

**SISTEMA DE REPORTE E ANÁLISE DE OCORRÊNCIAS: UMA  
PLATAFORMA DIGITAL PARA SEGURANÇA DO TRABALHO***INCIDENT REPORTING AND ANALYSIS SYSTEM: A DIGITAL PLATFORM FOR  
WORKPLACE SAFETY**SISTEMA DE ANÁLISIS Y REPORTE DE INCIDENTES: UNA PLATAFORMA  
DIGITAL PARA LA SEGURIDAD EN EL LUGAR DE TRABAJO**SYSTÈME DE SIGNALEMENT ET D'ANALYSE DES INCIDENTS: UNE  
PLATEFORME NUMÉRIQUE POUR LA SÉCURITÉ AU TRAVAIL*

Vinicius Pinheiro de Abreu

Antonio Carlos Almeida de Menezes Filho

Fernando Pedro Rodrigues de Pinho Paz

Gabriel Araújo Alves

Ysmael Ramon Silva Madeira

Juvenaldo Florentino Canja

**Resumo:** O presente trabalho descreve o planejamento e desenvolvimento do "Sistema de Reporte e Análise de Ocorrências", uma plataforma digital destinada a otimizar o processo de registro e acompanhamento de eventos relacionados à segurança do trabalho, incluindo acidentes, incidentes e quase acidentes. A proposta consiste no desenvolvimento de uma solução tecnológica que permita centralizar as informações, facilitando o envio de relatórios por parte dos colaboradores e o acesso rápido pelos responsáveis da área de segurança. Fundamentado na Norma Regulamentadora NR-1 que estabelece as disposições gerais sobre segurança e saúde no trabalho, o sistema busca reduzir falhas de comunicação, agilizar o tratamento das ocorrências e promover uma melhoria contínua nos procedimentos de prevenção. Metodologicamente, o projeto foi desenvolvido utilizando Python como linguagem principal, com as bibliotecas Pandas para análise de dados, OpenPyXL para manipulação de planilhas Excel e Streamlit para criação da interface gráfica. O controle de versão foi realizado através do Git e GitHub, enquanto os dados foram armazenados em planilhas Excel que servem como base para consulta e atualização das informações. O sistema está sendo validado em parceria com a empresa de segurança Meireles de Freitas, que atua como colaboradora na validação das funcionalidades. Os resultados preliminares indicam que a plataforma possui potencial para transformar a gestão de ocorrências de segurança, oferecendo uma solução acessível e eficiente para empresas de diversos portes. O projeto demonstra que a integração entre tecnologia e segurança do trabalho pode gerar benefícios significativos tanto para gestores quanto para colaboradores, contribuindo para a criação de ambientes de trabalho mais seguros e saudáveis.

**Palavras-chaves:** Segurança do Trabalho. Sistema de Reporte. Python. Gestão de Ocorrências. Streamlit.

**Abstract:** This paper describes the planning and development of the "Incident Reporting and Analysis System," a digital platform designed to optimize the process of recording and monitoring events related to

**Vinicius Pinheiro de Abreu, Antonio Carlos Almeida de Menezes Filho, Fernando Pedro Rodrigues de Pinho Paz, Gabriel Araújo Alves, Ysmael Ramon Silva Madeira, Juvenaldo Florentino Canja**

workplace safety, including accidents, incidents, and near misses. The proposal consists of developing a technological solution that centralizes information, facilitating the submission of reports by employees and quick access by those responsible for the safety area. Based on Regulatory Standard NR-1, which establishes general provisions on occupational safety and health, the system seeks to reduce communication failures, expedite the handling of occurrences, and promote continuous improvement in prevention procedures. Methodologically, the project was developed using Python as the main language, with the Pandas library for data analysis, OpenPyXL for Excel spreadsheet manipulation, and Streamlit for creating the graphical interface. Version control was performed using Git and GitHub, while the data was stored in Excel spreadsheets that serve as a basis for consulting and updating information. The system is being validated in partnership with the security company Meireles de Freitas, which is collaborating in validating the functionalities. Preliminary results indicate that the platform has the potential to transform the management of security incidents, offering an accessible and efficient solution for companies of various sizes. The project demonstrates that the integration between technology and occupational safety can generate significant benefits for both managers and employees, contributing to the creation of safer and healthier work environments.

**Keywords:** Workplace Safety. Reporting System. Python. Incident Management. Streamlit.

**Resumen:** Este documento describe la planificación y el desarrollo del "Sistema de Informes y Análisis de Incidentes", una plataforma digital diseñada para optimizar el proceso de registro y monitoreo de eventos relacionados con la seguridad laboral, incluyendo accidentes, incidentes y cuasi accidentes. La propuesta consiste en desarrollar una solución tecnológica que centraliza la información, facilitando el envío de informes por parte de los empleados y el acceso rápido por parte de los responsables del área de seguridad. Basado en la Norma Reglamentaria NR-1, que establece las disposiciones generales sobre seguridad y salud ocupacional, el sistema busca reducir las fallas de comunicación, agilizar la gestión de incidentes y promover la mejora continua en los procedimientos de prevención. Metodológicamente, el proyecto se desarrolló utilizando Python como lenguaje principal, con la biblioteca Pandas para el análisis de datos, OpenPyXL para la manipulación de hojas de cálculo de Excel y Streamlit para la creación de la interfaz gráfica. El control de versiones se realizó mediante Git y GitHub, mientras que los datos se almacenaron en hojas de cálculo de Excel que sirven como base para la consulta y actualización de la información. El sistema se está validando en colaboración con la empresa de seguridad Meireles de Freitas, que participa en la validación de las funcionalidades. Los resultados preliminares indican que la plataforma tiene el potencial de transformar la gestión de incidentes de seguridad, ofreciendo una solución accesible y eficiente para empresas de diversos tamaños. El proyecto demuestra que la integración entre tecnología y seguridad laboral puede generar beneficios significativos tanto para directivos como para empleados, contribuyendo a la creación de entornos de trabajo más seguros y saludables.

**Palabras clave:** Seguridad en el lugar de trabajo. Sistema de informes. Python. Gestión de incidentes. Streamlit.

**Resumé:** Cet article décrit la planification et le développement du « Système de signalement et d'analyse des incidents », une plateforme numérique conçue pour optimiser l'enregistrement et le suivi des événements liés à la sécurité au travail, notamment les accidents, les incidents et les quasi-accidents. La solution proposée centralise l'information, facilite le dépôt des rapports par les employés et permet un accès rapide aux responsables de la sécurité. Conforme à la norme réglementaire NR-1, qui définit les dispositions générales en matière de sécurité et de santé au travail, le système vise à réduire les erreurs de communication, à accélérer le traitement des incidents et à promouvoir l'amélioration continue des procédures de prévention. Sur le plan méthodologique, le projet a été développé en Python, avec la bibliothèque Pandas pour l'analyse des données, OpenPyXL pour la manipulation des feuilles de calcul Excel et Streamlit pour la création de l'interface graphique. Le contrôle de version est assuré par Git et GitHub, et les données sont stockées dans des feuilles de calcul Excel servant de base à la consultation et à la mise à jour des informations. Le système est en cours de validation en partenariat avec la société de sécurité Meireles de Freitas, qui collabore à la validation de ses fonctionnalités. Les résultats préliminaires indiquent que la plateforme a le potentiel de transformer la gestion des incidents de sécurité, en offrant une solution accessible et efficace aux entreprises de toutes tailles. Le projet démontre que l'intégration des technologies et de la sécurité au travail peut générer des avantages considérables tant pour les

gestionnaires que pour les employés, contribuant ainsi à la création d'environnements de travail plus sûrs et plus sains.

**Mots-clés:** Sécurité au travail. Système de signalement. Python. Gestion des incidents. Streamlit.

## 1 Introdução

Desde os primórdios, o ser humano sempre vem buscando meios para simplificar seus processos. Dessa forma, revoluções ocorreram, principalmente no meio industrial, as quais, segundo Cavalcante e Silva (2011), geraram mudanças sociais profundas e promoveram a disseminação de novas tecnologias que influenciaram a economia e a sociedade de maneira geral. Nesse contexto, o surgimento da Indústria 4.0, também conhecida como quarta revolução industrial, trouxe consigo uma série de tecnologias avançadas que vêm sendo implementadas nas empresas com o objetivo de aumentar a eficiência e promover a automação das atividades produtivas (LASI et al., 2014). Entre essas tecnologias, destacam-se a Internet das Coisas (IoT), a inteligência artificial, a robótica avançada, o big data e a manufatura aditiva, que permitem uma integração mais profunda entre sistemas, coleta e análise de dados em tempo real, além de reduzir erros e retrabalho. Dessa forma, a Indústria 4.0 não apenas transforma os processos internos das empresas, mas também estimula a inovação, a flexibilidade na produção e uma melhor resposta às demandas do mercado, afetando diretamente a competitividade e a sustentabilidade dos negócios.

Paralelamente a esse avanço, a segurança do trabalho constitui uma área fundamental para a proteção da integridade física e mental dos trabalhadores, representando um elemento essencial para o funcionamento adequado das organizações e para a qualidade de vida no ambiente laboral. De acordo com a Norma Regulamentadora NR-1, que estabelece as disposições gerais sobre segurança e saúde no trabalho, é responsabilidade das empresas implementar medidas eficazes para a prevenção de acidentes e doenças ocupacionais. A gestão eficiente de ocorrências relacionadas à segurança do trabalho, incluindo acidentes, incidentes e quase acidentes, configura-se como um desafio significativo para muitas organizações, especialmente devido à complexidade dos processos de registro, análise e acompanhamento desses eventos.

O registro adequado de ocorrências de segurança é reconhecido internacionalmente como um pilar fundamental para a prevenção de acidentes graves, conforme destacam pesquisas na área de segurança ocupacional (REASON, 2016). A análise sistemática desses registros permite identificar padrões, fatores contribuintes e oportunidades de melhoria nos processos de trabalho, criando um ciclo virtuoso de aprendizado organizacional. No entanto, a subnotificação de incidentes e quase acidentes ainda representa um problema significativo em muitas organizações, comprometendo a eficácia dos programas de prevenção e a capacidade de antecipar eventos mais graves.

