

Grupo 7: Code Fighters

Integrantes: Julia Von Dentz, Juliana Moises de Souza, Diego Puline, Fernando Zang

1. Levantamento de Tecnologias

1.1. Back-end

1.1.1. JavaScript

Essa linguagem foi considerada devido ao conhecimento prévio de um dos integrantes do grupo, porém logo descartada pois sua concorrente pareceu mais propícia. Seu nível de dificuldade foi definido como alto devido a necessidade de aprendizado da linguagem.

1.1.2. Kotlin

Essa linguagem foi escolhida devido ao conhecimento e domínio prévios que um dos integrantes do grupo possui na linguagem. Seu nível de dificuldade foi definido como alto pois, embora nem todo o grupo tenha conhecimento de seu funcionamento, por ser muito conhecida e semelhante ao Java, além de ser uma linguagem orientada a objetos (algo como qual todo grupo já possui algum contato), se torna possível aprender sobre ela ao longo do projeto sem grandes prejuízos.

1.2. Front-end

1.2.1. React

Essa linguagem foi escolhida devido a já ter sido utilizada anteriormente por alguns integrantes. Seu nível de dificuldade foi definido como médio-fácil pois, além da grande popularidade, tem semelhanças com outras linguagens nesse mesmo estilo e uma elevada curva de aprendizado inicial.

1.2.2. Flutter

Essa linguagem foi considerada por ter sido trabalhada em sala em uma das disciplinas anteriores, porém descartada devido a pouca experiência real do grupo com ela. Seu nível de dificuldade foi definido como médio-alto, pois embora seja uma linguagem emergente e alguns membros do grupo possuam uma base de conhecimento nela, está longe de ser algo com o qual tenham uma afinidade e familiaridade notável.

1.3. Banco de dados

1.3.1. MariaDB

Esse servidor foi escolhido devido a sua compatibilidade com o Oracle Database e com a linguagem SQL, cujos quais múltiplos integrantes do grupo já possuíam conhecimento prévio sobre devido a disciplinas cursadas. Seu nível de dificuldade foi definido como médio-fácil devido a esse conhecimento prévio, a grande quantidade de documentação disponível e a facilidade de aprendizado.

1.3.2. PostgreSQL

Esse servidor foi considerado devido a sua popularidade e compatibilidade com o SQL, mas rejeitado pois continha grande diversidade de fornecedores e ninguém no grupo continha experiência prévia. Seu nível de dificuldade foi definido como médio pois, embora compatível com SQL, a ausência de conhecimentos prévios dificultaria sua utilização adequada.

2. Pesquisa de Ambientes de Hospedagem

2.1. Front-end

Tabela de Comparação

Plataforma	Gratuidade/ Custo	Limitações de uso	Integração e Documentação
GitHub Pages	Grátis	Apenas sites estáticos (HTML/CSS/JS); sem backend.	Fácil integração com repositórios GitHub. Documentação simples.
Vercel	Grátis com plano Pro opcional	100 GB de largura de banda/mês, 1 GB de armazenamento.	Suporte a frameworks modernos (Next.js, React, etc.). CI/CD automático. Documentação excelente.

2.2 Back-end

Tabela de comparação

Plataforma	Gratuidade/Custo	Limitações de uso	Integração e documentação
Render	Gratuito (ou planos pagos)	Não concede acesso root	Implementação rápida e guiada, mais ágil, documentação simples com linguagem acessível
EC2	Gratuito (ou planos pagos)	Sem integração nativa com repositórios git para automação de deploy. Precisa usar Github Actions ou Jenkins	Mais liberdade na implementação, porém toma mais tempo e exige conhecimento de devops, documentação técnica e densa

2.3. Banco de dados

Tabela de comparação

Plataforma	Gratuidade/Custo	Limitações de uso	Integração e documentação
Railway	Gratuito (ou planos pagos)	Sem recursos nativos avançados	Controle total sobre o back end, documentação minimalista e direta
Supabase	Gratuito (ou planos pagos)	Recursos limitados, sem acesso root	Implementação rápida com ferramentas prontas, documentação bem escrita, com passo-a-passo

3. Proposta Final

1.1. Tecnologias escolhidas:

Kotlin, React, MariaDB

1.2. Provedores de Hospedagem propostos:

Vercel, Render, Supabase

1.3. Eventuais custos previstos:

Inicialmente pretende-se utilizar os planos gratuitos, mas não se pode negar a possibilidade de necessitar de um plano caso a quantidade de armazenamento de dados no sistema exceda o disponibilizado de forma grátis.

1.4. Justificativa baseada na realidade do projeto:

Por ter como foco auxiliar uma ONG de pequeno a médio porte cuja atuação é majoritariamente local, não se viu necessidade de adicionar mais custos de manutenção a ela, sendo que o sistema provavelmente não terá grandes exigências, ao menos inicialmente.