

## Mãos a obra: Filtrando resultados

Na classe `EstoqueWS`, no método `getItens()` coloque um parâmetro para filtrar os dados. No projeto já tem um classe `Filtro` e uma outra `Filtros` preparadas para receber os dados da requisição SOAP. O DAO também já está preparado para a pesquisa usando os dados do filtro. O método `getItens` fica assim:

```
@WebMethod(operationName="todosOsItens")
@WebResult(name="itens")
public ListaItens getItens(Filtros filtros) { //cuidado, plural
    System.out.println("Chamando getItens()");
    List<Filtro> lista = filtros.getLista();
    List<Item> itensResultado = dao.todosItens(lista);
    return new ListaItens(itensResultado);
}
```

Um filtro define um tipo do item ( `Livro`, `Celular` ou `Tablet` ) e o nome do item. A classe já criada é bem simples:

```
public class Filtro {

    private TipoItem tipo;
    private String nome;

    //get e sets
}
```

A classe filtro é um pequeno *wrapper* para embrulhar um filtro e deixar o XML mais expressivo:

```
@XmlRootElement()
@XmlAccessorType(XmlAccessType.FIELD)
public class Filtros {

    @XmlElement(name="filtro")
    private List<Filtro> filtros;

    //construtores

    public List<Filtro> getLista() {
        return filtros;
    }

}
```

Republique o serviço. Atualize o cliente no SoapUI. Se for preciso criar um novo `Request` abaixo da operação `todosOsItens`.

Como ficou o XML SOAP? Tente enviar uma requisição SOAP com apenas 1 filtro, 2 filtros e nenhum.