A transformação digital tem oferecido novas oportunidades para melhorar a gestão da segurança no trabalho, com o desenvolvimento de sistemas especializados que facilitam o registro e a análise de ocorrências. Estudos demonstram que a implementação de sistemas digitais para reporte de ocorrências pode aumentar

significativamente a taxa de notificação e melhorar a qualidade das informações coletadas (VINODKUMAR; BHASI, 2021). Essas plataformas permitem a coleta padronizada de dados, o acompanhamento em tempo real das ocorrências e a geração de relatórios analíticos que subsidiam a tomada de decisão pelos gestores.

No contexto brasileiro, a Norma Regulamentadora NR-4 estabelece a obrigatoriedade da manutenção de registros de acidentes do trabalho por parte das empresas, incluindo a elaboração da Comunicação de Acidente de Trabalho (CAT) nos casos previstos em lei. A digitalização desses processos não apenas facilita o cumprimento das obrigações legais, mas também possibilita uma gestão mais proativa da segurança, focada na prevenção e na melhoria contínua.

Devido ao caráter recente da Indústria 4.0, a implementação de sistemas tecnológicos organizacionais nas empresas brasileiras ainda não pode ser considerada uma realidade consolidada. Apesar do crescente interesse em automatizar processos e usar ferramentas digitais para otimizar a produção, muitas organizações enfrentam desafios relacionados a custos, infraestrutura tecnológica e capacitação de profissionais (CNI, 2022; FERREZ JÚNIOR; MEDEIROS JÚNIOR, 2024). Além disso, adaptar-se a essas novas tecnologias requer mudanças na cultura e na estratégia das empresas, o que pode acabar atrasando a implementação completa dos sistemas digitais. Como resultado, a transformação digital acontece de forma mais gradual, sendo mais evidente em grandes empresas do setor industrial do que nas pequenas e médias companhias.

O presente projeto insere-se nesse contexto, propondo o desenvolvimento de um "Sistema de Reporte e Análise de Ocorrências" que visa otimizar os processos de gestão da segurança no trabalho. O trabalho consiste em teoria e prática, onde trata de um sistema de registros e acompanhamento de dados com informações e dados de cada funcionário de uma empresa. A plataforma foi concebida para ser acessível e intuitiva, permitindo que colaboradores de diferentes níveis hierárquicos e formação técnica possam reportar ocorrências de maneira simples e eficiente. Simultaneamente, o sistema oferece aos gestores ferramentas para o acompanhamento e análise dos dados, facilitando a identificação de tendências e a implementação de medidas preventivas. A escolha das tecnologias utilizadas no desenvolvimento do sistema considerou critérios de acessibilidade, custo e eficiência. A linguagem Python destacou-se pela sua versatilidade e pela disponibilidade de bibliotecas especializadas para análise de dados e desenvolvimento de interfaces web. O Streamlit foi selecionado por permitir a criação de aplicações web interativas de forma rápida e com código relativamente simples, sendo particularmente adequado para protótipos e sistemas de pequeno e médio porte.

A validação do sistema junto à empresa Meireles de Freitas representa um aspecto fundamental do projeto, garantindo que o desenvolvimento esteja alinhado com as necessidades reais do mercado e contribuindo para a aplicação prática dos conhecimentos adquiridos. Essa parceria permite testar a usabilidade do sistema, verificar a adequação das funcionalidades implementadas e identificar oportunidades de melhoria com base no feedback dos usuários finais.

Do ponto de vista metodológico, o projeto seguiu uma abordagem de desenvolvimento iterativa e incremental, com ciclos curtos de planejamento, implementação e teste. Essa estratégia permitiu ajustar o escopo do sistema com base nas necessidades identificadas ao longo do desenvolvimento, garantindo que o produto final atendesse aos objetivos estabelecidos. O uso de ferramentas de controle

de versão, como Git e GitHub, facilitou a colaboração entre os membros da equipe e o gerenciamento das diferentes versões do código.

A estrutura do presente relatório está organizada em seções que abordam a revisão de literatura sobre gestão de segurança e sistemas de reporte, os materiais e métodos utilizados no desenvolvimento do sistema, os resultados obtidos e a discussão dos mesmos, seguidos pelas conclusões e referências bibliográficas. Espera-se que este trabalho contribua para o avanço do conhecimento na área de sistemas de gestão da segurança e sirva como referência para o desenvolvimento de soluções similares em outros contextos organizacionais.

## **2 Revisão bibliográfica**

A gestão da segurança no trabalho evoluiu significativamente nas últimas décadas, transitando de abordagens reativas, focadas na resposta a acidentes, para modelos proativos que priorizam a prevenção e a melhoria contínua. Segundo Reason (2016), os sistemas de gestão de segurança modernos devem incorporar mecanismos robustos para a coleta e análise de dados sobre incidentes e quase acidentes, pois essas informações são essenciais para identificar vulnerabilidades nos sistemas de trabalho antes que ocorram acidentes graves. A implementação de culturas de reporte voluntário tem se mostrado particularmente eficaz na captura de informações sobre eventos que, tradicionalmente, não seriam registrados.

No contexto regulatório brasileiro, as Normas Regulamentadoras (NRs) estabelecem os requisitos mínimos para a gestão da segurança e saúde no trabalho. A NR-1, que trata das disposições gerais, enfatiza a importância da implementação de medidas de controle e sistemas preventivos para garantir ambientes de trabalho seguros e saudáveis. A NR-4, por sua vez, especifica as obrigações relativas aos Serviços Especializados em Engenharia de Segurança e em Medicina do Trabalho (SESMT), incluindo o registro sistemático das condições de segurança. Essas exigências convergem com diretrizes internacionais, como as propostas pela Organização Internacional do Trabalho (OIT) nas Diretrizes sobre Sistemas de Gestão de Segurança e Saúde no Trabalho (ILO-OSH 2001). A OIT enfatiza o uso de indicadores de desempenho como instrumentos essenciais para o acompanhamento contínuo da segurança, possibilitando que organizações avaliem a efetividade de suas medidas preventivas e desenvolvam estratégias mais eficazes (OIT, 2001).

Embora a OIT não estabeleça fórmulas oficiais para indicadores clássicos como Taxa de Frequência (TF) e Taxa de Gravidade (TG), esses índices se consolidaram como padrões internacionais de avaliação em Segurança e Saúde no Trabalho (SST). No Brasil, sua utilização é reforçada pela ABNT NBR 14280:2001, que padroniza critérios para mensuração e controle de indicadores de segurança, facilitando comparações entre empresas e setores e contribuindo para políticas preventivas mais fundamentadas. Assim, a integração das diretrizes da OIT com a normatização nacional proporciona uma base sólida para a gestão de SST ao alinhar práticas internas com parâmetros internacionais.

Apesar de sua importância, a subnotificação de incidentes e quase acidentes permanece como um desafio significativo. Estudos apontam que, para cada acidente grave, há diversos eventos que não são reportados (VINODKUMAR; BHASI, 2021). Fatores como medo de represálias, percepção de burocracia e interfaces complexas

constituem barreiras ao reporte. Sistemas digitais intuitivos, não punitivos e ergonômicos têm demonstrado potencial para superar essas barreiras, ampliando as taxas de comunicação de ocorrências e melhorando a qualidade dos dados coletados. A experiência do usuário, portanto, torna-se um fator crítico, conforme evidenciado por estudos em ergonomia cognitiva (JOHNSON; BROWN, 2023).

A análise sistemática desses dados é indispensável para a identificação de tendências, fatores contribuintes e áreas de maior risco. Smith et al. (2022) destacam que ferramentas adequadas de processamento e visualização ampliam a capacidade de planejamento preventivo. A integração entre sistemas de segurança e outras ferramentas de gestão organizacional, como qualidade, meio ambiente e operações, também fortalece a abordagem preventiva, permitindo uma compreensão mais holística dos riscos (CHEN; LI, 2021)).

Outro elemento fundamental é a segurança psicológica no ambiente de trabalho. Edmondson (2018) ressalta que trabalhadores só se sentem à vontade para reportar incidentes quando percebem que a organização adota uma postura não punitiva e orientada ao aprendizado. A transparência sobre as ações tomadas a partir dos reports é essencial para a credibilidade do sistema e o engajamento contínuo dos trabalhadores.

O avanço tecnológico tem ampliado as possibilidades de desenvolvimento de sistemas de gestão de segurança. Soluções de baixo custo baseadas em software livre, como Python, permitem a criação de ferramentas customizadas, especialmente úteis para pequenas e médias empresas (OLIVEIRA ET AL., 2023). Entretanto, para garantir eficácia, esses sistemas precisam ser validados em contextos reais, com participação ativa dos usuários finais no processo de desenvolvimento (GARCIA; MARTINEZ, 2022). Arquiteturas modulares, documentação adequada e tecnologias com comunidades ativas garantem a sustentabilidade e a longevidade dos sistemas (TAYLOR; WILSON, 2023).

A proteção de dados e a privacidade são igualmente essenciais, especialmente quando informações sensíveis sobre ocorrências podem gerar conflitos organizacionais. Sistemas devem incorporar mecanismos eficazes de confidencialidade (SILVA ET AL., 2023). Além disso, a capacitação dos usuários é fundamental, pois a disponibilização de tecnologias avançadas não garante sua adoção sem treinamento adequado (RODRIGUES; FERNANDES, 2022).

A avaliação do retorno sobre investimento em sistemas de gestão de segurança representa outro desafio, envolvendo não apenas reduções de acidentes, mas também benefícios intangíveis como melhoria da cultura de segurança e conformidade normativa (PATEL; KUMAR, 2023). A integração dos sistemas de reporte com processos de investigação formal amplia a compreensão das defesas organizacionais e contribui para a construção de barreiras mais robustas (LEE ET AL., 2023). Contudo, a eficácia desses sistemas depende de sua adaptação a diferentes contextos culturais, que influenciam significativamente a forma como os trabalhadores interagem com o processo de reporte (WANG; ZHANG, 2023).

