



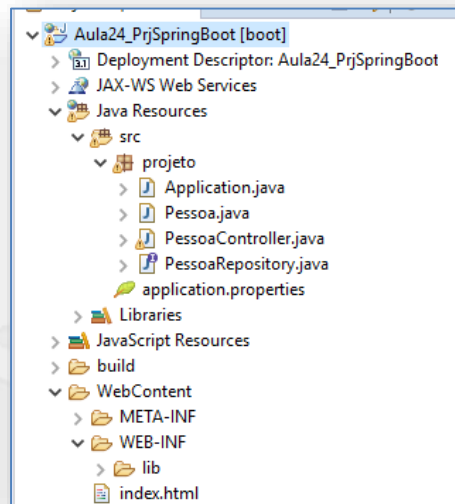
Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

PROJETO ECLIPSE:

Estrutura do projeto: (Dynamic Web Project)



LIBS:

antlr-2.7.7.jar
aspectjweaver-1.8.13.jar
attoparser-2.0.4.RELEASE.jar
classmate-1.3.4.jar
dom4j-1.6.1.jar
hibernate-commons-annotations-5.0.1.Final.jar
hibernate-core-5.2.14.Final.jar
hibernate-jpa-2.1-api-1.0.0.Final.jar
hibernate-validator-6.0.7.Final.jar
HikariCP-2.7.8.jar
jackson-annotations-2.9.0.jar
jackson-core-2.9.4.jar
jackson-databind-2.9.4.jar
jackson-datatype-jdk8-2.9.4.jar
jackson-datatype-jsr310-2.9.4.jar
jackson-module-parameter-names-2.9.4.jar
jandex-2.0.3.Final.jar
javassist-3.22.0-GA.jar
javax.annotation-api-1.3.2.jar
javax.transaction-api-1.2.jar
jboss-logging-3.3.2.Final.jar



Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

jul-to-slf4j-1.7.25.jar
log4j-api-2.10.0.jar
log4j-to-slf4j-2.10.0.jar
logback-classic-1.2.3.jar
logback-core-1.2.3.jar
mysql-connector-java-5.1.23-bin.jar
slf4j-api-1.7.25.jar
snakeyaml-1.19.jar
spring-aop-5.0.4.RELEASE.jar
spring-aspects-5.0.4.RELEASE.jar
spring-beans-5.0.4.RELEASE.jar
spring-boot-2.0.0.RELEASE.jar
spring-boot-autoconfigure-2.0.0.RELEASE.jar
spring-boot-starter-2.0.0.RELEASE.jar
spring-boot-starter-aop-2.0.0.RELEASE.jar
spring-boot-starter-data-jpa-2.0.0.RELEASE.jar
spring-boot-starter-jdbc-2.0.0.RELEASE.jar
spring-boot-starter-json-2.0.0.RELEASE.jar
spring-boot-starter-logging-2.0.0.RELEASE.jar
spring-boot-starter-thymeleaf-2.0.0.RELEASE.jar
spring-boot-starter-tomcat-2.0.0.RELEASE.jar
spring-boot-starter-web-2.0.0.RELEASE.jar
spring-context-5.0.4.RELEASE.jar
spring-core-5.0.4.RELEASE.jar
spring-data-commons-2.0.5.RELEASE.jar
spring-data-jpa-2.0.5.RELEASE.jar
spring-expression-5.0.4.RELEASE.jar
spring-jcl-5.0.4.RELEASE.jar
spring-jdbc-5.0.4.RELEASE.jar
spring-orm-5.0.4.RELEASE.jar
spring-tx-5.0.4.RELEASE.jar
spring-web-5.0.4.RELEASE.jar
spring-webmvc-5.0.4.RELEASE.jar
thymeleaf-3.0.9.RELEASE.jar
thymeleaf-extras-java8time-3.0.1.RELEASE.jar
thymeleaf-spring5-3.0.9.RELEASE.jar
tomcat-embed-core-8.5.28.jar
tomcat-embed-el-8.5.28.jar
tomcat-embed-websocket-8.5.28.jar
unobscure-1.1.5.RELEASE.jar
validation-api-2.0.1.Final.jar

package projeto;



SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="pessoa")
public class Pessoa implements
Serializable{

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;
    @Column(length=50)
    private String nome;
    @Column(length=80,unique=true)
    private String email;
    @Column(columnDefinition="enum('m','f') ")
    private String sexo;

    public Pessoa() {
    }

    public Pessoa(Long id, String nome, String email, String
sexo) {
        this.id = id;
        this.nome = nome;
        this.email = email;
        this.sexo = sexo;
    }

