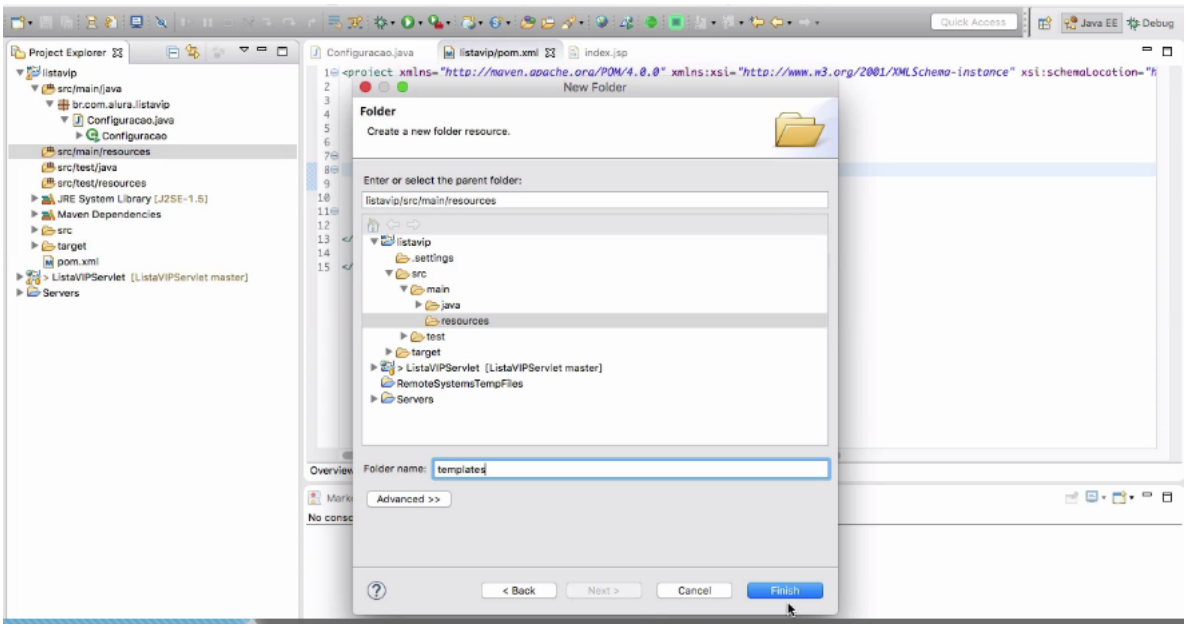
01

Criando nossa primeira tela

Transcrição

Na aplicação entregue pelo cliente, as páginas são criadas com *JSP* padrão. Mas como estamos utilizando *Spring Boot* com o *starter* web que nos disponibiliza o *Spring MVC*, utilizaremos outra *engine* de templates. Uma do próprio *Spring*, chamada *Thymeleaf*.

Iniciaremos a criação da nossa primeira página criando uma nova pasta dentro da pasta `resources`, e chamaremos esta de `templates`. Dentro desta, criaremos um arquivo *HTML* simples com uma mensagem de bem vindo.



O arquivo *HTML* se chamará `index.html` e terá o seguinte código.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <h1>Bem-vindo ao sistema Lista VIPs</h1>
</body>
</html>
```

O próximo passo será remover o código presente na classe de configuração que trata a requisição para uma outra classe de *controller*, visto que, a classe de configuração deve conter código apenas de configuração.

Criaremos uma classe chamada `ConvidadoController` e nesta criaremos um método chamado `index` que mapeará a requisição em `/` e retornará como *String* o nome do template que criamos, o `index.html`. Assim teremos.

