projetado para ser compacto e eficiente em termos de recursos. Ele possui um pequeno uso de memória e consome poucos recursos do sistema, o que o torna ideal para dispositivos com recursos limitados, como smartphones e dispositivos IoT. Além disso, o SQLite é conhecido por sua velocidade e desempenho rápido em operações de leitura. Oferece suporte a transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade), garantindo que as operações sejam executadas de forma segura e confiável. Isso é especialmente importante em aplicações que exigem integridade e consistência nos dados. Diferente de outros sistemas de gerenciamento de banco de dados, o SQ-Lite não requer uma configuração complexa ou administração contínua. Ele pode ser facilmente incorporado a uma aplicação sem a necessidade de configurações ou ajustes adicionais.

O SQLite não é recomendado para aplicações que exigem alta escalabilidade ou lidam com um grande volume de usuários simultâneos. Ele não é tão adequado para ambientes com várias gravações concorrentes intensas, pois a escrita simultânea pode diminuir o desempenho. Ao contrário dos bancos de dados clienteservidor, o SQLite não possui suporte nativo para acesso simultâneo de vários processos ou threads. Embora seja possível acessar um arquivo de banco de dados SQLite a partir de vários processos ou threads, é necessário tomar precauções adicionais para evitar conflitos de acesso concorrente. Como ele é incorporado, não oferece os mesmos recursos de gerenciamento avançados encontrados em sistemas de banco de dados mais robustos. Recursos como *backup* online, replicação e gerenciamento de usuários são limitados ou não estão disponíveis no SQLite. Além disso, possui um conjunto de tipos de dados mais limitado em comparação com outros sistemas de banco de dados, o que pode ser uma desvantagem em determinados casos. Por exemplo, não há suporte nativo para campos de data/hora ou campos binários grandes (BLOBs) diretamente no SQLite.

Este banco foi escolhido pois atende aos requisitos básicos e suporta os modelos de dados, apresentando suas tabelas na Figura 2.7.



**Figura 2.7** – Arquivo db.sqlite3 com as tabelas utilizadas no projeto

#### 2.3.3 Ferramentas de desenvolvimento

Ao combinar algumas tecnologias, é possível criar um sistema de gerenciamento de riscos ocupacionais completo, desde o acompanhamento e atribuição de tarefas no ClickUp, controle de versão e colaboração no GitHub, empacotamento e distribuição com Docker, até o desenvolvimento da aplicação web utilizando o Django. Essas ferramentas fornecem uma base sólida e flexível para o desenvolvimento de um programa de gerenciamento de riscos ocupacionais confiável e de alto desempenho.

## 2.3.3.1 ClickUp

O ClickUp é uma ferramenta de gerenciamento de projetos e tarefas que oferece uma variedade de recursos para facilitar a organização e colaboração de equipes. Possui uma interface intuitiva e flexível, que permite a criação e atribuição de tarefas, o acompanhamento do progresso, a colaboração em equipe, a integração com outras ferramentas, a automação de tarefas, além de fornecer relatórios e análises para auxiliar na tomada de decisões. Com recursos como esses, o ClickUp se destaca como uma solução eficaz para melhorar a produtividade e o gerenciamento de projetos. O ClickUp permite organizar e acompanhar as atividades relacionadas às atividades de desenvolvimento do programa de gerenciamento de riscos ocupacionais. Ele oferece recursos avançados, como atribuição de tarefas, prazos, acompanhamento de progresso e integração com outras ferramentas.

#### 2.3.3.2 GitHub

O GitHub é uma plataforma de hospedagem de código-fonte baseada em nuvem, que utiliza o sistema de controle de versão Git. Ele fornece recursos para colaboração, gerenciamento de projetos e controle de código. O Git é um sistema distribuído, e cada desenvolvedor tem uma cópia do repositório em seu próprio ambiente de trabalho. Isso permite que desenvolvedores trabalhem offline e façam alterações independentes antes de sincronizá-las com o repositório central no GitHub.

O versionamento de código é um processo importante na gestão de projetos de desenvolvimento de software, permitindo o controle das alterações feitas ao longo do tempo e envolvem:

Criação de um repositório do projeto para armazenar o código do trabalho e acompanhar as alterações.

Clonagem do repositório. Isso permite que possa trabalhar no código sem afetar diretamente o repositório principal.

Branches (ramificações), que é possível criar cópias isoladas do código para trabalhar em novos recursos ou corrigir bugs sem afetar o código principal. Isso facilita a colaboração em equipe, pois cada desenvolvedor pode trabalhar em sua própria ramificação e mesclar as alterações quando estiverem prontas.

Commits são as unidades lógicas de alterações do código. Ao realizar os commits, são salvos as alterações de código em um determinado momento. Cada commit tem uma mensagem descritiva que explica o que foi alterado.

No Pull Requests (solicitações de Pull), ao terminar de trabalhar em uma ramificação e o desenvolvedor quiser mesclar suas alterações de volta para o ramo principal (geralmente chamado de branch "master"ou "main"), é aberto uma solicitação de pull. Isso notificará os membros da equipe sobre as alterações e permitirá que eles revisem o código antes de fazer a mesclagem final.

O Merge (Mesclagem) é realizado após revisar e aprovar as alterações na solicitação de pull, onde mesclam as alterações com o ramo principal. Isso adiciona as alterações ao código principal do projeto.

Atualização (Pull ou Fetch): Para manter o repositório local sincronizado com o repositório remoto no GitHub, pode atualizar o código usando os comandos "pull"ou "fetch". Isso garantirá que você tenha as últimas alterações feitas por outros membros da equipe.

#### 2.3.3.3 Visual Studio Code

O Visual Studio Code é um ambiente de desenvolvimento integrado (IDE) altamente popular entre os desenvolvedores de software. Com interface intuitiva e recursos poderosos, oferece uma experiência de programação eficiente e produtiva, sendo a integração com o Docker e o GitHub um dos pontos fortes dessa ferramenta.

Com relação ao Docker, sua extensão dedicada permite criar, gerenciar e depurar containers Docker diretamente na IDE. Isso simplifica o processo de desenvolvimento de aplicativos em contêineres, pois os desenvolvedores podem criar e testar seus ambientes de desenvolvimento em contêineres isolados com facilidade. Além disso, o Visual Studio Code também oferece recursos para a construção de imagens Docker, execução de comandos dentro dos contêineres e até mesmo depuração de aplicativos em execução no Docker.

No que diz respeito à integração com o GitHub, ele fornece uma extensão que permite aos desenvolvedores interagir com seus repositórios do GitHub diretamente na interface do IDE. Com essa integração, é possível clonar repositórios, gerenciar branches, fazer commits e push de alterações, revisar e mesclar pull requests, além de receber notificações em tempo real sobre atividades nos seus projetos do GitHub. Essa integração aprimora a colaboração entre os desenvolvedores e simplifica o processo de controle de versão e gerenciamento de código-fonte, tornando o trabalho em equipe mais eficiente e fluido.

O Visual Studio Code é uma ferramenta valorizada por desenvolvedores, e sua integração com o Docker e o GitHub proporciona recursos adicionais que tornam o desenvolvimento de software mais ágil, organizado e colaborativo. Essa combinação é especialmente relevante para a monografia em questão, pois evidencia o uso de ferramentas eficientes, direcionando o sucesso do projeto de pesquisa e desenvolvimento em questão.

#### 2.3.3.4 Docker

O Docker é uma plataforma de virtualização que permite empacotar o sistema e suas dependências em um contêiner. Isso oferece vantagens, como a portabilidade do sistema, facilitando a implantação em diferentes ambientes de desenvolvimento e produção. Além disso, o Docker garante a consistência do ambiente, evitando problemas de compatibilidade entre diferentes sistemas operacionais e configurações de software. Com o Docker, é possível criar um ambiente isolado e replicável para o sistema, simplificando o processo de implantação e facilitando a colaboração entre desenvolvedores.

As imagens Docker são o ponto de partida para criar contêineres. Elas são compostas por várias camadas de arquivos somente leitura, empacotados juntos para formar um ambiente completo. As imagens são armazenadas em repositórios na web, e podem ser compartilhadas para serem reutilizadas por outros desenvolvedores. HYKES (2013), fundador do Docker, descreve a virtualização de ambiente como "uma tecnologia que permite que você empacote um aplicativo

com todas as suas dependências em uma unidade padronizada chamada contêiner"(HYKES, 2013).

Com essa alternativa de virtualização, o kernel da máquina hospedeira é compartilhado com a máquina virtualizada ou o software em operação, permitindo ao desenvolvedor agregar bibliotecas e outras dependências ao software de modo que haja uma menor perda de desempenho quando comparado com a virtualização completa do hardware de um servidor, Figura 2.8.

**Containerized Applications** 

# App App Docker Host Operating System Infrastructure

**Figura 2.8** – Conteinerização de aplicações com Docker

Quando o desenvolvimento requer muitas imagens ou containers, pode ser usado o Docker Compose que é uma ferramenta usada para definir e executar aplicativos de vários contêineres do Docker. Nele, um arquivo yaml é usado para configurar os serviços do aplicativo. Em seguida, se cria e inicia todos os serviços com base na sua configuração executando um comando no terminal.

Um arquivo do Docker permite o gerenciamento de todas as fases de um ciclo de vida de um serviço definido: início, interrupção e recompilação de serviços, exibição do status do serviço e streaming de log.

# Capítulo 3

## **DESENVOLVIMENTO**

A Metodologia ágil é uma abordagem de desenvolvimento de projetos que valoriza a flexibilidade, colaboração e adaptação contínua. Em contraste com métodos tradicionais, que seguem uma abordagem linear e sequencial, elas promovem a entrega de valor de forma incremental e iterativa. As metodologias ágeis mais conhecidas são o Scrum, o Kanban e o Lean. Cada uma delas possui suas próprias características e abordagens, mas todas compartilham os mesmos princípios fundamentais, o foco no planejamento de tarefas das pessoas, na colaboração, na entrega de valor e na melhoria contínua. Uma de suas principais vantagens é a capacidade de lidar com a incerteza e a volatilidade dos projetos, permitindo que a equipe se adapte rapidamente às mudanças de requisitos e prioridades. Isso resulta em uma maior flexibilidade, eficiência e qualidade do produto final.

As etapas do projeto foram divididas em 3(três), sendo elas intituladas: Planejamento, Back-end e Front-end. Durante o processo foi necessário que essas etapas tivessem uma interação entre si de acordo com a necessidade.

A etapa de planejamento do projeto tinham 3 (três) *sprints*, sendo a primeira, responsável pelo versionamento do código no github e configuração do ambiente do ambiente de desenvolvimento, a segunda, o levantamento de requisitos, e a terceira, a modelagem do banco de dados e prototipagem do sistema.

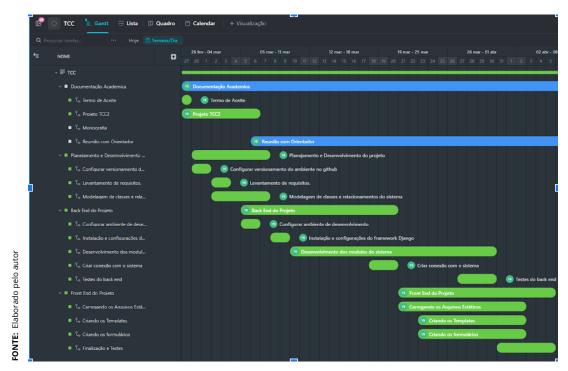


Figura 3.1 - Esquema de sprints utilizando o modelo de gráfico de Gantt

Na etapa de *Frontend* foram 3 (três) *sprints*, sendo elas, uma continuação das *sprints* anteriores, a criação dos templates e dos formulários com base nos modelos, finalizaram a sprint 2 com o pleno funcionamento da API, e para finalizar, foi o momento de desenvolver o *frontend* da aplicação.

Na etapa de *Backend* foram 4 (quatro) *sprints*, onde iniciei de fato o desenvolvimento de código, nas *sprints* 1 e 2 foram, configuração do ambiente utilizando o docker; a partir da *sprint* 3 (três) foi realizado o desenvolvimento das aplicações do sistema, onde o ecossistema *backend* foi sendo parametrizado e sincronizado para rodar o projeto; na *sprint* 4 (quatro) o foco foi trabalhar nas funcionalidades, definir as Models, as Rotas e criar as Views de acordo com os requisitos pré-estabelecidos para o sistema;

# 3.1 Planejamento e versionamento do projeto

#### 3.1.1 Gerenciador de tarefas

Como plataforma para o gerenciamento de tarefas foi adotado o ClickUp (https://app.clickup.com/), que utiliza quadros Kanban e listas para visualizar o andamento de *sprints* e acompanhar o progresso das tarefas durante cada sprint. Além disso, o ClickUp oferece recursos de colaboração em equipe, comentários e atribuições de tarefas, que são essenciais para a comunicação e colaboração ágeis.