    @Override
    public String toString() {
        return "Pessoa [id=" + id + ", nome=" + nome + ",
email=" + email + ", sexo=" + sexo + " ]";
    }
}
```



Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
}  
  
public Long getId() {  
    return id;  
}  
public void setId(Long id) {  
    this.id = id;  
}  
public String getNome() {  
    return nome;  
}  
public void setNome(String nome) {  
    this.nome = nome;  
}  
public String getEmail() {  
    return email;  
}  
public void setEmail(String email) {  
    this.email = email;  
}  
public String getSexo() {  
    return sexo;  
}  
public void setSexo(String sexo) {  
    this.sexo = sexo;  
}  
public static long getSerialVersionUID() {  
    return serialVersionUID;  
}  
}
```



Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
package projeto;
```

```
import java.util.List;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
public interface PessoaRepository  
extends CrudRepository<Pessoa, Long>{
```

```
    public Pessoa findByEmail(String email);
```

```
    public List<Pessoa> findBySexo(String sexo);
```

```
}
```

```
package projeto;
```

```
import java.util.Collections;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.http.MediaType;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PathVariable;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import org.springframework.web.bind.annotation.ResponseBody;
```

```
import org.springframework.web.servlet.ModelAndView;
```

```
@Controller
```

```
@RequestMapping("/rs")
```

```
public class PessoaController {
```

```
    @Autowired
```

```
    private PessoaRepository dao;
```

```
    @PostMapping(value="/pessoa",  
consumes=MediaType.APPLICATION_FORM_URLENCODED_VALUE,  
produces = MediaType.TEXT_HTML_VALUE )
```



Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
public @ResponseBody String salvar(@RequestParam String
nome, @RequestParam String email,
    @RequestParam String sexo ) {
    try {
        Pessoa p = new Pessoa(null, nome, email, sexo);
        dao.save(p);
        return "Dados Gravados Com Sucesso "
+ "<a href=http://localhost:8888/rs/pessoas>Listar</a>";
    }catch(Exception ex) {
        return "Error :"+ ex.getMessage()+ "<a
href=/pessoas>Listar</a>";
    }
}

@GetMapping(value="/pessoa/busca/{email}",
    produces=MediaType.APPLICATION_JSON_VALUE)
public @ResponseBody Pessoa busca(@PathVariable(name="email")
String email ) {
    try {
        return dao.findByEmail(email);
    }catch(Exception ex) {
        return null;
    }
}

@PostMapping(value="/pessoa/buscar",
    consumes=MediaType.APPLICATION_FORM_URLENCODED_VALUE,
    produces=MediaType.APPLICATION_JSON_VALUE
)
@ResponseBody
public Pessoa buscar(@RequestParam String email) {
    try {
        Pessoa p = new Pessoa(null, null, email, null);
        Pessoa busca = dao.findByEmail(p.getEmail());
        return busca;
    }catch(Exception ex) {
        return null;
    }
}

@GetMapping(value="/pessoas",
    produces=MediaType.APPLICATION_JSON_VALUE)
public @ResponseBody List<Pessoa> findAll() {
```



SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
        try {
            return (List<Pessoa>) dao.findAll();
        } catch (Exception ex) {
            return Collections.emptyList();
        }
    }

