

No código que será feito, haverá duas classes principais: Escalonador e o PageFetcher. O Escalonador é responsável a fila de urls e saberá quais urls já foram visitadas, além de outras atribuições descritas a seguir.

Lembrando da nossa prática onde aprendemos a fazer um programa multithread, o Escalonador é análogo ao Contador e, o PageFetcher, às MultithreadHelloWorld (ou seja, a Thread). Pois, haverá uma instancia (objeto) do Escalonador que será compartilhado para todas as instancias de PageFetcher criadas (cada instancia será uma Thread), que requisitarão uma url nova para baixar, e adicionará novas páginas ao coletor.

Haverá apenas uma questão a mais: caso o PageFetcher requisitar uma página e não haver páginas disponíveis na fila para ser baixada, o Escalonador deve fazer a Thread ficar no estado de espera (utilizando o metodo `this.wait()`). Quando estiver neste estado, outras Threads poderão adicionar mais páginas no Escalonador.

Atribuição de cada classe principal (lembrar de tudo que apresentamos na especificação do coletor):

- Escalonador
  - Para cada domínio, armazena o timestamp da última requisição
  - Todas as urls já visitadas
  - Armazenar instancias Record (o Record é a classe do jrobotx que possui o robots.txt em forma de um objeto java) de cada domínio
  - Métodos Principais:
    - Resgatar uma nova url (cujo domínio está há mais de 30 segundos sem ser requisitado)
    - Adicionar uma nova url (apenas se não foi visitada)
    - Resgatar uma instancia Record, dado um domínio
    - Dado um domínio adicionar uma instancia Record deste domínio
- PageFetcher
  - Requisitar uma nova URL ao Escalonador (sem baixá-la) para:
    - Após receber a URL do escalonador, solicitar o Record do domínio desta URL (este objeto solicitado é o robots.txt do dominio):
      - Caso ainda não exista esse Record, faça a requisição usando a API jrobotx para obter esse Record. Logo após, invoque o metodo do escalonador que adiciona para esse determinado domínio a instancia Record criada.
    - Logo após a obtenção do Record, verificar se essa url pode ser coletada. Caso seja possível:
      - Baixe página referente à esta URL
      - Extraia seus links (podendo utilizar uma API para isso)
      - Adicionar os links no escalonador. Transformando possíveis URL relativas em absolutas, há um método no ColetorUtil para identificar URL relativa.