O modelo de *sprints* foi importante para seguir o cronograma e as etapas préestabelecidas de forma que o desenvolvimento obteve um andamento dentro do programado. O projeto foi dividido em etapas, que foram segmentadas em *sprints*, com data início e previsão de conclusão, e o acompanhamento pelo orientador do projeto até a sua conclusão.

## 3.1.2 Diagrama de tarefas

A definição do escopo do projeto foram efetuadas nesta fase com a identificação dos objetivos, problemática e justificativas do projeto, juntamente com a listagem das tarefas e as entregas esperadas do projeto, sendo o prazo máximo de 4 meses para conclusão do projeto e para cada tarefa foi atribuído um prazo de acordo com a demanda e complexidade da tarefa variando de 3 a 15 dias.

## 3.1.3 Versionamento de código

Para iniciar foi configurado o Github para o versionamento e *backup* do código fonte do software. Foi criado um repositório online, Figura 3.4, e disponibilizado o link com todo o código do projeto. 3.4

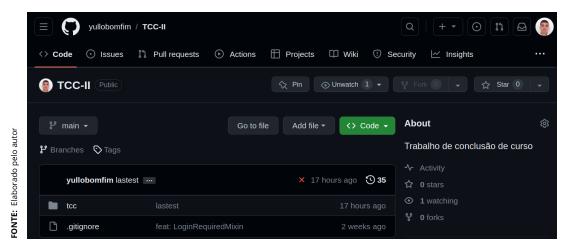


Figura 3.2 – Repositório no Github

Através de sucessivos "commits" foi possível analisar o progresso e também retornar a versões anteriores após a alterações ou erros durante o desenvolvimento do projeto, conforme Figura 3.4.

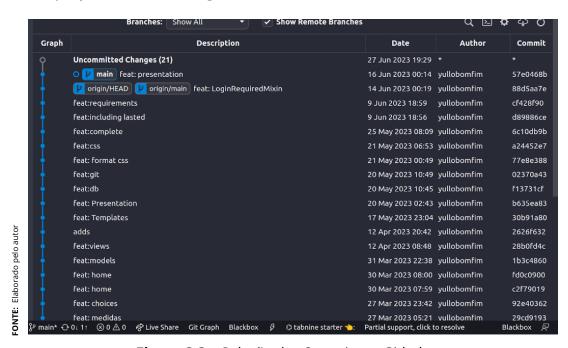


Figura 3.3 - Relação dos Commits no Github

## 3.2 Levantamento de requisitos

O levantamento de requisitos de um software é fundamental para o seu desenvolvimento, foram identificadas as principais classes e os relacionamentos necessários para dar seguimento no processo de modelagem do sistema, além disso, foi possível identificar os CRUD necessários para cada classe apontada.

#### 3.2.1 Necessidades dos clientes

Para identificar as necessidades do cliente, foi realizado uma pesquisa na Norma Regulamentadora (NR-01), que apresenta os requisitos para a implementação de um Programa de Gerenciamento de Riscos (PGR), a fim de compreender as necessidades dos usuários em relação ao sistema a ser desenvolvido. Essa norma abrange uma variedade de aspectos, como prevenção de acidentes, controle de riscos, saúde ocupacional e equipamentos de proteção individual.

Ao consultar essa legislação, foi possível identificar as responsabilidades da empresa e os requisitos legais que devem ser considerados no sistema a ser desenvolvido, como a necessidade de realização de avaliações de riscos, a implementação de medidas de prevenção, entre outros aspectos relevantes.

"1.5.3 Responsabilidades 1.5.3.1. A organização deve implementar, por estabelecimento, o gerenciamento de riscos ocupacionais em suas atividades. 1.5.3.1.2 O PGR pode ser atendido por sistemas de gestão, desde que estes cumpram as exigências previstas nesta NR e em dispositivos legais de segurança e saúde no trabalho. 1.5.3.1.1.1 A critério da organização, o PGR pode ser implementado por unidade operacional, setor ou atividade. 1.5.3.2 A organização deve: a) evitar os riscos ocupacionais que possam ser originados no trabalho; b) identificar os perigos e possíveis lesões ou agravos à saúde; c) avaliar os riscos ocupacionais indicando o nível de risco; d) classificar os riscos ocupacionais para determinar a necessidade de adoção de medidas de prevenção; e) implementar medidas de prevenção, de acordo com a classificação de risco e na ordem de prioridade estabelecida na alínea "g" do subitem 1.4.1; e f) acompanhar o controle dos riscos ocupacionais."(BRASIL, 2021).

Estes dados servem como base para o planejamento, o projeto e a implementação do sistema, permitindo que sejam tomadas as medidas adequadas para atender aos requisitos legais e às necessidades dos clientes. Dessa forma, a pesquisa na legislação de segurança do trabalho e a documentação dos requisitos foram etapas importantes para compreender as necessidades e garantir o desenvolvimento de um sistema em conformidade com as normas e regulamentos aplicáveis, promovendo a segurança e a saúde dos trabalhadores.

## 3.2.2 Prototipagem do sistema

A prototipagem utilizando o framework Django auxilia no desenvolvimento de aplicativos web pois é possível criar modelos de dados que representam os conceitos da gestão de segurança do trabalho. Além disso, o framework oferece recursos para a criação de visualizações e templates. Durante a prototipagem, é possível testar diferentes funcionalidades e fluxos de trabalho, aproveitando os recursos adicionais do Django, como autenticação de usuários. Essa abordagem facilita o desenvolvimento e a criação de um sistema eficiente para a gestão de segurança do trabalho.

#### ## Atores

#### ### Administrador

- Cadastrar um empregado
- Cadastrar uma empresa
- Cadastrar um inventário de riscos
- Cadastrar um plano de ação
- Editar um empregado
- Editar uma empresa
- Editar um inventário de riscos
- Editar um plano de ação
- Deletar um empregado
- Deletar uma empresa
- Deletar um inventário de riscos
- Deletar um plano de ação

#### **### Empregados**

- Visualizar a sua empresa
- Visualizar o seu inventário de riscos
- Visualizar o seu plano de ação

**Figura 3.4** – *Levantamento de Requisitos* 

FONTE: Elaborado pelo autor

A prototipação de uma aplicação com o Django envolve um processo que vai desde o planejamento até o desenvolvimento do protótipo. Inicialmente, é necessário definir os requisitos básicos e estruturar o projeto. Em seguida, ocorre a modelagem de dados, onde são criadas as classes Python que representam as tabelas do banco de dados. As rotas e URLs são definidas para direcionar as solicitações dos clientes para as views apropriadas, que são funções ou classes Python que processam os dados e retornam respostas em HTML usando templates. O processo de prototipagem é iterativo, permitindo a adição de novas funcionalidades e refinamento do aplicativo com base no feedback recebido. Ao concluir o protótipo, é possível avaliar os resultados, validar ideias e obter feedback dos usuários. O protótipo do Django fornece uma versão inicial funcional do aplicativo, permitindo testar a viabilidade e a funcionalidade básica do conceito. À medida que o desenvolvimento avança, é possível expandir e aprimorar o aplicativo adicionando mais recursos e detalhes finais.

# 3.3 Modelagem do sistema

## 3.3.1 Metodologia de modelagem

A modelagem de sistemas utilizando a UML (Unified Modeling Language) é uma abordagem amplamente utilizada para representar visualmente os diferentes aspectos de um sistema, incluindo sua estrutura, comportamento e interações. A UML oferece uma variedade de diagramas que podem ser usados para modelar diferentes aspectos do sistema, como o diagramas de classes.

No contexto do Django, a modelagem do banco de dados envolve a criação de classes de modelo que representam as tabelas e relacionamentos do banco de dados. O Django segue uma abordagem ORM (Object-Relational Mapping), onde as classes de modelo são mapeadas para tabelas no banco de dados e os objetos dessas classes representam as linhas nas tabelas. Cada classe de modelo é definida como uma subclasse da classe models. Model fornecida pelo Django. Os atributos da classe representam os campos da tabela no banco de dados, e o Django fornece uma variedade de tipos de campo, como CharField, IntegerField, ForeignKey, vide figura 3.5

```
# CADASTRO EMPREGADO

class Empregado(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)

empresa=models.ForeignKey(Empresa, verbose_name=("empresa"), on_delete=models.CASCADE)

funcao = models.ForeignKey(Funcao, verbose_name=("funcao"), on_delete=models.CASCADE)

def __str__(self):
    return f"{self.user}"

class Meta:
    verbose_name = "empregado"
```

Figura 3.5 - Modelo com 3 instâncias de objeto

•

## 3.3.2 Diagrama de Classes

O diagrama de classes é um dos diagramas da UML que pode ser utilizado para representar a estrutura e as relações entre as classes em um sistema. No contexto do Django, o diagrama de classes pode ser usado para representar as classes de modelo que definem as tabelas e os relacionamentos do banco de dados. No diagrama de classes do Django, cada classe de modelo é representada como uma caixa retangular com o nome da classe. Os atributos da classe, correspondem aos campos da tabela no banco e são listados dentro da caixa retangular, juntamente com os tipos de dados, conforme diagrama da Figura 3.6

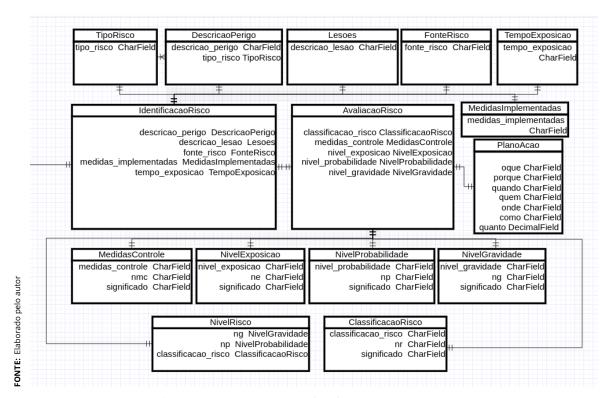


Figura 3.6 - Diagrama de Classes para o PGR

Criar o diagrama de classes para representar as entidades, atributos e relacionamentos, ajuda a visualizar a estrutura do sistema e definir as classes que serão implementadas durante a fase de implementação. Nesta fase foi criado uma classe para cada tabela a ser modelada, com uma subclasse da classe models. Model . Definindo os campos como atributos da classe usando os tipos de campo fornecidos pelo Django. Além disso, foi possível definir relacionamentos entre tabelas usando campos *foreign key*. Como um Protótipo tem por objetivo explorar aspectos críticos dos requisitos de um produto, implementando de forma rápida um subconjunto de funcionalidades deste produto, ele é indicado para estudar as alternativas de interface do usuário; problemas de comunicação com outros produtos; e a viabilidade de atendimento dos requisitos de desempenho.

Nesse sentido, as técnicas utilizadas na elaboração do protótipo com *framework* django foram com a interface de empregados com a criação de relatórios visuais. Alguns dos benefícios são a redução dos riscos na construção do sistema, pois o usuário chave já verificou o que o analista captou nos requisitos do produto. Para ter sucesso na elaboração dos protótipos é necessário o entendimento dos objetivos por todos os interessados no projeto, a focalização em áreas menos compreendidas e a rapidez na construção.

# 3.4 Implementação

Durante o desenvolvimento foi utilizado o editor de códigos Visual Studio Code. Também foi o momento de instalar o Docker, para posteriormente utilizá-lo através das linhas de comando do terminal no sistema operacional. Outra configuração necessária, foi autenticar os dados de login do Docker e do Github no Visual Studio Code, e assim sincronizar os códigos diretamente pela ferramenta. Após essas configurações, foi criado um diretório para guardar os arquivos do projeto, então foi possível passar para a próxima demanda listada no Click Up.

#### 3.4.1 Ambiente de desenvolvimento

Para codificar as primeiras linhas de código, com o Docker, através do arquivo Dockerfile, é possível passar comandos e atribuir serviços que serão necessários para a configuração e compilação do sistema, como é mostrado na Figura 3.7. A



**Figura 3.7** – Configuração do ambiente de desenvolvimento

partir desse código, foi possível realizar a configuração da imagem necessária para usar o Python na versão 3.8, juntamente com a instalação dos pacotes adicionados ao requirements.txt. Nesse arquivo também foi possível escrever comandos de atualização dos pacotes *pip* e *apt-get* providenciando um sistema com as bibliotecas mais atualizadas na criação do *container*. Nesta configuração foi realizada a migração inicial necessária para criar as tabelas padrão do django.