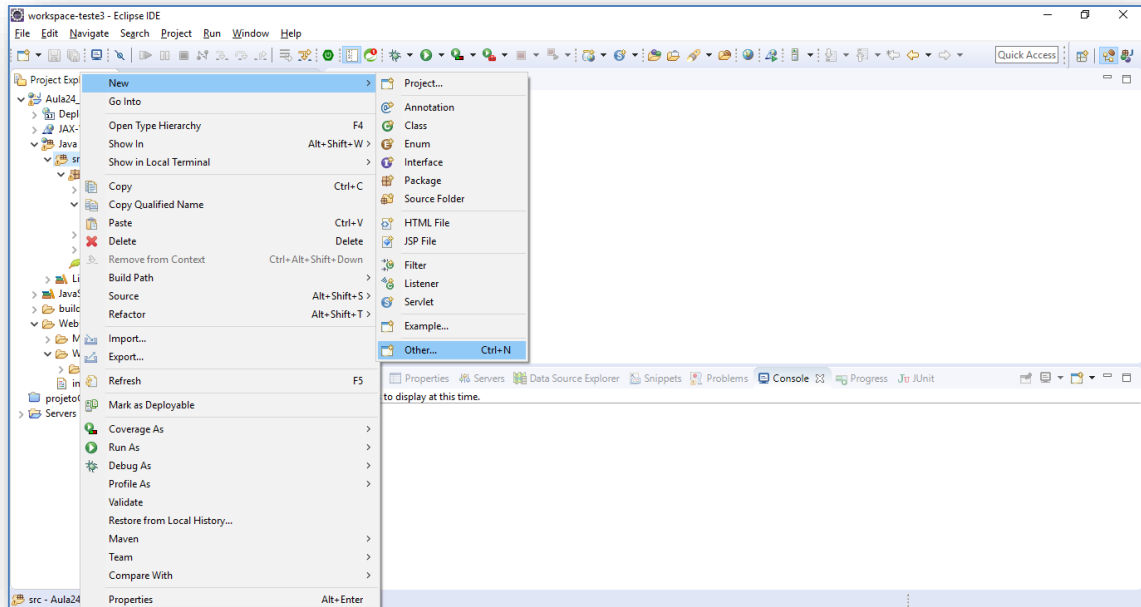
    @GetMapping(value="/pessoa/buscasexo/{sexo}",
        produces=MediaType.APPLICATION_JSON_VALUE)
    public @ResponseBody List<Pessoa> buscasexo(
        @PathVariable(name="sexo")
        String sexo ) {
        try {
            return dao.findBySexo(sexo);
        } catch (Exception ex) {
            return null;
        }
    }

    @PostMapping(value="/pessoa/buscarSexo",
        consumes=MediaType.APPLICATION_FORM_URLENCODED_VALUE,
        produces=MediaType.APPLICATION_JSON_VALUE
    )
    @ResponseBody
    public List<Pessoa> buscarSexo(@RequestParam String sexo) {
        try {
            Pessoa p = new Pessoa(null, null, null, sexo);
            List<Pessoa> busca = dao.findBySexo(p.getSexo());
            return busca;
        } catch (Exception ex) {
            return null;
        }
    }
}
```

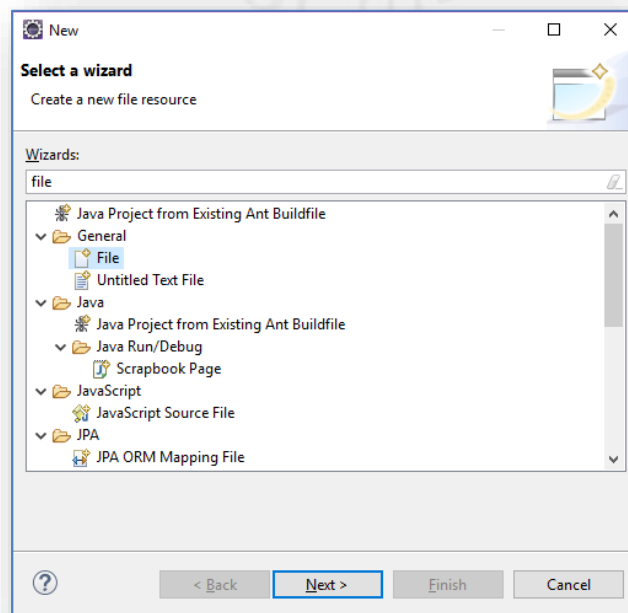
SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

Para criar o arquivo:

Clicar em “src” com o botão direito -> new -> other

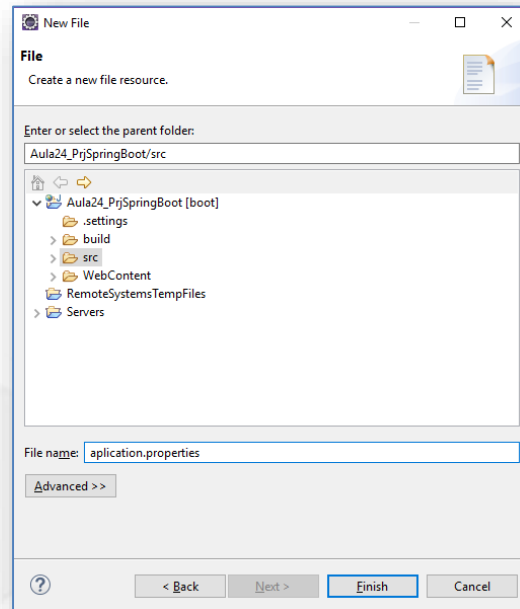


File -> next...



SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

Digitar o nome do arquivo “application.properties” -> finish...



application.properties

//ATENÇÃO PARA A PORTA QUE SERÁ INICIADA PELO TOMCAT

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/springboot?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=coti
server.port = 8888
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix= .jsp
```

