Cheat Sheet Laboratório de Engenharia de Software:

JDBC, JPA, Hibernate e Repository

```
EntityManagerFactory <objEntityManagerFactory> =
Persistence.createEntityManagerFactory(<nome do
persistence-unit>);
TypedQuery <objTypedQuery> =
<objEntityManager>.createQuery(<query>, <Tipo do objeto
de  resultado>);
String <valor> = <objeto ResultSet>.getString( <nome da
coluna> );
int <valor> = <objeto ResultSet>.getInt( <nome da
coluna> );
java.sql.Date <valor> = <objeto ResultSet>.getDate(
<nome da coluna> );
<objeto EntityManager>.persist( <objeto> );
ArrayList<Tipo do objeto resultado> <lista> =
<objTypedQuery>.getResultList();
ResultSet <objeto ResultSet> = <objeto
PreparedStatement>.executeQuery();
<Tipo do objeto resultado> <obj> =
<objTypedQuery>.getSingleResult();
@Column
@Entity
@Repository
@Id
@GeneratedValue
public interface BrinquedoRepository extends
CrudRepository<Entidade, Tipo do Id>{
    List<Entidade> findByNome(String nome);
 }
<Tipo> <objeto> =
<objeto EntityManager>.getReferenceById( <objeto id>,
<Tipo do objeto>);
EntityManager <objEntityManager> =
<objEntityManagerFactory>.createEntityManager();
 <objeto ResultSet>.next();
<objeto EntityManager>.getTransaction().begin();
<objeto EntityManager>.getTransaction().commit();
<objeto EntityManager>.remove( <objeto> );
int <numero inteiro> = <objeto
PreparedStatement>.executeUpdate();
PreparedStatement <objeto PreparedStatement> =
<objeto Connection>.prepareStatement( <String SQL> );
```

HttpServlet

```
protected void doGet(HttpServletRequest req,
HttpServletResponse res) { }
protected void doPost(HttpServletRequest req,
HttpServletResponse res) { }
<Objeto RequestDispatcher>.forward(<Objeto
HttpServletRequest>, <Objeto HttpServletResponse>);
RequestDispatcher <obj> =<Objeto
HttpServletRequest>.getRequestDispatcher( <nome da
pagina> );
Objeto <valor objeto> =
<Objeto HttpServletRequest>.getSession().getAttribute(
<nome do objeto> );
<Objeto HttpServletRequest>.getSession().setAttribute(
<nome do objeto> , <objeto>);
String <valor parametro> = <Objeto
HttpServletRequest>.getParameter( <nome do parametro>
);
class MeuServlet extends HttpServlet { }
```

Spring Boot

```
<obj ModelView>.addObject("<chave>", <objeto>);
@RequestMapping("/caminho")
ModelAndView <obj ModelView> = new
ModelAndView("<pagina>");
@ModelAttribute("<chave do objeto>") <Tipo> <nome
Variavel>
@Bean
@Autowired
```

Java Básico

```
ArrayList<Tipo da informacao> = new ArrayList<Tipo da
Informacao>();
CAST: <Tipo da Informacao> <variavel> = ( <Tipo da
Informacao> ) <Objeto do tipo Object> ;
java.util.Date <objeto Date>  =
new java.util.Date( <objeto java.sql.Date>.getTime() );
```

Java Server Faces

```
Object <obj> = <obj
context>.getApplication().evaluateExpressionGet(ctx,
"#{objeto}", Object.class);

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.org/ui">
<html xmlns:th="http://www.thymeleaf.org">
<h:body> <h:form > <h:panelGrid columns="N"> <h:panelGroup>
<h:input"[Text|Secret]" id="<nome>" value="#{} "/>
<h:commandButton action="#{}" id="<nome>" value="<valor> "/>
<h:dataTable value="#{…}"  var="<variavel>">
</h:dataTable>
<h:column> … </h:column>
<f:facet name="header"> … </facet>
<obj FacesContext>.getViewRoot().getViewId();
@ManagedBean
<obj navigation>.handleNavigation(<obj context>, null,
"<pagina>?faces-redirect=true");
@SessionScoped
<obj context>.renderResponse();
@ViewScoped
FacesContext <obj context> = <obj
phaseEvent>.getFacesContext();
NavigationHandler <obj navigation> = <obj
context>.getApplication().getNavigationHandler();
@RequestScoped

<button type="submit" name="cmd" value="salvar" class="btn
btn-primary">Salvar</button>
```

Spring MVC

```
public class Configuracao implements WebMvcConfigurer{ }
public class WebInitializer implements
WebApplicationInitializer {
    @Override
    public void onStartup(ServletContext servletContext)
throws ServletException {
        AnnotationConfigWebApplicationContext
        ctx = new AnnotationConfigWebApplicationContext();
        ctx.register(Configuracao.class);
        ctx.setServletContext(servletContext);
        Dynamic servlet = servletContext
        .addServlet("dispatcher", new
DispatcherServlet(ctx));
        servlet.addMapping("/");
        servlet.setLoadOnStartup(1);
    }
}
@Configuration
@EnableWebMvc
@ComponentScan("edu.curso.controller")
@EnableTransactionManagement
@EnableJpaRepositories(basePackages =
{"edu.curso.repository"})
```

Thymeleaf

```
<form th:action="@{/entidade/atualizar}" method="post"
th:object="${entidadeAtual}">
<input th:field="*{id}" name="id" id="txtId" class="form-
control"/>
<table class="table"
  th:unless="${#lists.isEmpty(entidadeLista)}">
  <thead class="thead-dark">
    <tr>
      <th>Id</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="e : ${entidadeLista}">
      <td th:text="${e.id}"/>
    </tr>
  </tbody>
</table>
```