

Desenvolvimento Web para Departamento Pessoal

1. Objetivos

- Facilitar preenchimento de dados para os usuários (clientes), de forma que esses dados também cheguem mais facilmente nos contribuintes do departamento pessoal. Além de, gerar visualização de dados relevantes para o cliente.
- Elaborar login e senha para que os usuários não visualizem dados de outra empresa e tenham seu próprio acesso privado, **OS USUÁRIOS NÃO PODEM DE FORMA ALGUMA VISUALIZAR DADOS DE OUTRAS EMPRESAS QUE NÃO SEJA A DELE(A) PRÓPRIO(A).**
- Será criado uma área onde o usuário vai ter opções para cada funcionalidade específica, exemplo: inserir dados novos sobre transporte dos funcionários; editar dados sobre transporte dos funcionários e outras funções que adicionaremos de acordo com demandas.
- Desenvolver uma interface Web responsiva, de fácil entendimento e objetiva (com base em conhecimentos UX e UI). Precisa ser intuitiva, com poucos processos nos casos de uso, para que o usuário alcance o objetivo o mais rápido possível. Desenvolver uma arquitetura modular que permita adicionar novas funcionalidades sem impactar o sistema existente (escalabilidade).

2. Desenvolvimento Front-End

- **Tecnologias:** HTML | CSS | JS | React
- **Login:** O site será totalmente bloqueado enquanto o usuário não estiver logado, a primeira página do usuário sempre vai ser a requisição de login para poder acessar os conteúdos, caso o usuário não esteja logado, esse sempre vai ser o caso (logar 1º), sem estar logado o usuário terá opção de criar cadastro, logar e recuperar conta.
- **Home:** Na página principal o usuário visualizará “caixas” que terão descrições para qual área aquele link vai levar o usuário, por exemplo: Transporte de funcionários;

Contrato de funcionários e outros casos. Toda página vai possuir header e foot, com links e navegações de forma padrão.

- **Navegações principais:** Quando o usuário clicar em uma das caixas para uma área específica, ele será direcionado para uma página que vai conter outras caixas com funcionalidades específicas daquele assunto/área, exemplo: clicando em editar dados transporte, o usuário vai visualizar todos os funcionários cadastrados (puxado do banco de dados) e poderá selecionar algum deles para edição. Adicionar modos de lidar com muitos dados na página, implementando máximo por lista e navegação entre listas. Importante ter avisos de “erro”, “sucesso” e loadings para retorno com o usuário.
- **Atualização:** Sempre mostrar os dados atualizados conforme o banco de dados.
- **Finalização:** O usuário sempre terá a opção de deslogar na barra do header, ele também poderá se manter logado no site caso não queira deslogar.

3. Desenvolvimento Back-End

- **Tecnologias:** PHP | Laravel | MySQL | XAMPP
- **Login:** Quando o usuário fizer cadastro, login ou requisição de reset de senha, todos esses dados serão enviados para o banco de dados e logs, as atualizações de senhas e novos cadastros serão guardados no banco de dados, logins e logouts serão gravados em logs também, com horário e o usuário responsável pelo login.
- **Dados:** Quando o usuário preencher, atualizar, editar algum dado, será feita requisição para atualizar o banco de dados conforme, e ele entregará a visualização dos dados corretamente para o usuário quase em tempo real. E será salvo um log da modificação feita.
- **Segurança:** Implementar tratamentos para tentativas maliciosas em relação ao site (use HTTPS, criptografia para dados sensíveis e env., validação de todas as entradas do usuário para prevenir SQL injection e XSS, e limite tentativas de login para evitar brute force).

4. Documentação

- **Documentação:** Desenvolver documentação completa de cada sessão da aplicação e o que for necessário.
- A documentação será dividida em três categorias principais:
 - **Documentação Técnica:** Detalhes sobre arquitetura, tecnologias, APIs e banco de dados.
 - **Documentação de Segurança e Processos:** Procedimentos de segurança e controle de acessos.
 - **Documentação do Usuário:** Guia de uso e funcionalidades do sistema.
- **Documentação Técnica**
 - Arquitetura do Sistema
 - Banco de Dados
 - APIs e Requisições
 - Login e Autenticação
- **Documentação do Usuário**
 - Cadastro e Login
 - Interface e Navegação
 - Funcionalidades Disponíveis
- **Documentação de Segurança e Processos**
 - Segurança do Sistema
 - Controle de Acessos
 - Procedimentos de Manutenção

5. Testes e Feedbacks

- **Testes de Funcionalidade:** Testes unitários e de integração para validar cada funcionalidade. Simular uso por diferentes tipos de usuários, correção de bugs e melhorias antes da implantação.
- **Testes de Segurança:** Validar vulnerabilidades contra SQL Injection e XSS, realizar testes de autenticação e controle de acesso. Além de, avaliar criptografia e armazenamento seguro de senhas.
- **Testes de Usabilidade:** Realizar avaliação com usuários finais para verificar a experiência com a aplicação. Ajustes na interface e fluxos conforme feedbacks coletados.
- **Coleta de Feedbacks:** Enviar formulários de avaliação para os usuários e implementar melhorias com base nos feedbacks recebidos. Também é necessário a análise de métricas de uso e identificação de pontos de melhoria.

6. Projeto em Produção

- **Implantação do Projeto:** Configuração do ambiente de produção; deploy da aplicação em um servidor confiável e monitoramento inicial para detectar possíveis problemas.
- **Manutenção e Melhoria Contínuas:** Atualizações de segurança regulares; correção de bugs e monitoramento de performance. Implementação de novas funcionalidades conforme necessidade; revisão periódica da experiência do usuário e adaptação a novas tecnologias e padrões de desenvolvimento.
- **Feedbacks Contínuos:** Realizar coleta ativa de sugestões e reclamações dos usuários, disponibilizar canal de comunicação aberto para suporte e melhorias. E também implementar atualizações iterativas com base nas necessidades do usuário.