Após a configuração do Dockerfile, foi iniciado a escrita do arquivo docker-compose.yml, conforme Figura 3.8.

```
docker-compose.yml
         version: '3'
          services:
            web:
              container name: pgr
              build: .
              command: 'python manage.py runserver 0.0.0.0:8000'
              volumes:
                - '.:/tcc'
                - './db.sqlite3:/tcc/db.sqlite3'
              ports:
                - '8000:8000'
            db:
FONTE: Elaborado pelo autor
              container name: pgr db
              image: alpine
              volumes:
                 - './db.sqlite3:/tcc/db.sqlite3'
              command: tail -f /dev/null
    19
```

**Figura 3.8** – *Docker-compose.yml* 

Essas configurações definem dois serviços em um ambiente Docker. O primeiro serviço é um servidor web que executa um servidor Django usando o arquivo Dockerfile presente no diretório atual. Ele mapeia o diretório atual e o arquivo "db.sqlite3" para o contêiner, permitindo que o código e o banco de dados sejam compartilhados entre o host e o contêiner. A porta 8000 do contêiner é exposta para o host, permitindo o acesso ao servidor web.

O segundo serviço é um banco de dados, que usa a imagem Alpine Linux e mapeia o arquivo "db.sqlite3" para o contêiner. O comando "tail -f /dev/null" é usado para manter o contêiner em execução.

Após configurado, já é possível executar o *build* dos *containers* e com isso manter o software ativo, essa configuração só se faz necessária uma única vez, sendo assim podemos utilizar todo o ambiente configurado sempre que for necessário com apenas o comando, "docker-compose up -d", dentro do terminal para que o conjunto de arquivos e diretórios sejam carregados no *container* docker.

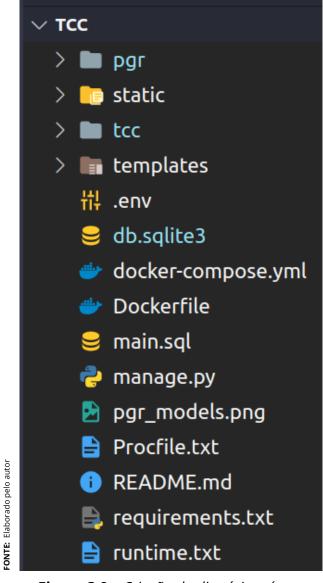


Figura 3.9 – Criação do diretório raíz

Foi especificado o diretório onde os arquivos serão armazenados, nesse caso, "tcc", e também um diretório chamado "tcc" foi criado para abrigar o core do projeto.

Arquivo **manage.py**: Este é um arquivo Python localizado no diretório raiz do projeto. Ele é usado para executar as tarefas de gerenciamento do projeto, como iniciar o servidor, executar migrações, criar superusuários, entre outras.

Diretórios do app: Dentro do diretório do projeto, podemos criar um ou mais aplicativos Django. Cada aplicativo tem seu próprio diretório. O Django segue a estrutura de um aplicativo como um pacote Python, então podemos encontrar um arquivo **\_\_init\_\_.py** dentro do diretório pgr do aplicativo.

Arquivo **settings.py**: Este arquivo é encontrado dentro do diretório **tcc** do projeto e contém as configurações principais do Django, como informações do banco de dados, configurações de middleware, aplicativos instalados, etc.

Arquivo **urls.py**: Também localizado no diretório do projeto, o arquivo urls.py é responsável por definir as rotas URL do projeto. Nele, foi possível mapear as URLs para as views correspondentes.

Esses foram os principais diretórios e arquivos criados durante a instalação do Django. À medida que o projeto foi desenvolvido, outros diretórios e arquivos foram adicionados para armazenar modelos, visualizações, arquivos estáticos, entre outros. Acessar via web browser o endereço "http://localhost:8000/", onde o localhost referência a máquina local seguindo da porta 8000 previamente configurada no arquivo docker-compose.

Para acessar o software é necessário realizar o cadastro inicial de um usuário, e além disso, deve ser criado um Superusuário que será responsável pela Administração do Sistema. Para isso será executado o comando para criar o superusuário:

#### python manage.py createsuperuser

Será solicitado a fornecer um nome de usuário (username), fornecer um endereço de e-mail (email address) opcional e uma senha (password). Se todas as informações forem fornecidas corretamente, você receberá uma mensagem informando que o superusuário foi criado com sucesso. Agora é possível usar o usuário e a senha do Superusuário para acessar a área de administração do seu projeto Django.

Ao executar o servidor de desenvolvimento com o comando **python manage.py runserver** e acessar a URL:**http://localhost:8000/admin/**. deve fazer o login usando as credenciais do superusuário e você terá acesso às funcionalidades de administração do Django.

#### 3.4.2 Criando Models

Utilizando o Django foi possível construir toda a base de dados com as configurações que o framework oferece. Para a criação do banco de dados com suas tabelas e colunas contendo todos os relacionamentos durante a modelagem do banco. No Django, os modelos são classes que representam tabelas em um banco de dados. Eles definem a estrutura dos dados e as operações possíveis. Para criar um modelo, deve definir uma classe que herda da classe Model do Django, especificar os campos do modelo e executar as migrações para criar as tabelas correspondentes no banco de dados. Os modelos no Django facilitam o trabalho com bancos de dados, permitindo que se crie, atualize, exclua e consulte registros de forma eficiente através da API do Django ORM.

#### 3.4.3 Definindo URLs

As URLs no Django são configuradas no arquivo urls.py e são responsáveis por mapear as requisições dos usuários para funções ou classes de visualização. O uso das funções path ou re-path permite definir padrões de URL e associá-los às funções ou classes de visualização adequadas. Com o mapeamento correto das URLs, o Django encaminha as requisições dos usuários para as visualizações corretas, permitindo a construção da estrutura de navegação do aplicativo, conforme Figura 3.10.

```
urlpatterns = [
    path('', views.home, name='home'),
    path('cadastro/', views.cadastro, name='cadastro'),

path('empresa/', EmpresaListView.as_view(), name='empresa_list'),
    path('empresa/create/', EmpresaCreateView.as_view(), name='empresa_create'),
    path('empresa/update/<int:pk>/', EmpresaUpdateView.as_view(), name='empresa_update'),
    path('empresa/delete/<int:pk>/', EmpresaDeleteView.as_view(), name='empresa_delete'),

path('funcao/', FuncaoListView.as_view(), name='funcao_list'),
    path('funcao/create/', FuncaoCreateView.as_view(), name='funcao_update'),
    path('funcao/delete/<int:pk>/', FuncaoDeleteView.as_view(), name='funcao_delete'),

path('tiporisco/', TiporiscoListView.as_view(), name='tiporisco_list'),
    path('tiporisco/create/', TiporiscoCreateView.as_view(), name='tiporisco_create'),
    path('tiporisco/update/<int:pk>/', TiporiscoUpdateView.as_view(), name='tiporisco_create'),
    path('tiporisco/delete/<int:pk>/', TiporiscoDeleteView.as_view(), name='tiporisco_create'),
    path('tiporisco/delete/<int:pk>/', TiporiscoDeleteView.as_view(), name='tiporisco_delete'),
```

**Figura 3.10** – Padrão utilizado para definir as rotas para diferentes views

O código define as URLs (rotas) para diferentes visualizações. Primeiro, são importados os módulos necessários para a definição das URLs:

Em seguida, são importadas as visualizações relevantes do arquivo views.py.

Em seguida, são definidas as URLs por meio da lista urlpatterns:

- path(", views.home, name='home') define uma rota vazia que corresponde à função home no arquivo views.py. O nome da rota é definido como 'home', o que permite referenciá-la usando o nome em outros lugares do código.
- ▶ path('signup/', views.signup, name='signup') define uma rota chamada 'signup/', que corresponde à função signup no arquivo views.py. O nome da rota é definido como 'signup'. As próximas linhas seguem uma série de padrões semelhantes para diferentes visualizações, seguindo o padrão prefixo>/nome/:
- /empresa/... para as visualizações relacionadas à empresa.

- ► /funcao/... para as visualizações relacionadas à função.
- ► /tiporisco/... para as visualizações relacionadas ao tipo de risco.
- /descricaoperigo/... para as visualizações relacionadas à descrição de perigo.
- ► /empregado/... para as visualizações relacionadas ao empregado.
- ► /inventario/... para as visualizações relacionadas ao inventário.
- ▶ /planoacao/... para as visualizações relacionadas ao plano de ação.
- /identificacaorisco/... para as visualizações relacionadas à identificação de risco.
- /avaliacaorisco/... para as visualizações relacionadas à avaliação de risco.
- ► /empregadoinventario/... para as visualizações relacionadas ao empregado no inventário.
- ▶ /empregadoplano/... para as visualizações relacionadas ao empregado no plano. Em cada rota, são definidas as visualizações correspondentes usando as classes de visualização importadas anteriormente. Cada rota tem um nome associado que pode ser usado para referenciar a rota em outros lugares do código e mapear as solicitações do cliente para as visualizações correspondentes do aplicativo.

#### 3.4.4 Criando Views

As views no Django são responsáveis por processar as requisições dos usuários e retornar uma resposta. Elas podem ser definidas como funções ou classes, sendo que as classes de visualização baseadas em classe fornecem uma estrutura organizada para lidar com a lógica do aplicativo. Ao herdar de classes base como View ou ListView, você implementa métodos correspondentes aos tipos de requisições, processando os dados da requisição e retornando a resposta apropriada. As views são associadas às URLs do aplicativo e desempenham um papel fundamental na construção da funcionalidade e interação com o usuário, vistos na 3.11.

```
🤚 views.py > ...
 @login required
 def home(request):
      return render(request, 'home.html')
 def cadastro(request):
      if request.method == 'POST':
          form = UserCreationForm(request.POST)
          if form.is valid():
              form.save()
              messages.success(request, "Usuário cadastrado com Sucesso")
              return redirect('home')
      else:
          form = UserCreationForm()
      return render(request, 'registration/cadastro.html', {'form': form})
> class EmpregadoListView(LoginRequiredMixin, ListView): --
> class EmpregadoCreateView(LoginRequiredMixin, CreateView): --
> class EmpregadoUpdateView(LoginRequiredMixin, UpdateView): --
> class EmpregadoDeleteView(LoginRequiredMixin, DeleteView): --
> class EmpresaListView(LoginRequiredMixin, ListView): ...
> class EmpresaCreateView(LoginRequiredMixin, CreateView): --
> class EmpresaUpdateView(LoginRequiredMixin, UpdateView): --
> class EmpresaDeleteView(LoginRequiredMixin, DeleteView): ...
> class FuncaoListView(LoginRequiredMixin, ListView): --
> class FuncaoCreateView(LoginRequiredMixin, CreateView): ---
> class FuncaoUpdateView(LoginRequiredMixin, UpdateView): --
```

Figura 3.11 – Criando as views

O trecho de código acima define o CRUD para o modelo Empresa. Sendo as classes EmpresaListView, EmpresaCreateView, EmpresaUpdateView, EmpresaDeleteView:

O **mixin LoginRequiredMixin** garante que somente usuários autenticados possam acessar essa visualização.

**login\_url = reverse\_lazy('login')** especifica a URL de login para redirecionar o usuário caso ele não esteja autenticado.

model = Empresa define o modelo a ser utilizado como Empresa.

**context\_object\_name = 'empresas'** define o nome do objeto de contexto que será disponibilizado no template.

Em consulta ao **template\_name = 'empresa\_list.html'**, **'empresa\_create.html' 'empresa\_update.html'** , **'empresa\_delete.html'** são especificados o nome do template a serem renderizado para essa visualização.