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
package projeto;
```

```
import org.springframework.boot.SpringApplication;  
import  
org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class Application {
```

```
//start no Tomcat ....
```

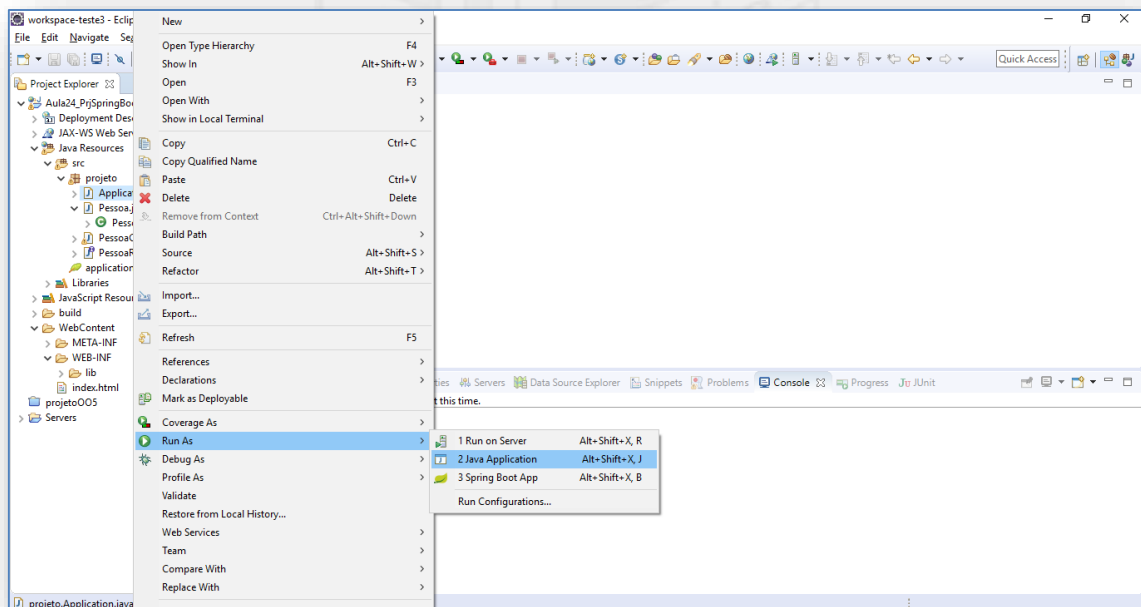
```
    public static void main(String args[]) {  
        SpringApplication.run(Application.class, args);
```

```
    }
```

```
}
```

Para rodar o projeto:

Clicar em application (Classe onde tem o main) -> run as -> java
application





Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

Aguardar executar...

