

Segurança da Informação

Prof. Claudinei Dias (Ney)

N3 - Trabalho: Assinatura Digital

Objetivo:

O objetivo deste trabalho é desenvolver um sistema de gestão de relatórios, utilizando **Assinaturas Digitais** para garantir a autenticidade, integridade e não repúdio dos documentos. Proporcionando segurança no uso do sistema e confiabilidade no processo de aprovação dos relatórios.

Pipeline de Desenvolvimento:

1. Configuração do Ambiente de Desenvolvimento:
 - a. Utilize ferramentas modernas (foco em tecnologias que garantam a segurança da informação) como Docker com “políticas de segurança” para configuração do ambiente, incluindo contêineres para backend, banco de dados e servidores web.
 - b. Recomendações: Node.js para backend, MongoDB para banco de dados, e Next.js para frontend.
2. Banco de Dados:
 - a. Crie uma estrutura de banco de dados (Firebase ou MongoDB) para armazenar colaboradores, relatórios de despesas e assinaturas digitais.
 - b. Utilização de banco de dados em nuvem como estratégia de transferir os riscos de manter a segurança ao servidor de banco de dados à terceiros.
3. Desenvolvimento do Frontend (React/Next.js):
 - a. Crie interfaces de usuário interativas e responsivas para login, submissão de relatórios e aprovação de despesas.
 - b. Utilização de Material-UI para componentes de interface e Tailwind CSS para estilização.
4. Implementação do Backend (Node.js/Express):
 - a. Desenvolva a lógica de negócios, processamento de formulários e interações com o banco de dados.
 - b. Implementação de “autenticação” JWT (JSON Web Token) para segurança.
5. Integração com Bibliotecas de Criptografia:
 - a. Utilize bibliotecas para “operações de criptografia” e geração de “chaves”, como Node.js Crypto e Web Cryptography API.
 - b. Implementação de “assinaturas digitais” e “verificação” com Web Cryptography API.
6. Implementação de Segurança:
 - a. Implemente “medidas de segurança” como proteção contra injeção de SQL, XSS, CSRF, etc.
 - b. Utilização de Helmet para configurar cabeçalhos “HTTP de segurança”.
 - c. Publicar no GH-PAGES com https. Certificado TLS/SSL.
7. Requisitos mínimos:
 - **login.js (Next.js):** Página de Login para autenticação de usuários com senhas criptografadas.
 - **dashboard.js (Next.js):** Página Principal (Após Login) com opções para CRUD de funcionários e cadastro de relatórios de despesas.
 - **employees.js (Next.js):** CRUD de Funcionários, página para listar, cadastrar e editar funcionários.
 - **submitExpense.js (Next.js):** Cadastro de Relatório de Despesa, página para submissão de relatórios de despesas com upload de recibos.
 - **Notificação por E-mail:** Utilize Nodemailer para enviar notificações por e-mail.
 - **pendingExpenses.js (Next.js):** Página para listar e validar relatórios pendentes pelo Gerente.
 - **validateExpense.js (Next.js):** Página para visualizar e validar relatórios pelo Gerente.
 - **signExpense.js (Next.js):** Página para Assinatura Digital dos relatórios validados.
 - **signedExpenses.js (Next.js):** Página para listar relatórios assinados para verificação pelo Diretor.
 - **verifySignature.js (Next.js):** Página para o Diretor verificar a autenticidade das assinaturas digitais.
8. Tecnologias Recomendadas:
 - **Backend:** Node.js com Express.
 - **Frontend:** React com Next.js.
 - **Banco de Dados:** MongoDB ou Firebase.
 - **Criptografia:** Node.js Crypto e Web Cryptography API.
 - **E-mail:** Nodemailer.
 - **Autenticação:** JWT (JSON Web Token).