

CV Builder AI - Requisitos do Projeto

(Parte 1)

Gerador de Currículo Inteligente com Preview em Tempo Real

Este documento contém os requisitos para desenvolvimento da aplicação CV Builder AI. A aplicação permite aos usuários criar currículos profissionais com preview instantâneo e melhorias por inteligência artificial.

Tecnologias

React 19 + TypeScript + TailwindCSS v4 + Vite

Conceito da Aplicação

Layout Split-Screen

- **Esquerda (50%):** Formulário completo para entrada de dados
- **Direita (50%):** Preview do currículo atualizado em tempo real
- **Layout fixo:** Otimizado para telas desktop

Seções do Formulário

1. **Dados Pessoais** - Nome, email, telefone, LinkedIn, resumo
2. **Habilidades** - Lista dinâmica com níveis de proficiência
3. **Experiências** - Lista dinâmica de experiências profissionais

Funcionalidades Core

- **Preview Instantâneo** - Mudanças aparecem em tempo real
- **Listas Dinâmicas** - Adicionar/remover habilidades e experiências
- **Layout Desktop** - Interface otimizada para desktop
- **Design Profissional** - Interface clean e moderna

Estrutura da Aplicação

```
src/  
├── App.tsx  
├── components/  
│   ├── Layout/  
│   │   ├── FormSection.tsx  
│   │   └── PreviewSection.tsx  
│   ├── Form/  
│   │   ├── PersonalInfo.tsx  
│   │   ├── Skills.tsx  
│   │   ├── Experience.tsx  
│   │   └── AIEnhanceButton.tsx  
│   ├── Preview/  
│   │   ├── CVPreview.tsx  
│   │   ├── PersonalHeader.tsx  
│   │   ├── SkillsSection.tsx  
│   │   └── ExperienceSection.tsx  
│   └── UI/  
│       ├── LoadingSpinner.tsx  
│       ├── ErrorBoundary.tsx  
│       └── Toast.tsx  
├── services/  
│   └── aiService.ts  
├── hooks/  
│   ├── useCVData.ts  
│   ├── useAIEnhancement.ts  
│   └── useToast.ts  
├── utils/  
│   ├── validation.ts  
│   └── textProcessing.ts  
├── types/  
│   ├── cv.types.ts  
│   └── api.types.ts  
└── index.css
```

Requisitos Obrigatórios

01 - Layout Split-Screen

- Tela dividida em duas colunas iguais
- Scroll independente em cada seção
- Layout fixo para desktop

02 - Formulário de Dados Pessoais

- Campos: Nome, Email, Telefone, LinkedIn
- Textarea para resumo profissional
- Validação em tempo real
- Contador de caracteres para resumo

03 - Gerenciamento de Habilidades

- Lista dinâmica de habilidades
- Input para nome da habilidade
- Seletor de nível (Básico/Intermediário/Avançado)
- Botões adicionar/remover

04 - Gerenciamento de Experiências

- Lista dinâmica de experiências
- Campos: Empresa, Cargo, Período, Descrição
- Checkbox "Trabalho atual"
- Validação de datas

05 - Preview em Tempo Real

- Atualização instantânea ao digitar
- Layout profissional do currículo
- Indicação visual para campos vazios

Requisitos Opcionais

Exportação PDF Profissional

Para quem quiser ir além, pode implementar a funcionalidade de exportação em PDF:

Funcionalidades PDF

- **Formato:** Layout A4 otimizado para impressão
- **Design:** Tipografia profissional e espaçamento adequado
- **Download:** Geração automática com nome personalizado
- **Customização:** Diferentes temas de cores e layouts

Estrutura Adicional para PDF

```
|— components/  
| |— Preview/  
| |   └─ ExportButton.tsx  
|— services/  
|   └─ pdfService.ts
```

Tecnologias PDF

- **jsPDF** - Geração de PDFs no frontend
- **html2canvas** - Captura de elementos para PDF

Melhorias Avançadas de UX

- Temas de cores personalizáveis
- Atalhos de teclado
- Modo de visualização completa
- Histórico de versões (undo/redo)

Funcionalidades de Persistência

- Salvar dados no localStorage
- Importar/exportar dados em JSON

- Múltiplos currículos salvos

Conceitos Técnicos a Implementar

Estado Compartilhado Avançado

- Estado central para todo o currículo
- Sincronização entre formulário e preview
- Gerenciamento de listas dinâmicas

Componentes Controlados

- Inputs controlados com validação
- Textareas com formatação
- Componentes de lista reutilizáveis

Props e Lifting State Up

- Comunicação entre componentes pai/filho
- Callbacks para atualização de estado
- Props drilling e soluções

Renderização Condicional

- Layout condicional baseado em estado
- Estados vazios e de loading
- Validação visual de campos

Observação: Este projeto foi projetado especificamente para desktop. Não é necessário implementar layout responsivo ou otimizações para mobile.