```
workspace-teste3 - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Problems Console Progress JUnit
Application (3) [Java Application] C:\Program Files\Java\jre1.8.0_172\bin\javaw.exe (25 de set de 2018 10:51:32)
2018-09-25 10:51:48.053 INFO 13232 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfr
name: default
...
2018-09-25 10:51:48.327 INFO 13232 --- [main] org.hibernate.Version : HHH000412: Hibernate Core [5.2.14.Final]
2018-09-25 10:51:48.332 INFO 13232 --- [main] org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found
2018-09-25 10:51:48.508 INFO 13232 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations [5.4.12.Final]
2018-09-25 10:51:48.905 INFO 13232 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
2018-09-25 10:51:50.627 INFO 13232 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2018-09-25 10:51:53.691 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.web.servlet.mvc.annotation.AnnotationMethodMappingHandlerAdapter
2018-09-25 10:51:53.951 WARN 13232 --- [main] aWebConfiguration$JpaWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. This causes the opening of one JDBC connection per servlet. [See https://github.com/spring-projects/spring-data-jpa/issues/155 for details]
2018-09-25 10:51:54.160 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto handler with method [GET]
2018-09-25 10:51:54.168 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto handler with method [POST]
2018-09-25 10:51:54.169 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto handler with method [POST]
2018-09-25 10:51:54.171 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto handler with method [POST]
2018-09-25 10:51:54.172 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto handler with method [POST]
2018-09-25 10:51:54.188 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto handler with method [POST]
2018-09-25 10:51:54.190 INFO 13232 --- [main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "[]" onto handler with method [POST]
2018-09-25 10:51:54.414 INFO 13232 --- [main] s.w.s.h.handler.SimpleUrlHandlerMapping : Mapped URL path [/*] onto handler with type [org.springframework.web.servlet.mvc.annotation.AnnotationMethodMappingHandlerMapping]
2018-09-25 10:51:54.633 INFO 13232 --- [main] o.s.j.e.a.AnnotationMethodMapping : Adding welcome page: ServletContext resource [/]
2018-09-25 10:51:54.799 WARN 13232 --- [main] ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:
2018-09-25 10:51:55.077 INFO 13232 --- [main] o.s.j.e.w>WelcomePageHandlerMapping : Adding welcome page: ServletContext resource [/]
2018-09-25 10:51:55.950 INFO 13232 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
2018-09-25 10:51:55.953 INFO 13232 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'dataSource' has been auto-detected
2018-09-25 10:51:55.977 INFO 13232 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Located MBean 'dataSource': registering with JMX
2018-09-25 10:51:56.120 INFO 13232 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8888 (http)
2018-09-25 10:51:56.129 INFO 13232 --- [main] projeto.Application : Started Application in 21.683 seconds
```

index.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
</head>
<body>
<h2>Cadastrar Pessoa</h2>
<form method="post" action="rs/pessoa">
  Nome<br/>
  <input type="text" name="nome" id="nome" value="" size="50"
  required="required" />
<br/>Email<br/>
  <input type="text" name="email" id="email" value=""
  size="50"
  required="required" />
<br/>Sexo<br/>
  <select name="sexo" id="sexo" required="required">
    <option value="m">Masculino</option>
    <option value="f">Feminino</option>
  </select>
  <button type="submit">Enviar os Dados</button>
```



Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
<br/>
<p th:text="{msg}" />
<br/>
<p th:text="{pessoa}" />
</form>
<br/>
<a href="/rs/pessoas">Listar Pessoas</a>

<br/>
<hr/>
</body>
<hr/>
<form method="post" action="/rs/pessoa/buscarSexo">
<br/>Email<br/>
<input type="text" name="sexo" id="sexo" value="" size="5"
required="required" />

<button type="submit">Buscar pelo sexo</button>
<br/>
<p th:text="{msg}" />
<br/>
<p th:text="{pessoa}" />
</form>
</body>
</body>
<hr/>
<form method="post" action="/rs/pessoa/buscarSexo">
<br/>Email<br/>
<input type="text" name="sexo" id="sexo" value="" size="5"
required="required" />

<button type="submit">Buscar pelo sexo</button>
<br/>
<p th:text="{msg}" />
<br/>
<p th:text="{pessoa}" />

<br/>
<p th:text="{busca}" />
</form>
</body>
```



Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

