



## Conhecendo o projeto

### Transcrição

[00:00] Continuando nosso treinamento de APIs REST com Spring Boot. Agora, já temos nosso projeto concluído: criamos um projeto no site do Spring inicializr e criamos nosso Hello World. Tudo funcionando. Podemos construir nossa API.

[00:17] O projeto que vamos utilizar no curso vai ser a API do fórum da Alura. Vocês, como alunos, já conhecem a Alura, talvez já tenham entrado no fórum e postado alguma dúvida ou lido algo que os alunos postaram. O fórum funciona assim: temos algumas categorias de cursos (como Mobile, Programação e Front-end) e, dentro de cada categoria, estão os cursos, onde os alunos podem postar as dúvidas. Basicamente, eles criam um novo tópico com um título e uma descrição da dúvida. Aparece também o aluno que postou a dúvida, a data, e outros alunos podem responder. Esse vai ser o projeto do nosso treinamento: criar uma API REST do fórum da Alura.

[01:26] Basicamente, nesse fórum vamos ter os tópicos abertos pelos alunos (os tópicos têm o status de quando foram criados, se já foram resolvidos), as respostas, os cursos e categorias de cada curso.

[01:40] Como o foco do treinamento não é criar essas classes de modelo, vamos disponibilizar um link para vocês baixarem as classes de domínio da aplicação. Nós só vamos importar as classes prontas para não perdermos muito tempo com isso e focarmos na API REST em si.

[01:57] Depois que vocês fizerem o download, vocês vão baixar um arquivo chamado `modelo.zip`. Esse arquivo tem as classes de domínio da aplicação.

Vocês vão baixar o link que vamos fornecer. Depois, vão selecionar o projeto "forum", pressionar o botão direito e, em seguida, a opção "import". Na próxima tela, dentro da pasta "General", temos a opção "Archive File". Vocês vão selecionar essa opção e pressionar "Next". No "browser", vocês vão escolher o arquivo `modelo.zip` que vocês baixaram na atividade. Verificar se está marcado ao lado da pasta um "check" indicando estar tudo certinho. Verificar se em "Into folder:" está o projeto "forum" e apertar "Finish". Se você olhar no "src/main/Java", além do pacote "controller", vocês vão ver o pacote "modelo". Nele, estão as classes de domínio da nossa aplicação.

[02:48] Passando um pouco por essas classes. Temos a classe `Usuário.java`, que vai representar um aluno ou aluna cadastrado no fórum e que tem as informações: "*id*", "*nome*", "*e-mail*" e "*senha*" de acesso ao fórum.

```
package br.com.alura.forum.modelo;
```

```
public class Usuario {
```

```
    private Long id;  
    private String nome;  
    private String email;  
    private String senha;  
    //...  
}
```

[COPIAR CÓDIGO](#)

Temos a classe `Tópico.java`, que é a classe principal do projeto. E esse tópico tem: um "*id*"; o "*título*"; a "*mensagem*"; a "*data de criação*", que é um "LocalDateTime, API de datas do java 8; o "*status*", que é um `enum`, sendo que, toda vez que você cria um tópico, o primeiro status em que ele se encontra é "NAO-RESPONDIDO" (ou seja, acabei de criar um tópico e ele ainda não tem nenhuma resposta); o "*usuário*", que é o autor desse tópico; o "*curso*", isto é, de que curso é essa dúvida; e uma "*lista*" com as respostas postadas.

```
public class Topico {  
  
    private Long id;  
    private String titulo;  
    private String mensagem;  
    private LocalDateTime dataCriacao = LocalDateTime.now();  
    private StatusTopico status =  
StatusTopico.NAO_RESPONDIDO;  
    private Usuario autor;  
    private Curso curso;  
    private List<Resposta> respostas = new ArrayList<>();  
    //...  
}
```

[COPIAR CÓDIGO](#)

[03:42] Temos o `StatusTopico`, que é um `enum` com algumas constantes (quais os status que esse tópico pode ter). Quando é criado, ele está no status `"NAO_RESPONDIDO"`. Se alguém posta uma resposta, ele vai para o `"NAO_SOLUCIONADO"`. Quando o autor ou administrador do fórum marca a resposta como solução, ele passa para o status `"SOLUCIONADO"`. Depois, se ele for fechado (quando ninguém pode mais postar respostas), vai para o status `"FECHADO"`.

```
package br.com.alura.forum.modelo;
```

```
public enum StatusTopico {  
  
    NAO_RESPONDIDO,  
    NAO_SOLUCIONADO,  
    SOLUCIONADO,  
    FECHADO;  
  
}
```

[COPIAR CÓDIGO](#)

[04:10] Também temos a classe `Resposta.java`, que apresenta a resposta de um tópico. Nessa classe, temos: o `"id"`; a `"mensagem"`; o `"tópico"`; `"dataCriação"` (quando essa resposta foi criada); `"autor"` (quem é o autor); `"solucao"` (se essa resposta é a solução ou não, pois, quando eu crio uma resposta, ela não é a solução até alguém marcar como solucionado, por isso, no caso, está como `"false"`).

```
package br.com.alura.forum.modelo;

import java.time.LocalDateTime;

public class Resposta {

    private Long id;
    private String mensagem;
    private Topico topico;
    private LocalDateTime dataCriacao = LocalDateTime.now();
    private Usuario autor;
    private Boolean solucao = false;

    //...
}
```

[COPIAR CÓDIGO](#)

[04:31] Por último, temos a classe `Curso.java`, que representa o curso onde o tópico foi aberto. O curso tem: o `"id"`; o `"nome"`; e a `"categoria"` (se é programação, design, front end, etc.).

```
package br.com.alura.forum.modelo;

public class Curso {

    private Long id;
    private String nome;
    private String categoria;
```

```
public Curso(String nome, String categoria) {  
    this.nome = nome;  
    this.categoria = categoria;  
}  
//...  
}
```

[COPIAR CÓDIGO](#)

[04:45] São essas cinco classes ( `Usuario.java` , `Topico.java` , `StatusTopico.java` , `Resposta.java` , `Curso.java` ) que representam o domínio da nossa aplicação. Para não perdermos muito tempo digitando o código, vocês vão fazer a atividade para importar no projeto. Cada classe é uma classe Java. Não tem nenhuma anotação, não é uma classe do Spring - não tem nada ver - é só uma classe Java pura (Java Bean, POJO). Nessas classes só estão os atributos, o método `quals()` , o método `hashCode()` , e os métodos `getId()` e `setId()` . Nada muito complicado.

[05:18] Depois que vocês fizerem o import e estiverem com as classes prontas, bonitinhas, vamos passar a construir nosso primeiro endpoint, nossa primeira URL, que é para devolver a lista com os tópicos que estão abertos no nosso projeto. Terminando essa atividade, vocês vão para o próximo curso para montarmos de fato nossa API. Isto é, começarmos a montar nossa API com o primeiro endpoint para listar todos os tópicos que estão criados no fórum.