infosimples

Take-Home Coding Challenge

Processo Seletivo Infosimples 2026

Infosimples Processamento de Dados LTDA
Avenida Paulista, 1636, sala 1504. CEP 01310-200. São Paulo, SP. Brasil.
Email: vagas@infosimples.com.br

Sumário

1. Sobre a Infosimples

A Infosimples foi fundada em 2011, e é especialista em desenvolvimento de projetos de Web Scraping e Inteligência Artificial. Dominamos as ferramentas no estado da arte em deep learning (redes neurais), automação de navegação na Internet e processamento de dados.

Os principais clientes da Infosimples tipicamente são organizações que valorizam atendimento de qualidade e precisam automatizar processos de governança cadastral, concessão de crédito, prevenção a fraude, e-commerce e enriquecimento de dados. Muitos clientes da área de logística também utilizam a Infosimples para otimizar seus processos fiscais e suas frotas de veículos.

O principal serviço fornecido pela Infosimples é uma plataforma que conta com APIs que automatizam o acesso a mais de 850 portais de órgãos públicos brasileiros, tais como Receita Federal, DETRANs e Portais de Prefeituras. Este serviço é oferecido em infosimples.com.

2. O que é Web Scraping?

Web Scraping é o processo de coletar, de maneira automatizada, informações de páginas web, e retornar essas informações de maneira estruturada (seja para salvar num Banco de Dados, ou devolver os dados como JSON ou XML). Resumindo: **Web Scraping te dá informações estruturadas vindas de sites na web**.[1]

Um Web Scraper é uma ferramenta desenvolvida para realizar o Web Scraping. Ele é responsável pela lógica de como acessar uma página web e como extrair as informações dela. As informações podem ser extraídas de um arquivo HTML através de seletores CSS ou com o uso de regex, por exemplo.

Fluxo de funcionamento de um Web Scraper[2]

A finalidade de um scraper é coletar informação de maneira automática, rápida e em diferentes volumes, dispensando a necessidade de ter um humano coletando manualmente essa informação. Essas informações coletadas podem ser usadas, por exemplo, em processos de *onboarding* e *know-your-client* (no caso de dados oriundos de portais governamentais), ou podem ser usadas para análises de mercado (no caso de dados extraídos de e-commerces).

1 https://www.zyte.com/learn/what-is-web-scraping/
2 https://www.thewindowsclub.com/what-is-web-scraping/

3. A vaga de desenvolvedor na Infosimples

Trabalhando na Infosimples, você vai desenvolver web scrapers de diversos portais públicos utilizando a linguagem de programação Ruby. Não tem problema se você não sabe programar em Ruby, pois nós vamos passar as primeiras semanas te ensinando!

Você também vai entender como o fluxo de navegação dos sites funcionam, aprendendo conceitos relacionados às chamadas HTTP, endereços de IP, proxies, headers, concorrência, JavaScript, provedores em nuvem, e muito mais.

Também temos alguns serviços internos que utilizam Machine Learning e Visão Computacional para resolver alguns problemas específicos. Eventualmente, você pode ter a oportunidade de se envolver com atividades dessa natureza.

4. O desafio

4.1. Descrição

Você foi escalado para fazer uma pesquisa de mercado para naves espaciais. O Birô de Inteligência Imperial quer saber o preço médio das naves espaciais na galáxia.

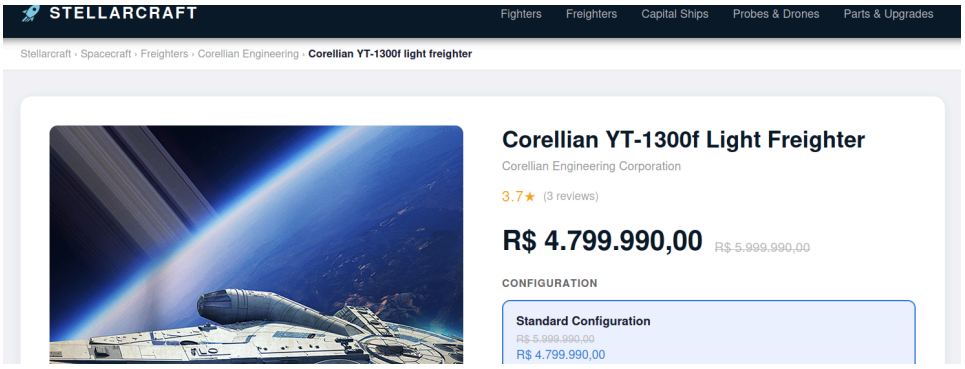
Seu objetivo é fazer um scraper de uma simples página web. Não se preocupe se você nunca fez isso! Este documento tem vários exemplos e tutoriais que te ensinam como fazer um scraper a partir do zero. Você pode usar a linguagem de programação que quiser para isso, e mais à frente vamos te mostrar exemplos em JavaScript, Ruby e Python.

