LINKS:

[GitHub – ambientes para desenvolvimento](https://docs.github.com/pt/actions/deployment/targeting-different-environments/using-environments-for-deployment)

[Github – Iniciando GIthub Actions](https://docs.github.com/pt/actions/quickstart)

Para implementar o conceito de ambientes (production, staging e development) em um repositório GitHub para o desenvolvimento de um site, você pode seguir os passos a seguir:

**1. Crie Branches para Cada Ambiente:**

- Crie três branches principais no seu repositório para representar os diferentes ambientes: `production`, `staging` e `development`. Você pode criar essas branches diretamente no GitHub ou usando o Git localmente e, em seguida, fazer o push para o repositório remoto.

**2. Configurações Específicas para Cada Ambiente:**

- Configure variáveis de ambiente ou arquivos de configuração específicos para cada ambiente. Isso pode incluir configurações de banco de dados, chaves de API, URLs e outras configurações relacionadas ao ambiente.

**3. Gerencie as Dependências:**

- Use um sistema de gerenciamento de dependências, como npm para projetos JavaScript ou pip para projetos Python, para garantir que as dependências sejam consistentes em cada ambiente. Certifique-se de que as dependências corretas estejam instaladas em cada ambiente.

**4. Implante Automaticamente nos Ambientes:**

- Configure integração contínua (CI) e implantação contínua (CD) para automatizar o processo de implantação em cada ambiente. As ferramentas populares para isso incluem GitHub Actions, Travis CI, CircleCI e Jenkins. Crie fluxos de trabalho de CI/CD separados para cada ambiente e especifique as etapas necessárias para implantar o site em cada ambiente.

**5. Proteja as Branches de Produção e Staging:**

- Proteja as branches de `production` e `staging` para evitar que alterações não autorizadas sejam mescladas diretamente nelas. Isso pode ser feito através das configurações de proteção de branch no GitHub.

**6. Desenvolvimento no Ambiente de Desenvolvimento:**

- A equipe de desenvolvimento deve trabalhar principalmente na branch `development`. Todas as novas funcionalidades e correções de bugs devem ser desenvolvidas nessa branch.

**7. Teste em Staging:**

- Quando uma nova versão do site estiver pronta para testes, crie um pull request (PR) da branch `development` para a branch `staging`. Isso permitirá que a equipe revise e teste as alterações no ambiente de staging antes de implantá-las em produção.

**8. Implantação em Produção:**

- Após testar e aprovar as alterações em staging, crie um PR da branch `staging` para a branch `production`. Esta ação realizará a implantação no ambiente de produção.

**9. Monitore e Faça Manutenção:**

- Monitore o ambiente de produção e faça manutenção regularmente. Corrija bugs e implemente novas funcionalidades na branch `development` e siga o ciclo de teste, revisão e implantação nos ambientes de staging e produção conforme necessário.

Lembre-se de documentar adequadamente seu processo e configurar notificações para que a equipe esteja ciente de implantações e problemas nos ambientes. Com a configuração adequada, você terá um fluxo de trabalho de desenvolvimento seguro e organizado que permite testar e implantar alterações com confiança em cada ambiente.