Ao iniciar um novo projeto Django, no arquivo de configuração **settings.py** contém uma seção dedicada às configurações do banco de dados,vide Figura 3.12, onde é possível especificar o SQLite3 como banco de dados padrão. A base de dados seguiu exatamente como proposto nas etapas de modelagem de dados, uma vez concluída a criação da base de dados com seus devidos relacionamentos, tendo toda essa parte de estruturação e configuração completa, chegou o momento de desenvolver os apps do sistema.

```
# Database

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE_DIR / 'db.sqlite3',

}
}
```

Figura 3.12 – Configurando o banco de dados

## 3.4.5 Criando Templates

O template base de nossa aplicação é um modelo HTML usando a linguagem de template Django. Ele define a estrutura básica da página base e inclui elementos como cabeçalho, título, estilos CSS, scripts JavaScript e áreas de conteúdo. O template também faz uso do framework Bootstrap para criar um layout responsivo. Além disso, ele utiliza a funcionalidade de blocos do Django, permitindo que outros templates estendam e substituam partes específicas do código. O arquivo também inclui arquivos HTML adicionais, como o cabeçalho, barra de navegação e o rodapé, que podem ser reutilizados em diferentes páginas. No geral, o template serve como um ponto de partida para o desenvolvimento de páginas da web consistentes e dinâmicas.

```
class EmpregadoListView(LoginRequiredMixin, ListView):
          login url = reverse lazy('login')
         model = Empregado
         template name = 'empregado list.html'
         context object name = 'empregados'
     class EmpregadoCreateView(LoginRequiredMixin, CreateView):
          login url = reverse lazy('login')
         model = Empregado
         form class = EmpregadoForm
         template name = 'empregado create.html'
          success url = reverse lazy('empregado list')
     class EmpregadoUpdateView(LoginRequiredMixin, UpdateView):
          login url = reverse lazy('login')
         model = Empregado
          form class = EmpregadoForm
          template_name = 'empregado_update.html'
ONTE: Elaborado pelo autor
          success_url = reverse_lazy('empregado list')
     class EmpregadoDeleteView(LoginRequiredMixin, DeleteView):
          login url = reverse lazy('login')
         model = Empregado
         template name = 'empregado delete.html'
          success_url = reverse_lazy('empregado_list')
```

Figura 3.13 – Criando os templates

#### 3.4.6 Criando Formulários

Um *ModelForm* é uma classe do Django que é automaticamente gerada com base em um modelo existente. Ele fornece uma maneira conveniente de criar formulários que estão vinculados a um modelo específico. Aqui está um exemplo de criação utilizando os formulários vinculados aos modelos:

```
🥏 forms.py > ...
     class CadastroForm(UserCreationForm):
         username = forms.CharField(max length=50)
         class Meta:
             model = User
             fields = ' all '
     class EmpresaForm(ModelForm):
         class Meta:
             model= Empresa
             fields = ' all '
     class FuncaoForm(ModelForm):
         class Meta:
             model= Funcao
             fields = '_all '
:ONTE: Elaborado pelo autor
     class EmpregadoForm(ModelForm):
         class Meta:
             model= Empregado
              fields = ' all '
```

Figura 3.14 - Exemplo de utilização do ModelForm

Foi realizado testes de uso do sistema, a fim de contemplar todos os requisitos propostos no início do projeto e validar a versão 1.0 do sistema, após os testes o resultado foi satisfatório no que diz respeito a intenção da proposta da aplicação, por ser tratar de uma prototipação foi necessário ao decorrer dos testes alguns ajustes pontuais ou de ajustes para melhorar a experiência do usuário no que diz respeito a usabilidade da ferramenta desenvolvida.

É possível acessar os códigos do projeto bem como a documentação, as *sprints*, bem como o código do sistema através dos links disponibilizados na tabela a seguir.

**Tabela 3.1** – *Links referentes ao software desenvolvido.* 

| Serviços                               | Links  |
|--|--|
| REPOSITÓRIO GITHUB                     | https://github.com/yullobomfim/TCC-II                |
| PLANEJAMENTO DAS SPRINTS               | https://app.clickup.com/9007024432/v/li/900700267395 |
| DEMONSTRATIVO DO PAINEL ADMINISTRATIVO | https://youtu.be/XSWTSOAn9Is                         |
| DEMONSTRATIVO DO SISTEMA               | https://youtu.be/lAanCEzh11A                         |

Fonte: Autor(2023)

# Capítulo 4

## TRABALHOS CORRELATOS

A segurança do trabalho é uma parte crucial de uma empresa e requer organização dos processos envolvidos. Portanto, ter um software que auxilie nesse objetivo é essencial para ter o melhor resultado.

Existem alguns serviços web disponíveis e suas funcionalidades podem variar em cada empresa e, para se contratar, pode ser difícil escolher um que atenda a todos os requisitos. Entre os principais softwares disponíveis e que realizam funções semelhantes ao proposto neste trabalho temos: o Bem.care; SOC Software de Saúde e Segurança do Trabalho; On Safety; Software SGG; entre outros. Em consulta extraída dos domínios de seus sites podemos relacionar algumas de suas principais características.

#### 4.1 Bem.care

O Bem.care é um software de gestão de segurança do trabalho que oferece uma variedade de recursos para gerenciar e acompanhar a saúde e segurança dos colaboradores. Ele permite a criação de documentos, como programas de prevenção de riscos, controle de acidentes e treinamentos. Além disso, o Bem.care auxilia na análise de indicadores de saúde ocupacional e na gestão de exames médicos.O Site está disponível sob o domínio https://bem.care/ .

# 4.2 SOC Software de Saúde e Segurança do Trabalho

O SOC é uma solução abrangente de gestão de saúde e segurança do trabalho. Ele oferece recursos para a criação de documentos legais, como PPRA (Programa de Prevenção de Riscos Ambientais) e PCMSO (Programa de Controle Médico de Saúde Ocupacional), além de permitir o registro e análise de acidentes de trabalho, gestão de treinamentos e controle de EPIs (Equipamentos de Proteção Individual).O Site está disponível sob o domínio https://soc.com.br/ .

# 4.3 On Safety

On Safety: O On Safety é um software que auxilia na gestão de segurança do trabalho e saúde ocupacional. Ele permite a elaboração de documentos, como PPRA, PCMSO e LTCAT (Laudo Técnico das Condições Ambientais do Trabalho), além de oferecer recursos para o controle de exames médicos, gestão de riscos, monitoramento de EPIs e treinamentos. O Site está disponível sob o dominio https://onsafety.com.br/.

### 4.4 Software SGG

Software SGG: O Software SGG (Sistema de Gestão de Segurança) é uma solução que visa simplificar a gestão da segurança do trabalho. Ele oferece recursos para o controle de documentos legais, análise e gerenciamento de riscos, acompanhamento de acidentes de trabalho, gestão de EPIs e treinamentos. O SGG também permite a geração de relatórios e indicadores para auxiliar na tomada de decisões.O Site está disponível sob o domínio http://www.sgg.com.br/.

# 4.5 Comparativo dos Sistemas

Durante o desenvolvimento deste trabalho foi identificado que houve similaridade com algumas funcionalidades apresentadas no On Safety, SOC e SGG quanto a análise e gerenciamento de riscos, possibilidade de elaboração de documentos técnicos, em nosso caso, apenas o PGR.

Analisando outros documentos legais, como cadastramento de dados do PPRA, PCMSO, LTCAT e outros programas de ergonomia ou específicos segurança, os softwares SOC e SGG se destacaram possuindo maior possibilidade de geração de relatórios com base nos dados cadastrados na plataforma.

Já em contraposição, verificou-se que o software Bem. Care não possui funcionalidades similares com o projeto desenvolvido, já que esse buscou um foco maior na saúde ocupacional, e o principal foco de nosso trabalho ficou com o inventário de riscos e planejamento de suas ações.

**Tabela 4.1** – Resumo das principais funcionalidades e diferenças entre os softwares de gestão de segurança do trabalho mencionados:

| Software          | Principais Funcionalidades  | Diferenciais   | Site            |
|-------------------|---|--|-----------------|
| Bem.care          | - Criação de documentos<br>de prevenção de riscos,<br>controle de acidentes, trei-<br>namentos e indicadores de<br>saúde ocupacional.               | Enfoque na gestão<br>da saúde ocupacio-<br>nal.  | bem.care        |
| SOC Soft-<br>ware | - Criação de documentos legais (PPRA, PCMSO, entre outros), registro e análise de acidentes de trabalho, gestão de treinamentos e controle de EPIs. | - Ampla gama de re-<br>cursos para gerenci-<br>amen to de saúde e<br>segurança do traba-<br>lho. | soc.com.br      |
| On Safety         | - Criação de documentos<br>de prevenção de riscos,<br>controle de acidentes, trei-<br>namentos e indicadores de<br>saúde ocupacional.               | - Ênfase na gestão<br>de riscos ocupacio-<br>nais.   | onsafety.com.br |
| Software<br>SGG   | - Controle de documentos legais, análise e gerenciamento de riscos, acompanhamento de acidentes de trabalho, gestão de EPIs e treinamentos.         | - Foco na análise e<br>gerenciamento de<br>riscos.   | sgg.com.br      |

Fonte: Autor(2023)

Importante reforçar que cada software tem suas próprias características e funcionalidades específicas em seu tempo. Assim, recomenda-se consultar os sites indicados para obter informações mais detalhadas sobre cada software, incluindo suas características, recursos adicionais, suporte oferecido e possíveis variações nos preços e planos de assinatura.

Desta forma, é relevante avaliar as necessidades específicas da empresa para escolher o software que melhor atenda às demandas de gestão de segurança do trabalho. Já que os sistemas de dados estão cada vez mais digitais, é importante seguir essas mudanças. Portanto, investir em um software que auxilie as empresas a realizar a gestão de saúde e segurança do trabalho trará grandes ganhos para as empresas. Durante o desenvolvimento deste projeto, podemos destacar que a sua interface simples e objetiva apresentando aos usuários informações atualizadas e precisas que tornam o sistema proposto muito acessível a qualquer empregado.

## Capítulo 5

### **RESULTADOS**

Dentre os resultados do desenvolvimento deste sistema, podemos destacar àqueles que auxiliaram a atingir os objetivos do projeto. O uso das tecnologias web deste projeto permitem que informações que auxiliam no processo de decisão para as empresas e empregados estejam acessíveis a qualquer momento.

O sistema desenvolvido proporcionou às empresas acessar as legislações relacionadas à segurança do trabalho, promovendo o cumprimento das obrigações legais, conscientização dos trabalhadores e cultura de segurança no ambiente laboral. Permite ainda que as empresas monitorem o inventário de riscos de forma eficiente. Isso significa que as informações sobre exposição aos agentes nocivos em diferentes funções e locais de trabalho foram coletadas e analisadas de maneira adequada.

A combinação das ferramentas de planejamento e desenvolvimento utilizadas trouxeram uma série de benefícios significativos para o gerenciamento deste projeto. Essas ferramentas permiram uma gestão eficiente, controle de versão e colaboração eficazes, interação dinâmica com o usuário e desenvolvimento ágil. Além disso, o Docker ofereceu padronização e portabilidade de ambientes , enquanto o Django, com Python, proporcionam um ambiente robusto e produtivo. Com essa combinação, equipes podem aumentar a produtividade, melhorar a qualidade dos projetos e oferecer uma experiência aprimorada aos usuários finais.

A apresentação dos dados sobre a identificação dos perigos no Programa de Gerenciamento de Riscos PGR de forma rápida e clara, facilita na identificação e análise dos diferentes perigos presentes em um ambiente de trabalho, vide Figura 5.1. Ela consiste em uma lista de perigos potenciais, suas fontes e as possíveis consequências associadas a cada um deles. Além disso, os dados sugerem medidas preventivas que podem ser adotadas para mitigar ou eliminar esses riscos. Ao utiliza-los, as empresas podem identificar de forma antecipada os perigos existentes e implementar estratégias eficazes para garantir a segurança e a saúde dos trabalhadores, prevenindo acidentes e danos relacionados ao trabalho.



Figura 5.1 – Identificação dos perigos

A avaliação dos riscos do PGR permite identificar, classificar e avaliar os riscos relacionados a atividades, projetos ou processos dentro de uma empresa, vide Figura 5.2. Ela consiste em uma matriz que combina a probabilidade de ocorrência do risco com o impacto resultante, permitindo a atribuição de um nível de risco para cada situação analisada. Com base nessa classificação, as medidas preventivas e corretivas podem ser implementadas de forma prioritária, concentrando recursos e esforços nas áreas de maior risco. Além disso, a avaliação dos riscos facilita a comunicação e o entendimento das partes interessadas sobre os riscos envolvidos, auxiliando na tomada de decisões informadas e eficazes.



**Figura 5.2** – *Avaliação dos riscos* 

O plano de ação do PGR apresenta o registro e acompanhamento das ações necessárias para mitigar ou controlar os riscos identificados em um projeto ou processo, vide Figura 5.3. Ela lista as ações específicas a serem tomadas, os responsáveis por cada ação e os prazos para sua conclusão. Com essa visão clara das ações a serem executadas, os responsáveis pelo gerenciamento de riscos podem monitorar o progresso, realizar ajustes e garantir que todas as medidas sejam implementadas de forma adequada e dentro do prazo estabelecido, contribuindo para o sucesso do projeto ou processo. O sistema auxiliou na execução das medidas propostas no plano de ação, garantindo que as ações preventivas necessárias fossem implementadas. Isso contribui para a tomada de decisões relacionadas à segurança do trabalho e para a redução de riscos ocupacionais.



