

# Guia de Configuração - Destinote

---

## Índice

---

1. [Problema do CSV Resolvido](#)
  2. [Como Adicionar Seus Links de Contato](#)
  3. [Como Adicionar URL do Portfólio](#)
  4. [Como Atualizar o Texto da Seção “Sobre”](#)
  5. [Como Atualizar o CSV](#)
- 

## Problema do CSV Resolvido

---

### O Problema

Inicialmente, o sistema estava carregando apenas **789 itens** ao invés dos **~1000 esperados**.

### A Causa

O parser CSV original usava uma função simples `split(',')` que não lidava adequadamente com:

- Campos entre aspas duplas contendo vírgulas
- Caracteres especiais e acentuação
- Diferentes encodings de linha (Windows vs Unix)





### Exemplo de linha problemática:

```
28,"Fazer uma trilha em Paranapiacaba (São Paulo, Brasil)",,Experiência,14,,,28
```

O parser simples interpretava incorretamente as vírgulas dentro das aspas, quebrando os dados.

### A Solução

Implementamos a biblioteca `csv-parse` que:

-  Lida corretamente com campos entre aspas
-  Suporta caracteres especiais e acentuação
-  É tolerante com diferentes formatos de linha
-  Pula linhas vazias automaticamente

**Resultado:** Agora **1075 itens** são carregados corretamente!

## Código da Solução

```
import { parse } from 'csv-parse/sync';

function parseCSV(filePath: string): any[] {
  const fileContent = fs.readFileSync(filePath, 'utf-8');

  const records = parse(fileContent, {
    columns: true, // Usa primeira linha como cabeçalho
    skip_empty_lines: true, // Pula linhas vazias
    trim: true, // Remove espaços
    relax_quotes: true, // Tolerante com aspas
    relax_column_count: true // Tolera colunas inconsistentes
  });

  return records;
}
```

## Como Adicionar Seus Links de Contato

Edite o arquivo `components/AboutSection.tsx` :

### Localize a seção de contatos (linha ~23):

```
const contacts = {
  whatsapp: '', // Adicionar número com código do país: 5511999999999
  instagram: '', // Adicionar username: @seu_usuario
  linkedin: '', // Adicionar URL completo
  email: '', // Adicionar email
};
```

### Preencha com suas informações:

```
const contacts = {
  whatsapp: '5511999887766', // Código do país + DDD + número (sem espaços ou caracteres especiais)
  instagram: '@seu_usuario', // Seu @ do Instagram
  linkedin: 'https://linkedin.com/in/seu-perfil', // URL completa do LinkedIn
  email: 'seu@email.com', // Seu email
};
```



### Importante:

- **WhatsApp:** Use o formato internacional sem espaços: 55 (Brasil) + 11 (DDD) + 999887766
- **Instagram:** Inclua o @ ou não, o código já trata isso
- **LinkedIn:** Use a URL completa começando com https://
- **Email:** Email completo e válido



## Como Adicionar URL do Portfólio

Edite o arquivo `components/AboutSection.tsx` :

## Localize a linha ~11:

```
const [portfolioUrl, setPortfolioUrl] = useState(''); // Para ser configurado pelo usuário
```

## Modifique para:

```
const [portfolioUrl, setPortfolioUrl] = useState('https://seu-portfolio.com');
```

Ou se preferir, configure como constante:

```
const portfolioUrl = 'https://seu-portfolio.com';
```

A foto ficará clicável e abrirá seu portfólio em uma nova aba!



## Como Atualizar o Texto da Seção “Sobre”

O texto atual é um **placeholder**. Para personalizar:

### 1. Extraia o texto da imagem `head.png`

Se você tem a imagem com o texto original, você pode:

- Transcrevê-lo manualmente
- Usar uma ferramenta de OCR online

### 2. Edite o arquivo `components/AboutSection.tsx`

Localize a seção de texto (linha ~116):

```
<div className="text-white/90 space-y-4 text-lg leading-relaxed">
  <p>
    <strong className="text-purple-400">Destinote</strong> é mais do que uma lista -
    é uma jornada de autodescoberta e realização de sonhos.
  </p>
  { /* ... mais parágrafos ... */ }
</div>
```