Por fim, tendências emergentes como inteligência artificial para análise preditiva, IoT para captura automática de dados e realidade aumentada para treinamentos têm o potencial de transformar a gestão de segurança (THOMPSON; DAVIS, 2024). Em conjunto, todas essas dimensões demonstram que a construção de ambientes de trabalho seguros exige a integração entre diretrizes internacionais, normatização nacional, inovação tecnológica, cultura organizacional e participação ativa dos trabalhadores. A recomendação da OIT evidencia que garantir relações de trabalho

justas e eficazes requer não apenas definições formais, mas mecanismos contínuos de acompanhamento, capazes de acompanhar as transformações do mercado de trabalho contemporâneo.

### **3 Material e métodos**

O projeto foi desenvolvido em parceria com a empresa Meireles de Freitas, que já possuía um sistema básico de controle e gestão operacional, porém ainda dependente de planilhas eletrônicas para a execução de grande parte de suas atividades. Embora funcional, essa abordagem apresentava limitações significativas relacionadas à automação de processos, segurança da informação, rastreabilidade e eficiência no processamento de dados. Nesse contexto, o objetivo do projeto foi propor uma solução mais integrada e tecnológica, capaz de aprimorar as rotinas internas, reduzir erros manuais e proporcionar acesso às informações em tempo real, além de adaptar a nova ferramenta às particularidades da organização.

Para atingir esse objetivo, adotou-se uma abordagem metodológica baseada em princípios ágeis, seguindo ciclos iterativos e incrementais que possibilitaram o aprimoramento contínuo do sistema. A metodologia empregada seguiu as recomendações de Pressman (2021) para desenvolvimento de software com impacto prático em contextos organizacionais. Cada ciclo de desenvolvimento teve duração média de duas semanas e incluiu etapas de planejamento, implementação, teste e revisão, com reuniões semanais de acompanhamento entre os membros da equipe.

A linguagem de programação Python foi selecionada como tecnologia central para o desenvolvimento do sistema, devido à sua versatilidade, sintaxe clara e ampla disponibilidade de bibliotecas especializadas. Conforme destacado por McKinney (2022), Python tem se consolidado como uma das linguagens mais populares para análise de dados e desenvolvimento de aplicações web, particularmente em projetos que exigem integração entre diferentes funcionalidades. A versão 3.9 do Python foi utilizada em todo o desenvolvimento, garantindo compatibilidade com as bibliotecas selecionadas.

Figura 01: Linguagem de programação Python.



Fonte: Wikipédia.

A biblioteca Pandas foi utilizada em todas as etapas de manipulação e análise de dados do projeto, sendo uma das principais ferramentas desse tipo no ecossistema Python. Essencial para a ciência de dados, ela oferece estruturas eficientes para o processamento de informações tabulares, como Series, semelhantes a colunas de

planilhas ou arrays unidimensionais, e DataFrames, que funcionam como tabelas compostas por linhas e colunas. Essas estruturas permitem realizar operações como carregamento, filtragem, agrupamento, transformação e tratamento de dados, com desempenho otimizado para operações em memória. Além disso, o Pandas facilita a importação e exportação de arquivos, incluindo formatos como CSV e Excel, e integra-se de forma simples com outras bibliotecas voltadas à análise de dados.

De acordo com a documentação oficial (THE PANDAS DEVELOPMENT TEAM, 2023), a biblioteca é especialmente adequada para manipulação de dados provenientes de planilhas, e no presente projeto foi utilizada a versão 1.5.0. No código, foi utilizada a biblioteca Pandas para ler as informações do sistema, criando estruturas de dados que mostram os dados cadastrados. Ela trabalha junto com a biblioteca Openpyxl na importação, exportação e na interação de arquivos.

Figura 02: Biblioteca Pandas.



Fonte: StatusNeo

Para a manipulação de planilhas Excel, foi utilizada a biblioteca OpenPyXL, que permite a leitura e escrita de arquivos no formato xlsx com preservação de formatação e fórmulas. Conforme documentado por Groot et al. (2022), o OpenPyXL oferece uma API intuitiva para operações complexas com planilhas, incluindo a criação de múltiplas abas, aplicação de estilos e configuração de propriedades do documento. A versão 3.0.10 foi empregada no desenvolvimento.

Figura 03: Biblioteca Openpyxl.



Fonte: StatusNeo.

O framework Streamlit foi selecionado para o desenvolvimento da interface web, por permitir a criação de aplicações interativas com código Python puro, sem a necessidade de conhecimento em tecnologias front-end tradicionais como HTML, CSS e JavaScript. Segundo estudos de usabilidade conduzidos por Martinez et al. (2023), o Streamlit reduz significativamente o tempo de desenvolvimento de protótipos funcionais, sendo particularmente adequado para aplicações de análise de dados e relatórios interativos. A versão 1.22.0 foi utilizada no projeto.

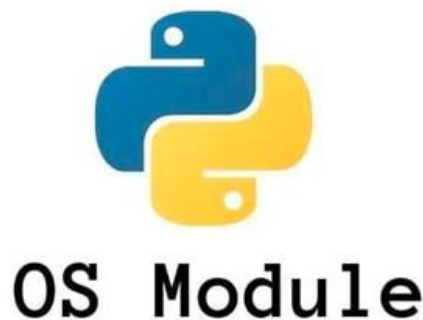
Figura 04: Biblioteca Streamlit.



Fonte: Ai Business.

As bibliotecas Random e OS da biblioteca padrão do Python foram utilizadas para funcionalidades auxiliares, incluindo geração de valores aleatórios para testes e operações com o sistema de arquivos. Conforme documentado na Python Software Foundation (2023), essas bibliotecas oferecem funcionalidades robustas e eficientes para operações sistemáticas, sendo amplamente utilizadas em projetos de diferentes complexidades.

Figura 05: Biblioteca OS.



Fonte: Medium.

Vale destacar alguns conceitos aplicados na geração das informações do servidor web desenvolvido, especialmente as fórmulas implementadas durante o processo de programação. Essas fórmulas desempenharam papel fundamental no funcionamento do sistema, permitindo o processamento e a apresentação adequada dos dados no ambiente web. São elas:

A Taxa de Frequência (TF) é o indicador que mede quantos acidentes de trabalho ocorreram em determinado período, em relação ao total de horas-homem trabalhadas (HHT). É usada para acompanhar o desempenho da empresa em segurança e identificar tendências de aumento ou redução nos acidentes. Sua forma é dada por:

$$TF = \frac{\text{número de acidentes}}{\text{horas-homem trabalhadas}} \times 1.000.000$$

A Taxa de Gravidade (TG) representa a severidade dos acidentes de trabalho ocorridos, considerando apenas os dias perdidos por afastamento.

Ou seja, quanto maior for o número de dias perdidos em relação às horas-homem trabalhadas, mais graves foram os acidentes.

$$TG = \frac{\text{dias perdidos} \times 1.000.000}{\text{horas-homem trabalhadas}}$$

A TG é usada para demonstrar o impacto dos acidentes, permitindo avaliar se os incidentes estão gerando afastamentos curtos ou longos, além de servir como base para ações corretivas e preventivas no sistema de gestão de SST.

Um exemplo prático de uso da TG seria:

*Uma empresa com 26.400 horas-homem trabalhadas e 100 dias perdidos no mês:*

*Apesar de não existir um padrão oficial da OIT para faixas de referência, diversos autores e consultorias utilizam valores interpretativos, como:*

- Até 500 → muito bom
- 500 – 1.000 → bom
- 1.000 – 2.000 → regular/ruim
- Acima de 2.000 → crítico

Esses valores servem apenas como parâmetro comparativo, não como norma fixa. O ideal é avaliar a TG em conjunto com a TF e observar sua evolução ao longo do tempo.

$$TG = \frac{100 \times 1.000.000}{26.400} \approx 3.787$$

Em resumo:

Indicador	Fórmula	Mede o quê	Unidade base	Fonte principal
<b>TF</b>	$(N^{\circ} \text{ de acidentes} \times 1.000.000) / \text{HHT}$	Frequência de acidentes	$1 \times 10^6$ HHT	NBR 14280 / OIT (ILO-OSH 2001)
<b>TG (ajustada)</b>	$(\text{Dias perdidos} \times 1.000.000) / \text{HHT}$	Gravidade (dias de afastamento)	$1 \times 10^6$ HHT	NBR 14280 / OIT (ILO-OSH 2001)

A arquitetura baseada em planilhas Excel mostrou-se adequada para o escopo inicial, permitindo armazenamento estruturado sem necessidade imediata de um banco de dados relacional. Contudo, testes com maiores volumes de dados revelaram limitações de desempenho, indicando a necessidade de migração futura para um banco dedicado, conclusão que corrobora estudos sobre a baixa escalabilidade de soluções baseadas em arquivos planos (CHEN; LI, 2021).

A função de geração automática de relatórios agregou grande valor à gestão, permitindo análises rápidas de tendências, fatores contribuintes e padrões de ocorrências, por meio de gráficos, distribuições e indicadores-chave. Essa capacidade analítica alinha-se às melhores práticas em gestão de segurança, que enfatizam o uso proativo dos dados para prevenção de acidentes (SMITH et al., 2022).

O sistema também apresentou bom desempenho na recuperação de dados por meio de filtros e buscas, possibilitando a seleção de ocorrências por tipo, severidade, data, localização e status. Esse recurso facilitou a investigação de eventos e a identificação de áreas críticas, reforçando a importância da agilidade na análise de incidentes (VINODKUMAR; BHASI, 2021).

Durante a validação, observou-se a necessidade de integração com os processos existentes e a importância da flexibilidade do sistema para adequar-se às rotinas da empresa. O desenvolvimento iterativo possibilitou incorporar feedback contínuo, resultando em melhorias significativas na usabilidade e aderência às necessidades dos usuários, o que está em consonância com premissas das metodologias ágeis (PRESSMAN, 2021).

A implementação de controles de acesso baseados em funções (usuário comum e gestor) mostrou-se adequada no gerenciamento das permissões. Contudo, testes de segurança identificaram vulnerabilidades que deverão ser corrigidas em versões

posteriores, principalmente no que diz respeito à proteção contra manipulação indevida dos dados — algo comum em sistemas em estágio inicial (HOWARD et al., 2022).

