

*\*Essa é uma versão dos Guias de Estudo desenvolvida para auxiliar o processo de impressão e para facilitar a leitura dos guias por programas que fornecem a leitura automatizada para suporte a todos os alunos que necessitem. Dessa forma, não apresentaremos ilustrações nesse arquivo. **#ParaTodosLerem**.*

## Módulo #5

### Guias de estudo

#### Deploy

Você deve ter percebido que um usuário que não é da área da computação teria dificuldade em executar os projetos que você criou até aqui - da maneira que fazemos, por meio de uma IDE. Seria então possível importar os projetos que criamos e disponibilizá-los para uso?

A resposta é sim! Esta etapa de um projeto se chama **implantação** ou **Deploy**. A grosso modo, a implantação refere-se a deixar uma aplicação entregue para os usuários utilizarem. E é importante você saber que alguns erros podem ocorrer neste processo, tais como:

1. Alguma biblioteca que usamos no projeto já possui uma versão mais recente e a que usamos foi deletada;
2. O uso de algum programa externo que pode parar de funcionar;
3. O programa depende de alguns arquivos localizados em uma determinada pasta, e a mesma foi deletada, renomeada ou está em outra pasta agora.

Enfim, tudo isso pode ocorrer e ocasionar um erro em nosso sistema. Para cada um destes problemas citados temos as seguintes soluções:

1. Atualizar na implementação o código utilizando a nova versão da biblioteca;
2. Esperar que o sistema externo volte a funcionar e, enquanto isso, mostre uma mensagem para o usuário, como por exemplo: "funcionalidade indisponível no momento, tente mais tarde!";
3. Modificar a referência ao arquivo no código.

Resumindo, sempre há uma possibilidade do nosso programa falhar quando estiver em produção. Para evitar erros de lógica ou algum tipo de erro na implementação, além de fazer testes após a criação dos códigos, existem algumas técnicas/ferramentas que podemos utilizar durante o processo de implementação do projeto que visam melhorar a qualidade deles, são elas:

**PEP:** do inglês **Python Enhancement Proposal**, que em português significa proposta de aprimoramento do Python. Trata-se de um documento composto por sugestões de boas práticas para desenvolver em Python, deste modo mantemos um padrão das implementações.

**LINTER:** ferramenta que verifica se os códigos implementados seguem os padrões determinados. Por exemplo: em um projeto foi adotado como boas práticas o **PEP8**, então utilizamos **LINTER** para verificar se os códigos seguiram as normas da PEP8.

**GITFLOW:** Sabemos que utilizar repositórios ajuda muito a manter sempre todas as versões dos nossos códigos, o que evita diversos transtornos. Sabemos também que toda equipe acessa estas versões. Imagine se enviamos o projeto errado e outro colega de equipe utiliza este projeto para acrescentar os seus códigos? Para evitar este tipo de problema, ao invés de usar o **GIT**, podemos usar o **GITFLOW** que permite criar várias ramificações, dentre elas, uma ramificação de desenvolvedor que você poderá conferir antes de compartilhar na ramificação que todos irão acessar.

Quando utilizamos repositório uma prática importante é criar uma rotina de submissão dos projetos atualizados para o repositório ou para atualizar o sistema em produção com as novas modificações. Uma forma de fazer isso, de maneira automatizada, é utilizando **CI/CD**.

## **Lembre-se**

Sempre que utilizar projetos desenvolvidos por terceiros, leia o arquivo **README** para ter um norte de como instalar e usar a aplicação. Isso evita erros e você ganhará tempo.