

🚀 Contabilease: Technical Roadmap para Construção com Cursor AI

📋 Visão Geral Técnica

****Stack Principal:**** Next.js 14, TypeScript, Supabase, Vercel

****Ferramenta de Desenvolvimento:**** Cursor AI

****Data de Início:**** 27/05/2024

****Engenheiro Líder:**** [Seu Nome/Nome do Tech Lead]

🎯 Objetivos Técnicos Principais

1. Implementar arquitetura escalável com i18n desde o início
2. Garantir cálculos financeiros com precisão de 99.9%
3. Implementar segurança robusta com RLS desde o dia 1
4. Configurar pipeline CI/CD totalmente automatizado

🛠️ Stack Técnica Detalhada

Frontend

| Tecnologia | Versão | Propósito |

| :--- | :--- | :--- |

| Next.js | 14.2+ | Framework React com App Router |

| TypeScript | 5.0+ | Tipagem estática para segurança |

| Tailwind CSS | 3.4+ | Estilização utilitária |

| React Hook Form | 7.50+ | Gerenciamento de formulários |
| Zod | 3.22+ | Validação de esquemas |

Backend & Infra

Tecnologia	Propósito
Supabase	Banco de dados, Auth e Storage
Vercel	Deploy, hospedagem e serverless functions
Stripe	Processamento de pagamentos

Ferramentas de Desenvolvimento

Tecnologia	Propósito
Cursor AI	Aceleração de desenvolvimento
GitHub Actions	CI/CD pipeline
Sentry	Monitoramento de erros
Vercel Analytics	Análise de performance

🌐 Roadmap Técnico Detalhado

Fase 1: Setup Inicial e Arquitetura (Semana 1)

```mermaid

graph LR

title Fase 1: Setup Inicial e Arquitetura

dateFormat YYYY-MM-DD

section Projeto

Inicialização Next.js 14 :2024-05-27, 2d

Configuração TypeScript :2024-05-29, 2d

Configuração Supabase :2024-05-29, 3d

section i18n

Estrutura de internacionalização :2024-05-30, 4d

Configuração next-intl :2024-06-03, 3d

...

**\*\*Tarefas Críticas:\*\***

1. `npx create-next-app@latest --typescript --tailwind --app --src-dir`
2. Configurar absolute imports no tsconfig.json
3. Integrar Supabase SDK e configurar ambiente
4. Criar estrutura de pastas para i18n
5. Configurar middleware de internacionalização

---

**### Fase 2: Sistema de Autenticação e Banco de Dados (Semana 2)**

```mermaid

gantt

title Fase 2: Autenticação e Banco de Dados

dateFormat YYYY-MM-DD

section Banco de Dados

Schema inicial :2024-06-05, 3d

RLS Policies :2024-06-07, 4d

section Autenticação

Supabase Auth setup :2024-06-05, 2d

UI de login/registro :2024-06-07, 3d

...

****Esquema do Banco de Dados:****

```sql

-- Tabela countries para i18n

CREATE TABLE countries (

id UUID DEFAULT gen\_random\_uuid() PRIMARY KEY,

iso\_code VARCHAR(2) UNIQUE NOT NULL,

name VARCHAR(100) NOT NULL,

currency\_code VARCHAR(3) NOT NULL,

date\_format VARCHAR(20) NOT NULL DEFAULT 'DD/MM/YYYY',

is\_active BOOLEAN DEFAULT true

);

-- Tabela de usuários extendida

CREATE TABLE profiles (

id UUID REFERENCES auth.users(id) PRIMARY KEY,

email VARCHAR(255) NOT NULL,

full\_name VARCHAR(255),

country\_id UUID REFERENCES countries(id),

subscription\_tier VARCHAR(20) DEFAULT 'free',

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

```
);
...

```

#### Fase 3: Núcleo de Cálculos Financeiros (Semanas 3-4)

```
```mermaid
```

```
graph TD
```

```
    title Fase 3: Núcleo de Cálculos Financeiros
```

```
    dateFormat YYYY-MM-DD
```

```
    section Cálculos
```

```
        Engine de taxas :2024-06-12, 5d
```

```
        Cálculo lease liability :2024-06-17, 5d
```

```
    section Testes
```

```
        Testes unitários :2024-06-19, 4d
```

```
        Testes de integração :2024-06-24, 3d
```

```
...  
  
**Estrutura de Cálculo:**  
  
...  
  
src/  
  
lib/  
  
calculations/  
  
    index.ts      # Interface principal  
  
    lease-liability.ts # Cálculos específicos  
  
    discount-rate.ts # Lógica de taxa de desconto
```

tests/

calculations.test.ts # Testes com Jest

...