Figura 5.3 - Plano de Ação

# Capítulo 6

## **CONCLUSÕES**

Ao longo deste trabalho, foram abordados conceitos fundamentais da gestão de riscos ocupacionais e a importância do programa de gerenciamento de riscos para garantir a saúde e a segurança dos colaboradores. Foi destacado o papel dos sistemas de gestão especializados como ferramentas úteis nesse processo, permitindo o registro, a análise e o monitoramento dos riscos presentes no ambiente de trabalho. A utilização deste sistema web possibilitou a identificação, avaliação e controle dos riscos, além de permitir o acompanhamento dos planos de ação e a implementação de medidas preventivas de forma mais ágil e integrada. Com essas informações, os gestores podem identificar tendências, padrões e áreas de maior risco, auxiliando na tomada de decisões embasadas em evidências.

Quanto aos aspectos tecnológicos, o controle de autenticação de usuários do framewirk também foi muito útil. Ele possui uma camada de autenticação integrada, que lida com o registro de usuários, login, logout e gerenciamento de permissões. Com apenas algumas linhas de código, foi possível adicionar recursos de autenticação, garantindo que apenas usuários autorizados tenham acesso às funcionalidades restritas. Além disso, utiliza os mecanismos de segurança, como o hashing de senhas, para proteger as informações dos usuários.

O uso desse framework para o desenvolvimento do sistema de gestão do Programa de Gerenciamento de Riscos foi uma experiência bastante positiva. O Django é um muito poderoso e flexível, e proporcionou uma série de recursos e facilidades ao longo do processo de desenvolvimento.

Outra vantagem foi o seu sistema de formulários integrados. Com eles, foi possível criar interfaces para coletar e validar dados dos usuários. Outro recurso

que utilizamos foram as visões baseadas em classes que simplificaram a criação de diferentes tipos de visualizações. As CBVs nos permitiram definir comportamentos padrão para as diferentes páginas do sistema, reduzindo a duplicação de código e facilitando a manutenção do sistema. Além disso, as CBVs oferecem métodos específicos para diferentes etapas do ciclo de vida de uma requisição, como processamento de formulários, renderização de templates e autenticação de usuários. Isso tornou o processo de desenvolvimento mais organizado e eficiente, permitindo um melhor controle sobre as funcionalidades do sistema.

O Docker permitiu criar e gerenciar contêineres que encapsulam todas as dependências e configurações necessárias para o sistema funcionar corretamente. Isso inclui o ambiente de execução, como versão específica do Python e bibliotecas, além do banco de dados, instalação do django e outras ferramentas utilizadas no projeto. Em resumo, a utilização do Docker proporcionou uma experiência altamente positiva. A capacidade de lidar com contêineres facilita a reprodução do ambiente em diferentes estágios do ciclo de vida do sistema. O Docker, portanto, se mostrou uma ferramenta poderosa e eficiente no desenvolvimento desse sistema de gestão de riscos de segurança do trabalho.

O sistema utilizou recursos de versionamento, possibilitando o controle das alterações realizadas ao longo do desenvolvimento do projeto. Isso proporcionou uma visão histórica das modificações e facilitou a identificação e resolução de problemas, garantindo maior transparência e evolução de todo o processo.

É importante salientar que este projeto está no repositório do github disponível em https://www.github.com/yullobomfim/TCC-II para sugestões, e assim contribuir na evolução, escalabilidade e usabilidade deste sistema.

# Capítulo 7

### TRABALHOS FUTUROS

No uso do software foi identificada algumas funcionalidades que aparentam ser relevantes para desenvolver uma melhor experiência de uso do sistema, e, servem como sugestão de trabalhos futuros:

- Criar funcionalidade para auditoria nos inventários de riscos e nos planos de ação.
- ► Realizar testes do sistema em uma empresa na prática para avaliar a usabilidade do sistema pelos usuários.
- ► Implementar novas funcionalidades para incrementar o sistema, como o cadastramento de fotos do ambiente de trabalho, usuários, EPIs e etc;
- ► Desenvolver blog com materiais de treinamentos;
- Criar uma funcionalidade para o envio de e-mail de boas vindas aos novos usuários com informações cadastrais;
- ► Implantar funcionalidade para gerar um arquivo em .pdf contendo um relatórios das ações do programa.

## **REFERÊNCIAS**

BRASIL. *Consolidação das Leis do Trabalho. Decreto-Lei nº 5.452, de 1º de maio de 1943.* 2021. <a href="https://www.planalto.gov.br/ccivil\_03/decreto-lei/del5452.htm">https://www.planalto.gov.br/ccivil\_03/decreto-lei/del5452.htm</a> [Acessado em 20 de junho de 2023].

CURTIN, B. Django Cookbook: Web Development with Django. [S.l.: s.n.], 2016.

GONçALVES, D. A. *Segurança do Trabalho: Conceitos e Aplicações*. [S.l.]: Editora Ergo, 2017.

HOLDENER, A. T. *Ajax: The Definitive Guide*. [S.l.]: 1st ed. Sebastopol: O'Reilly Media., 2008.

HYKES. Why we built Docker? 2013. <a href="https://www.youtube.com/watch?v=3N3n9FzebAA">https://www.youtube.com/watch?v=3N3n9FzebAA</a>> [Acessado em 27 de novembro de 2022].

JUNIOR, A. N. Gestão Integrada de QSMS: Qualidade, Segurança, Meio Ambiente e Saúde Ocupacional. [S.l.]: Editora Brasport, 2021.

LAUDON, K. J. *Sistemas de informação gerenciais: administrando a empresa digital.* [S.l.]: Editora São Paulo, 2004. v. 5.

LEIGH, J. P. *Economia da segurança e saúde no trabalho: Uma revisão da literatura.* [S.l.]: Revista Brasileira de Saúde Ocupacional, 2019. 44 p.

NUNES, I. L. *Gestão de riscos ocupacionais: Conceitos e metodologias*. [S.l.]: Editora LTr, 2010.

OIT. Segurança e Saúde Ocupacional: Conceitos e definições. 2011. <a href="https://www.ilo.org/brasilia/convencoes/WCMS\_236163/lang--pt/index.htm">https://www.ilo.org/brasilia/convencoes/WCMS\_236163/lang--pt/index.htm</a> [Acessado em 20 de junho de 2023].

RESENDE, G. *O que é HTML? Entenda para que ele funciona.* 2021. <a href="https://portaldeplanos.com.br/artigos/html/">https://portaldeplanos.com.br/artigos/html/</a>> [Acessado em 10 de novembro de 2022].

ROBBINS, J. N. Learning Web Design: A Beginner's Guide to Html, Css, Javascript, and Web Graphics. [S.l.]: O'Reilly Media, 2018.

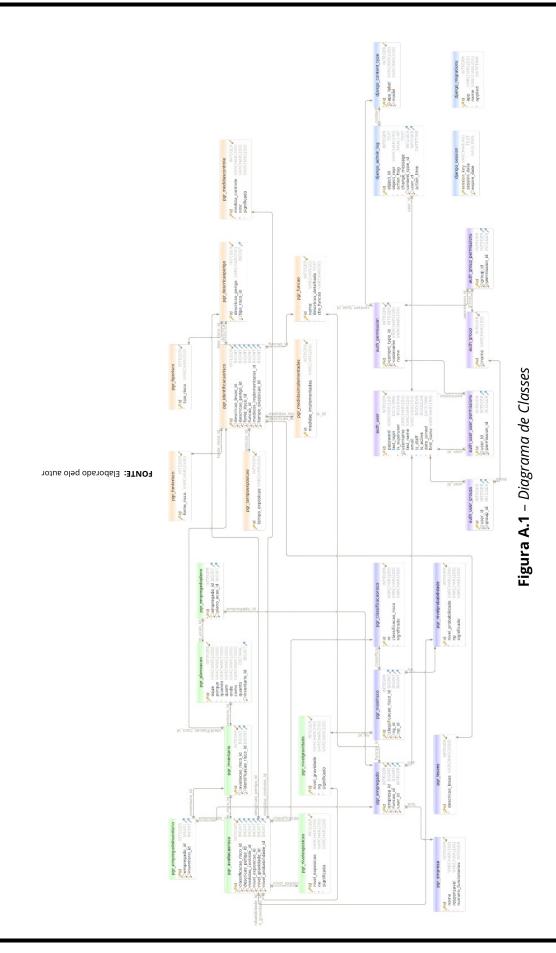
RUPPENTHAL, J. E. Gerenciamento de riscos. [S.l.]: Rede e-Tec Brasil, 2013.

SILVA, A. P. Benefícios da implementação de programas de segurança do trabalho: Uma revisão sistemática da literatura. [S.l.]: Revista de Gestão Ambiental e Sustentabilidade, 2019. v. 8. 258-276 p.

# APÊNDICE A

# Modelagem do banco de dados

**APÊNDICE** Diagrama de classes



71

## ANEXO I

# Norma Regulamentadora n.º 01 -Disposições Gerais e Gerenciamento de Riscos Ocupacionais

**ANEXO** Norma Regulamentadora NR01

#### NORMA REGULAMENTADORA N.º 01 - DISPOSIÇÕES GERAIS e GERENCIAMENTO DE RISCOS **OCUPACIONAIS**

Publicação D.O.U.

Portaria MTb n.º 3.214, de 08 de junho de 1978 06/07/78

> **Atualizações** D.O.U.

Portaria SSMT n.º 06, de 09 de março de 198314/03/83 Portaria SSMT n.º 03, de 07 de fevereiro de 1988 10/03/88 Portaria SSST n.º 13, de 17 de setembro de 1993 21/09/93 Portaria SIT n.º 84, de 04 de março de 2009 12/03/09 Portaria SEPRT n.º 915, de 30 de julho de 2019 31/07/19 Portaria SEPRT n.º 6.730, de 09 de março de 2020 12/03/20

#### (Redação dada pela Portaria SEPRT n.º 6.730, de 09/03/20)

#### Sumário

- 1.1 Objetivo
- 1.2 Campo de aplicação
- 1.3 Competências e estrutura
- 1.4 Direitos e deveres
- 1.5 Gerenciamento de riscos ocupacionais
- 1.6 Da prestação de informação digital e digitalização de documentos
- 1.7 Capacitação e treinamento em Segurança e Saúde no Trabalho
- 1.8 Tratamento diferenciado ao Microempreendedor Individual MEI, à Microempresa ME e à Empresa de Pequeno Porte - EPP
- 1.9 Disposições finais

Anexo I - Termos e definições

Anexo II - Diretrizes e requisitos mínimos para utilização da modalidade de ensino a distância e semipresencial.

#### 1.1 Objetivo

- O objetivo desta Norma é estabelecer as disposições gerais, o campo de aplicação, os termos e as definições comuns às Normas Regulamentadoras - NR relativas a segurança e saúde no trabalho e as diretrizes e os requisitos para o gerenciamento de riscos ocupacionais e as medidas de prevenção em Segurança e Saúde no Trabalho - SST.
- Para fins de aplicação das Normas Regulamentadoras NR, consideram-se os termos e definições constantes no Anexo I.
- 1.2 Campo de aplicação
- **1.2.1** As NR obrigam, nos termos da lei, empregadores e empregados, urbanos e rurais.

- **1.2.1.1** As NR são de observância obrigatória pelas organizações e pelos órgãos públicos da administração direta e indireta, bem como pelos órgãos dos Poderes Legislativo, Judiciário e Ministério Público, que possuam empregados regidos pela Consolidação das Leis do Trabalho —
- **1.2.1.2** Nos termos previstos em lei, aplica-se o disposto nas NR a outras relações jurídicas.
- **1.2.2** A observância das NR não desobriga as organizações do cumprimento de outras disposições que, com relação à matéria, sejam incluídas em códigos de obras ou regulamentos sanitários dos Estados ou Municípios, bem como daquelas oriundas de convenções e acordos coletivos de trabalho.

