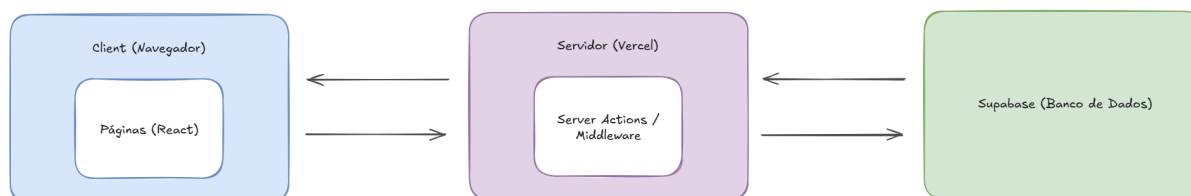


Design Técnico do Projeto Final do Sistema

4.1 Arquitetura de Aplicação

O sistema foi projetado para ser agnóstico à infraestrutura, rodando atualmente em ambiente de desenvolvimento (Localhost), mas com arquitetura preparada para implantação em nuvem na **AWS (Amazon Web Services)**. O Frontend e o Backend (Server Actions) serão hospedados utilizando o serviço **AWS Amplify**, que oferece suporte nativo ao framework Next.js.

- **Front-end:** Next.js (React) com Tailwind CSS.
- **Plataforma:** Web (Responsiva).
- **Backend:** Next.js Server Actions (Node.js runtime).
- **Banco de Dados:** PostgreSQL (Supabase - acessado via HTTPS).
- **Hospedagem / Servidor:** **AWS Amplify** (Gerenciamento de Hosting e CI/CD).



4.2 Armazenamento de Dados e Modelos

- **Tipo de Banco de Dados: SQL (Relacional).**
 - **Motivo da Escolha:** Escolhemos o **PostgreSQL** (via Supabase) devido à necessidade de relacionamentos complexos e integridade dos dados. O sistema precisa relacionar Usuários com Comitês, Comitês com Eventos e controlar listas de presença, o que é ideal para bancos relacionais.
- **Estratégia de Persistência:**
 - **Dados do Usuário:** Armazenados na tabela **profiles** e **auth.users** (gerenciado pelo Supabase).
 - **Logs/Histórico:** O Supabase mantém logs de acesso e transações internamente.
- **Segurança e LGPD:**
 - **Criptografia:** Senhas são criptografadas (Hash) automaticamente pelo Supabase Auth. A comunicação é toda via HTTPS (TLS).
 - **Controle de Acesso (RLS):** Utilizamos **Row Level Security (RLS)** do PostgreSQL. Isso garante que um usuário só possa editar os comitês que ele criou, ou que apenas membros possam ver certos detalhes. É a segurança aplicada direto no banco.
 - **Consentimento:** O cadastro exige ação ativa do usuário.

Resumo da Estrutura da Tabela:

- **profiles:** Dados públicos dos usuários (nome, foto).
- **committees:** Informações dos comitês (nome, descrição, criador_id).
- **committee_members:** Tabela pivô ligando usuários aos comitês.
- **events:** Eventos com data, hora e local.
- **event_attendance:** Lista de presença (quem vai em qual evento).

Estratégia de Backup:

- O Supabase realiza backups diários automáticos do banco de dados (Point-in-Time Recovery) garantindo a recuperação em caso de falhas.

4.3 Integrações e APIs

Serviços Externos Integrados:

- **Supabase Client:** Usado para toda a persistência de dados, autenticação e armazenamento de imagens (Storage).
- Google Maps API ou Mapbox será adicionado futuramente para a funcionalidade de "Mapa".

Padrão de Comunicação:

- A comunicação interna usa **JSON** via protocolo HTTPS.
- Embora usemos a biblioteca cliente do Supabase, ela se comunica com o banco através de uma interface **RESTful** (PostgREST) gerada automaticamente.

Formato de Dados: JSON.

Autenticação Segura:

- Uso de **Tokens JWT (JSON Web Tokens)**. As chaves de API (**NEXT_PUBLIC_SUPABASE_ANON_KEY**) são usadas no frontend, mas protegidas pelas regras de segurança do banco (RLS).

4.4 Nuvem & Implantação

- **Provedor Escolhido: AWS (Amazon Web Services).**
- **Método de Implantação:**
 - **Front-end e Back-end:** Utilização do **AWS Amplify**. Este serviço se conecta ao repositório do GitHub e, a cada atualização (push) na branch **main**, inicia automaticamente o processo de build e deploy (Pipeline CI/CD integrado).

- **Banco de Dados:** Mantido no **Supabase** (PostgreSQL Gerenciado), garantindo escalabilidade e segurança sem necessidade de manutenção de servidor de banco de dados (RDS).
- **Uso do pipeline CI/CD:**
 - **Sim (Nativo da AWS).** O AWS Amplify monitora o repositório Git. Ao detectar mudanças, ele provisiona um contêiner, instala as dependências (**npm install**), constrói o projeto (**npm run build**) e distribui para a rede global da AWS (CloudFront).

Planejamento de Hospedagem (Detalhes):

- O ambiente utilizará a infraestrutura **Serverless** da AWS. Isso significa que não pagamos por servidores ociosos; os recursos de computação (para rodar as Server Actions) são alocados apenas quando um usuário faz uma requisição.