Este desafio cobre aspectos básicos de web scraping e não vai tomar muito do seu tempo. Vai exigir de você conhecimentos de chamadas HTTP, seletores CSS, manipulação de texto e geração de strings JSON.

Você pode utilizar LLMs e agentes/assistentes de código para te auxiliar no seu desenvolvimento. Recomendamos que você entenda profundamente o que está sendo desenvolvido. Não se esqueça de enviar os prompts e respostas utilizados no formulário de resposta.

4.2. Objetivos

Seu primeiro objetivo é construir um programa que acessa e coleta dados da seguinte página: https://infosimples.com/vagas/desafio/stellarcraft/product.html



Essa página foi feita especialmente para este desafio. Ela imita uma página de produto de e-commerce.

Após acessar essa página, você vai precisar fazer o parsing e extração de dados dessa página, e estruturá-los como um JSON, que será salvo como um arquivo.

A seguir, uma lista de quais informações você precisa extrair, juntamente com seus tipos JSON:

<table>
<tr><th>Nome do campo</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>title</td><td>STRING</td><td>Título principal do produto</td></tr>
<tr><td>brand</td><td>STRING</td><td>Nome da marca do produto</td></tr>
<tr><td>categories</td><td>ARRAY DE STRINGS</td><td>Categorias do produto</td></tr>
<tr><td>description</td><td>STRING</td><td>Texto que descreve o produto</td></tr>
<tr><td>skus</td><td>ARRAY DE OBJETOS COM O SEGUINTE FORMATO:
<table><tr><th>Nome do campo</th><th>Tipo</th></tr><tr><td>name</td><td>STRING</td></tr><tr><td>current_price</td><td>FLOAT/NULL</td></tr><tr><td>old_price</td><td>FLOAT/NULL</td></tr><tr><td>available</td><td>BOOLEAN</td></tr></table></td><td>Lista com detalhes de cada uma das variações do produto.
name: Nome da variação
current_price: Preço atual do produto. Pode ser NULL se não estiver disponível.
old_price: Preço antigo do produto. Pode ser NULL se não estiver disponível.
available: true/false se o produto está ou não disponível em estoque.</td></tr>
<tr><td>specification</td><td>ARRAY DE OBJETOS COM O SEGUINTE FORMATO:
<table><tr><th>Nome do campo</th><th>Tipo</th></tr><tr><td>label</td><td>STRING</td></tr><tr><td>value</td><td>STRING</td></tr></table></td><td>Lista com as propriedades do produto.
label: Nome da propriedade.
value: Texto da propriedade.</td></tr>
<tr><td>reviews</td><td>ARRAY DE OBJETOS COM O SEGUINTE FORMATO:
<table><tr><th>Nome do campo</th><th>Tipo</th></tr><tr><td>name</td><td>STRING</td></tr><tr><td>date</td><td>STRING</td></tr><tr><td>score</td><td>INT</td></tr><tr><td>text</td><td>STRING</td></tr></table></td><td>Lista com as avaliações do produto.
name: Nome da pessoa.
date: Data da avaliação
score: Número de estrelas dadas.
text: Texto da avaliação.</td></tr>
<tr><td>reviews_average_score</td><td>FLOAT</td><td>Nota média das avaliações do produto</td></tr>
<tr><td>url</td><td>STRING</td><td>URL da página do produto</td></tr>
</table>

Para exemplificar, o JSON abaixo mostra como deveriam ficar os campos **title** e **categories**:

```
{
 "title": "Corellian YT-1300f Light Freighter",
 "categories": [
   "Stellarcraft",
   "Spacecraft",
   "Freighters",
   "Corellian Engineering",
   "Corellian YT-1300f light freighter"
 ]
}
```

Por fim, você deverá salvar a resposta final em um arquivo chamado **produto.json**.

4.3. Entrega

Após terminar seu código, para o envio você pode zipá-lo ou colocá-lo em um repositório do GitHub. Então, com o arquivo (ou URL do repositório git) em mãos, você deverá acessar o seguinte formulário:

https://forms.gle/9kPwbrdisgPVds2H7

Não tem problema se você não conseguiu completar tudo, ou se você acha que seu código poderia ser melhor. Mande o código mesmo assim!

Nesse formulário, além de submeter seu código, você irá preencher seu nome e seu email usado durante todo o processo seletivo. Além disso, você deverá escrever um pequeno texto sobre a seguinte pergunta:

Por que você acha que web scraping é um processo relevante, hoje em dia? Quais são as principais dificuldades de manter um scraper funcionando?

Após isso, é só submeter o formulário e aguardar pelo retorno de alguém da Infosimples.

Atenção!

