

Como Aplicar a Correção Manualmente

Nota Importante

Devido a limitações de permissões do GitHub App, não foi possível fazer push automático das alterações.

Por favor, aplique as correções manualmente seguindo as instruções abaixo.

Arquivos que Precisam ser Modificados

1. package.json

Alteração 1: Seção `resolutions` (linha ~77)

ANTES:

```
"resolutions": {  
  "react": "19.2.1",  
  "react-dom": "19.2.1",  
  "@types/react": "18.3.23",  
  "@types/react-dom": "19.2.3"  
},
```

DEPOIS:

```
"resolutions": {  
  "react": "^19.0.0",  
  "react-dom": "^19.0.0",  
  "@types/react": "^19.0.0",  
  "@types/react-dom": "^19.0.0"  
},
```

Alteração 2: Seção `devDependencies` (linha ~217)

ANTES:

```
"@types/react": "^19.2.7",  
"@types/react-dom": "^19.2.3",
```

DEPOIS:

```
"@types/react": "^19.0.6",  
"@types/react-dom": "^19.0.3",
```

Alteração 3: Adicionar novo script (linha ~72, após “`validate:patch-540`”)

ADICIONAR:

```
"fix:react-error": "bash scripts/fix-react-error.sh"
```

O resultado final deve ser:

```
"validate:patch-540": "./scripts/validate-patch-540.sh",  
"fix:react-error": "bash scripts/fix-react-error.sh"
```

2. vite.config.ts

Alteração: Seção `optimizeDeps` (linha ~441)

ANTES:

```
optimizeDeps: {  
  include: [  
    "react",  
    "react-dom",  
    "react-dom/client",  
    "react/jsx-runtime",  
    "react/jsx-dev-runtime",  
    "react-router-dom",  
    "@supabase/supabase-js",  
    "@tanstack/react-query-devtools",  
    "react-helmet-async",  
    "scheduler",  
    "mqtt"  
  ],  
  exclude: [],  
  // CRITICAL: Force rebuild of optimized deps to clear corrupted cache  
  force: true,  
}
```

DEPOIS:

```
optimizeDeps: {  
  include: [  
    "react",  
    "react-dom",  
    "react-dom/client",  
    "react/jsx-runtime",  
    "react/jsx-dev-runtime",  
    "react-router-dom",  
    "@supabase/supabase-js",  
    "@tanstack/react-query",  
    "@tanstack/react-query-devtools",  
    "react-helmet-async",  
    "scheduler",  
    "mqtt"  
  ],  
  exclude: [],  
  // CRITICAL: Force rebuild of optimized deps to clear corrupted cache  
  force: false,  
}
```

Mudanças:

- Adicionar `"@tanstack/react-query"`, na lista de include
 - Mudar `force: true` para `force: false`
-

3. Criar Novo Arquivo: `scripts/fix-react-error.sh`

Crie o arquivo `scripts/fix-react-error.sh` com o seguinte conteúdo:

```
#!/bin/bash

# Script de correção do erro "Cannot read properties of null (reading 'useEffect')"
# Este script limpa caches e reinstala dependências com as versões corretas

set -e

echo "🔧 Iniciando correção do erro React useEffect..."
echo ""

# Cores para output
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
NC='\033[0m' # No Color

# Função para imprimir mensagens coloridas
print_step() {
    echo -e "${GREEN}✓${NC} $1"
}

print_warning() {
    echo -e "${YELLOW}⚠️${NC} $1"
}

print_error() {
    echo -e "${RED}✗${NC} $1"
}

# Verificar se estamos no diretório correto
if [ ! -f "package.json" ]; then
    print_error "Erro: package.json não encontrado. Execute este script na raiz do projeto."
    exit 1
fi

print_step "Passo 1: Removendo node_modules e caches..."
rm -rf node_modules
rm -rf .vite-cache-v4
rm -rf .vite
rm -rf dist
rm -rf .next
rm -rf .vercel_cache
print_step "Caches removidos com sucesso"

echo ""
print_step "Passo 2: Limpando cache do npm..."
npm cache clean --force
print_step "Cache do npm limpo"

echo ""
print_step "Passo 3: Reinstalando dependências..."
npm install
print_step "Dependências instaladas com sucesso"

echo ""
print_step "Passo 4: Verificando versões do React..."
REACT_VERSION=$(npm list react --depth=0 2>/dev/null | grep react@ | awk -F@ '{print $2}')
REACT_DOM_VERSION=$(npm list react-dom --depth=0 2>/dev/null | grep react-dom@ | awk -F@ '{print $2}')
TYPES_REACT_VERSION=$(npm list @types/react --depth=0 2>/dev/null | grep @types/re-
```

```
act@ | awk -F@ '{print $2}'

echo "  React: $REACT_VERSION"
echo "  React-DOM: $REACT_DOM_VERSION"
echo "  @types/react: $TYPES_REACT_VERSION"

echo ""
print_step "✅ Correção aplicada com sucesso!"
echo ""
print_warning "Próximos passos:"
echo "  1. Execute: npm run dev"
echo "  2. Limpe o cache do navegador (Ctrl+Shift+Delete)"
echo "  3. Recarregue a página com Ctrl+F5"
echo ""
print_step "Para mais informações, consulte: FIX_REACT_ERROR.md"
```

Depois de criar o arquivo, torne-o executável:

```
chmod +x scripts/fix-react-error.sh
```

4. Criar Novo Arquivo: FIX_REACT_ERROR.md

Este arquivo já está disponível no repositório local. Copie-o para o repositório remoto.



Passos para Aplicar a Correção

Opção 1: Aplicar Manualmente (Recomendado)

1. **Edite os arquivos** conforme descrito acima
2. **Crie o script** `scripts/fix-react-error.sh`
3. **Execute o script de correção:**

```
bash
```

```
npm run fix:react-error
```

Ou diretamente:

```
bash
```

```
bash scripts/fix-react-error.sh
```

Opção 2: Usar Git Apply (Se tiver o patch)

Se você tiver acesso ao arquivo de patch:

```
git apply react-useeffect-fix.patch
npm run fix:react-error
```



Verificação

Após aplicar as correções:

1. ✅ Verifique se não há erros de sintaxe nos arquivos editados

2. ✓ Execute `npm run fix:react-error`
 3. ✓ Execute `npm run dev`
 4. ✓ Limpe o cache do navegador
 5. ✓ Verifique se a aplicação carrega sem erros
-

Suporte

Se encontrar problemas ao aplicar as correções:

1. Verifique se todas as alterações foram aplicadas corretamente
 2. Consulte o arquivo `FIX_REACT_ERROR.md` para mais detalhes
 3. Execute `npm run fix:react-error` novamente
 4. Limpe completamente o cache do navegador
-

Permissões do GitHub

Para permitir que o Abacus.AI GitHub App faça push automático no futuro, visite: [GitHub App Permissions](https://github.com/apps/abacusai/installations/select_target) (https://github.com/apps/abacusai/installations/select_target)

E conceda as permissões necessárias para o repositório `travel-hr-buddy`.