

## Relatorio de Casos de Teste - API PokeAPI Excel Generator

### **Caso de teste:** test\_post\_valid\_ids

Descrição: Envia IDs válidos [1, 4, 7] e espera o retorno de um Excel com os dados desses Pokemons.

Saída esperada: Status 200 e campo 'excel\_base64' presente com conteúdo de um arquivo .xlsx.

Saída retornada: Status 200 e conteúdo base64 iniciado com assinatura ZIP válida (b'PK\x03\x04').

Observação: Teste passou com sucesso, confirmando o fluxo principal da API.

### **Caso de teste:** test\_post\_invalid\_id

Descrição: Envia ID inexistente [999999] para verificar se a API lida com falhas da PokeAPI.

Saída esperada: Status 200 e Excel vazio, ou status 500 com mensagem de erro clara.

Saída retornada: Status 500 com mensagem de erro retornada pela exceção da PokeAPI.

Observação: A API poderia ser ajustada para ignorar IDs inválidos e retornar status 200 com planilha parcial ou vazia.

### **Caso de teste:** test\_post\_empty\_list

Descrição: Envia uma lista vazia para verificar se a API trata corretamente a ausência de dados.

Saída esperada: Status 200 e planilha gerada (vazia).

Saída retornada: Status 200 e campo 'excel\_base64' presente.

Observação: Passou corretamente gerando um arquivo excel sem dados

### **Caso de teste:** test\_post\_negative\_id

Descrição: Envia IDs negativos para simular erro.

Saída esperada: Status 500 e mensagem de erro informando falha na chamada da PokeAPI.

Saída retornada: Status 500 e campo 'error' presente no JSON.

Observação: A resposta está correta para uma entrada inválida, mas o tratamento poderia ser mais resiliente (retornar 200 e ignorar inválidos).

### **Caso de teste:** test\_post\_not\_all\_valid\_ids

Descrição: Envia uma mistura de IDs válidos e inválidos [1, 4, 7, 0, 150].

Saída esperada: Status 200 com planilha gerada apenas com os dados válidos.

Saída retornada: Status 500 causado pelos IDs inválidos.

Observação: Ideal seria a API continuar com os válidos e ignorar os inválidos. Pode ser ajustado em get\_pokemon\_data. Pode-se observar que quando o teste foi realizado pelo pytest retornou uma falha, entretanto o mesmo teste realizado no Postman retornou um sucesso, visto que o Postman aguarda até o final da execução para avaliar o teste, e o Python ao encontrar o primeiro erro já cancela sua execução.

### **Caso de teste:** test\_post\_decimal\_id

Descrição: Envia IDs com ponto flutuante [1.0, 2.2].

Saída esperada: Status 500 ou erro de validação.

Saída retornada: Status 500 e erro retornado pela API ao tentar processar ID decimal.

Observação: A entrada deveria ser validada previamente como inteiros.

**Caso de teste:** test\_post\_invalid\_type

Descrição: Envia uma string no lugar de uma lista.

Saída esperada: Status 500 ou 400 com mensagem de erro indicando tipo invalido.

Saída retornada: Status 500 e mensagem de erro capturada pela exceção.

Observação: Poderia retornar 400 com validação mais clara do tipo de dado esperado.