Você não deve submeter arquivos executáveis ou diretórios contendo bibliotecas externas. Apenas o código-fonte deverá ser entregue. O envio de binários ou pastas contendo bibliotecas podem acarretar na sua desclassificação.

5. Material de estudo

A Infosimples disponibiliza gratuitamente um curso com conceitos básicos de programação, chamado de Estágio em Programação.

O capítulo relevante para fazer o desafio neste documento está disponível em

https://infosimples.github.io/estagio-em-programacao/aulas/15/01_extra_scraping/

Você pode usar qualquer outro material de estudo que quiser também.

6. Como começar a resolver o desafio?

Se você ainda não faz ideia de como começar a resolver o desafio, a seguir resolvemos para você o começo do desafio em três linguagens de programação populares: JavaScript, Ruby e Python.

Independentemente de qual linguagem de programação você escolher, o seu programa vai funcionar na seguinte sequência:

1. Fazer uma requisição HTTP GET para o site do produto;
2. Parsear o body HTML da resposta;
3. Extrair os dados necessários da página;
4. Salvar os dados num arquivo **produto.json**.

Nas próximas páginas, mostraremos exemplos em JavaScript, Ruby e Python.

6.1. Exemplo em JavaScript

Nós testamos este tutorial com o Node.js v22.14.0. Você pode usar outras versões.

Libs que sugerimos você instalar (você pode usar outras se quiser):

- Cheerio (para fazer o parse do HTML)
- Request (para acessar a página web e coletar o HTML)

```
// Bibliotecas que nós instalamos manualmente
const cheerio = require('cheerio');
const request = require('request');

// Bibliotecas nativas do Node.js
const fs = require('fs');

// URL do site
const url = 'https://infosimples.com/vagas/desafio/stellarcraft/product.html';

// Objeto contendo a resposta final
const respostaFinal = {};

// Faz o request e manipula o corpo de resposta
request(url, function (error, response, body) {
  const parsedHtml = cheerio.load(body);

  // Vamos pegar o título do produto, na tag H1, com ID "product_title"
  respostaFinal['title'] = parsedHtml('h1#product_title').text();


  // Aqui você adiciona os outros campos...


  // Gera string JSON com a resposta final
  const jsonRespostaFinal = JSON.stringify(respostaFinal);

  // Salva o arquivo JSON com a resposta final
  fs.writeFile('produto.json', jsonRespostaFinal, function (err) {
    if (err) {
      // Loga o erro (caso ocorra)
      console.log(err);
    } else {
      console.log('Arquivo salvo com sucesso!');
    }
  });
});
```

6.2. Exemplo em Ruby

Nós testamos este tutorial com o Ruby 3.4.3. Você pode usar outras versões.

Gems que sugerimos você instalar (você pode usar outras se quiser):

- Mechanize (para acessar a página web, coletar e fazer o parse do HTML)

```ruby
# Bibliotecas que nós instalamos manualmente
require 'mechanize'

# Bibliotecas nativas do Ruby
require 'json'

# URL do site
url = 'https://infosimples.com/vagas/desafio/stellarcraft/product.html'

# Agent do Mechanize
agent = Mechanize.new

# Objeto contendo a resposta final
resposta_final = {}

# Faz o request
agent.get(url)

# Parse do response
parsed_html = agent.page.parser

# Vamos pegar o título do produto, na tag H1, com ID "product_title"
resposta_final['title'] = parsed_html.css('h1#product_title').text

# Aqui você adiciona os outros campos...

# Salva o arquivo JSON com a resposta final
File.open('produto.json', 'w') { |f| f.write(JSON.dump(resposta_final)) }
```

6.3. Exemplo em Python

Nós testamos este tutorial com o Python 3.13.3. Você pode usar outras versões.

Libs que sugerimos você instalar (você pode usar outras se quiser):

- BeautifulSoup (para fazer o parse do HTML)
- Requests (para acessar a página web e coletar o HTML)

```python
# Bibliotecas que nós instalamos manualmente
from bs4 import BeautifulSoup
import requests

# Bibliotecas nativas do Python
import json

# URL do site
url = 'https://infosimples.com/vagas/desafio/stellarcraft/product.html'

# Objeto contendo a resposta final
resposta_final = {}

# Faz o request
response = requests.get(url)

# Parse do responses
parsed_html = BeautifulSoup(response.content, 'html.parser')

# Vamos pegar o título do produto, na tag H1, com ID "product_title"
resposta_final['title'] = parsed_html.select_one('h1#product_title').get_text()


# Aqui você adiciona os outros campos...


# Gera string JSON com a resposta final
json_resposta_final = json.dumps(resposta_final)

# Salva o arquivo JSON com a resposta final
with open('produto.json', 'w') as arquivo_json:
    arquivo_json.write(json_resposta_final)
```