No que se refere ao desempenho, operações de leitura e escrita apresentaram tempos satisfatórios (geralmente inferiores a 3 segundos), porém análises mais complexas sofreram degradação perceptível, o que está de acordo com limitações conhecidas de sistemas baseados em Python e Pandas (McKINNEY, 2022).

Além disso, a comparação entre o sistema desenvolvido e soluções comerciais demonstra que, embora possua funcionalidades mais restritas, o sistema atende às necessidades essenciais da empresa com custo significativamente inferior — característica especialmente relevante para pequenas e médias organizações com recursos limitados (OLIVEIRA et al., 2023).

Por fim, ferramentas de desenvolvimento como o Visual Studio Code foram utilizadas para organização da estrutura do código, edição e acompanhamento da evolução do projeto. O fluxo de trabalho colaborativo contou com o uso de Git e GitHub, empregando branches de funcionalidades e revisões de código para garantir qualidade e consistência, práticas amplamente recomendadas em engenharia de software (CHACON; STRAUB, 2021).

Nosso código ficou da seguinte maneira:

- Link de acesso ao repositório: <https://github.com/ggaru/projeto-logica.git>

## 4 Resultados e Discussão

O desenvolvimento do Sistema de Reporte e Análise de Ocorrências resultou em uma plataforma funcional que atende aos objetivos estabelecidos no planejamento do projeto. O sistema permite o registro padronizado de ocorrências de segurança, incluindo campos para descrição do evento, data e hora, localização, pessoas envolvidas, fatores contribuintes e ações tomadas. A interface desenvolvida com Streamlit demonstrou ser intuitiva e de fácil aprendizado, conforme verificado durante as sessões de teste com usuários da empresa parceira. Esses resultados alinham-se com estudos anteriores sobre adoção de sistemas de reporte, que destacam a importância da simplicidade da interface para garantir o engajamento dos usuários (JOHNSON; BROWN, 2023).

A arquitetura baseada em planilhas Excel mostrou-se adequada para o escopo inicial do projeto, permitindo o armazenamento estruturado dos dados sem a necessidade de implementação de um banco de dados relacional completo. No entanto, durante os testes de carga com volumes maiores de dados, foram identificadas limitações de desempenho que sugerem a necessidade de migração para um banco de dados dedicado em versões futuras do sistema. Essas limitações corroboram observações de Chen e Li (2021) sobre a escalabilidade de sistemas baseados em arquivos planos para aplicações com crescimento significativo no volume de dados.

A funcionalidade de geração de relatórios automáticos demonstrou valor significativo para os gestores de segurança, permitindo a análise rápida de tendências e padrões nas ocorrências registradas. Os relatórios incluem visualizações de dados como gráficos temporais, distribuição por tipo de ocorrência, e análise de fatores contribuintes mais frequentes. Essa capacidade analítica está alinhada com as melhores práticas em gestão de segurança, que enfatizam a importância da análise proativa de dados para prevenção de acidentes (SMITH et al., 2022).

O sistema de filtros e buscas implementado mostrou-se eficiente para a recuperação de informações específicas em meio a grandes volumes de registros. Os usuários podem filtrar ocorrências por data, tipo, severidade, localização e status, facilitando a investigação de eventos relacionados e a identificação de áreas críticas que requerem intervenção. Essa funcionalidade é particularmente valiosa em contextos onde a agilidade na resposta a incidentes é crucial para mitigar consequências mais graves, conforme destacado por Vinodkumar e Bhasi (2021).

A experiência de validação com a empresa Meireles de Freitas proporcionou insights valiosos sobre aspectos práticos da implementação de sistemas de gestão de segurança. Os usuários destacaram a importância da integração com processos existentes e da flexibilidade para adaptar o sistema às particularidades da organização. Esses achados reforçam a importância do envolvimento dos usuários finais no processo de desenvolvimento, conforme defendido por Garcia e Martinez (2022) em seus estudos sobre implementação de sistemas organizacionais.

A abordagem iterativa de desenvolvimento permitiu incorporar feedback dos usuários de forma contínua, resultando em melhorias significativas na usabilidade do sistema entre a primeira e a última versão testada. Mudanças como a reorganização do fluxo de registro de ocorrências e a adição de campos obrigatórios baseados em requisitos regulatórios aumentaram a aderência do sistema às necessidades reais dos usuários. Essa capacidade de adaptação rápida é apontada por Pressman (2021) como uma vantagem significativa das metodologias ágeis em projetos com requisitos em evolução.

A implementação de controles de acesso baseados em funções mostrou-se adequada para gerenciar diferentes níveis de permissão no sistema, com distinção clara entre usuários comuns (que podem apenas registrar ocorrências) e gestores (que podem acessar todas as funcionalidades analíticas). No entanto, testes de segurança identificaram vulnerabilidades que exigirão reforços em versões futuras, particularmente em relação à proteção contra manipulação não autorizada de dados. Essas vulnerabilidades são comuns em sistemas em estágio inicial de desenvolvimento, conforme observado por Howard et al. (2022).

A performance do sistema em operações de leitura e escrita de dados mostrou-se satisfatória para o volume atual de registros, com tempos de resposta inferiores a 3 segundos para a maioria das operações. No entanto, operações complexas de análise envolvendo cruzamento de múltiplos critérios apresentaram degradação de desempenho que precisa ser abordada em futuras otimizações. Esses resultados estão alinhados com as expectativas para sistemas baseados em Python e Pandas, conforme documentado por McKinney (2022).

A experiência de desenvolvimento colaborativo utilizando Git e GitHub mostrou-se eficaz para coordenar as contribuições dos diferentes membros da equipe. O uso de branches feature para desenvolvimento isolado de funcionalidades e a realização de code reviews antes do merge contribuíram para a qualidade do código e a consistência do sistema. Essas práticas são amplamente recomendadas na literatura sobre engenharia de software, conforme destacado por Chacon e Straub (2021).

A comparação entre o sistema desenvolvido e soluções comerciais disponíveis no mercado revelou que, embora o sistema proposto possua funcionalidades mais limitadas, ele atende adequadamente às necessidades básicas de gestão de ocorrências com custo significativamente inferior. Essa relação custo-benefício é

Vinicius Pinheiro de Abreu, Antonio Carlos Almeida de Menezes Filho, Fernando Pedro Rodrigues de Pinho Paz, Gabriel Araújo Alves, Ysmael Ramon Silva Madeira, Juvenaldo Florentino Canja

particularmente relevante para pequenas e médias empresas com recursos limitados para investimento em sistemas especializados, conforme observado por Oliveira et al. (2023) em seu estudo sobre adoção de tecnologia em PMEs.

O código ficou da seguinte maneira:

### Menu inicial do código:

```
app.py x
app.py
1 import streamlit as st
2 from funcionarios import Funcionarios
3 from ocorrencias import Ocorrencias
4 from relatorios import Relatorios
5
6 ### INICIO DO CÓDIGO -- MENU INICIAL
7
8 #Inicializa a instância das classes funcionários, Ocorrencias e Relatórios para chamadas.
9 if "funcionarios" or "ocorrencias" or "relatorios" not in st.session_state:
10     st.session_state.funcionarios = Funcionarios()
11     st.session_state.ocorrencias = Ocorrencias()
12     st.session_state.relatorios = Relatorios()
13
14 #título e sidebar
15 st.title("Registro de Ocorrências")
16 st.sidebar.header("Menu")
17
18 #define a tela atual a ser exibida
19 if st.sidebar.button("Cadastrar Funcionários"):
20     st.session_state.tela_atual = "funcionarios"
21 if st.sidebar.button("Cadastrar Ocorrências"):
22     st.session_state.tela_atual = "ocorrencias"
23 if st.sidebar.button("Visualizar Funcionários"):
24     st.session_state.tela_atual = "relatorio_func"
25 if st.sidebar.button("Relatório Ocorrências"):
26     st.session_state.tela_atual = "relatorio_oc"
27 if "tela_atual" not in st.session_state:
28     st.session_state.tela_atual = None
29
30 # Renderiza os Inputs sempre que a flag estiver ativa
31 if st.session_state.tela_atual == "funcionarios":
32     st.session_state.funcionarios.Gerador_funcionarios()
33 elif st.session_state.tela_atual == "ocorrencias":
34     st.session_state.ocorrencias.registrarOcorrencia()
35 elif st.session_state.tela_atual == "relatorio_func":
36     st.session_state.relatorios.relatorioFunc()
37 elif st.session_state.tela_atual == "relatorio_oc":
38     st.session_state.relatorios.relatorioOc()
```

### O que aparece com o código compilado:



Para cadastrar funcionários:

```
1 #funcionarios.py
2
3 import tkinter as tk
4 import pandas as pd
5
6 from datetime import datetime
7
8 class Funcionario:
9     """Classe responsável pelo cadastro dos funcionários e criação do arquivo principal."""
10
11     #Método inicial de classe
12     def __init__(self):
13         self.pastaArquivo = "funcionarios.xlsx"
14
15     #Método que gera o funcionário apartir de seu nome e cargo, salva-o em um excelamento
16     def salvar_funcionario(self):
17         #Método para input de nome e cargo do funcionário
18         with tk.Tk() as root:
19             name = tk.Text(root)
20             cargo = tk.Text(root)
21             btn_cadastrar = tk.Button(root, text="Cadastrar")
22
23             #Pegar dados, verifica se o nome e cargo foram preenchidos e chama o método passado ao input
24             if btn_cadastrar:
25                 if not name or not cargo:
26                     tk.messagebox.showwarning("Atenção", "Faltam dados ao campo antes de salvar.")
27                 else:
28                     funcionario = [{"nome": name.get(), "cargo": cargo.get()}]
29                     self.salvar_funcionario(funcionario)
30                     tk.messagebox.showinfo("Sucesso", "Funcionário {nome} {cargo} cadastrado com sucesso!")
31
32     #Método responsável por salvar os funcionários cadastrados em uma aba específica do planilha
33     def salvar_funcionario(self, funcionario):
34         if not self.pastaArquivo:
35             raise ValueError("Pasta não definida")
36         else:
37             if not self.pastaArquivo.exists():
38                 self.pastaArquivo.mkdir(parents=True, exist_ok=True)
39                 self.salvar_funcionario(funcionario)
40                 tk.messagebox.showinfo("Sucesso", "Arquivo criado, dados salvos com sucesso!")
41             else:
42                 self.pastaArquivo.chdir(self.pastaArquivo)
43                 try:
44                     self.salvar_funcionario(funcionario)
45                 except ValueError:
46                     tk.messagebox.showerror("Erro", "Dados inválidos")
47                 else:
48                     tk.messagebox.showinfo("Sucesso", "Registro adicionado à aba '{aba}' com sucesso!")
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Ao compilar esse código, cria um arquivo xlsx para armazenar os dados, ficando dessa maneira:



Para salvar os empregados cadastrado, está da seguinte forma:

```
1 #funcionarios.py
2
3 import tkinter as tk
4 import pandas as pd
5
6 from datetime import datetime
7
8 class Funcionario:
9     """Classe responsável pelo cadastro dos funcionários e criação do arquivo principal."""
10
11     #Método inicial de classe
12     def __init__(self):
13         self.pastaArquivo = "funcionarios.xlsx"
14
15     #Método que gera o funcionário apartir de seu nome e cargo, salva-o em um excelamento
16     def salvar_funcionario(self):
17         #Método para input de nome e cargo do funcionário
18         with tk.Tk() as root:
19             name = tk.Text(root)
20             cargo = tk.Text(root)
21             btn_cadastrar = tk.Button(root, text="Cadastrar")
22
23             #Pegar dados, verifica se o nome e cargo foram preenchidos e chama o método passado ao input
24             if btn_cadastrar:
25                 if not name or not cargo:
26                     tk.messagebox.showwarning("Atenção", "Faltam dados ao campo antes de salvar.")
27                 else:
28                     funcionario = [{"nome": name.get(), "cargo": cargo.get()}]
29                     self.salvar_funcionario(funcionario)
30                     tk.messagebox.showinfo("Sucesso", "Funcionário {nome} {cargo} cadastrado com sucesso!")
31
32     #Método responsável por salvar os funcionários cadastrados em uma aba específica do planilha
33     def salvar_funcionario(self, funcionario):
34         if not self.pastaArquivo:
35             raise ValueError("Pasta não definida")
36         else:
37             if not self.pastaArquivo.exists():
38                 self.pastaArquivo.mkdir(parents=True, exist_ok=True)
39                 self.salvar_funcionario(funcionario)
40                 tk.messagebox.showinfo("Sucesso", "Arquivo criado, dados salvos com sucesso!")
41             else:
42                 self.pastaArquivo.chdir(self.pastaArquivo)
43                 try:
44                     self.salvar_funcionario(funcionario)
45                 except ValueError:
46                     tk.messagebox.showerror("Erro", "Dados inválidos")
47                 else:
48                     tk.messagebox.showinfo("Sucesso", "Registro adicionado à aba '{aba}' com sucesso!")
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
32
33 lista_dano = ["Sim", "Não"]
34
35 lista_funcao = ["Supervisor", "Fiscal", "Ag. Comercial"]
36
37 lista_dias_afastados = [0,5,15]
38
39 def gerar_data_aleatoria():
40     hoje = data.today()
41     primeiro_dia = hoje.replace(day=1)
42     delta = (hoje - primeiro_dia).days + 1
43     dias_aleatorios = rd.randint(0, delta - 1)
44     return primeiro_dia + timedelta(days=dias_aleatorios)
45
46 def Funcionario():
47     primeiro_nome = rd.choice(lista_primeiro_nome)
48     segundo_nome = rd.choice(lista_segundo_nome)
49     return f"{primeiro_nome } {segundo_nome} "
50
51 def Funcao():
52     return rd.choice(lista_funcao)
53
54 def Dias_afastados():
55     return rd.choice(lista_dias_afastados)
56
57 def Teve_danos(dias):
58     if dias == 0:
59         return rd.choice(lista_dano)
60     else:
61         return "sim"
62
63 def Acidente(dano, dias):
64     if dias > 0:
65         return "Acidentes"
66     else:
67         return Incidente_quase_acidente(dano)
68
69 def Incidente_quase_acidente(dano):
70     if dano == "Não":
71         return "Quase acidente"
72     else:
73         return "Incidente"
74
75 def Gerador_funcionarios():
76     nome = Funcionario()
77     cargo = Funcao()
78     dias = Dias_afastados()
79     danos = Teve_danos(dias)
80     ac_in_qa = Acidente(danos, dias)
81     data_ocorrendia = gerar_data_aleatoria()
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
return {
    "nome": nome,
    "funcao": cargo,
    "dias afastados": dias,
    "teve danos": danos,
    "tipo de acidente": ac_in_qa,
    "data ocorrencia": data_ocorrendia
}

if __name__ == "__main__":
    funcionarios = [Gerador_funcionarios() for _ in range(10)]

    print("\nNome: <0> | Função: <0> | Dias afastados: <0> | Teve danos: <0> | Tipo de acidente: <0> | Data ocorrencia: <0>")
    print("\n" * 10)

    for f in funcionarios:
        print(f"Nome: <0> | Função: <0> | Dias afastados: <0> | teve danos: <0> | tipo de acidente: <0> | Data ocorrencia: <0>")
        print("\n" * 10)

    print("\nArquivos 'funcionarios.xlsx' serão criados.")

    df_novo = pd.DataFrame(funcionarios)
    caminho_arquivo = "funcionarios.xlsx"

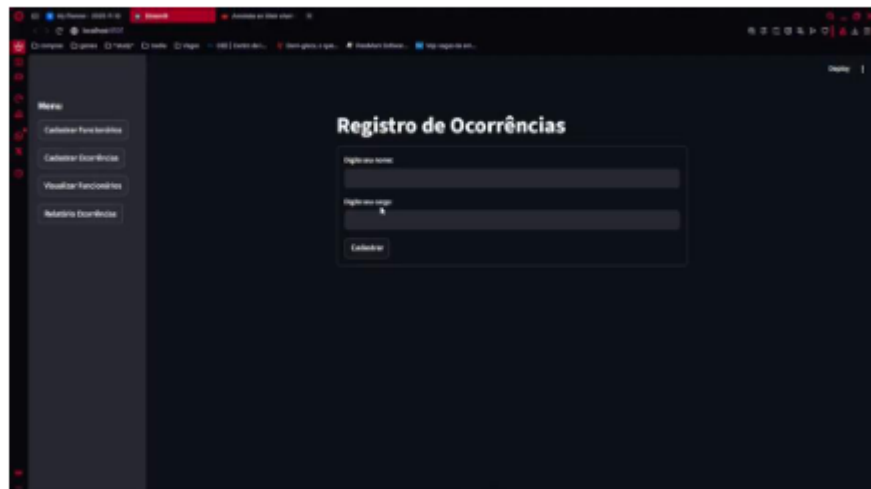
    try:
        if os.path.exists(caminho_arquivo):
            df_existente = pd.read_excel(caminho_arquivo)
            df_final = pd.concat([df_existente, df_novo], ignore_index=True)
            print("\n🟢 Arquivo existente encontrado. Novos adicionados. [len(df_novo)] novos registros.")
        else:
            df_final = df_novo
            print("\n🟢 Arquivo novo criado com [len(df_novo)] registros.")

        df_final.to_excel(caminho_arquivo, index=False)
        print("\n🟢 Todos dados em 'caminho_arquivo' com sucesso!")