```
@Controller
public class ConvidadoController {
```

```
    @RequestMapping("/")
    public String index(){
        return "index";
    }

}
```

Esta classe deve estar no mesmo pacote da classe de configuração.

Note que retornamos apenas o nome do template, sem a extensão do arquivo. Isto por que o *engine* de *template* fará as associações. Se iniciarmos a aplicação agora, teremos a mesma tela de erro que tivemos quando iniciamos o projeto. Isto por que não adicionamos o *engine* de *templates* na nossa aplicação. No arquivo `pom.xml`, teremos então que adicionar a dependência do `thymeleaf starter`.

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
    <version>1.4.1.RELEASE</version>
</dependency>
```

A versão utilizada no vídeo é a `1.3.6`, porém, você pode utilizar a versão mais recente sem problema algum.

Por último, resolveremos alguns problemas do nosso template que por enquanto está muito simples, sem nenhum estilo. O *HTML* da aplicação antiga, que estamos migrando tinha o seguinte *HTML* em sua *index*.

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>ListaVIP</title>
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <div class="jumbotron" align="center" style="margin-top: 50px;">
            <h1>Seja bem vindo ao ListaVIPs</h1>
            <div align="center">
                <a href="listavip" class="btn btn-lg btn-primary">Clique aqui para ver a lista
            </div>
        </div>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

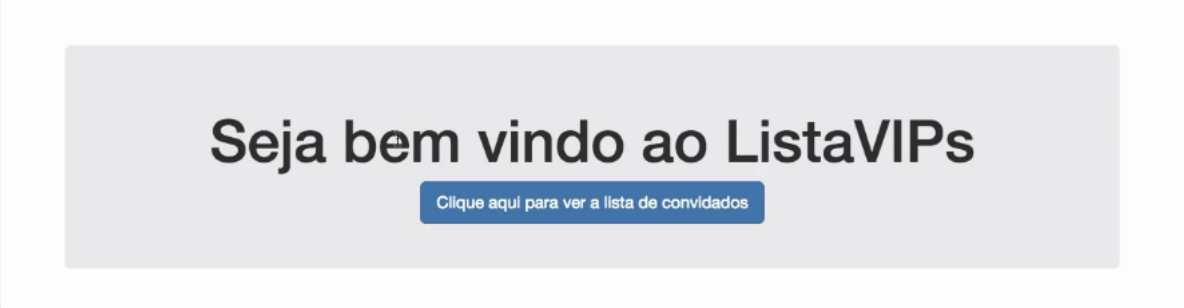
O primeiro ponto a ser observado é que ele usa o *Bootstrap*. O segundo ponto é que a *engine* que adicionamos no nosso projeto é bem detalhista. Teremos erros por exemplo, se esquecermos de fechar alguma *tag*. Por isso cuidado. **Sempre feche todas as tags HTML.**

Para a adição dos arquivos estáticos do *Bootstrap*, criaremos uma nova pasta chamada `static`, dentro da pasta `resources`. Baixaremos o *Bootstrap* em getbootstrap.com (http://www.getbootstrap.com) e o descompactaremos nesta pasta que criamos.

Copiaremos o código do antigo *index*, fecharemos as tags corretamente e o colaremos no lugar do *index* atual. Que terá ao final, o seguinte código.

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>ListaVIP</title>
        <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" />
    </head>
    <body>
    <div class="container">
        <div class="jumbotron" align="center" style="margin-top: 50px;">
            <h1>Seja bem-vindo ao ListaVIPs</h1>
            <div align="center">
                <a href="listavip" class="btn btn-lg btn-primary">Clique aqui para ver a lista
            </div>
        </div>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>
    </body>
</html>
```

Agora, ao iniciarmos a aplicação, teremos uma página de bem-vindo mais agradável.



Observações: Muito do que vimos até agora, está dentro de um conceito muito comum presente nos *frameworks* mais modernos que é a Convenção sobre a Configuração. Antes precisaríamos configurar uma série de recursos para ter a aplicação funcionando. Agora apenas seguindo algumas convenções, pulamos todas as configurações e apenas focamos no que é importante.

Das convenções que vimos até aqui temos: Os *templates* das páginas são guardados na pasta *templates* dentro de *resources* e também a convenção de onde armazenar os arquivos estáticos (*css, js, imagens, etc.*) que ficam dentro da pasta *static*.