

Prova prática (1h) — Autonomia na web + Python (Colab + API)

O que você pode usar

- Google Colab (rodando no navegador, sem instalar nada). [Google Colab](#)
- Um tutorial curto de *requests* para consultar APIs web (pode ler durante a prova): W3Schools — *Python Requests*. [w3schools.com](#) OU [DocsPython](#)
- Documentação de uma API pública sem chave: PokéAPI. [pokeapi.co+1](#)

Observação: você não precisa instalar bibliotecas. *requests* já funciona no Colab.

Objetivo

Seguir (e adaptar) o tutorial de *requests* para consumir a PokéAPI e construir um mini-relatório em um notebook do Colab com:

1. busca por um Pokémon,
 2. extração e formatação de alguns dados,
 3. exibição da imagem (sprite) no notebook,
 4. (bônus) análise simples com laço e dicionário.
-

Passo a passo (entregáveis)

1) Preparar o ambiente (5 min)

- Abra colab.research.google.com, crie um Novo notebook e renomeie para prova_api_<seu_nome>. [Google Colab](#)
- Na 1ª célula, escreva um comentário: “*Fonte consultada: W3Schools Requests + PokéAPI*” com os links.

2) Teste do tutorial (10 min)

- A partir do tutorial de Requests, faça um GET simples a qualquer URL de exemplo e mostre o status_code e os 120 primeiros caracteres do text.
[w3schools.com](#)
(isso prova que você leu e executou o básico do tutorial)

3) Consulta à PokéAPI (25 min)

- Pergunte ao usuário (com input) o nome de um Pokémon (ex.: pikachu, bulbasaur etc.).
- Monte a URL <https://pokeapi.co/api/v2/pokemon/<nome>> e faça o GET.
[pokeapi.co](#)

- Se `status_code != 200`, mostre “Pokémon não encontrado” e peça outro nome.
- Se der certo, extraia e imprima:
 - `id`, `name`, `height` e `weight`
 - tipos (ex.: `electric`, `grass/poison`)
 - estatísticas base (exibir como pares `stat: base_stat`)
- Mostre a imagem (sprite padrão) no notebook (dica: o JSON tem `sprites["front_default"]`; pesquise como exibir imagem por URL no Colab com `IPython.display.Image`). pokeapi.co

4) Bônus — análise rápida (10 min)

- Para os 20 primeiros Pokémons (end-point <https://pokeapi.co/api/v2/pokemon?limit=20>), conte a frequência de tipos (por ex., quantos têm `water`, `fire`, etc.) e imprima um ranking do tipo mais comum ao menos comum.
(Use laço + dicionário; não precisa gráfico.) pokeapi.co

5) Fechamento (5 min)

- No topo do notebook, escreva 3–5 linhas respondendo:
 - O que você pesquisou para resolver?
 - Qual parte foi mais difícil?
 - Como você testou que deu certo?

Regras

- Pode consultar estes materiais (Colab + tutorial Requests + docs PokéAPI). Outras fontes também são permitidas, desde que cite no topo.
- Não usar IA geradora de código (o objetivo é avaliar sua capacidade de leitura de docs e adaptação).

O que será avaliado

1. Funciona: consulta a 1 Pokémon com tratamento de erro
2. Dados corretos: campos pedidos + sprite exibido
3. Código claro: variáveis nomeadas, comentários curtos, organização em células
4. Autonomia demonstrada: referência às fontes e relato final

5. Bônus: contagem de tipos e ranking

Entrega

- Compartilhe o link do Colab com acesso “Qualquer pessoa com o link: Leitor” para os emails: asartori@furb.br e dalton@furb.br
- Prazo: 1 hora a partir do início.