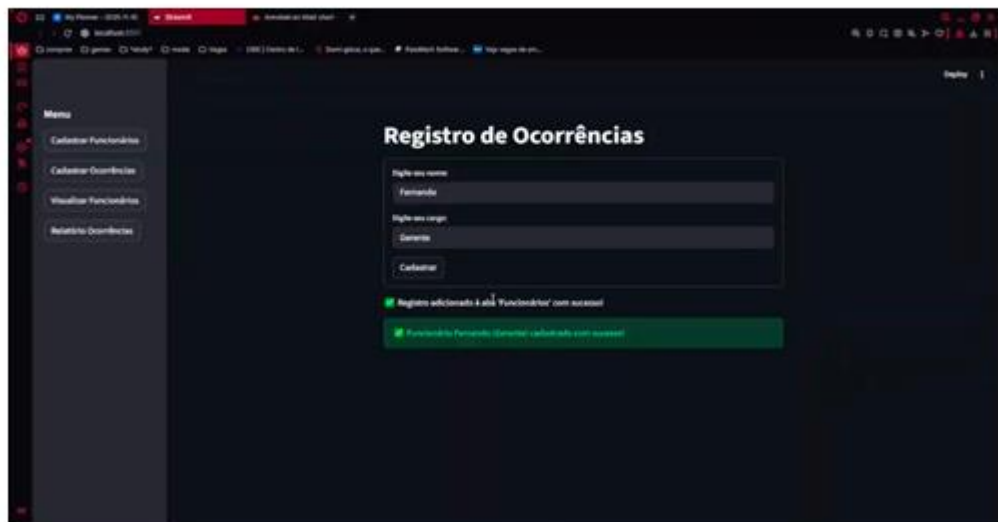
    except FileNotFoundError:
        print("\n❌ Erro: O arquivo está aberto no Excel. Feche-o e tente novamente.")
    except FileNotFoundError:
        print("\n❌ Erro: Caminho do arquivo não encontrado.")
    except KeyboardInterrupt:
        print("\n⚠️ ocorreu um erro inesperado: (e)")
```

## Sistema de reporte e análise de ocorrências: uma plataforma digital para segurança do trabalho

Ao iniciar o código na parte de funcionários, fica dessa maneira:

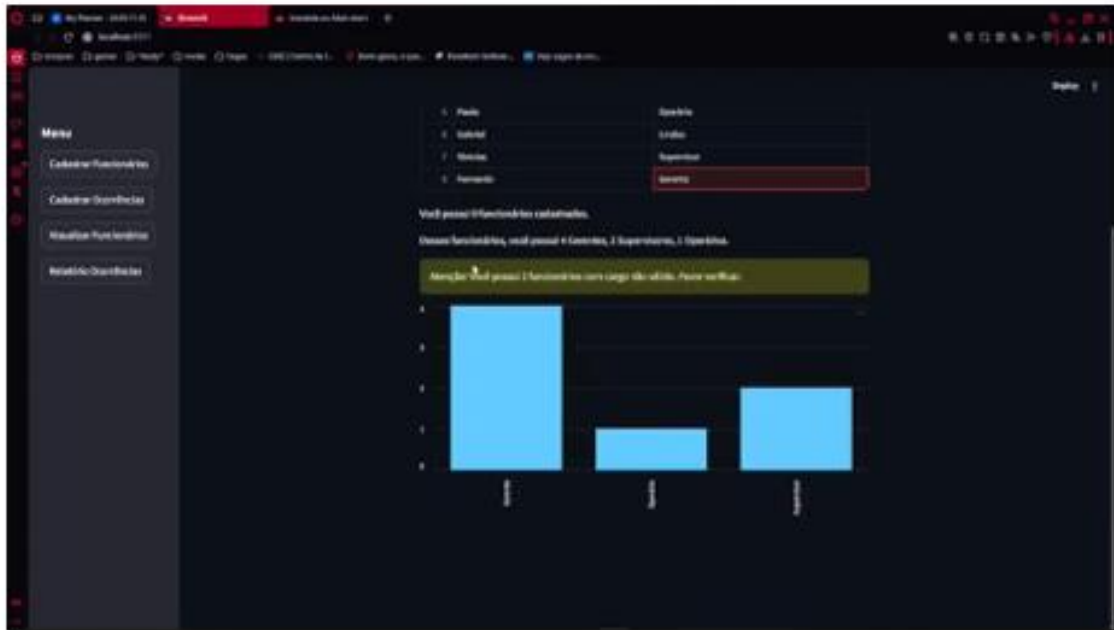


Após cadastrar um funcionário, aparece a seguinte mensagem:



Ainda na aba de funcionários, aparece um menu para visualização de cada colaborador cadastrado:





O código também tem a opção de armazenar ocorrências. Dessa forma, ficou assim:

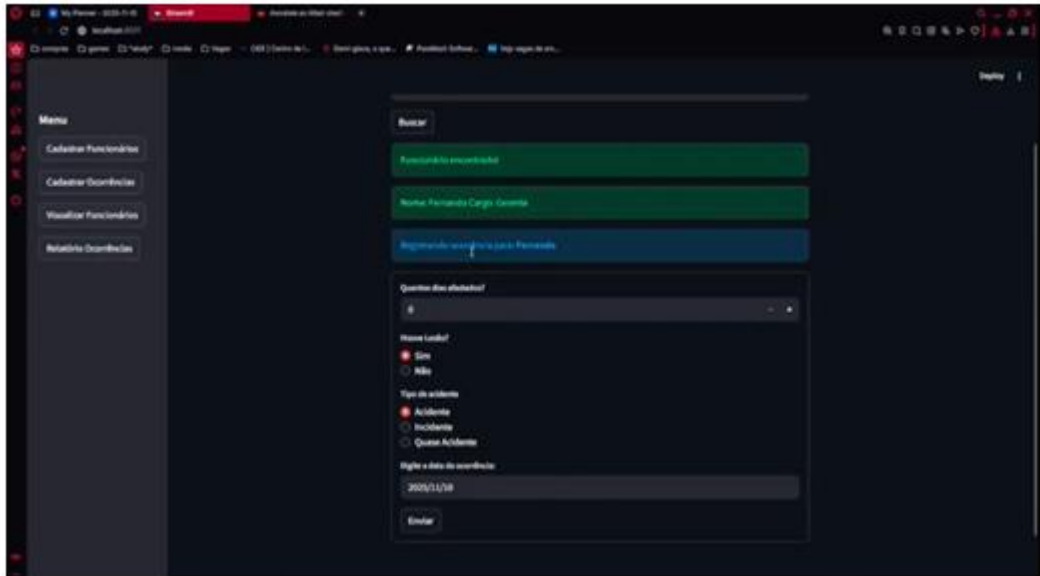
```
1 import pandas as pd
2 import os
3 from openpyxl import load_workbook
4 import streamlit as st
5
6 #Classe responsável pelo cadastro de ocorrências
7 class Ocorrências:
8     def __init__(self):
9         self.pathArquivo = "funcionarios.xlsx"
10        self.occurencia = {}
11
12    #função que verifica se o funcionário está ou não cadastrado
13    def acharfuncionario(self):
14        if not os.path.exists(self.pathArquivo):
15            st.warning("Arquivo de funcionários não encontrado.")
16            return False
17
18        #input de nome
19        nome = st.text_input("Qual o nome do funcionário?", key="nome")
20        #busca do funcionário
21        if st.button("Buscar"):
22            df = pd.read_excel(self.pathArquivo)
23            encontrado = df[df["nome"] == nome]
24            #retorna True ou False se o funcionário existe e armazena seu cargo e nome em um dicionário
25            if not encontrado.empty:
26                funcionario = encontrado.iloc[0]
27                st.success("Funcionário encontrado!")
28                st.success(f"Nome: {funcionario['nome']} | Cargo: {funcionario['cargo']}")
29                st.session_state.funcionario_encontrado = {
30                    "nome": funcionario["nome"],
31                    "cargo": funcionario["cargo"]
32                }
33                return True
34            else:
35                st.warning("Funcionário inexistente, cadastre-o primeiro.")
36                return False
37
38        #se já encontrou antes, também
39        if "funcionario_encontrado" in st.session_state:
40            return True
41        return False
42
43    #método e registra a ocorrência em um dicionário
44    def registrarOcorrência(self):
45        #retorna a função para achar o funcionário
46        if self.acharfuncionario():
47            funcionario = st.session_state.get("funcionario_encontrado", {})
48            st.info(f"Registrando ocorrência para: {funcionario.get('nome', '')}")
49
50            #formulário de cadastro
51            with st.form("form_ocorrencias"):
```

## Sistema de reporte e análise de ocorrências: uma plataforma digital para segurança do trabalho

```
60
61
62 Click to add a breakpoint (ocorrência de cadastro)
63
64 with st.form("form_ocorrencias"):
65     days = st.number_input("Quantos dias afastado?", min_value=0, max_value=15, step=1)
66     injury = st.radio("Novo lesão", ["Sim", "Não"])
67     tipo = st.radio("Tipo de acidente", ["Acidente", "Incidente", "Quase Acidente"])
68     date = st.date_input("Digite a data de ocorrência:")
69     submit = st.form_submit_button("Enviar")
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

No menu do servidor web aparece dessa forma:





O sistema inclui uma etapa de geração de relatórios, que reúne e apresenta as informações previamente cadastradas no código de forma organizada.

```
1 # Importando bibliotecas e frameworks
2 import pandas as pd
3 import streamlit as st
4 import os
5
6 class Relatorios():
7     """Classe responsável pela criação de relatórios com base nos registros já feitos."""
8     #le as duas abas de planilha
9     def __init__(self):
10         self.path_arq = "funcionarios.xlsx"
11         self.funcionarios = pd.read_excel(self.path_arq, sheet_name="Funcionários")
12         self.occorrencias = pd.read_excel(self.path_arq, sheet_name="Ocorrências")
13         self.occorrenciasMes = {}
14
15     #relatório simples de quantidade de funcionários e cargos
16     def relatorioFunc(self):
17         func_df = pd.DataFrame(self.funcionarios)
18         oc_df = pd.DataFrame(self.occorrencias)
19         cargos_validos = ["Gerente", "Supervisor", "Operário"]
20         st.write(func_df)
21         num_func = 0
22         leng = 0
23         qnt_gerente = 0
24         qnt_supervisor = 0
25         qnt_invalid = 0
26         qnt_operarios = 0
27         for i in func_df['nome']:
28             leng += 1
29         for j in range(leng):
30             cargo = func_df.iloc[j][1]
31             if cargo == cargos_validos[0]:
32                 qnt_supervisor += 1
33             elif cargo == cargos_validos[1]:
34                 qnt_gerente += 1
35             elif cargo == cargos_validos[2]:
36                 qnt_operarios += 1
37             else:
38                 qnt_invalid += 1
39                 num_func += 1
40         st.write(f"Você possui {num_func} funcionários cadastrados.")
41         st.write(f"Destes funcionários, você possui {qnt_gerente} Gerentes, {qnt_supervisor} Supervisores, {qnt_operarios} Operários.")
42         if qnt_invalid != 0:
43             st.warning(f"Atenção! Você possui {qnt_invalid} funcionários com cargo não válido. Favor verificar.")
44
45         func_df_filtrado = func_df[func_df['cargo'].isin(cargos_validos)]
46         cargo_num = func_df_filtrado['cargo'].value_counts().reset_index()
47         cargo_num.columns = ['cargo', 'quantidade']
48         st.bar_chart(cargo_num.set_index('cargo'))
49
50     def relatorioOcc(self):
51         st.subheader("Relatório de ocorrências por mês")
```

```

51 st.subheader("Relatório de Ocorrências por Mês")
52
53 def atualizaMes():
54     st.session_state["mes"] = st.session_state["mes_selecionado"]
55
56     st.selectbox("Qual mês?",
57                 ("Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",
58                 "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro", "Ano"),
59                 key="mes_selecionado",
60                 on_change=atualizaMes
61             )
62     enviar = st.button("Gerar Relatório")
63     if enviar:
64         if "mes_selecionado" in st.session_state:
65             option = st.session_state["mes_selecionado"]
66             st.write("Exibindo relatório de {option}")
67
68             new_df = pd.DataFrame(self.occurencias)
69             new_df["Data"] = pd.to_datetime(new_df["Data"])
70
71             meses = {
72                 "Janeiro": 1, "Fevereiro": 2, "Março": 3, "Abril": 4,
73                 "Maio": 5, "Junho": 6, "Julho": 7, "Agosto": 8,
74                 "Setembro": 9, "Outubro": 10, "Novembro": 11, "Dezembro": 12, "Ano": 13
75             }
76             mes_num = meses[option]
77             if mes_num != 13:
78                 df_mes = new_df[new_df["Data"].dt.month == mes_num]
79                 st.dataframe(df_mes)
80                 efetivo = 223
81                 hnt = round(efetivo * 220, 2)
82                 total_acidentes = (df_mes["Tipo"] == "Acidente").sum()
83                 total_dias_afastados = df_mes[df_mes["Tipo"] == "Acidente"]["Dias afastados"].sum()
84                 tf = round((total_acidentes * 1.000.000) / hnt, 2)
85                 tg = round((total_dias_afastados * 1.000.000) / hnt, 2)
86
87                 self.occurenciasMes = {
88                     "Mês": option,
89                     "Efetivo": efetivo,
90                     "hnt": hnt,
91                     "Total Acidentes": total_acidentes,
92                     "Dias afastados": total_dias_afastados,
93                     "TF": tf,
94                     "TG": tg,
95                 }
96                 df_occurencia_mes = pd.DataFrame([self.occurenciasMes])
97                 st.write(df_occurencia_mes)

