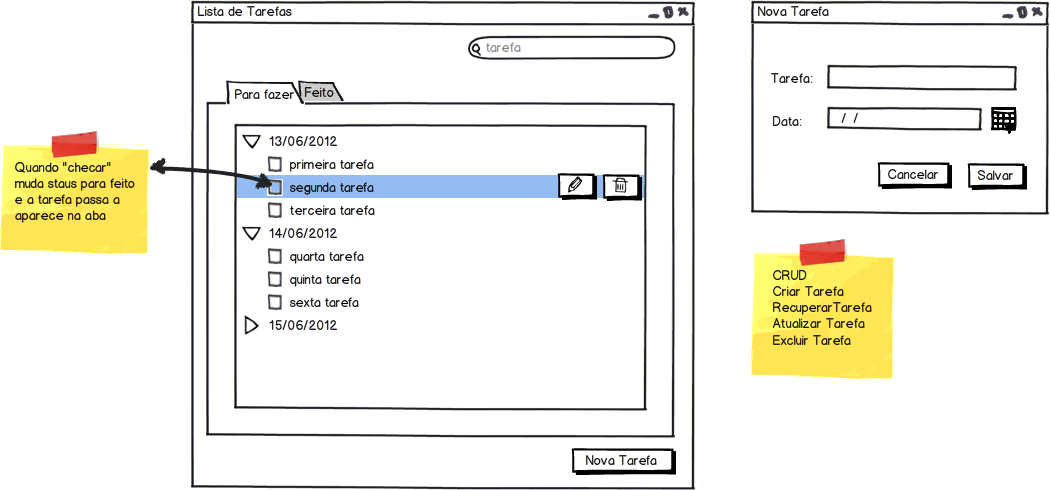
**Exercício 09 – Serviço Tarefa Rest**

**Valor: 02 pontos.**

**Roteiro:**

Imagine a interface para um TodoList – Lista de tarefas



Nosso objetivo é criar serviços REST que possam ser consumidos por qualquer outra aplicação. Não é para criar a interface com usuário.

Para esse exemplo não vamos utilizar banco de dados. Todos os dados ficarão na memória em uma lista como o exemplo List<Tarefa>list = new LinkedList<Tarefa>()

Referência para o exercício: <http://www.mkyong.com/tutorials/jax-rs-tutorials/>

Para o serviço vocês deverão implementar métodos para as seguintes chamadas:

|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP** | **URL** | **Anotação utilizada** | **Objetivo** |
| POST | <http://localhost:8080/tarefa> | @POST  @Produces  @Consumes | Cria uma nova tarefa e retorna para o cliente |
| GET | http://localhost:8080/tarefa/{id} | @GET  @Produces | Retorna em formato json uma tarefa que possui o “id” informado |
| GET | <http://localhost:8080/tarefa/query>?  descricao=”alguma\_coisa” | @GET  @Path  @Produces  @QueryParam | Executa uma query para retornar as tarefas que possuem parte da descrição. Exemplo de chamada:  <http://localhost:8080/tarefa/query>?  Descricao=primeira\_tarefa |
| GET | http://localhost:8080/tarefas | @GET  @Path  @Produces | Retorna em formato json todas as tarefas |
| GET | http://localhost:8080/tarefas/{from}/{to} | @GET  @Path  @Produces | Retorna todas as tarefas que estão dentro do intervalo from/to |
| PUT | <http://localhost:8080/tarefa> | @PUT  @Produces  @Consumes | Edita o conteúdo de uma determinada tarefa e retorna em formato json |
| DELETE | http://localhost:8080/tarefa/{id} | @DELETE  @Produces | Excluir uma tarefa pelo seu id |