

Desafio Técnico — Boletim do Trabalho e Emprego

O [Boletim do Trabalho e Emprego \(BTE\)](#) é uma publicação oficial do Ministério do Trabalho, Solidariedade e Segurança Social (MTSSS) de Portugal. Esta publicação inclui dois tipos de documentos: 1^a Série e Separatas. Considera os seguintes exemplos, um de cada tipo:

- [BTE n.º 44, Vol. 92, de 29 novembro 2025](#) (1^a Série)
- [BTE Separata n.º 19, de 27 novembro 2025](#) (Separata)

Objetivo

O desafio consiste em criar um processo **totalmente automático** composto por duas fases. Na primeira, deves recolher todos os links para documentos publicados no BTE. Para cada link, deves ser capaz de carregar para memória o PDF correspondente. Nesta fase, **não é permitido utilizar soluções de headless browser** (e.g.: Puppeteer), apenas HTTP requests. Na segunda fase, deverás transformar cada PDF num documento JSON em **formato de árvore**, ignorando elementos acessórios como o índice do documento e os números de página.

A raiz do documento deverá seguir esta estrutura:

```
{  
  "type": "...", // tipo do documento  
              // i.e. "issue" (1a série) ou "offprint" (separata)  
  "reference": "...", // referência do documento, e.g.  
                  // "BTE n.º 44, Vol. 92, de 29 novembro 2025"  
  "date": "...", // data de publicação do documento  
          // no formato ISO 8601  
  "url": "...", // link para o documento original  
  "root": { ... } // nó raiz da árvore  
}
```

E cada nó da árvore deve ter o seguinte formato:

```
{  
  "type": "...", // tipo do nó e.g. "root", "article", etc.  
  "header": "...", // cabeçalho do nó, se aplicável  
             // e.g. "Cláusula 1.º\nÂmbito de atividade"  
  "text": "...", // texto do nó, se for uma folha  
  "children": "...", // nós filhos, se o nó não for uma folha  
  "metadata": { ... } // outras propriedades próprias do tipo de nó  
                // que não caibam noutro campo  
}
```

Stack

Deverás usar as seguintes ferramentas na tua solução. No caso das bibliotecas, desde que apresentes uma **justificação fundamentada**.

- **Linguagem:** TypeScript (obrigatório).
- **Runtime:** Bun (obrigatório).
- **Bibliotecas:** Axios (crawling), pdfjs-dist (parsing de PDFs).

Critérios de avaliação

A tua solução será avaliada segundo:

- **Robustez:** O código lida corretamente com diferenças inesperadas e inconsistências nos PDFs?
- **Qualidade do código:** Tipagem forte, clareza, arquitetura, modularidade e qualidade geral.
- **Qualidade do output:** O texto é bem extraído, limpo e corretamente estruturado no JSON?
- **Testes (bónus):** Existência e qualidade dos testes automatizados.