Developing Applications with Java EE 6 on WebLogic Server 12c

- **Formato do curso:** Presencial
- **Preço:** 2250€
- **Nível:** Avançado
- **Duração:** 30 horas

The Developing Applications with Java EE 6 on WebLogic Server 12c course provides students with the knowledge to build and deploy enterprise applications that comply with Java Platform,Enterprise Edition 6 Web Profile. The technologies presented in this course include annotations,Session Enterprise JavaBeans (EJBs), the Java Persistence API (JPA), servlets, JavaServer Pages (JSPs) , JavaServer Faces (JSF), Contexts and Dependency Injection (CDI), and Bean Validation.

The primary focus is the creation of a web-based application that is accessible from desktop and mobile web browsers using JSF technology.

Students gain hands-on experience through labs that build an end-to-end application. Students will create web-based user interfaces, primarily using JSF Facelet pages along with JSPs and servlets. Web-based interfaces for both desktop and mobile devices are developed by students.User input is validated using the Bean Validation API and data is persisted using JPA and optimistic locking. The labs explore session EJB components which are used to enable container managed transactions and enhance application performance through the caching of data.

Students learn to assemble an application and how to deploy an application into an application server (Java EE platform runtime environment). Lab exercises are performed using the NetBeans IDE and Oracle WebLogic Server.

Pré-requisitos

Required Prerequisites

- Experience with the Java programming language
- Familiarity with HTML and CSS
- Familiarity with relational database theory and the basics of structured query language (SQL)
- Familiarity with the use of an IDE
- Java SE 7 Programming

Suggested Prerequisites

- Java SE 7: Develop Rich Client Applications

Objectivos

- Use the Java EE Web Profile
- Develop and run an EJB technology application
- Develop basic Java Persistence API entity classes to enable database access
- Develop a web-based user interface using JSF, Servlets, and JSPs
- Design applications to use CDI
- Use IDEs and Application Servers for Java EE development
- Create mobile web applications
- Validate data using Bean Validation
- Secure Enterprise Applications
- Use Logging
- Install Oracle WebLogic Sever 12c zip file distribution

Programa

Java Platform, Enterprise Edition

- Describe the purpose of the Java EE Platform
- Describe the needs of enterprise applications
- List the various Java EE specifications
- Compare services and libraries
- Describe the Java EE Web Profile
- Describe the EE application tiers and layers

Enterprise Development Tools and Application Servers

- Describe the purpose of an application server
- Identify the potential selection criteria used when choosing an application server
- Install the Oracle WebLogic Server 12c Zip Distribution
- Describe the properties of Java EE components
- Describe the process of developing a Java EE application
- Describe how to configure and package Java EE applications
- List EE supporting features provided by integrated development environments (IDEs)

JavaBeans, Annotations, and Logging

- Describe the Java SE features that are used extensively in enterprise applications
- Create POJO JavaBeans components
- Log application activity and errors
- Write to server logs
- Describe common Java SE annotations and features
- Develop Java annotations
- Describe the role of annotations in Java EE

Web Component Model

- Describe the HTTP request-response model
- Define the difference between Java Servlets, JSP, and JSF components
- Implement application layering and the MVC Pattern
- Avoid thread safety issues in web components
- Use the Expression Language

Developing with JavaServer Faces Technology

- Evaluate the role of JavaServer Faces (JSF) technology as a presentation mechanism
- Describe the flow of the JSF life cycle
- Author JSF pages using Facelets
- Process form submissions and use JSF managed beans
- Describe the use of JSF tag libraries
- Use the appropriate annotation to control the scope of a bean instance
- Use a component to iterate over values in a collection

Using AJAX and Composite Components with JSF

- Define Asynchronous JavaScript and XML (AJAX)
- Describe how JSF Components can be enhanced with AJAX
- Use the tag
- Describe how AJAX request integrates with the JSF life cycle
- Define a composite component
- Create a JSF composite component

Apache Trinidad JSF Component Library and Mobile Development

- Create JavaServer Faces (JSF) pages that use Apache Trinidad components
- Create a JSF-based mobile application
- Dynamically apply Cascading Style Sheets (CSS) with Trinidad Skinning
- Use the HTML5 video tag

Dependency Injection With CDI

- Create managed bean compatible classes
- Inject managed beans
- Qualify the bean being requested at an injection point
- Use CDI alternatives

Using JSF and Bean Validation

- Define the approach JSF uses to convert and validate input data
- Use built-in validation constraints provided with JSF
- Use built-in validation constraint annotations provided by Bean Validation
- Create a custom Bean Validation constraint

Developing Servlets

- Describe the servlet API
- Use the request and response APIs
- Set response headers
- Create text and binary response bodies
- Process file uploads using servlets
- Forward to JSPs using RequestDispatcher
- Use the session management API

Developing with JavaServer Pages Technology

- Evaluate the role of JSP technology as a presentation mechanism
- Author JSP pages
- Process data received from servlets in a JSP page
- Describe the use of tag libraries

EJB Component Model

- Describe the role of EJB components in a Java EE application
- Describe the benefits of EJB components
- Describe the operational characteristics of a stateless, stateful, and singleton session beans
- Create session beans
- Create session bean clients

The Java Persistence API

- Describe the role of the Java Persistence API (JPA) in a Java EE application
- Explain the basics of object-relational mapping
- Describe the elements and environment of an entity component
- Describe the life cycle and operational characteristics of entity components

Implementing a Transaction Policy

- Describe transaction semantics
- Compare programmatic and declarative transaction scoping
- Use JTA to scope transactions programmatically
- Implement a container-managed transaction policy
- Support optimistic locking with the versioning of entity components
- Support pessimistic locking using EntityManager APIs
- Describe the effect of exceptions on transaction state

Web Service and Integration Technology Survey

- Describe the purpose of integration technologies
- Define the integration layer in a multilayered application architecture
- List various Java EE integration technologies
- Describe the benefit of Web Services over other integration technologies

Implementing a Security Policy

- Leverage container-managed security
- Define user roles and responsibilities
- Create a role-based security policy
- Using Declarative Security
- Configure authentication in the web tier