

Quais os próximos passos de melhoria? O que é a Fase 3?

A "Fase 3" é exatamente o que você descreveu: pegar nossa aplicação, que já é funcional, e transformá-la em um produto **excepcional**. É a fase de polimento, robustez e refinamento.

Aqui está um roadmap do que podemos fazer, cobrindo os pontos que você mencionou:

A. Para Deixar o Sistema Mais ROBUSTO:

1. **Validação de Entrada na API (Prioridade Alta):** Hoje, se o front-end enviar um nome vazio ou um preço negativo, nossa API pode aceitar. Precisamos ensinar o back-end a validar todos os dados que chegam, rejeitando informações inválidas. Isso garante a integridade total do nosso banco de dados. Usaremos uma biblioteca profissional chamada `express-validator` para isso.
2. **Melhorar Feedback de Erro e Carregamento no Front-end:** Em vez de `alert()`, podemos implementar um sistema de notificações mais elegante (chamados de "toasts") que aparecem no canto da tela. E, enquanto os dados carregam, podemos exibir "spinners" para que o usuário saiba que o sistema está trabalhando.

B. Para Deixar o Sistema Mais SEGURO:

1. **Proteção de Rotas no Front-end (Route Guards):** Nossa API já impede que um "Operador" execute ações de "Admin". Mas podemos ir além: podemos impedir que ele sequer **acesse a URL** de uma página de admin. O Vue Router tem uma ferramenta chamada "navigation guards" para isso.
2. **Página "Meu Perfil" (Sua Sugestão):** Esta é uma feature de segurança e usabilidade fundamental. Vamos criar uma página onde o usuário logado possa alterar a própria senha (de forma segura, claro, pedindo a senha antiga e a nova).

C. Para Deixar o Sistema Mais USÁVEL:

1. **Refatorar para Componentes:** Nossas páginas estão ficando grandes. Podemos começar a quebrar partes delas em componentes menores e reutilizáveis. Sua ideia da `Composable` para exportação foi o primeiro passo perfeito nessa direção. Podemos fazer o mesmo para tabelas, modais, etc.
2. **Responsividade:** Testar e garantir que a aplicação seja agradável de usar em telas de celular. O Tailwind CSS facilita muito isso.

Vai precisar rodar testes?

Sim. Em um projeto profissional, testes são essenciais. Eles são como o "seguro" do nosso código. Eles garantem que, ao adicionar uma nova funcionalidade (como a Folha de Pagamento), nós não quebramos acidentalmente uma antiga (como o login).

Não precisamos buscar 100% de cobertura agora, mas seria uma excelente prática começar a escrever alguns testes para as partes mais críticas da nossa API, como o cálculo da folha de pagamento, usando ferramentas como o `Vitest` (que é do mesmo ecossistema do Vite).

Dá pra hospedar em uma Hostinger?

Sim, com certeza. A Hostinger geralmente oferece hospedagem mais tradicional (servidores compartilhados ou VPS), onde você tem um ambiente Linux rodando 24/7.

É uma abordagem diferente da "serverless" da Vercel/Netlify, mas totalmente viável. O processo seria:

1. Fazer o deploy do nosso back-end Node.js para rodar como um serviço contínuo no servidor da Hostinger.
2. Fazer o deploy dos arquivos "construídos" (`dist`) do nosso front-end para serem servidos pelo mesmo servidor.

É uma ótima alternativa a se considerar se os problemas com as plataformas serverless persistirem.

Nosso Fluxo de Trabalho (Git)

Sua sugestão está perfeita. **Vamos trabalhar na `develop`.** Todo o nosso progresso a partir de agora será salvo em `feature` branches que serão mescladas na `develop`. A `main` só será atualizada quando tivermos um conjunto de funcionalidades estáveis e prontas para um "lançamento".

Nosso Próximo Passo Concreto

Dada essa lista de melhorias, minha sugestão é começarmos pelo pilar da **robustez**, que é o mais fundamental.

O que acha de começarmos implementando a **Validação de Entrada na API**? Ensinaremos nosso back-end a rejeitar dados inválidos, protegendo nosso banco de dados. Este é um passo gigantesco para tornar a aplicação verdadeiramente profissional.

Se estiver de acordo, nosso primeiro passo será criar uma nova branch a partir da `develop` e instalar o `express-validator` no nosso back-end.