Uma imagem contendo Texto

Descrição gerada automaticamente

### **FACULDADE ESTÁCIO DE SÁ**

**DOUGLAS SIQUEIRA TINOCO**

Curso Sistemas de Informação

**SISTEMA DE GERAÇÃO DE RELATÓRIOS**

**VILA VELHA – ES**

**2025**

### **FACULDADE ESTÁCIO DE SÁ**

**DOUGLAS SIQUEIRA TINOCO**

**SISTEMA DE GERAÇÃO DE RELATÓRIOS:**

**Uma solução automatizada para consultas SQL frequentes**

**Trabalho de Conclusão de Curso apresentado ao curso de Sistemas de Informação, da FACULDADE ESTÁCIO DE SÁ, como requisito parcial para a Obtenção do grau de Bacharel em Sistemas de Informação.**

**VILA VELHA – ES**

**2025**

### **RESUMO**

Este trabalho aborda o desenvolvimento de um sistema automatizado para a geração de relatórios em formato CSV, voltado para organizações que demandam consultas frequentes a bancos de dados. O sistema propõe uma solução que permite o cadastro e execução de consultas SQL sob demanda, associado a um módulo de controle de acesso que garante a segurança das informações, restringindo o uso apenas a usuários autorizados. O objetivo principal é otimizar o processo de extração de dados, reduzindo o tempo gasto em atividades manuais e aumentando a eficiência operacional dos setores envolvidos.

O sistema foi projetado para oferecer uma interface amigável e funcionalidades robustas, integrando mecanismos de autenticação e autorização, logs e dashboard para garantir o compliance organizacional.

Os resultados esperados incluem a redução de retrabalho, a minimização de erros humanos no processo de geração de relatórios e a melhoria na segurança da informação. Além disso, o sistema tem o potencial de aumentar a produtividade dos usuários finais ao automatizar tarefas repetitivas. Conclui-se que a solução apresentada é viável e pode ser aplicada em diversos cenários corporativos, especialmente em empresas que lidam com grandes volumes de dados e relatórios frequentes.

**Palavras-chave:** geração de relatórios. controle de acesso. segurança da informação. automatização de processos.

### **SUMÁRIO**

[1. INTRODUÇÃO 5](#_Toc897004313)

[2. O PROBLEMA 6](#_Toc1961651073)

[2.1 Tema do Trabalho 6](#_Toc414082331)

[2.2 Contextualização 6](#_Toc469402723)

[2.3 A Situação-Problema 6](#_Toc1545212292)

[2.4 Breve Descrição da Solução 7](#_Toc157933474)

[3. A SOLUÇÃO 8](#_Toc1106103454)

[3.1 O Sistema Proposto 8](#_Toc667508267)

[3.1.1 Justificativas para o Novo Sistema 8](#_Toc913937950)

[3.1.2 Situação Desejada 8](#_Toc1883285917)

[3.2 Escopo da Solução 8](#_Toc1512007)

[3.2.1 Lista de Requisitos do Sistema 8](#_Toc445851650)

[4. DESENVOLVIMENTO 9](#_Toc457002902)

[4.1 Ferramentas Utilizadas 9](#_Toc897062852)

[4.2 Arquitetura do Sistema 9](#_Toc979907017)

[4.1. Camada de Apresentação (Frontend) 9](#_Toc805496901)

[4.2. Camada de Lógica de Negócio (Backend) 9](#_Toc832788819)

[4.3. Camada de Persistência de Dados (Banco de Dados) 10](#_Toc659853344)

[5. FLUXO DE DADOS 11](#_Toc1982763241)

[6. SEGURANÇA E AUTENTICAÇÃO 12](#_Toc1219879219)

[CONCLUSÃO 13](#_Toc306822885)

[REFERÊNCIAS 14](#_Toc690193594)

### **1. INTRODUÇÃO**

A geração de relatórios é uma atividade essencial em diversos setores organizacionais, sendo amplamente utilizada para suportar tomadas de decisão e monitoramento de indicadores. Contudo, muitas empresas ainda dependem de processos manuais e ineficientes, que consomem tempo e aumentam a probabilidade de erros. Neste contexto, o presente trabalho propõe o desenvolvimento de um sistema automatizado para geração de relatórios em formato CSV, oferecendo uma alternativa prática e segura para o processamento de dados.

O tema é relevante devido à crescente necessidade de otimizar operações organizacionais, especialmente em um ambiente corporativo que demanda maior agilidade e segurança na gestão da informação. A proposta deste trabalho busca solucionar problemas relacionados à execução manual de consultas SQL e à falta de mecanismos de controle de acesso, garantindo que somente usuários autorizados possam acessar informações sensíveis.

