

at_S17_A3_SL6_backend Atividade prática

Nome:	Turma:
vorrie	Turria

Título da atividade: Dependências sob controle -Gerenciamento com npm e pip

Explicação inicial

O gerenciamento de dependências permite que desenvolvedores integrem bibliotecas e pacotes ao projeto com eficiência, garantindo a funcionalidade e a compatibilidade do sistema.

- 1. npm (Node Package Manager):
 - o Gerenciador de pacotes para aplicações Node.js.
 - o Facilita a instalação, a atualização e a remoção de bibliotecas JavaScript.
 - o Pacotes populares: Express, Lodash, Axios.
- 2. pip (Python Package Installer):
 - o Ferramenta-padrão para gerenciar pacotes Python.
 - o Utilizada para instalar bibliotecas como Django, Flask, Pandas.

Exemplo prático

 Cenário: um projeto back-end precisa de autenticação. O desenvolvedor utiliza npm para instalar uma biblioteca OAuth para Node.js ou pip para adicionar uma biblioteca de autenticação no Django.



Situação-problema

Uma equipe está desenvolvendo um sistema que utiliza várias dependências externas. É necessário garantir que todas as dependências sejam instaladas e mantidas de forma consistente entre os membros da equipe.

Situação fictícia produzida pela SEDUC-SP.

Roteiro

- 1. Analisem como gerenciadores de pacotes (npm ou pip) ajudam no controle de dependências.
- 2. Planejem como documentar as dependências necessárias para o projeto em um arquivo (ex.: package.json ou requirements.txt).
- 3. Identifiquem as melhores práticas para atualizar pacotes sem quebrar o sistema.

Perguntas dissertativas:

- 1. Qual é a função principal de um gerenciador de pacotes como npm ou pip?
- 2. Por que é importante documentar as dependências de um projeto?
- 3. Quais cuidados devem ser tomados ao atualizar dependências?

4. Como um arquivo como package.json ou require	ments.txt beneficia a
equipe?	