```
</html>
```

No browser iniciar pela porta que foi configurada.

<http://localhost:8888/>

The screenshot shows a web browser window with the URL localhost:8888/. The page content is as follows:

Cadastrar Pessoa

Nome

Email

Sexo
Masculino

[Listar Pessoas](#)

Email

Email

Escola de Nerds
www.cotiinformatica.com.br



Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

Cadastrando...

The screenshot shows a web browser window at localhost:8888. The page title is "Cadastrar Pessoa". The form contains the following fields and controls:

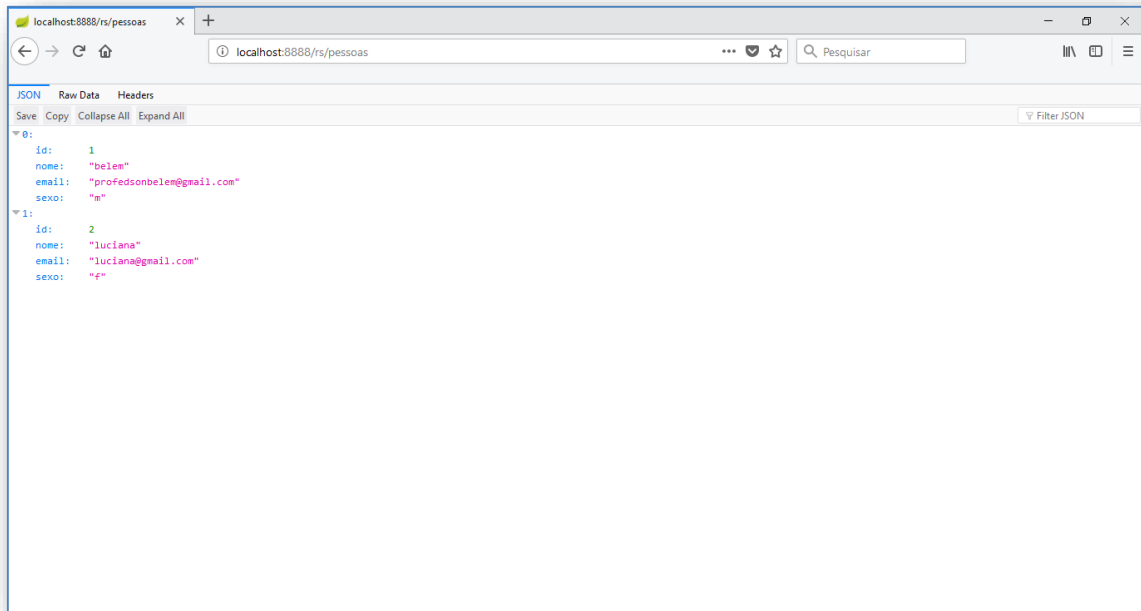
- Nome:** Input field containing "belem".
- Email:** Input field containing "profedsonbelem@gmail.com".
- Sexo:** Dropdown menu with "Masculino" selected.
- Enviar os Dados:** Button.
- Listar Pessoas:** A link below the form.
- Search Section:** Two identical sections, each with an "Email" input field and a "Buscar pelo sexo" button.

Gravado...

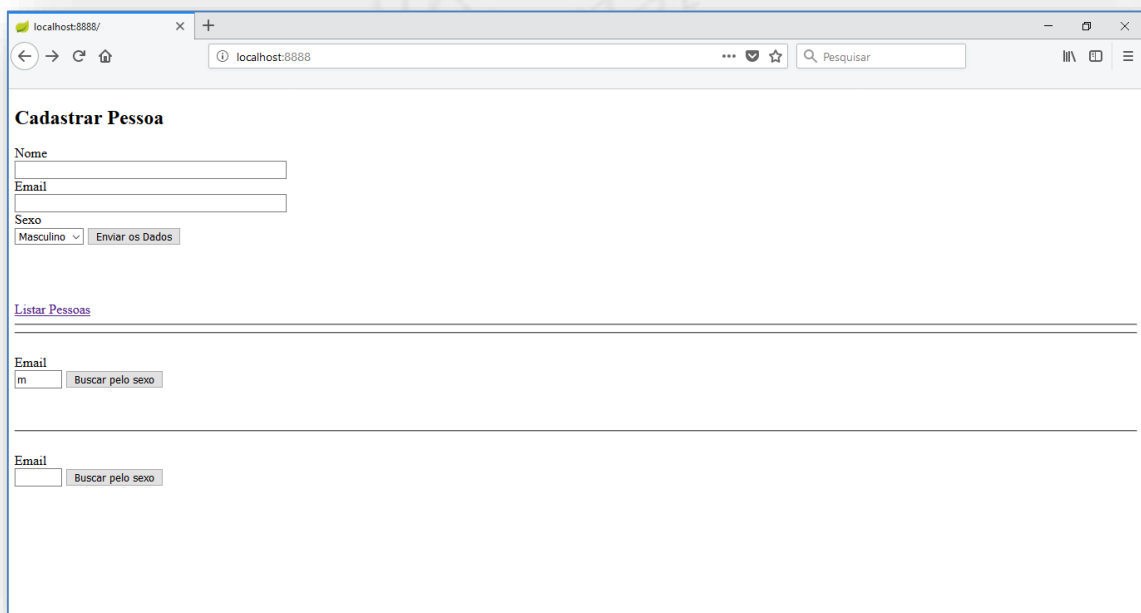
The screenshot shows a web browser window at localhost:8888/rs/pessoa. The page displays the message "Dados Gravados Com Sucesso" followed by a link to "Listar".

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

Clicando em listar...



Buscando pelo sexo (m)...





Java WebDeveloper - Prof. Edson Belém

Aula

SpringBoot2.0, Hibernate5.2, MySql, TomCat8.5, gravar,
listar, buscar pelo sexo

Resultado...

```
localhost:8888/rs/pessoa/buscarSexo
{
  "id": 1,
  "nome": "belem",
  "email": "profedsonbelem@gmail.com",
  "sexo": "m"
}
```