#### 1.3 Competências e estrutura

- **1.3.1** A Secretaria de Trabalho STRAB, por meio da Subsecretaria de Inspeção do Trabalho SIT, é o órgão de âmbito nacional competente em matéria de segurança e saúde no trabalho para:
- a) formular e propor as diretrizes, as normas de atuação e supervisionar as atividades da área de segurança e saúde do trabalhador;
- b) promover a Campanha Nacional de Prevenção de Acidentes do Trabalho CANPAT;
- c) coordenar e fiscalizar o Programa de Alimentação do Trabalhador PAT;
- d) promover a fiscalização do cumprimento dos preceitos legais e regulamentares sobre Segurança e Saúde no Trabalho SST em todo o território nacional;
- e) participar da implementação da Política Nacional de Segurança e Saúde no Trabalho PNSST; e
- f) conhecer, em última instância, dos recursos voluntários ou de ofício, das decisões proferidas pelo órgão regional competente em matéria de segurança e saúde no trabalho, salvo disposição expressa em contrário.
- **1.3.2** Compete à SIT e aos órgãos regionais a ela subordinados em matéria de Segurança e Saúde no Trabalho, nos limites de sua competência, executar:
- a) fiscalização dos preceitos legais e regulamentares sobre segurança e saúde no trabalho; e
- b) as atividades relacionadas com a CANPAT e o PAT.
- **1.3.3** Cabe à autoridade regional competente em matéria de trabalho impor as penalidades cabíveis por descumprimento dos preceitos legais e regulamentares sobre segurança e saúde no trabalho.

#### 1.4 Direitos e deveres

#### **1.4.1** Cabe ao empregador:

- a) cumprir e fazer cumprir as disposições legais e regulamentares sobre segurança e saúde no trabalho;
- b) informar aos trabalhadores:
- I. os riscos ocupacionais existentes nos locais de trabalho;
- II. as medidas de prevenção adotadas pela empresa para eliminar ou reduzir tais riscos;
- III. os resultados dos exames médicos e de exames complementares de diagnóstico aos quais os próprios trabalhadores forem submetidos; e
- os resultados das avaliações ambientais realizadas nos locais de trabalho.
- c) elaborar ordens de serviço sobre segurança e saúde no trabalho, dando ciência aos trabalhadores;

- d) permitir que representantes dos trabalhadores acompanhem a fiscalização dos preceitos legais e regulamentares sobre segurança e saúde no trabalho;
- e) determinar procedimentos que devem ser adotados em caso de acidente ou doença relacionada ao trabalho, incluindo a análise de suas causas;
- f) disponibilizar à Inspeção do Trabalho todas as informações relativas à segurança e saúde no trabalho; e
- g) implementar medidas de prevenção, ouvidos os trabalhadores, de acordo com a seguinte ordem de prioridade:
- eliminação dos fatores de risco;
- II. minimização e controle dos fatores de risco, com a adoção de medidas de proteção coletiva;
- III. minimização e controle dos fatores de risco, com a adoção de medidas administrativas ou de organização do trabalho; e
- IV. adoção de medidas de proteção individual.

#### **1.4.2** Cabe ao trabalhador:

- a) cumprir as disposições legais e regulamentares sobre segurança e saúde no trabalho, inclusive as ordens de serviço expedidas pelo empregador;
- b) submeter-se aos exames médicos previstos nas NR;
- c) colaborar com a organização na aplicação das NR; e
- d) usar o equipamento de proteção individual fornecido pelo empregador.
- **1.4.2.1** Constitui ato faltoso a recusa injustificada do empregado ao cumprimento do disposto nas alíneas do subitem anterior.
- **1.4.3** O trabalhador poderá interromper suas atividades quando constatar uma situação de trabalho onde, a seu ver, envolva um risco grave e iminente para a sua vida e saúde, informando imediatamente ao seu superior hierárquico.
- **1.4.3.1** Comprovada pelo empregador a situação de grave e iminente risco, não poderá ser exigida a volta dos trabalhadores à atividade enquanto não sejam tomadas as medidas corretivas.
- **1.4.4** Todo trabalhador, ao ser admitido ou quando mudar de função que implique em alteração de risco, deve receber informações sobre:
- a) os riscos ocupacionais que existam ou possam originar-se nos locais de trabalho;
- b) os meios para prevenir e controlar tais riscos;
- c) as medidas adotadas pela organização;
- d) os procedimentos a serem adotados em situação de emergência; e
- e) os procedimentos a serem adotados, em conformidade com os subitens 1.4.3 e 1.4.3.1.

#### **1.4.4.1** As informações podem ser transmitidas:

- a) durante os treinamentos; e
- b) por meio de diálogos de segurança, documento físico ou eletrônico.

#### **1.5** Gerenciamento de riscos ocupacionais

- **1.5.1** O disposto neste item deve ser utilizado para fins de prevenção e gerenciamento dos riscos ocupacionais.
- **1.5.2** Para fins de caracterização de atividades ou operações insalubres ou perigosas, devem ser aplicadas as disposições previstas na NR-15 Atividades e operações insalubres e NR-16 Atividades e operações perigosas.

#### 1.5.3 Responsabilidades

- **1.5.3.1.** A organização deve implementar, por estabelecimento, o gerenciamento de riscos ocupacionais em suas atividades.
- **1.5.3.1.1** O gerenciamento de riscos ocupacionais deve constituir um Programa de Gerenciamento de Riscos PGR.
- **1.5.3.1.1.1** A critério da organização, o PGR pode ser implementado por unidade operacional, setor ou atividade.
- **1.5.3.1.2** O PGR pode ser atendido por sistemas de gestão, desde que estes cumpram as exigências previstas nesta NR e em dispositivos legais de segurança e saúde no trabalho.
- **1.5.3.1.3** O PGR deve contemplar ou estar integrado com planos, programas e outros documentos previstos na legislação de segurança e saúde no trabalho.

#### 1.5.3.2 A organização deve:

- a) evitar os riscos ocupacionais que possam ser originados no trabalho;
- b) identificar os perigos e possíveis lesões ou agravos à saúde;
- c) avaliar os riscos ocupacionais indicando o nível de risco;
- d) classificar os riscos ocupacionais para determinar a necessidade de adoção de medidas de prevenção;
- e) implementar medidas de prevenção, de acordo com a classificação de risco e na ordem de prioridade estabelecida na alínea "g" do subitem 1.4.1; e
- f) acompanhar o controle dos riscos ocupacionais.
- 1.5.3.2.1 A organização deve considerar as condições de trabalho, nos termos da NR-17.
- **1.5.3.3** A organização deve adotar mecanismos para:
- a) consultar os trabalhadores quanto à percepção de riscos ocupacionais, podendo para este fim ser adotadas as manifestações da Comissão Interna de Prevenção de Acidentes CIPA, quando houver: e
- b) comunicar aos trabalhadores sobre os riscos consolidados no inventário de riscos e as medidas de prevenção do plano de ação do PGR.
- 1.5.3.4 A organização deve adotar as medidas necessárias para melhorar o desempenho em SST.
- 1.5.4 Processo de identificação de perigos e avaliação de riscos ocupacionais
- **1.5.4.1** O processo de identificação de perigos e avaliação de riscos ocupacionais deve considerar o disposto nas Normas Regulamentadoras e demais exigências legais de segurança e saúde no trabalho.

#### 1.5.4.2 Levantamento preliminar de perigos

- **1.5.4.2.1** O levantamento preliminar de perigos deve ser realizado:
- a) antes do início do funcionamento do estabelecimento ou novas instalações;
- b) para as atividades existentes; e
- c) nas mudanças e introdução de novos processos ou atividades de trabalho.
- **1.5.4.2.1.1** Quando na fase de levantamento preliminar de perigos o risco não puder ser evitado, a organização deve implementar o processo de identificação de perigos e avaliação de riscos ocupacionais, conforme disposto nos subitens seguintes.
- **1.5.4.2.1.2** A critério da organização, a etapa de levantamento preliminar de perigos pode estar contemplada na etapa de identificação de perigos.

- 1.5.4.3 Identificação de perigos
- **1.5.4.3.1** A etapa de identificação de perigos deve incluir:
- a) descrição dos perigos e possíveis lesões ou agravos à saúde;
- b) identificação das fontes ou circunstâncias; e
- c) indicação do grupo de trabalhadores sujeitos aos riscos.
- **1.5.4.3.2** A identificação dos perigos deve abordar os perigos externos previsíveis relacionados ao trabalho que possam afetar a saúde e segurança no trabalho.

#### 1.5.4.4 Avaliação de riscos ocupacionais

- **1.5.4.4.1** A organização deve avaliar os riscos ocupacionais relativos aos perigos identificados em seu(s) estabelecimento(s), de forma a manter informações para adoção de medidas de prevenção.
- **1.5.4.4.2** Para cada risco deve ser indicado o nível de risco ocupacional, determinado pela combinação da severidade das possíveis lesões ou agravos à saúde com a probabilidade ou chance de sua ocorrência.
- **1.5.4.4.2.1** A organização deve selecionar as ferramentas e técnicas de avaliação de riscos que sejam adequadas ao risco ou circunstância em avaliação.
- **1.5.4.4.3** A gradação da severidade das lesões ou agravos à saúde deve levar em conta a magnitude da consequência e o número de trabalhadores possivelmente afetados.
- **1.5.4.4.3.1** A magnitude deve levar em conta as consequências de ocorrência de acidentes ampliados.
- **1.5.4.4.4** A gradação da probabilidade de ocorrência das lesões ou agravos à saúde deve levar em conta:
- a) os requisitos estabelecidos em Normas Regulamentadoras;
- b) as medidas de prevenção implementadas;
- c) as exigências da atividade de trabalho; e
- d) a comparação do perfil de exposição ocupacional com valores de referência estabelecidos na NR-09.
- **1.5.4.4.5** Após a avaliação, os riscos ocupacionais devem ser classificados, observado o subitem 1.5.4.4.2, para fins de identificar a necessidade de adoção de medidas de prevenção e elaboração do plano de ação.
- **1.5.4.4.6** A avaliação de riscos deve constituir um processo contínuo e ser revista a cada dois anos ou quando da ocorrência das seguintes situações:
- a) após implementação das medidas de prevenção, para avaliação de riscos residuais;
- b) após inovações e modificações nas tecnologias, ambientes, processos, condições, procedimentos e organização do trabalho que impliquem em novos riscos ou modifiquem os riscos existentes;
- c) quando identificadas inadequações, insuficiências ou ineficácias das medidas de prevenção;
- d) na ocorrência de acidentes ou doenças relacionadas ao trabalho;
- e) quando houver mudança nos requisitos legais aplicáveis.
- **1.5.4.4.6.1** No caso de organizações que possuírem certificações em sistema de gestão de SST, o prazo poderá ser de até 3 (três) anos.

#### 1.5.5. Controle dos riscos

#### 1.5.5.1. Medidas de prevenção

- **1.5.5.1.1** A organização deve adotar medidas de prevenção para eliminar, reduzir ou controlar os riscos sempre que:
- a) exigências previstas em Normas Regulamentadoras e nos dispositivos legais determinarem;
- b) a classificação dos riscos ocupacionais assim determinar, conforme subitem 1.5.4.4.5;
- c) houver evidências de associação, por meio do controle médico da saúde, entre as lesões e os agravos à saúde dos trabalhadores com os riscos e as situações de trabalho identificados.
- **1.5.5.1.2** Quando comprovada pela organização a inviabilidade técnica da adoção de medidas de proteção coletiva, ou quando estas não forem suficientes ou encontrarem-se em fase de estudo, planejamento ou implantação ou, ainda, em caráter complementar ou emergencial, deverão ser adotadas outras medidas, obedecendo-se a seguinte hierarquia:
- a) medidas de caráter administrativo ou de organização do trabalho;
- b) utilização de equipamento de proteção individual EPI.
- **1.5.5.1.3** A implantação de medidas de prevenção deverá ser acompanhada de informação aos trabalhadores quanto aos procedimentos a serem adotados e limitações das medidas de prevenção.

#### **1.5.5.2.** Planos de ação

- **1.5.5.2.1** A organização deve elaborar plano de ação, indicando as medidas de prevenção a serem introduzidas, aprimoradas ou mantidas, conforme o subitem 1.5.4.4.5.
- **1.5.5.2.2** Para as medidas de prevenção deve ser definido cronograma, formas de acompanhamento e aferição de resultados.
- 1.5.5.3 Implementação e acompanhamento das medidas de prevenção
- **1.5.5.3.1** A implementação das medidas de prevenção e respectivos ajustes devem ser registrados.
- **1.5.5.3.2** O desempenho das medidas de prevenção deve ser acompanhado de forma planejada e contemplar:
- a) a verificação da execução das ações planejadas;
- b) as inspeções dos locais e equipamentos de trabalho; e
- c) o monitoramento das condições ambientais e exposições a agentes nocivos, quando aplicável.
- **1.5.5.3.2.1** As medidas de prevenção devem ser corrigidas quando os dados obtidos no acompanhamento indicarem ineficácia em seu desempenho.
- **1.5.5.4** Acompanhamento da saúde ocupacional dos trabalhadores
- **1.5.5.4.1** A organização deve desenvolver ações em saúde ocupacional dos trabalhadores integradas às demais medidas de prevenção em SST, de acordo com os riscos gerados pelo trabalho.

