**Atividade 1**

O inferno Jarbas das dependências é algo terrível pois quanto mais o sistema cresce, mais pacotes são adicionados a ele, por isso será maior a possibilidade de, um dia, você encontrar-se em grande desespero. Em sistemas com muitas dependências, lançar novos pacotes de versões pode se tornar rapidamente um pesadelo. Se as especificações das dependências são muito amarradas você corre o risco de um bloqueio de versão.

No caso Jarbas era usado o PyUp para manter a lista de dependências atualizadas automaticamente.

Um pacote que usamos, o **reprint,** lançou uma versão nova e rodando os testes unitários com a versão recente, todos os testes continuaram passando.

Porem ao tentar rodar o comando que popula o banco com os dados do reembolsos, um novo contribuidor se deparou com um erro ao tentar carregar os dados de reembolso do Jarbas.

O comando que dias antes funcionava como esperado não funcionava mais. Duas coisas chamaram atenção:

* Nenhuma alteração tinha sido feita nesse comando ou em algo que esse comando dependesse.
* O traceback mostrava que o erro acontecia exatamente na linha que usava uma função de uma biblioteca externa.

Depois ficou claro de que o que quebrou nosso código foi algo “de fora” do projeto. E um pouco de busca mostrou que o **reprint** havia mudado parte do código da função que estava sendo utilizado.

O jeito mais fácil de testar e confirmar foi reinstalar a versão anterior do pacote e tentar rodar novamente o comando. Feito isso, confirmamos que a versão recente quebra o Jarbas.

Se o **reprint** não versionasse as suas alterações de códigos além do tradicional git, seria muito difícil voltar a usar a versão anterior. Nesse caso a mudança do código tornou possível e fácil de voltar a usar localmente a versão anterior.

PyUp além de ser uma ferramenta que atualiza todos os arquivos de dependência do Python de seu projeto através de solicitações pull no GitHub / GitLab. É construído para resolver estruturas especificas, e ambientes Python complexos. Por exemplo, analisa arquivos requirements.txt, setup.cfg, tox.ini, Pipfiles e Conda.

Ele contribuiu na análise de arquivos, mantendo sua base de código segura. Atualizada ele aciona as integrações no repositório do GitHub, executa testes de CI automaticamente e é altamente configurável.

Ele pode colabora no relato da situação avisando outras pessoas das mudanças que foram feitas.

Ele ajudou no problema alterando o código permitindo usar a versão anterior.