O problema central investigado é a falta de sistemas acessíveis e robustos que possibilitem a geração automatizada de relatórios com controle de acesso integrado. Dentre os objetivos deste trabalho, destacam-se: desenvolver uma interface intuitiva para o cadastro e execução de consultas SQL, implementar funcionalidades que permitam o download seguro de relatórios e criar mecanismos de autenticação e autorização para assegurar o compliance organizacional.

A metodologia empregada envolve a definição de requisitos funcionais e não funcionais, o uso de boas práticas de engenharia de software e o planejamento detalhado das etapas de desenvolvimento. Ferramentas modernas foram utilizadas para a implementação do sistema, incluindo tecnologias que garantem escalabilidade e segurança.

Este estudo é relevante porque, ao automatizar processos repetitivos e garantir a segurança dos dados, contribui para a eficiência operacional e para a redução de custos organizacionais. A solução proposta, ao aliar praticidade e segurança, atende às necessidades de empresas que lidam com grandes volumes de dados e relatórios recorrentes, demonstrando potencial para ser aplicada em diversos cenários corporativos.

### **2. O PROBLEMA**

#### **2.1 Tema do Trabalho**

Desenvolvimento de um sistema que automatiza a geração e o download de relatórios a partir de queries SQL, com controle de acesso para garantir segurança e gerenciamento eficiente.

#### **2.2 Contextualização**

Empresas que dependem de relatórios frequentes para decisões estratégicas frequentemente enfrentam dificuldades em organizar e acessar informações rapidamente. Os processos manuais são demorados, propensos a erros e não oferecem um controle adequado sobre quem pode acessar informações sensíveis.

#### **2.3 A Situação-Problema**

Um setor da empresa necessita frequentemente de relatórios baseados em consultas SQL para suas operações. No cenário atual, o processo envolve a seleção manual de relatórios previamente cadastrados, permitindo que os usuários façam o download diretamente da interface do sistema. Embora funcional, o modelo atual apresenta desafios relacionados à eficiência, rastreabilidade e segurança no acesso às informações.

Um ponto crítico é a ausência de um mecanismo automatizado e robusto que registre de forma detalhada as interações dos usuários com o sistema. Apesar de permitir o download, o controle de acesso aos relatórios e o rastreamento das ações realizadas (como horário do download, nome do responsável e relatório baixado) ainda não são gerenciados de forma centralizada e segura.

Essas limitações podem gerar gargalos operacionais, dificultar auditorias e aumentar os riscos associados ao acesso não autorizado ou uso indevido das informações.

O problema, portanto, consiste na necessidade de uma solução que otimize o acesso aos relatórios, assegure a rastreabilidade completa das operações realizadas e garanta a segurança das informações corporativas, alinhando-se às melhores práticas de governança de dados

#### **2.4 Breve Descrição da Solução**

Propõe-se o desenvolvimento de um sistema automatizado que:

1. Permite o cadastro de consultas SQL.
2. Executa essas consultas sob demanda.
3. Gera os relatórios no formato CSV para download.
4. Garante controle de acesso por meio de autenticação e permissões específicas.
5. Guarda log de acesso ao sistema, e log de download de cada relatório.

### **3. A SOLUÇÃO**

#### ****3.1 O Sistema Proposto****

##### **3.1.1 Justificativas para o Novo Sistema**

* Reduzir o tempo gasto na geração de relatórios.
* Aumentar a segurança com controle de acesso.
* Melhorar a eficiência operacional ao automatizar tarefas repetitivas.

##### **3.1.2 Situação Desejada**

O sistema permitirá que os usuários autenticados selecionem e baixem relatórios previamente configurados, de maneira rápida e segura, sem intervenção manual do analista.

##### **3.2 Escopo da Solução**

* Sistema web com autenticação de usuários.
* Cadastro de consultas SQL no sistema.
* Permissões específicas para usuários.
* Geração de relatórios em CSV sob demanda.

##### **3.2.1 Lista de Requisitos do Sistema**

* **Funcionais:**
  + Login e autenticação de usuários.
  + Cadastro e execução de consultas SQL.
  + Geração de arquivos CSV.
  + Log de Acesso
  + Histórico de downloads de cada relatório
  + Dashboard simples
* **Não Funcionais:**
  + Interface intuitiva.
  + Resposta rápida para consultas de até 1 milhão de registros.
  + Segurança contra injeções SQL.

### **4. DESENVOLVIMENTO**

#### **4.1 Ferramentas Utilizadas**

* **Linguagem de Programação:** Python.
* **Frameworks:** Flask para o backend, HTML / CSS / Bootstrap para o frontend.
* **Banco de Dados:** PostgreSQL.
* **Outros:** Pandas para manipulação de dados e SQLAlchemy para ORM.

#### **4.2 Arquitetura do Sistema**

A arquitetura do sistema pode ser dividida em três camadas principais:

#### **4.1. **Camada de Apresentação (Frontend)****

* **HTML**: Utilizado para estruturar as páginas web que são exibidas ao usuário.
* **CSS**: Usado para o design e estilização das páginas HTML, proporcionando uma interface agradável.
* **JavaScript (Opcional)**: Pode ser usado para interações dinâmicas na interface do usuário, como validação de formulários ou ações em tempo real.