### 3. Substitua pelos seus parágrafos:

```
<div className="text-white/90 space-y-4 text-lg leading-relaxed">
  <p>
    Seu primeiro parágrafo aqui...
  </p>
  <p>
    Seu segundo parágrafo aqui...
  </p>
  <p>
    <strong className="text-purple-400">Destaque importante</strong> em roxo...
  </p>
</div>
```

## 4. Remova a nota de placeholder:

Apague ou comente o parágrafo:

```
<p className="text-sm text-white/70 italic">
  💡 Dica: Para editar este texto...
</p>
```



## Como Atualizar o CSV

### Estrutura Esperada do CSV

O CSV deve ter **9 colunas** nesta ordem:

```
ID,Title,Local (opcional),Categories,Category ID,Cod 2,Cod 3,,REF base
```

### Colunas Importantes:

1. **ID** - Número único do objetivo
2. **Title** - Título do objetivo (pode conter vírgulas se entre aspas)
3. **Local (opcional)** - Local relacionado ao objetivo (FUTURO: será exibido)
4. **Categories** - Nome da categoria (deve corresponder ao mapeamento)
5. **REF base** - Referência base (atualmente não usado, foi removido da UI)

### Como Adicionar o Campo “Local”:

1. **No CSV**: Preencha a coluna `Local (opcional)` com o local desejado
2. **Preparado automaticamente**: O sistema já está preparado para exibir este campo
3. **Na UI**: Aparecerá automaticamente na janela de detalhes

### Como Adicionar Thumbnails:

1. **No CSV**: Adicione uma nova coluna `ImageUrl` (o schema já suporta)
2. **No seed.ts**: Mapeie a coluna na criação do goal:

```
imageUrl: row['ImageUrl'] || (isTopTen ? `/images/placeholder-${i + 1}.jpg` : null),
```

1. **As imagens**: Coloque em `public/images/`

### Categorias Disponíveis:

```
Lifestyle, Aprender, Participar, Visitar, Comer, Assistir, Ter,
Criativos, Amor, Humanitário, Ler, Esporte, Hard, Experiência,
Top, Ouvir/jogar, Cômico/Comico, Simples, Milestone, Nerd,
Adulto +18, Adulto, Premium, Play
```

### Após Atualizar o CSV:

1. **Substitua o arquivo**:

```
cp /caminho/para/novo-csv.csv /home/ubuntu/Uploads/1000\ Main\ -\ online\ -\ Lista\ Principal.csv
```

### 1. Resete o banco de dados:

```
cd /home/ubuntu/destinote  
npx prisma migrate reset --force
```

### 1. Verifique os resultados:

- O seed mostrará quantos itens foram carregados
- Avisos sobre categorias não encontradas aparecerão em amarelo

---

## Executando o Projeto

### Modo Desenvolvimento:

```
cd /home/ubuntu/destinote  
npm run dev
```

O site estará disponível em `http://localhost:3000`

### Resetar o Banco de Dados:

```
npx prisma migrate reset --force
```

### Ver o Banco de Dados:

```
npx prisma studio
```

---

## Notas Importantes

1. **Top 8:** Os primeiros 8 itens do CSV são automaticamente marcados como destaque
  2. **Responsividade:** Todas as telas são responsivas (mobile, tablet, desktop)
  3. **Animações:** GSAP e Framer Motion são usados para animações suaves
  4. **Imagens:** Coloque todas as imagens em `public/images/`
  5. **Localhost:** Este localhost refere-se ao computador onde o código está rodando. Para acessar de outro dispositivo, você precisará fazer deploy em um servidor.
-

## Troubleshooting

---

### **Problema: “Missing required environment variable: DATABASE\_URL”**

**Solução:** Execute com `export DATABASE_URL="file:./prisma/dev.db"` antes dos comandos

### **Problema: CSV não carrega todos os itens**

**Solução:** Verifique se:

- O arquivo está em UTF-8
- Campos com vírgulas estão entre aspas duplas
- Não há linhas completamente vazias no meio do arquivo

### **Problema: Categoria não encontrada**

**Solução:** Adicione a categoria nova no arquivo `prisma/seed.ts` no mapeamento `categoryColorMap`

---

## Suporte

---

Se tiver dúvidas, edite este guia ou consulte o README.md principal do projeto!

**Projeto criado com ❤️ usando Next.js 14, TypeScript, TailwindCSS, Framer Motion e GSAP**