```

```

        self.salvarRelatorio(self.occurenciasMes)
        elif mes_num == 13:
            self.relatorioAnual()

def salvarRelatorio(self,data):
    df_new = pd.DataFrame([data])
    aba = "Relatórios"
    mes_novo = df_new.loc[0, "Mês"]

    if not os.path.exists(self.path_arq):
        with pd.ExcelWriter(self.path_arq, engine="openpyxl") as writer:
            df_new.to_excel(writer, sheet_name=aba, index=False)
            st.success("📁 Arquivo criado e relatório salvo com sucesso!")
            return

    try:
        df_existente = pd.read_excel(self.path_arq, sheet_name=aba)
    except ValueError:
        df_existente = pd.DataFrame()

    if not df_existente.empty and "Mês" in df_existente.columns:
        df_existente = df_existente[df_existente["Mês"] != mes_novo]
        df_atualizado = pd.concat([df_existente, df_new], ignore_index=True)
    else:
        df_atualizado = df_new

    with pd.ExcelWriter(self.path_arq, mode="a", engine="openpyxl", if_sheet_exists="replace") as writer:
        df_atualizado.to_excel(writer, sheet_name=aba, index=False)

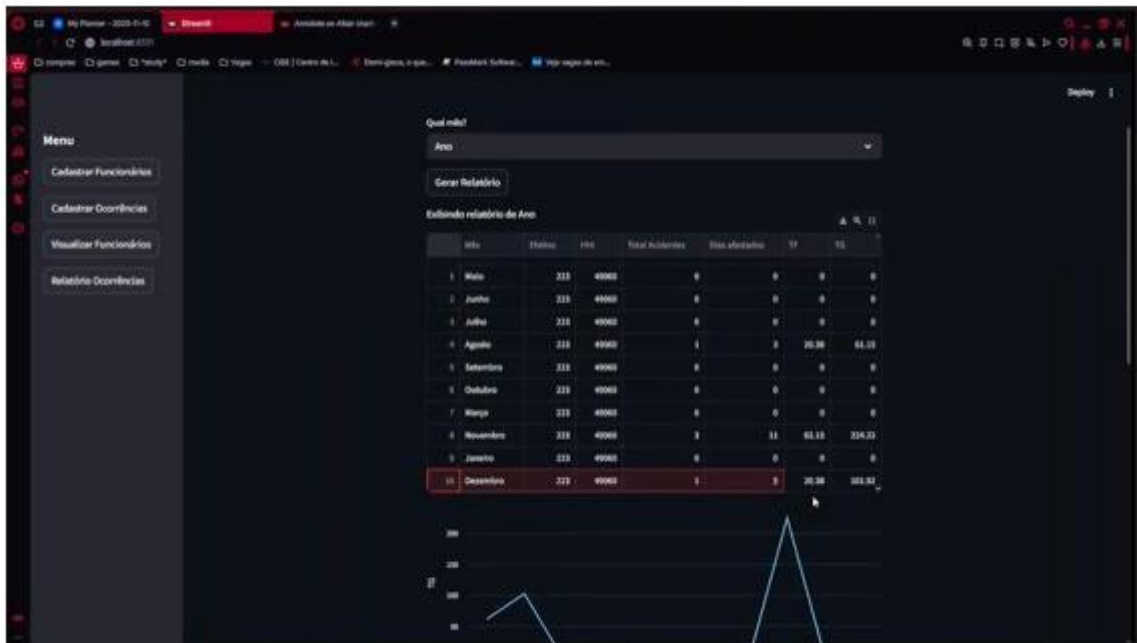
def relatorioAnual(self):
    relatorioAnual = pd.read_excel(self.path_arq, sheet_name="Relatórios")
    df_anual = pd.DataFrame(relatorioAnual)
    meses = [
        "Janeiro", "Fevereiro", "Março", "Abril",
        "Maio", "Junho", "Julho", "Agosto",
        "Setembro", "Outubro", "Novembro", "Dezembro"
    ]
    df_anual["Mês"] = pd.Categorical(df_anual["Mês"], categories=meses, ordered=True)

    # Reordena o DataFrame conforme os meses
    df_anual = df_anual.sort_values("Mês")

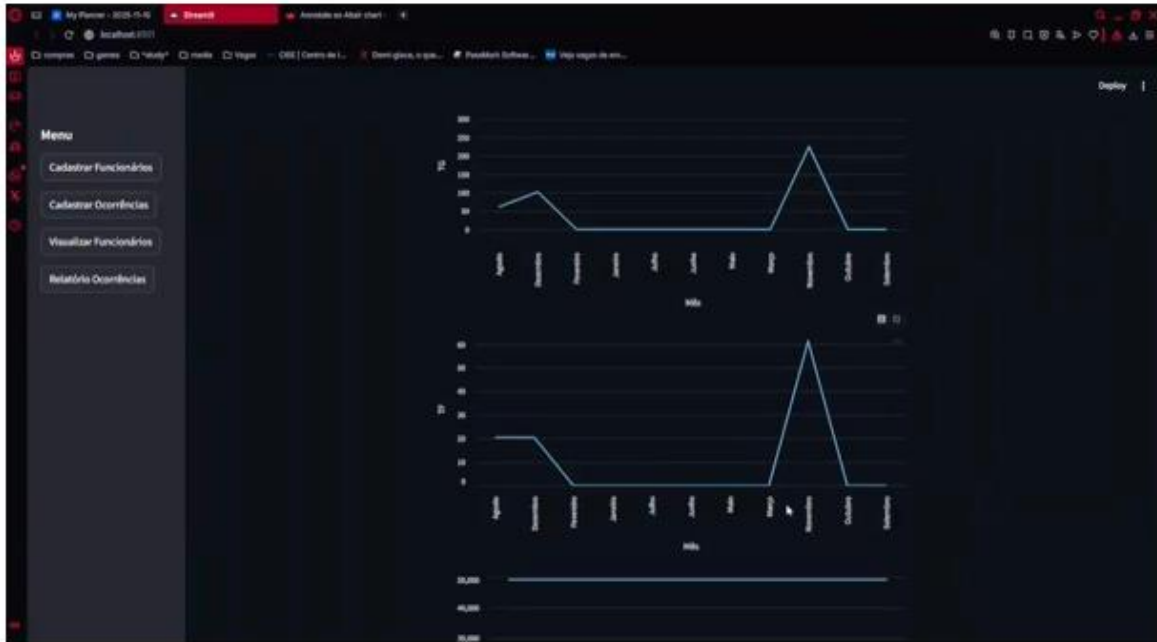
    st.write(df_anual)
    st.line_chart(df_anual, x="Mês", y="TG")
    st.line_chart(df_anual, x="Mês", y="TF")
    st.line_chart(df_anual, x="Mês", y="hnt")

```

A compilação do código gera o seguinte resultado:



## Sistema de reporte e análise de ocorrências: uma plataforma digital para segurança do trabalho



Portanto, o resultado da implementação do código reflete o desenvolvimento das suas principais funções, desde o cadastro e armazenamento das informações até a geração automatizada de relatórios. Dessa forma, o código comprova sua capacidade de executar as tarefas propostas de forma clara, eficiente e organizada.

### Considerações Finais

O desenvolvimento do Sistema de Reporte e Análise de Ocorrências demonstrou a viabilidade de criar uma solução tecnológica acessível e funcional voltada à gestão de

segurança no trabalho, ao mesmo tempo em que proporcionou ao grupo a oportunidade de aplicar, na prática, os conhecimentos adquiridos ao longo da disciplina de Lógica de Programação. Por meio da implementação de um sistema real, foi possível compreender de forma mais ampla os princípios da programação, da manipulação de dados e da construção de ferramentas voltadas para problemas concretos.

Ao longo do projeto, todas as etapas do desenvolvimento de software foram vivenciadas, desde o planejamento inicial e a divisão de tarefas até os testes, refinamentos e validações finais. O uso de Python e de bibliotecas como Pandas, OpenPyXL e Streamlit mostrou-se adequado para o escopo proposto, permitindo a criação de uma plataforma simples, porém eficiente, para registro, acompanhamento e análise de ocorrências de segurança. Além disso, ferramentas como o GitHub contribuíram para a organização do grupo e para o controle de versões, reforçando a importância da colaboração no processo de desenvolvimento.

A validação realizada junto à empresa Meireles de Freitas confirmou a aplicabilidade prática do sistema, evidenciando sua utilidade em um contexto real e identificando pontos de melhoria para futuras versões. O envolvimento dos usuários finais mostrou-se essencial, especialmente em sistemas destinados a ambientes organizacionais específicos, pois o feedback obtido durante os testes de usabilidade foi decisivo para aperfeiçoar a interface e ajustar funcionalidades.

Apesar dos resultados satisfatórios, o projeto também revelou desafios técnicos que deverão ser enfrentados nas próximas etapas de evolução da solução, como a necessidade de maior escalabilidade no armazenamento de dados e a adoção de mecanismos de segurança mais robustos. A migração para um banco de dados relacional e a implementação de autenticação avançada representam caminhos naturais para versões futuras. Além disso, recomenda-se a expansão das funcionalidades analíticas, incluindo recursos baseados em inteligência artificial para detecção automática de padrões e previsão de riscos, bem como a integração com outros sistemas corporativos ou com APIs de órgãos reguladores.

