

Estrutura do Projeto: Aplicativo de Ajuda em Eventos Extremos

1. Título do Projeto

ExtremeHelp: Resposta Rápida em Eventos Extremos

2. Introdução e Justificativa

Eventos climáticos extremos têm se tornado mais frequentes e intensos, causando impactos significativos em comunidades. Em momentos de crise, a agilidade na comunicação e na coordenação de ajuda é fundamental. Este projeto propõe o desenvolvimento de um aplicativo móvel simples e intuitivo que conecte pessoas necessitando de auxílio com voluntários dispostos a ajudar durante e após a ocorrência de desastres naturais ou situações críticas. A plataforma também servirá como um canal para alertas preventivos e informações úteis.

3. Objetivo Geral

Desenvolver um aplicativo móvel (MVP) que facilite a solicitação e o oferecimento de ajuda mútua em contextos de eventos extremos, além de prover informações e alertas relevantes para a preparação e segurança dos usuários.

4. Objetivos Específicos

- Permitir que usuários solicitem ajuda, informando sua localização e a natureza do problema.
- Permitir que voluntários visualizem os pedidos de ajuda próximos e se ofereçam para auxiliar.
- Disponibilizar uma seção de alertas sobre eventos críticos iminentes.
- Oferecer informações básicas sobre como se preparar para diferentes tipos de eventos extremos.
- Garantir uma interface simples e de fácil utilização, considerando o contexto de urgência.

5. Público-Alvo

- **Pessoas Afetadas:** Indivíduos e famílias que necessitam de ajuda imediata devido a eventos extremos (ex: desabrigados, isolados, precisando de suprimentos básicos, resgate leve).
- **Voluntários:** Cidadãos dispostos a oferecer ajuda (ex: doação de itens, transporte, auxílio em abrigos, pequenas tarefas de reparo).
- **Comunidade em Geral:** Usuários que buscam informações preventivas e alertas.

6. Funcionalidades Principais (MVP - Foco nos 11 dias)

6.1. Para Quem Precisa de Ajuda (Solicitante):

1. **Cadastro/Login Simplificado:**
 - Campos essenciais: Nome, contato (telefone/email), senha.
2. **Solicitar Ajuda:**
 - Formulário simples:
 - **Descrição do Problema:** Campo de texto para detalhar a situação.
 - **Localização:**
 - Uso do GPS do dispositivo para obter a localização atual (prioridade).
 - Opção de inserir o endereço manualmente como alternativa.
 - **Tipo de Ajuda (Categorias Simples):** Lista pré-definida para facilitar (ex: Alimento/Água, Abrigo, Medicamentos, Resgate Leve, Roupas/Cobertor, Outro).
 - Submissão do pedido.
3. **Meus Pedidos:**
 - Listagem dos pedidos feitos pelo usuário.
 - Status simplificado do pedido (ex: "Aguardando Voluntário", "Voluntário a Caminho", "Ajuda Recebida").

6.2. Para Voluntários:

1. **Cadastro/Login Simplificado:** (Similar ao solicitante).
2. **Visualizar Pedidos de Ajuda:**
 - Lista de pedidos de ajuda ativos.
 - Priorização por proximidade (baseado na localização do voluntário, se ele permitir o acesso).
 - Informações exibidas: Tipo de ajuda, breve descrição, distância aproximada.
3. **Aceitar Pedido de Ajuda:**
 - Botão para se voluntariar para um pedido específico.
 - Ao aceitar, o status do pedido é atualizado para o solicitante.
4. **Meus Atendimentos:**
 - Lista de pedidos que o voluntário aceitou.

6.3. Funcionalidades Gerais / Informativas:

1. **Painel de Alertas:**
 - Seção para exibir alertas importantes sobre eventos críticos (ex: risco de enchente, alerta de vendaval).
 - **Implementação MVP:** Os alertas podem ser cadastrados manualmente por um administrador em um painel simples (ou diretamente no banco de dados, se mais rápido).
2. **Dicas de Preparação:**
 - Conteúdo estático com informações e guias básicos sobre como se preparar para diferentes tipos de eventos (ex: o que ter em um kit de emergência, como agir em caso de enchente).

7. Funcionalidades Adicionais (Pós-MVP / Ideias Futuras)

- Sistema de recompensas/gamificação para voluntários (via parcerias).
- Chat integrado para comunicação entre solicitante e voluntário.
- Mapa interativo para visualização de pedidos e voluntários.
- Notificações push para novos pedidos (para voluntários) e status de pedidos (para solicitantes).
- Avaliação mútua entre solicitantes e voluntários.
- Cadastro de abrigos e pontos de coleta.
- Integração com APIs de órgãos oficiais para alertas automáticos (Defesa Civil, INMET).

