

# Implementando CI/CD com GitHub e Vercel

Por Escola Dnc

## Introdução

Este ebook aborda a implementação prática de um processo de Integração Contínua e Entrega Contínua (CI/CD) utilizando GitHub e Vercel. Veremos como configurar um fluxo de trabalho automatizado para desenvolver, testar e implantar uma aplicação web React, desde o ambiente de desenvolvimento local até a hospedagem em produção.

## Configurando o Repositório no GitHub

O primeiro passo é criar e configurar um repositório no GitHub para armazenar o código-fonte do projeto.

### Criando um Novo Repositório

1. Acesse sua conta no GitHub
2. Clique em "New repository"
3. Nomeie o repositório (ex: "dnc-arc-[seu-nome]")
4. Deixe o repositório público
5. Clique em "Create repository"

### Conectando o Repositório Local ao GitHub

Após criar o repositório, siga estes passos no terminal:

```
git initgit add .git commit -m "First commit"git branch -M maingit remote add origin https://github.com/[seu-usuario]/[nome-do-repo].gitgit push -u origin main
```

**Importante:** Certifique-se de ter o Git instalado localmente e configurado com suas credenciais do GitHub.

## Integrando com a Vercel

A Vercel é uma plataforma de hospedagem em nuvem que oferece integração nativa com o GitHub, facilitando o processo de CI/CD.

### Configurando a Conta na Vercel

1. Acesse [vercel.com](https://vercel.com)
2. Faça login ou crie uma nova conta
3. Conecte sua conta do GitHub à Vercel

### Importando o Projeto

1. Na dashboard da Vercel, clique em "Add New" > "Project"
2. Selecione o repositório criado anteriormente
3. Mantenha as configurações padrão
4. Clique em "Deploy"

A Vercel iniciará automaticamente o processo de build e deploy da aplicação.

## O Processo de CI/CD em Ação

O fluxo de CI/CD implementado funciona da seguinte forma:

1. **Desenvolvimento Local:** Você faz alterações no código em seu ambiente de desenvolvimento
2. **Push para o GitHub:** As alterações são enviadas para o repositório no GitHub
3. **Trigger Automático:** A Vercel detecta as mudanças no repositório
4. **Build e Testes:** A Vercel inicia o processo de CI, executando o build e testes automatizados
5. **Deploy:** Se o build for bem-sucedido, a Vercel realiza o deploy automático da nova versão

## Exemplo Prático

Vamos fazer uma alteração no componente Hero para demonstrar o processo:

1. Modifique o arquivo do componente Hero, adicionando um link:

```
<Link to="/about">Learn More</Link>
```

2. Teste localmente com `npm run dev`

3. Commit e push das alterações:

```
git add .git commit -m "Adding About route on Hero component"git push origin main
```

#### 4. Observe o processo de deploy na dashboard da Vercel

## Monitoramento e Gerenciamento de Deploys

A Vercel oferece ferramentas para acompanhar e gerenciar os deploys:

- **Dashboard de Deployments:** Visualize todos os deploys realizados
- **Logs de Build:** Acesse logs detalhados do processo de build
- **Preview Deployments:** Teste novas versões antes de ir para produção
- **Rollbacks:** Reverta facilmente para versões anteriores se necessário

**Dica:** Utilize a aba "Deployments" na Vercel para acompanhar o histórico e status de cada deploy.

## Conclusão

A implementação de CI/CD com GitHub e Vercel oferece diversos benefícios:

- Automação do processo de deploy
- Detecção precoce de erros
- Facilidade de colaboração em equipe
- Rápida entrega de novas funcionalidades

Ao dominar este fluxo de trabalho, você estará preparado para desenvolver e manter aplicações web de forma eficiente e profissional.

## Recursos Adicionais

- [Documentação oficial da Vercel](#)
- [Guia de CI/CD do GitHub](#)
- [Melhores práticas de deploy contínuo](#)

Lembre-se de explorar a documentação fornecida e praticar regularmente para aprimorar suas habilidades em CI/CD e desenvolvimento web moderno.