- **1.5.5.4.2** O controle da saúde dos empregados deve ser um processo preventivo planejado, sistemático e continuado, de acordo com a classificação de riscos ocupacionais e nos termos da NR-07.
- 1.5.5.5. Análise de acidentes e doenças relacionadas ao trabalho
- 1.5.5.1 A organização deve analisar os acidentes e as doenças relacionadas ao trabalho.
- 1.5.5.2 As análises de acidentes e doenças relacionadas ao trabalho devem ser documentadas e:
- a) considerar as situações geradoras dos eventos, levando em conta as atividades efetivamente desenvolvidas, ambiente de trabalho, materiais e organização da produção e do trabalho;
- b) identificar os fatores relacionados com o evento; e
- c) fornecer evidências para subsidiar e revisar as medidas de prevenção existentes.
- **1.5.6.** Preparação para emergências
- **1.5.6.1** A organização deve estabelecer, implementar e manter procedimentos de respostas aos cenários de emergências, de acordo com os riscos, as características e as circunstâncias das atividades.
- 1.5.6.2 Os procedimentos de respostas aos cenários de emergências devem prever:
- a) os meios e recursos necessários para os primeiros socorros, encaminhamento de acidentados e abandono; e
- b) as medidas necessárias para os cenários de emergências de grande magnitude, quando aplicável.
- **1.5.7** Documentação
- **1.5.7.1** O PGR deve conter, no mínimo, os seguintes documentos:
- a) inventário de riscos; e
- b) plano de ação.
- **1.5.7.2** Os documentos integrantes do PGR devem ser elaborados sob a responsabilidade da organização, respeitado o disposto nas demais Normas Regulamentadoras, datados e assinados.
- **1.5.7.2.1** Os documentos integrantes do PGR devem estar sempre disponíveis aos trabalhadores interessados ou seus representantes e à Inspeção do Trabalho.
- 1.5.7.3 Inventário de riscos ocupacionais
- **1.5.7.3.1** Os dados da identificação dos perigos e das avaliações dos riscos ocupacionais devem ser consolidados em um inventário de riscos ocupacionais.
- **1.5.7.3.2** O Inventário de Riscos Ocupacionais deve contemplar, no mínimo, as seguintes informações:
- a) caracterização dos processos e ambientes de trabalho;
- b) caracterização das atividades;
- c) descrição de perigos e de possíveis lesões ou agravos à saúde dos trabalhadores, com a identificação das fontes ou circunstâncias, descrição de riscos gerados pelos perigos, com a indicação dos grupos de trabalhadores sujeitos a esses riscos, e descrição de medidas de prevenção implementadas;
- d) dados da análise preliminar ou do monitoramento das exposições a agentes físicos, químicos e biológicos e os resultados da avaliação de ergonomia nos termos da NR-17.

- e) avaliação dos riscos, incluindo a classificação para fins de elaboração do plano de ação; e f) critérios adotados para avaliação dos riscos e tomada de decisão.
- **1.5.7.3.3** O inventário de riscos ocupacionais deve ser mantido atualizado.
- **1.5.7.3.3.1** O histórico das atualizações deve ser mantido por um período mínimo de 20 (vinte) anos ou pelo período estabelecido em normatização específica.
- **1.5.8** Disposições gerais do gerenciamento de riscos ocupacionais
- **1.5.8.1** Sempre que várias organizações realizem, simultaneamente, atividades no mesmo local de trabalho devem executar ações integradas para aplicar as medidas de prevenção, visando à proteção de todos os trabalhadores expostos aos riscos ocupacionais.
- **1.5.8.2** O PGR da empresa contratante poderá incluir as medidas de prevenção para as empresas contratadas para prestação de serviços que atuem em suas dependências ou local previamente convencionado em contrato ou referenciar os programas d contratadas.
- **1.5.8.3** As organizações contratantes devem fornecer às contratadas informações sobre os riscos ocupacionais sob sua gestão e que possam impactar nas atividades das contratadas.
- **1.5.8.4** As organizações contratadas devem fornecer ao contratante o Inventário de Riscos Ocupacionais específicos de suas atividades que são realizadas nas dependências da contratante ou local previamente convencionado em contrato.
- **1.6** Da prestação de informação digital e digitalização de documentos
- **1.6.1** As organizações devem prestar informações de segurança e saúde no trabalho em formato digital, conforme modelo aprovado pela STRAB, ouvida a SIT.
- **1.6.1.1** Os modelos aprovados pela STRAB devem considerar os princípios de simplificação e desburocratização.
- **1.6.2** Os documentos previstos nas NR podem ser emitidos e armazenados em meio digital com certificado digital emitido no âmbito da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil), normatizada por lei específica.
- **1.6.3** Os documentos físicos, assinados manualmente, inclusive os anteriores à vigência desta NR, podem ser arquivados em meio digital, pelo período correspondente exigido pela legislação própria, mediante processo de digitalização conforme disposto em Lei.
- **1.6.3.1** O processo de digitalização deve ser realizado de forma a manter a integridade, a autenticidade e, se necessário, a confidencialidade do documento digital, com o emprego de certificado digital emitido no âmbito da Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil).
- **1.6.3.2** Os empregadores que optarem pela guarda de documentos prevista no *caput* devem manter os originais conforme previsão em lei.
- **1.6.4** O empregador deve garantir a preservação de todos os documentos nato digitais ou digitalizados por meio de procedimentos e tecnologias que permitam verificar, a qualquer tempo, sua validade jurídica em todo território nacional, garantindo permanentemente sua autenticidade, integridade, disponibilidade, rastreabilidade, irretratabilidade, privacidade e interoperabilidade.
- **1.6.5** O empregador deve garantir à Inspeção do Trabalho amplo e irrestrito acesso a todos os documentos digitalizados ou nato digitais.

- **1.6.5.1** Para os documentos que devem estar à disposição dos trabalhadores ou dos seus representantes, a organização deverá prover meios de acesso destes às informações, de modo a atender os objetivos da norma específica.
- 1.7 Capacitação e treinamento em Segurança e Saúde no Trabalho
- **1.7.1** O empregador deve promover capacitação e treinamento dos trabalhadores, em conformidade com o disposto nas NR.
- **1.7.1.1** Ao término dos treinamentos inicial, periódico ou eventual, previstos nas NR, deve ser emitido certificado contendo o nome e assinatura do trabalhador, conteúdo programático, carga horária, data, local de realização do treinamento, nome e qualificação dos instrutores e assinatura do responsável técnico do treinamento.

#### **1.7.1.2** A capacitação deve incluir:

- a) treinamento inicial;
- b) treinamento periódico; e
- c) treinamento eventual.
- **1.7.1.2.1** O treinamento inicial deve ocorrer antes de o trabalhador iniciar suas funções ou de acordo com o prazo especificado em NR.
- **1.7.1.2.2** O treinamento periódico deve ocorrer de acordo com periodicidade estabelecida nas NR ou, quando não estabelecido, em prazo determinado pelo empregador.

#### **1.7.1.2.3** O treinamento eventual deve ocorrer:

- a) quando houver mudança nos procedimentos, condições ou operações de trabalho, que impliquem em alteração dos riscos ocupacionais;
- b) na ocorrência de acidente grave ou fatal, que indique a necessidade de novo treinamento; ou
- c) após retorno de afastamento ao trabalho por período superior a 180 (cento e oitenta) dias.
- **1.7.1.2.3.1** A carga horária, o prazo para sua realização e o conteúdo programático do treinamento eventual deve atender à situação que o motivou.

#### 1.7.1.3 A capacitação pode incluir:

- a) estágio prático, prática profissional supervisionada ou orientação em serviço;
- b) exercícios simulados; ou
- c) habilitação para operação de veículos, embarcações, máguinas ou equipamentos.
- **1.7.2** O tempo despendido em treinamentos previstos nas NR é considerado como de trabalho efetivo.
- **1.7.3** O certificado deve ser disponibilizado ao trabalhador e uma cópia arquivada na organização.
- **1.7.4** A capacitação deve ser consignada nos documentos funcionais do empregado.
- **1.7.5** Os treinamentos previstos em NR podem ser ministrados em conjunto com outros treinamentos da organização, observados os conteúdos e a carga horária previstos na respectiva norma regulamentadora.

Aproveitamento de conteúdos de treinamento na mesma organização

**1.7.6** É permitido o aproveitamento de conteúdos de treinamentos ministrados na mesma organização desde que:

- a) o conteúdo e a carga horária requeridos no novo treinamento estejam compreendidos no treinamento anterior;
- b) o conteúdo do treinamento anterior tenha sido ministrado no prazo inferior ao estabelecido em NR ou há menos de 2 (dois) anos, quando não estabelecida esta periodicidade; e
- c) seja validado pelo responsável técnico do treinamento.
- **1.7.6.1** O aproveitamento de conteúdos deve ser registrado no certificado, mencionando o conteúdo e a data de realização do treinamento aproveitado.
- **1.7.6.1.1** A validade do novo treinamento passa a considerar a data do treinamento mais antigo aproveitado.

#### Aproveitamento de treinamentos entre organizações

- **1.7.7** Os treinamentos realizados pelo trabalhador podem ser avaliados pela organização e convalidados ou complementados.
- **1.7.7.1** A convalidação ou complementação deve considerar:
- a) as atividades desenvolvidas pelo trabalhador na organização anterior, quando for o caso;
- b) as atividades que desempenhará na organização;
- c) o conteúdo e carga horária cumpridos;
- d) o conteúdo e carga horária exigidos; e
- e) que o último treinamento tenha sido realizado em período inferior ao estabelecido na NR ou há menos de 2 (dois) anos, nos casos em que não haja prazo estabelecido em NR.
- **1.7.8** O aproveitamento de treinamentos anteriores, total ou parcialmente, não exclui a responsabilidade da organização de emitir a certificação da capacitação do trabalhador, devendo mencionar no certificado a data da realização dos treinamentos convalidados ou complementados.
- **1.7.8.1** Para efeito de periodicidade de realização de novo treinamento, é considerada a data do treinamento mais antigo convalidado ou complementado.

#### Dos treinamentos ministrados na modalidade de ensino a distância ou semipresencial

- **1.7.9** Os treinamentos podem ser ministrados na modalidade de ensino a distância ou semipresencial, desde que atendidos os requisitos operacionais, administrativos, tecnológicos e de estruturação pedagógica previstos no Anexo II desta NR.
- **1.7.9.1** O conteúdo prático do treinamento pode ser realizado na modalidade de ensino a distância ou semipresencial, desde que previsto em NR específica.
- **1.8** Tratamento diferenciado ao Microempreendedor Individual MEI, à Microempresa ME e à Empresa de Pequeno Porte EPP
- 1.8.1 O Microempreendedor Individual MEI está dispensado de elaborar o PGR
- **1.8.1.1** A dispensa da obrigação de elaborar o PGR não alcança a organização contratante do MEI, que deverá incluí-lo nas suas ações de prevenção e no seu PGR, quando este atuar em suas dependências ou local previamente convencionado em contrato.
- **1.8.2** Serão expedidas pela Secretaria Especial de Previdência e Trabalho SEPRT fichas com orientações sobre as medidas de prevenção a serem adotadas pelo MEI.
- **1.8.3** As microempresa e empresas de pequeno porte que não forem obrigadas a constituir SESMT e optarem pela utilização de ferramenta(s) de avaliação de risco a serem disponibilizada(s)

pela SEPRT, em alternativa às ferramentas e técnicas previstas no subitem 1.5.4.4.2.1, poderão estruturar o PGR considerando o relatório produzido por esta(s) ferramenta(s) e o plano de ação.

- **1.8.4** As microempresas e empresas de pequeno porte, graus de risco 1 e 2, que no levantamento preliminar de perigos não identificarem exposições ocupacionais a agentes físicos, químicos e biológicos, em conformidade com a NR9, e declararem as informações digitais na forma do subitem 1.6.1, ficam dispensadas da elaboração do PGR.
- **1.8.4.1** As informações digitais de segurança e saúde no trabalho declaradas devem ser divulgadas junto aos trabalhadores.
- **1.8.5** A dispensa prevista nesta Norma é aplicável quanto à obrigação de elaboração do PGR e não afasta a obrigação de cumprimento por parte do MEI, ME e EPP das demais disposições previstas em NR.
- **1.8.6** O MEI, a ME e a EPP, graus de risco 1 e 2, que declararem as informações digitais na forma do subitem 1.6.1 e não identificarem exposições ocupacionais a agentes físicos, químicos, biológicos e riscos relacionados a fatores ergonômicos, ficam dispensados de elaboração do Programa de Controle Médico de Saúde Ocupacional PCMSO.
- **1.8.6.1** A dispensa do PCMSO não desobriga a empresa da realização dos exames médicos e emissão do Atestado de Saúde Ocupacional ASO.
- **1.8.7** Os graus de riscos 1 e 2 mencionados nos subitens 1.8.4 e 1.8.6 são os previstos na Norma Regulamentadores nº 04 Serviços Especializados em Engenharia de Segurança e em Medicina do Trabalho SESMT.
- **1.8.8** O empregador é o responsável pela prestação das informações previstas nos subitens 1.8.4 e 1.8.6.