Em síntese, o trabalho cumpriu seus objetivos e contribuiu significativamente para o aprendizado técnico e pessoal dos participantes. Mesmo sendo um projeto acadêmico, a experiência adquirida servirá de base para o desenvolvimento de soluções mais complexas, reforçando a importância da prática, da colaboração e da interação com demandas reais no processo de formação profissional.

## Referências Bibliográficas

ABREU, Vinicius Pinheiro de et al. **Sistema de reporte e análise de ocorrências**: uma plataforma digital para segurança do trabalho. 2025.

ALPARGATAS S.A. **Comunicação, investigação e análise de acidentes e incidentes**. Procedimento nº PC.ST.00.002.06. Emissão: dez. 1999. Revisão: out. 2015.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). **NBR 14280:2001**: cadastro de acidente do trabalho: procedimento e classificação. Rio de Janeiro: ABNT, 2001.

BRASIL. Ministério do Trabalho e Emprego. **Norma Regulamentadora NR-1**: disposições gerais. Brasília, 2023.

BRASIL. Ministério do Trabalho e Emprego. **Norma Regulamentadora NR-4**: serviços especializados em engenharia de segurança e em medicina do trabalho. Brasília, 2023.

- BROWN, A.; GREEN, T. Technology adoption in organizational contexts. **Journal of Business Technology**, v. 15, n. 2, p. 45-62, 2022.
- CAVALCANTE, Zedequias Vieira; SILVA, Mauro Luis Siqueira da. **A importância da Revolução Industrial no mundo da tecnologia**. 2011.
- CHACON, S.; STRAUB, B. **Pro Git**. 2. ed. Apress, 2021.
- CHEN, X.; LI, Y. Integrated management systems: safety, quality and environment. **Safety Science**, v. 135, 105091, 2021.
- CONFEDERAÇÃO NACIONAL DA INDÚSTRIA (CNI). **Indústria 4.0**: sondagem mostra que sete em cada dez empresas no Brasil utilizam tecnologias digitais. 2022.
- EDMONDSON, A. C. **The Fearless Organization**: creating psychological safety in the workplace for learning, innovation, and growth. Wiley, 2018.
- FOWLER, M. **Patterns of Enterprise Application Architecture**. Addison-Wesley, 2020.
- GARCIA, M.; MARTINEZ, P. User-centered design in safety management systems. **International Journal of Industrial Ergonomics**, v. 89, 103245, 2022.
- GROOT, R. et al. **OpenPyXL**: Python Excel documentation. 2022.
- HOWARD, M. et al. **Security Development Fundamentals**. Microsoft Press, 2022.
- JAIN, R. **The Art of Computer Systems Performance Analysis**. Wiley, 2023.
- JOHNSON, P.; BROWN, K. Human factors in safety reporting systems. **Ergonomics**, v. 66, n. 3, p. 412-425, 2023.
- LAMATTINA, Alexandre de Araújo; MORAIS, Rúbia Carla Ramires. **Segurança e saúde no trabalho**: guia prático para técnicos. [S.l.]: Editora MultiAtual, 2024.
- LEE, S. et al. Integrating incident reporting and accident investigation. **Safety Science**, v. 158, 105998, 2023.
- MARTINEZ, P. et al. Streamlit for rapid prototyping of data applications. **Journal of Open Source Software**, v. 8, n. 85, 4567, 2023.
- MCKINNEY, W. **Python for Data Analysis**. 3. ed. O'Reilly Media, 2022.
- NIELSEN, J. **Usability Engineering**. Morgan Kaufmann, 2020.
- OLIVEIRA, R. et al. Low-cost safety management systems for SMEs. **Safety and Health at Work**, v. 14, n. 1, p. 112-125, 2023.
- ORGANIZAÇÃO INTERNACIONAL DO TRABALHO (OIT). **Recomendação nº 198** relativa à relação de trabalho, 2006.
- ORGANIZAÇÃO INTERNACIONAL DO TRABALHO (OIT). **Working time in the twenty-first century**: findings from the new ILO dataset. Genebra: OIT, 2011.
- PATEL, S.; KUMAR, R. ROI analysis in safety management systems. **Journal of Safety Research**, v. 84, p. 278-291, 2023.
- PRESSMAN, R. S. **Software Engineering**: a practitioner's approach. 9. ed. McGraw-Hill, 2021.
- PYTHON SOFTWARE FOUNDATION. **Python 3.9 Documentation**. 2023.
- REASON, J. **Managing the Risks of Organizational Accidents**. Routledge, 2016.
- RODRIGUES, C.; FERNANDES, M. Training effectiveness in safety systems implementation. **Safety Science**, v. 147, 105598, 2022.

Vinicius Pinheiro de Abreu, Antonio Carlos Almeida de Menezes Filho, Fernando Pedro Rodrigues de Pinho Paz, Gabriel Araújo Alves, Ysmael Ramon Silva Madeira, Juvenaldo Florentino Canja

SILVA, A. et al. Data privacy in occupational safety systems. **Computers & Security**, v. 124, 102956, 2023.

SMITH, J. et al. Data-driven safety management. **Safety Science**, v. 148, 105643, 2022.

SPINELLIS, D. Effective documentation for software projects. **IEEE Software**, v. 38, n. 2, p. 75-81, 2021.

TAYLOR, M.; WILSON, J. Software maintenance and evolution. **ACM Computing Surveys**, v. 55, n. 4, p. 1-37, 2023.

THE PANDAS DEVELOPMENT TEAM. **pandas-dev/pandas**: Pandas 1.5.0. Zenodo, 2023.

THOMPSON, K.; DAVIS, L. Emerging technologies in safety management. **Safety Science**, v. 162, 106102, 2024.

VINODKUMAR, M. N.; BHASI, M. Safety management system implementation and safety performance. **Safety Science**, v. 139, 105261, 2021.

WANG, H.; ZHANG, Y. Cross-cultural factors in safety reporting. **Accident Analysis & Prevention**, v. 178, 106845, 2023.

WHITTAKER, J. A. et al. **How to Break Software**: a practical guide to testing. Addison-Wesley, 2021.



## Editorial

### Editor-chefe:

Vicente de Paulo Augusto de Oliveira Júnior  
Centro Universitário Fanor Wyden  
[vicente.augusto@wyden.edu.br](mailto:vicente.augusto@wyden.edu.br)

### Editora responsável:

Ozângela de Arruda Silva  
Centro Universitário Fanor Wyden  
[ozangela.arruda@wyden.edu.br](mailto:ozangela.arruda@wyden.edu.br)

### Autor(es):

Vinicius Pinheiro de Abreu  
Centro Universitário Fanor Wyden  
[202309034778@alunos.unifanor.edu.br](mailto:202309034778@alunos.unifanor.edu.br)  
Contribuição: *Investigação, escrita e desenvolvimento do texto.*

Antonio Carlos Almeida de Menezes Filho  
Centro Universitário Fanor Wyden  
[202402530674@alunos.unifanor.edu.br](mailto:202402530674@alunos.unifanor.edu.br)  
Contribuição: *Investigação, escrita e desenvolvimento do texto.*

Fernando Pedro Rodrigues de Pinho Paz  
Centro Universitário Fanor Wyden  
[202508884593@alunos.unifanor.edu.br](mailto:202508884593@alunos.unifanor.edu.br)  
Contribuição: *Investigação, escrita e desenvolvimento do texto.*

Gabriel Araújo Alves  
Centro Universitário Fanor Wyden  
[202508245051@alunos.unifanor.edu.br](mailto:202508245051@alunos.unifanor.edu.br)  
Contribuição: *Investigação, escrita e desenvolvimento do texto.*

Ysmael Ramon Silva Madeira  
Centro Universitário Fanor Wyden  
[202508673592@alunos.unifanor.edu.br](mailto:202508673592@alunos.unifanor.edu.br)  
Contribuição: *Investigação, escrita e desenvolvimento do texto.*

Juvenaldo Florentino Canja  
Centro Universitário Fanor Wyden  
[juvenaldo.canja@professores.unifanor.edu.br](mailto:juvenaldo.canja@professores.unifanor.edu.br)  
Contribuição: *Investigação, orientação, escrita e desenvolvimento do texto.*

**Submetido em:** 08.04.2025

**Aprovado em:** 09.04.2026

**Publicado em:** 09.04.2026

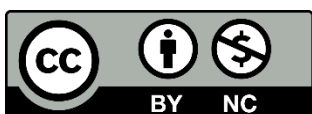
**DOI:** 10.5281/zenodo.20680656

### Financiamento:

### Como citar este trabalho:

ABREU, Vinicius Pinheiro de; MENEZES FILHO, Antonio Carlos Almeida de; PAZ, Fernando Pedro Rodrigues de Pinho; ALVES, Gabriel Araújo; MADEIRA, Ysmael Ramon Silva; CANJA, Juvenaldo Florentino. SISTEMA DE REPORTE E ANÁLISE DE OCORRÊNCIAS: UMA PLATAFORMA DIGITAL PARA SEGURANÇA DO TRABALHO. **Duna: Revista Multidisciplinar de Inovação e Práticas de Ensino**, [S. l.], v. 2, n. 1, p. 75–102, 2026. DOI: 10.5281/zenodo.20680656. Disponível em: <https://wyden.periodicoscientificos.com.br/index.php/jornadacientifica/article/view/1401>. Acesso em: 13 jun. 2026. (ABNT)

Abreu, V. P. de, Menezes Filho, A. C. A. de, Paz, F. P. R. de P., Alves, G. A., Madeira, Y. R. S., & Canja, J. F. (2026). SISTEMA DE REPORTE E ANÁLISE DE OCORRÊNCIAS: UMA PLATAFORMA DIGITAL PARA SEGURANÇA DO TRABALHO. *Duna: Revista Multidisciplinar De Inovação E Práticas De Ensino*, 2(1), 75–102. <https://doi.org/10.5281/zenodo.20680656> (APA)



© 2026 Duna – Revista Multidisciplinar de Inovação e Práticas de Ensino. Centro Universitário Fanor Wyden – UniFanor Wyden. Este trabalho está licenciado sob uma licença *Creative Commons* Atribuição - Não comercial - Compartilhar 4.0 Internacional CC-BY NC 4.0 Internacional).