Fase 4: UI/UX e Componentes (Semanas 5-6)

```mermaid

gantt

title Fase 4: UI/UX e Componentes

dateFormat YYYY-MM-DD

section Componentes

Design system :2024-06-26, 7d

Formulários :2024-07-03, 5d

section Dashboard

Lista contratos :2024-07-08, 4d

Detalhe contrato :2024-07-10, 4d

...

**\*\*Componentes Principais:\*\***

- `ContractForm` - Formulário de criação de contratos
- `CalculationResult` - Exibição de resultados
- `DashboardLayout` - Layout principal do app

---

#### ### Fase 5: Exportação e Funcionalidades Avançadas (Semanas 7-8)

```mermaid

gantt

title Fase 5: Exportação e Funcionalidades Avançadas

dateFormat YYYY-MM-DD

section Exportação

Geração PDF :2024-07-17, 5d

Geração Excel :2024-07-22, 5d

section Avançado

Histórico de versões :2024-07-24, 4d

Sistema de alertas :2024-07-29, 3d

```

---

#### ### Fase 6: Preparação para Produção (Semanas 9-10)

```mermaid

gantt

title Fase 6: Preparação para Produção

dateFormat YYYY-MM-DD

section DevOps

CI/CD Pipeline :2024-08-05, 5d

Monitoramento :2024-08-09, 3d

section Otimização

Performance :2024-08-12, 4d

Security audit :2024-08-15, 3d

...

🛠️ Configuração do Cursor AI para Máxima Produtividade

1. Contexto do Projeto

Crie um arquivo `.cursorsrules`` na raiz do projeto:

```
```json
```

```
{
```

```
 "projectContext": {
```

```
 "techStack": ["nextjs", "typescript", "supabase", "tailwind"],
```

```
 "patterns": {
```

```
 "components": "src/components",
```

```
 "styles": "src/styles",
```

```
 "lib": "src/lib"
```

```
 },
```

```
 "rules": {
```

```
 "preferFunctionalComponents": true,
```

```
 "useTypeScriptStrict": true,
```

```
 "namingConvention": "camelCaseForFunctions PascalCaseForComponents"
```

```
 }
```

```
 }
```

```
}
```

```
```
```


2. Templates para Cursor

Crie templates comuns em `.cursor/templates/`:

```
**Component Template** (`.cursor/templates/component.tsx`):
```

```
``typescript
```

```
import { useTranslations } from 'next-intl';
```

```
interface Props {
```

```
  // Defina suas props aqui
```

```
}
```

```
export const {{componentName}} = ({ /* props */}: Props) => {
```

```
  const t = useTranslations('{{translationKey}}');
```

```
  return (
```

```
    <div>
```

```
      /* Seu código aqui */
```

```
    </div>
```

```
  );
```

```
};
```

```
...
```

3. Comandos Úteis para Cursor

```
``bash
```

```
# Gerar componente com i18n
```

```
/c Create a responsive form component for contract creation with react-hook-form  
and zod validation, using our design system and i18n
```

Gerar cálculo financeiro

/c Create a function to calculate lease liability based on IFRS 16 with detailed comments and TypeScript types

Gerar teste unitário

/c Create comprehensive Jest tests for the discount rate calculation function covering edge cases

...

🔍 Métricas de Qualidade de Código

| Métrica | Meta | Ferramenta |

| :--- | :--- | :--- |

| **Coverage de testes** | > 90% | Jest |

| **TypeScript strict** | 100% | tsc |

| **Performance (LCP)** | < 1.5s | Vercel Analytics |

| **Acessibilidade** | > 95% | Lighthouse CI |

| **Security vulnerabilities** | 0 | Snyk/GitHub Dependabot |

⚠️ Riscos Técnicos e Mitigações

| Risco | Impacto | Mitigação |

| :--- | :--- | :--- |

| **Complexidade cálculos** | Alto | Testes exaustivos e validação com contadores |

| **Performance de cálculos** | Médio | Web Workers para cálculos pesados |

| **Vazamento de dados** | Crítico | RLS rigoroso e auditorias regulares |

| **Problemas i18n** | Médio | Testes de localização e pseudolocalização |

📊 Sistema de Monitoramento

1. **Sentry** para erro tracking
2. **Vercel Analytics** para performance
3. **Supabase Logs** para queries lentas
4. **Cron jobs** para saúde do sistema

```typescript

// Exemplo de monitoramento de erros

import \* as Sentry from '@sentry/nextjs';

export const registerError = (error: Error, context: Record<string, any>) => {

Sentry.withScope(scope => {

scope.setExtras(context);

Sentry.captureException(error);

});

};

...

---

Este roadmap técnico fornece uma base sólida para desenvolvimento com Cursor AI, priorizando qualidade, segurança e escalabilidade desde o início. A integração do i18n desde a fase inicial permitirá expansão internacional sem refatoração significativa.