

# Padrao de Projeto adotado - Strategy (Crawler)

## Por que Strategy aqui?

O coletor precisa lidar com duas rotas equivalentes do GDE: a resposta AJAX (rapida) e a pagina completa (robusta). Estrategias separadas permitem selecionar a melhor opcao e acionar fallback automatico quando a rota principal falha, evitando quebras na coleta.

## Onde esta no codigo

- crawler/src/crawler\_app/collectors:strategies/base.py
- crawler/src/crawler\_app/collectors:strategies/ajax\_strategy.py
- crawler/src/crawler\_app/collectors:strategies/fullpage\_strategy.py
- crawler/src/crawler\_app/collectors:enumerate\_pipeline.py (fetch\_with\_strategy)

## Diagrama de classes

```
classDiagram
    class ICurriculumFetchStrategy {
        +fetch(session, settings, params, raw_dir) str
    }
    class AjaxStrategy {
        +fetch(session, settings, params, raw_dir) str
    }
    class FullPageStrategy {
        +fetch(session, settings, params, raw_dir) str
    }
    ICurriculumFetchStrategy <|.. AjaxStrategy
    ICurriculumFetchStrategy <|.. FullPageStrategy
```

ASCII: interface -> {AjaxStrategy, FullPageStrategy}

## Sequencia de execucao

```
sequenceDiagram
    participant CLI
    participant Pipeline as enumerate_pipeline
    participant Ajax as AjaxStrategy
    participant Full as FullPageStrategy

    CLI->>Pipeline: fetch_with_strategy(params)
    Pipeline->>Ajax: fetch(session, settings, params)
    Ajax-->>Pipeline: caminho RAW
    alt Falha AJAX
        Ajax-->>Pipeline: excecao
        Pipeline->>Full: fetch(session, settings, params)
        Full-->>Pipeline: caminho RAW
```

```
end
Pipeline-->CLI: HTML persistido / JSON produzido
```

ASCII: CLI -> Pipeline -> AjaxStrategy (erro?) -> FullPageStrategy -> Pipeline -> CLI.

## Criterios de aceitacao (A4)

- ☐ `fetch_with_strategy` respeita `CRAWLER_STRATEGY=auto|ajax|full`.
- ☐ Logs exibem quando o fallback para FullPage ocorre.
- ☐ `cli curriculum` salva RAW antes de parsear disciplinas.
- ☐ Teste manual cobre ambos os caminhos com curso real (ex.: `--course-id 34 --year 2022`).

## Issue / Branch / PR template

### Issue

- **Titulo sugerido:** AvaliacaoA4 - Strategy no crawler (AJAX <-> FullPage)
- **Labels:** AvaliacaoA4
- **Checklist:**
  - ☐ Descrever cenários de uso (auto, ajax, full) com exemplos de comando.
  - ☐ Anexar trecho de log demonstrando fallback ou sucesso na estratégia escolhida.
  - ☐ Confirmar destino de arquivos RAW (`crawler/data/raw`) e JSON (`crawler/data/json`).

### Branch sugerida

`feature/crawler-strategy-arvore`

### PR template

```
## Contexto
- Curso/catalogo usados nos testes:
- Strategy configurada (`CRAWLER_STRATEGY` ou flag CLI):

## Validacao manual
- [ ] `python -m src.crawler_app.cli collect --strategy auto`
- [ ] `python -m src.crawler_app.cli curriculum --course-id 34 --year 2022 --strategy ajax`
- [ ] Verifiquei arquivos RAW em `crawler/data/raw` e JSON em `crawler/data/json`
- [ ] Registrei print/log do fallback (se aplicavel)

## Evidencias
- Screenshots ou logs relevantes:

## Checklist geral
- [ ] Atualizei documentacao pertinente (se necessario)
- [ ] Avaliei impacto em dependencias externas (GDE)
```