8. Tecnologias Sugeridas (Considerando o prazo e as diretrizes FIAP)

- **Frontend (Aplicativo Móvel):**
 - **React Native:** Conforme sugerido no material da FIAP (página 33), permite desenvolvimento multiplataforma (iOS e Android).
 - Componentes: `react-native-maps` para funcionalidades de mapa (se o MVP avançar para isso), `axios` ou `fetch` para chamadas API.
- **Backend (API):**
 - **Java (Spring Boot) ou .NET:** Conforme as disciplinas da FIAP (páginas 16 e 28). Escolher a que o grupo tem mais familiaridade para agilizar.
 - Requisitos: API RESTful, persistência em banco de dados relacional, autenticação (JWT se possível, ou mais simples para o MVP).
- **Banco de Dados:**
 - **Oracle ou outro BD Relacional (PostgreSQL, MySQL):** Conforme disciplina "Mastering Relational and Non-Relational Database" (página 30).
 - **Alternativa para MVP rápido (se permitido e o foco for no app):** Firebase/Firestore pela facilidade de configuração e recursos em tempo real (mas verificar se atende aos requisitos da disciplina de BD).
- **Autenticação:**
 - Implementação própria com JWT (se usando Java/.NET).
 - Firebase Authentication (se optar por Firebase como BaaS - Backend as a Service).
- **Hospedagem (Deploy):**
 - Serviços de nuvem como Heroku, AWS, Google Cloud, Azure (verificar requisitos da disciplina de Java Advanced - página 28).
- **Outras Ferramentas:**
 - Git/GitHub para controle de versão.
 - Swagger para documentação da API.
 - Ferramentas de design para wireframes/protótipos (Figma, Adobe XD - opcional, rascunhos no papel podem ser suficientes para o MVP).

9. Escopo Simplificado para 11 Dias (Foco no Essencial)

- **Fluxo Principal:**
 - Usuário (solicitante) se cadastra/loga.
 - Usuário (solicitante) cria um pedido de ajuda com localização e descrição.
 - Usuário (voluntário) se cadastra/loga.
 - Usuário (voluntário) visualiza lista de pedidos.
 - Usuário (voluntário) aceita um pedido.
 - Solicitante vê que seu pedido foi aceito (mudança de status).
- **Telas Essenciais:**
 - Login/Cadastro.
 - Home (com acesso a Pedir Ajuda / Ver Pedidos / Alertas).
 - Formulário de Solicitação de Ajuda.
 - Lista de Pedidos de Ajuda (para voluntários).
 - Detalhes do Pedido (para voluntário decidir se aceita).
 - Meus Pedidos (para solicitante).
 - Meus Atendimentos (para voluntário).
 - Tela de Alertas.
 - Tela de Dicas (conteúdo estático).
- **Backend:** Endpoints para CRUD de usuários, CRUD de pedidos de ajuda, e listagem de alertas.
- **Interface:** Limpa, funcional, sem animações complexas ou design elaborado. Foco na usabilidade.

10. Plano de Desenvolvimento Sugerido (11 Dias)

- **Dia 1-2: Planejamento e Configuração**
 - Refinamento final do escopo do MVP.
 - Desenho dos wireframes básicos das telas essenciais.
 - Modelagem do banco de dados (entidades principais: Usuário, PedidoAjuda, Alerta).
 - Configuração dos ambientes de desenvolvimento (frontend, backend, banco de dados).
 - Criação do repositório Git.
- **Dia 3-5: Desenvolvimento do Backend (API)**
 - Implementação da autenticação de usuários (cadastro, login).
 - Desenvolvimento dos endpoints CRUD para Pedidos de Ajuda.
 - Desenvolvimento do endpoint para listar/cadastrar Alertas.
 - Testes unitários básicos da API (Postman/Insomnia).
- **Dia 6-9: Desenvolvimento do Frontend (React Native)**
 - Criação das telas de Login/Cadastro.
 - Desenvolvimento da tela e formulário para Solicitar Ajuda.
 - Desenvolvimento da tela para listar Pedidos de Ajuda (para voluntários).
 - Implementação da funcionalidade de "Aceitar Pedido".
 - Criação das telas "Meus Pedidos" e "Meus Atendimentos".
 - Criação das telas de Alertas e Dicas (conteúdo estático ou consumindo API simples de alertas).
 - Integração das telas com os endpoints da API.
- **Dia 10: Integração, Testes e Refinamentos**
 - Testes completos do fluxo principal (ponta a ponta).
 - Correção de bugs identificados.
 - Pequenos ajustes de usabilidade e interface.
 - Revisão dos requisitos das disciplinas da FIAP.
- **Dia 11: Documentação e Preparação para Entrega**
 - Finalização da documentação (README no GitHub, diagramas, etc.).
 - Preparação do vídeo de demonstração e pitch (conforme requisitos da FIAP).
 - Verificação dos artefatos de entrega para cada disciplina.

11. Considerações Importantes para o Sucesso em 11 Dias

- **Comunicação Constante na Equipe:** Se for em grupo, alinhem-se diariamente.
- **Priorização Rigorosa:** Se uma funcionalidade está tomando muito tempo, reavalie sua necessidade para o MVP. O "feito é melhor que perfeito" dentro do prazo.
- **Simplicidade Acima de Tudo:** Evite complexidade desnecessária tanto no código quanto na interface.
- **Reutilização de Código:** Crie componentes reutilizáveis no frontend.
- **Foco nos Requisitos da FIAP:** Tenha sempre em mente os critérios de avaliação de cada disciplina.
- **Testes Contínuos:** Teste as funcionalidades à medida que são desenvolvidas.