Paginação de registros sob demanda.

Recupere grande quantidade de registros corretamente consumindo menos recursos.

|  |  |
| --- | --- |
| C:\DevMedia\Revistas\Java Magazine\_Edicoes\jm 105\pablo.png |  |
| **Pablo Bruno de Moura Nóbrega** *<http://pablonobrega.wordpress.com>* | **Joel Xavier Rocha**  <http://joelxr.github.io> |
| [pablonobrega2004@gmail.com](mailto:pablonobrega2004@gmail.com) | [joelxr@gmail.com](mailto:joelxr@gmail.com) |
| Líder de Projeto Java, certificado OCJP e OCWCD, Graduado em Ciências da Computação pela Universidade de Fortaleza – UNIFOR, Mestre em Computação pela Universidade Estadual do Ceará – UECE, cursa MBA em Gerenciamento de Projetos na UNIFOR, trabalha na Secretaria Municipal de Finanças de Fortaleza – SEFIN e desenvolve sistemas há cerca de oito anos. | Analista de Sistemas Java, certificado OCJP, Graduado em Engenharia da Computação pelo Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE). Trabalha na Secretária de Finanças do Município de Fortaleza no Ceará (SEFIN) e desenvolve e mantem sistemas há mais de cinco anos. |

**Cenário**

Considere uma aplicação que recupera grande volume de dados e pretende exibi-los para o usuário em uma tabela, trata-se – por exemplo – de um sistema de listagem de clientes que mostra todos os clientes em um *datatable* cujo cada linha é representada por um cliente. Obviamente, para não tornar a listagem muito longa, eles são exibidos em partes cujo tamanho é determinado pelo desenvolvedor. Essas partes são chamadas de páginas desse *datatable*. Vamos analisar a forma como o conteúdo dessa tabela é gerenciado, de que maneira as páginas dessa tabela são exibidas e como carregar o conteúdo para elas. Para isso temos duas possibilidades.

A primeira possibilidade consiste no cenário em que todos os clientes cadastrados na base são carregados e colocados em memória. Facilmente encontramos aplicações que fazem seleção em todos os registros da base de dados (sejam Clientes, Usuários entre outros) e inserem esses objetos em uma coleção para serem visualizados pelo usuário final. Muitas vezes alguma ação do usuário no *datatable* faz com que os registros sejam carregados novamente. Por exemplo, qualquer filtro aplicado ou algum critério de ordenação dos registros.

O segundo cenário que vamos apresentar nesse artigo é aquele que é recuperado da base de dados apenas os registros que estão sendo visualizados pelo usuário, por exemplo, suponhamos que existam 4 mil clientes na base, com essa abordagem vamos selecionar apenas a parte que compõe a primeira página do *datatable,* ignorando o restante dos registros que serão carregados a medida que alguma ação do *datatable* for realizada (mudar a página, aplicar filtro, buscar, ordenar entre outras).

Introdução

* discutir sobre o problema que será tratado no artigo
* descrever brevemente a solução comumente utilizada e que está inadequada
* descrever brevemente a solução que resolve o problema de uma forma mais adequada

Cenário para analisar o problema e a solução

* descrever em detalhes o estudo de caso que será utilizado para analisar a “má prática” e a “boa prática”

**Estudo de caso considerando a má prática**

* descrever em detalhes o desenvolvimento do estudo de caso considerando a forma inadequada
* deixar claro os problemas trazidos com o uso da má prática

**Estudo de caso considerando a boa prática**

* descrever em detalhes o desenvolvimento do estudo de caso considerando a forma adequada
* deixar claro as vantagens trazidas com o uso da boa prática

**Entendendo o porquê desta ser a melhor solução**

* comparar: boa prática VS má prática