#### 1.9 Disposições finais

- **1.9.1** O não-cumprimento das disposições legais e regulamentares sobre segurança e saúde no trabalho acarretará a aplicação das penalidades previstas na legislação pertinente.
- **1.9.2** As dúvidas suscitadas e os casos omissos verificados no cumprimento das NR serão decididos pela Secretaria de Trabalho, ouvida a SIT.

#### Anexo I da NR-01

#### Termos e definições

Agente biológico: Microrganismos, parasitas ou materiais originados de organismos que, em função de sua natureza e do tipo de exposição, são capazes de acarretar lesão ou agravo à saúde do trabalhador. Exemplos: bactéria **Bacillus anthracis**, vírus linfotrópico da célula T humana, príon agente de doença de **Creutzfeldt-Jakob**, fungo **Coccidioides immitis**.

Agente físico: Qualquer forma de energia que, em função de sua natureza, intensidade e exposição, é capaz de causar lesão ou agravo à saúde do trabalhador. Exemplos: ruído, vibrações, pressões anormais, temperaturas extremas, radiações ionizantes, radiações não ionizantes.

Observação: Critérios sobre iluminamento, conforto térmico e conforto acústico da NR-17 não constituem agente físico para fins da NR-09.

Agente químico: Substância química, por si só ou em misturas, quer seja em seu estado natural, quer seja produzida, utilizada ou gerada no processo de trabalho, que em função de sua natureza, concentração e exposição, é capaz de causar lesão ou agravo à saúde do trabalhador. Exemplos: fumos de cádmio, poeira mineral contendo sílica cristalina, vapores de tolueno, névoas de ácido sulfúrico.

Canteiro de obra: área de trabalho fixa e temporária, onde se desenvolvem operações de apoio e execução à construção, demolição ou reforma de uma obra.

Empregado: a pessoa física que presta serviços de natureza não eventual a empregador, sob a dependência deste e mediante salário.

Empregador: a empresa individual ou coletiva que, assumindo os riscos da atividade econômica, admite, assalaria e dirige a prestação pessoal de serviços. Equiparam-se ao empregador as organizações, os profissionais liberais, as instituições de beneficência, as associações recreativas ou outras instituições sem fins lucrativos, que admitam trabalhadores como empregados.

Estabelecimento: local privado ou público, edificado ou não, móvel ou imóvel, próprio ou de terceiros, onde a empresa ou a organização exerce suas atividades em caráter temporário ou permanente.

Evento perigoso: Ocorrência ou acontecimento com o potencial de causar lesões ou agravos à saúde.

Frente de trabalho: área de trabalho móvel e temporária.

Local de trabalho: área onde são executados os trabalhos.

Obra: todo e qualquer serviço de engenharia de construção, montagem, instalação, manutenção ou reforma.

Ordem de serviço de segurança e saúde no trabalho: instruções por escrito quanto às precauções para evitar acidentes do trabalho ou doenças ocupacionais. A ordem de serviço pode estar contemplada em procedimentos de trabalho e outras instruções de SST.

Organização: pessoa ou grupo de pessoas com suas próprias funções com responsabilidades, autoridades e relações para alcançar seus objetivos. Inclui, mas não é limitado a empregador, a tomador de serviços, a empresa, a empreendedor individual, produtor rural, companhia, corporação, firma, autoridade, parceria, organização de caridade ou instituição, ou parte ou combinação desses, seja incorporada ou não, pública ou privada.

Perigo ou fator de risco ocupacional/ Perigo ou fonte de risco ocupacional: Fonte com o potencial de causar lesões ou agravos à saúde. Elemento que isoladamente ou em combinação com outros tem o potencial intrínseco de dar origem a lesões ou agravos à saúde.

Prevenção: o conjunto das disposições ou medidas tomadas ou previstas em todas as fases da atividade da organização, visando evitar, eliminar, minimizar ou controlar os riscos ocupacionais.

Responsável técnico pela capacitação: profissional legalmente habilitado ou trabalhador qualificado, conforme disposto em NR específica, responsável pela elaboração das capacitações e treinamentos.

Risco ocupacional: Combinação da probabilidade de ocorrer lesão ou agravo à saúde causados por um evento perigoso, exposição a agente nocivo ou exigência da atividade de trabalho e da severidade dessa lesão ou agravo à saúde.

Setor de serviço: a menor unidade administrativa ou operacional compreendida no mesmo estabelecimento.

Trabalhador: pessoa física inserida em uma relação de trabalho, inclusive de natureza administrativa, como os empregados e outros sem vínculo de emprego.

#### Anexo II da NR-01

# Diretrizes e requisitos mínimos para utilização da modalidade de ensino a distância e semipresencial.

#### Sumário:

- 1. Objetivo
- 2. Disposições gerais
- 3. Estruturação pedagógica
- 4. Requisitos operacionais e administrativo
- 5. Requisitos tecnológicos
- 6. Glossário

#### 1. Objetivo

**1.1** Estabelecer diretrizes e requisitos mínimos para utilização da modalidade de ensino à distância e semipresencial para as capacitações previstas nas NR, disciplinando tanto aspectos relativos à estruturação pedagógica, quanto exigências relacionadas às condições operacionais, tecnológicas e administrativas necessárias para uso desta modalidade de ensino.

#### 2. Disposições gerais

- **2.1** O empregador que optar pela realização das capacitações por meio das modalidades de ensino a distância ou semipresencial poderá desenvolver toda a capacitação ou contratar empresa ou instituição especializada que a oferte, devendo em ambos os casos observar os requisitos constantes deste Anexo e da NR-01.
- **2.1.1** A empresa ou instituição especializada que oferte as capacitações previstas nas NR na modalidade de ensino à distância e semipresencial, deve atender aos requisitos constantes deste Anexo e da NR-01 para que seus certificados sejam considerados válidos.
- **2.2** O empregador que optar pela contratação de serviços de empresa ou instituição especializada deve fazer constar na documentação que formaliza a prestação de serviços a obrigatoriedade pelo prestador de serviço do atendimento aos requisitos previstos neste Anexo e nos itens relativos à capacitação previstos nas NR.
- **2.3** As capacitações que utilizam ensino a distância ou semipresencial devem ser estruturadas com, no mínimo, a duração definida para as respectivas capacitações na modalidade presencial.
- **2.4** A elaboração do conteúdo programático deve abranger os tópicos de aprendizagem requeridos, bem como respeitar a carga horária estabelecida para todos os conteúdos.
- **2.5** As atividades práticas obrigatórias devem respeitar as orientações previstas nas NR e estar descritas no Projeto Pedagógico do curso.

#### 3. Estruturação pedagógica

- **3.1** Sempre que a modalidade de ensino a distância ou semipresencial for utilizada, será obrigatória a elaboração de projeto pedagógico que deve conter:
- a) objetivo geral da capacitação;
- b) princípios e conceitos para a proteção da segurança e da saúde dos trabalhadores, definidos nas NR;
- c) estratégia pedagógica da capacitação, incluindo abordagem quanto à parte teórica e prática, quando houver;
- d) indicação do responsável técnico pela capacitação;
- e) relação de instrutores, quando aplicável;
- f) infraestrutura operacional de apoio e controle;
- g) conteúdo programático teórico e prático, quando houver;
- h) objetivo de cada módulo;
- i) carga horária;
- j) estimativa de tempo mínimo de dedicação diária ao curso;
- k) prazo máximo para conclusão da capacitação;
- l) público alvo;
- m) material didático;
- n) instrumentos para potencialização do aprendizado; e
- o) avaliação de aprendizagem.
- **3.2** O projeto pedagógico do curso deverá ser validado a cada 2 (dois) anos ou quando houver mudança na NR, procedendo a sua revisão, caso necessário.

#### 4. Requisitos operacionais e administrativos

- **4.1** O empregador deve manter o projeto pedagógico disponível para a Inspeção do Trabalho, para a representação sindical da categoria no estabelecimento e para a Comissão Interna de Prevenção de Acidentes CIPA.
- **4.1.1** A empresa ou instituição especializada deve disponibilizar aos contratantes o projeto pedagógico.
- **4.2** Deve ser disponibilizado aos trabalhadores todo o material didático necessário para participar da capacitação, conforme item 3.1 deste Anexo.
- **4.3** Devem ser disponibilizados recursos e ambiente que favoreça a concentração e a absorção do conhecimento pelo empregado, para a realização da capacitação.
- **4.4** O período de realização do curso deve ser exclusivamente utilizado para tal fim para que não seja concomitante com o exercício das atividades diárias de trabalho.
- **4.5** Deve ser mantido canal de comunicação para esclarecimento de dúvidas, possibilitando a solução das mesmas, devendo tal canal estar operacional durante o período de realização do curso.
- **4.6** A verificação de aprendizagem deve ser realizada de acordo com a estratégia pedagógica adotada para a capacitação, estabelecendo a classificação com o conceito satisfatório ou insatisfatório.
- **4.6.1** A avaliação da aprendizagem se dará pela aplicação da prova no formato presencial, obtendo, dessa forma, o registro da assinatura do empregado, ou pelo formato digital, exigindo a sua identificação e senha individual.

- **4.6.2** Quando a avaliação da aprendizagem for online, devem ser preservadas condições de rastreabilidade que garantam a confiabilidade do processo.
- **4.6.3** O processo de avaliação da aprendizagem deve contemplar situações práticas que representem a rotina laboral do trabalhador para a adequada tomada de decisões com vistas à prevenção de acidentes e doenças relacionadas ao trabalho.
- **4.7** Após o término do curso, as empresas devem registrar sua realização, mantendo o resultado das avaliações de aprendizagem e informações sobre acesso dos participantes (logs).
- **4.7.1** O histórico do registro de acesso dos participantes (logs) deve ser mantido pelo prazo mínimo de 2 (dois) anos após o término da validade do curso.

#### 5. Requisitos tecnológicos

**5.1** Somente serão válidas as capacitações realizadas na modalidade de ensino à distância ou semipresencial que sejam executadas em um Ambiente Virtual de Aprendizagem apropriado à gestão, transmissão do conhecimento e aprendizagem do conteúdo.

#### 6. Glossário

Ambiente exclusivo: espaço físico distinto do posto de trabalho que disponibilize ao trabalhador os recursos tecnológicos necessários à execução do curso e condições de conforto adequadas para a aprendizagem.

Ambiente Virtual de Aprendizagem (AVA): espaço virtual de aprendizagem que oferece condições para interações (síncrona e assíncrona) permanentes entre seus usuários. Pode ser traduzida como sendo uma "sala de aula" acessada via web. Permite integrar múltiplas mídias, linguagens e recursos, apresentar informações de maneira organizada, desenvolver interações entre pessoas e objetos de conhecimento, elaborar e socializar produções, tendo em vista atingir determinados objetivos.

Avaliação de Aprendizagem: visa aferir o conhecimento adquirido pelo trabalhador e o respectivo grau de assimilação após a realização da capacitação.

EAD: segundo Decreto n.º 9.057/2017, caracteriza-se a Educação a Distância como modalidade educacional na qual a mediação didático-pedagógica nos processos de ensino e aprendizagem ocorre com a utilização de meios e tecnologias de informação e comunicação, com estudantes e professores desenvolvendo atividades educativas em lugares ou tempos diversos.

Ensino semipresencial: conjugação de atividades presenciais obrigatórias com outras atividades educacionais que podem ser realizadas sem a presença física do participante em sala de aula, utilizando recursos didáticos com suporte da tecnologia, de material impresso e/ou de outros meios de comunicação.

Projeto pedagógico: instrumento de concepção do processo ensino-aprendizagem. Nele deve-se registrar o objetivo da aprendizagem, a estratégia pedagógica escolhida para a formação e capacitação dos trabalhadores, bem como todas as informações que estejam envolvidas no processo.

Instrumentos para potencialização do aprendizado: recursos, ferramentas, dinâmicas e tecnologias de comunicação que tenham como objetivo tornar mais eficaz o processo de ensino-aprendizagem.

Log: registro informatizado de acesso ao sistema. Ex.: log de acesso: registro de acessos; *login*: registro de entrada;

Logoff: registro de saída.