#### **4.2. **Camada de Lógica de Negócio (Backend)****

* **Flask (Python)**: Framework leve utilizado para o desenvolvimento do backend. O Flask é responsável por gerenciar as requisições HTTP, realizar a comunicação com o banco de dados, gerar relatórios e servir esses relatórios ao usuário.
* **Roteamento (Routing)**: O Flask define as rotas para as requisições (GET, POST, etc.), como por exemplo, /gerar\_relatorio ou /download\_relatorio/<id>.
* **Geração de Relatórios**: A geração dos relatórios pode ser feita utilizando bibliotecas Python (como Pandas).
* **Processamento de Dados**: O Flask interage com o banco de dados PostgreSQL para recuperar ou manipular os dados necessários para a criação dos relatórios.

#### **4.3. **Camada de Persistência de Dados (Banco de Dados)****

* **PostgreSQL**: Banco de dados relacional utilizado para armazenar as informações necessárias para os relatórios. Ele pode conter tabelas com dados de usuários, transações ou qualquer outra informação relevante para o sistema.
* **Consultas SQL**: O Flask irá executar consultas SQL no banco PostgreSQL para recuperar os dados, que serão então utilizados na geração dos relatórios.

### **5. **FLUXO DE DADOS****

1. **Requisição do Usuário**: O usuário faz uma requisição através da interface (frontend) para gerar um relatório específico.
2. **Processamento Backend**: O Flask recebe a requisição, verifica os parâmetros e, se necessário, consulta o banco de dados PostgreSQL para obter os dados que serão utilizados na geração do relatório.
3. **Geração do Relatório**: O sistema processa os dados e cria um relatório no formato csv
4. **Download**: O relatório gerado é oferecido para o download ao usuário.

### **6. **SEGURANÇA E AUTENTICAÇÃO****

* **Autenticação**: O sistema conterá autenticação de usuários, utilizando, por exemplo, a extensão Flask-Login para gerenciar sessões de usuários.
* **Validação de Dados**: O Flask pode validar os dados recebidos do frontend para garantir que as requisições estejam no formato correto.
* **Proteção contra ataques**: Será implementado medidas de segurança como proteção contra CSRF (Cross-Site Request Forgery) e XSS (Cross-Site Scripting).
* **Logs e Auditorias:** O sistema contará com logs de acesso informando datas e horários, e auditorias de relatórios baixados para atender a LGP de dados

### **CONCLUSÃO**

O desenvolvimento de um sistema automatizado para a geração de relatórios frequentes, conforme proposto neste trabalho, apresenta-se como uma solução eficaz para otimizar processos organizacionais, especialmente em empresas que lidam com grandes volumes de dados. Através da implementação de funcionalidades como controle de acesso, geração de relatórios no formato CSV e autenticação de usuários, o sistema proporciona maior segurança, eficiência e praticidade na gestão da informação.

Ao longo do projeto, foram adotadas ferramentas modernas, como Flask, PostgreSQL e Pandas, que possibilitaram a construção de uma solução escalável e segura. A divisão clara entre as camadas de apresentação, lógica de negócio e persistência de dados garantiu a modularidade e a manutenção simplificada do sistema.

O impacto esperado é significativo, incluindo a redução do tempo gasto em atividades manuais, a diminuição de erros humanos e o aumento da produtividade dos usuários finais. Além disso, ao integrar mecanismos de autenticação e autorização robustos, o sistema contribui para o compliance organizacional e a proteção de informações sensíveis.

Portanto, conclui-se que o sistema proposto é viável, eficiente e alinhado às demandas de organizações que necessitam de soluções práticas para a extração e análise de dados. Sua aplicação pode ser estendida a diferentes cenários corporativos, com potencial para agregar valor estratégico às operações e decisões empresariais.

### **REFERÊNCIAS**

1. PYTHON SOFTWARE FOUNDATION. **Python 3 Documentation.** Disponível em: <https://docs.python.org/3/>. Acesso em: 12 jan. 2025.
2. POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL Documentation**. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 12 jan. 2025.
3. GITHUB. **Flask – Web Framework Documentation**. Disponível em: <https://flask.palletsprojects.com/>. Acesso em: 12 jan. 2025.
4. FLASK. **Flask Documentation.** Disponível em: https://flask.palletsprojects.com. Acesso em: 12 jan. 2025.
5. **FLASK-LOGIN. Flask-Login Documentation.** Disponível em: <https://flask-login.readthedocs.io/en/latest/>. Acesso em: 12 jan. 2025.
6. **PANDAS. Pandas Documentation.** Disponível em: <https://pandas.pydata.org/pandas-docs/stable/>. Acesso em: 12 jan. 2025.