Cate **COLABORADORES**

Detalhes Técnicos do Aplicativo Cate Para colaboradores

O aplicativo desenvolvido para a Igreja Católica está disponível de forma pública e pode ser acessado por qualquer pessoa para visualização e colaboração no código-fonte. O repositório está hospedado no GitHub e pode ser encontrado no seguinte link: https://github.com/AgenSoftwere/catebr.git. O código-fonte do aplicativo é aberto, permitindo que desenvolvedores contribuam com melhorias e atualizações, ajudando a aprimorar a plataforma continuamente.

O aplicativo está escrito utilizando o framework **Next.js**, uma tecnologia moderna e poderosa para a criação de aplicativos web, oferecendo performance otimizada e uma experiência de usuário eficiente. Ele está hospedado na **Vercel**, uma plataforma de deployment focada em performance e escalabilidade para projetos Next.js, garantindo que o app seja acessível e seguro para todos os usuários.

Atualmente, o aplicativo está na versão **1.0.0 Alpha**, marcando o início do ciclo de desenvolvimento. A versão Alfa indica que o aplicativo está em um estágio inicial de testes, e melhorias serão implementadas ao longo do tempo com base no feedback dos usuários e colaboradores.

O aplicativo é licenciado sob a **Licença MIT**, o que significa que qualquer pessoa pode usar, modificar e distribuir o código, desde que respeite os termos desta licença. Ao colaborar com o projeto, pedimos que todos os contribuintes testem minuciosamente o código para garantir a qualidade e estabilidade da aplicação. Além disso, é fundamental que todos os colaboradores se atentem aos aspectos de segurança ao implementar novas funcionalidades, corrigir bugs ou alterar qualquer parte do sistema. A segurança dos dados dos usuários é uma prioridade, e quaisquer mudanças que possam afetar a integridade ou a privacidade dos dados devem ser cuidadosamente revisadas e testadas antes de serem integradas ao código final.

Ao utilizar qualquer API no aplicativo, é fundamental garantir que ela **não exija chaves privadas** para autenticação, a menos que seja absolutamente necessário, e que esteja configurada de maneira a **garantir a segurança** dos dados trafegados. APIs externas devem ser seguras, com criptografia adequada e conformidade com as melhores práticas de segurança. Além disso, é importante que as APIs utilizadas sejam **públicas**, para que qualquer colaborador ou desenvolvedor possa acessá-las, integrá-las ou modificar seu funcionamento, caso seja necessário, sem comprometer a transparência ou a acessibilidade do projeto.

Ao colaborar no desenvolvimento do aplicativo, é imprescindível que a **linguagem utilizada seja formal e técnica**, com clareza e objetividade em todas as comunicações, incluindo commits, comentários de código e documentação. Para garantir uma estrutura de front-end organizada e modular, todos os arquivos de **CSS devem utilizar o padrão** `module.css`,

promovendo a encapsulação dos estilos e evitando conflitos entre componentes. O **JavaScript** deve seguir as melhores práticas de **escrita modular** e **legível**, utilizando componentes reutilizáveis no estilo de **programação declarativa**. Além disso, o uso de **React** (e outros componentes do Next.js) deve ser feito de maneira eficiente, com **uso de hooks** para gerenciar o estado e **componentes funcionais** sempre que possível.

O design do front-end deve ser responsivo, utilizando o conceito de **design mobile-first**, garantindo que o aplicativo seja acessível e funcional em diferentes dispositivos. O uso de bibliotecas de UI, como **Material-UI** ou **Tailwind CSS**, deve ser considerado para padronizar o design e acelerar o desenvolvimento. Qualquer implementação de **grande porte** ou a criação de **novas páginas** no aplicativo deve ser **notificada aos desenvolvedores principais** para garantir que a equipe esteja alinhada e que as modificações estejam de acordo com a arquitetura do sistema.

Em relação à organização do código, é importante seguir as melhores práticas de desenvolvimento, como a **manutenção de código limpo e legível**, a **utilização de commits significativos e frequentes** e a **aplicação de testes automatizados** sempre que possível. A **modularização do código** é essencial para garantir a reutilização e manutenção fácil dos componentes. Também é recomendado o uso de **linters** e **formatadores automáticos** para manter o código consistente e fácil de ler. Essas práticas são fundamentais para assegurar que o projeto se mantenha escalável, seguro e profissional à medida que cresce.

Ao realizar commits no repositório, é obrigatório que o **nome completo do colaborador** seja incluído na mensagem de commit, para que possamos reconhecer e incluir seu nome nos créditos do projeto. Essa prática é essencial para garantir o devido reconhecimento de todos os colaboradores que contribuem para o desenvolvimento do aplicativo. Além disso, as mensagens de commit devem ser claras e descritivas, refletindo as mudanças feitas no código de forma objetiva e informativa, para que o histórico do projeto seja bem documentado e fácil de entender para todos os membros da equipe.