

Objetivo

O curso visa explorar a geração de conteúdo dinâmico para páginas WEB através de recursos da linguagem Java que são a base da plataforma Java Enterprise Edition (Java EE), e mostrar os frameworks mais conhecidos e requisitados no mercado para aplicar o padrão arquitetural MVC (Model View Controller).

Pré-requisitos

Curso "Programação Java" ou conhecimento equivalente de Java (Anotação, Enums, Genéricos). Escrever aplicativos em Java, demonstrando significativa capacidade de programação. Integrar código Java existente (por exemplo, reutilizar classes existentes criadas por outros membros da equipe). Usar um navegador da Web, como o Firefox, HTML e XML básicos. Sólidos conhecimentos de Orientação a Objetos.

Público Alvo

Desenvolvedores da linguagem de programação Java, Analistas de Sistemas e estudantes de informática que desejam se especializar em Java no desenvolvimento para web.

Material Didático



Livro:

Struts 2 : Projeto e Programação

Editora: Ciência Moderna

Autor: Antunes Budi Kurniawan

Número de páginas: 600

Tudo Sobre a JSP - com o NetBeans em Aplicações Distribuídas

Editora: Editora Visual Books

Autor: Fernando Anselmo

Número de páginas: 222

Duração
52 horas/aula

Diferenciais X25

- Instrutores altamente qualificados
- Livros como Material Didático
- Coffee-break
- Estacionamento coberto gratuito
- Computadores de última geração
- Salas com projetores multimídia
- Somente 01 aluno por computador
- Certificado diferenciado pelo aproveitamento do aluno
- Parcerias internacionais
- Treinamentos in-company
- Treinamentos revisados periodicamente

Conteúdo Programático

Conceitos de Programação Java para WEB

- Nomenclatura de pacotes, métodos, variáveis para classes Java
- Nomenclatura de arquivos e diretórios
- Apresentação do projeto que será desenvolvido
- Texto descritivo do problema apresentado pelo cliente
- Texto descritivo com as funcionalidades do projeto
- Wireframe das telas do projeto
- Apresentação do diagrama de classe de domínio do projeto
- Ambiente de desenvolvimento e estrutura do laboratório e todos os recursos disponíveis
- **Exercício: Codificar as classes do diagrama UML**

Conceitos de Banco de Dados

- Introdução ao MySQL
- Principais comandos utilizados
 - SELECT
 - INSERT
 - DELETE
 - UPDATE
- Modelagem básica do banco de dados com base no diagrama de classe, apresentado no exercício anterior
- Inserção de registros no banco de dados
- Integração do Java com o Banco de dados
 - Download do Driver JDBC do MySQL
 - JDBC e suas principais classes
 - DriverManager
 - Driver
 - Connection
 - PreparedStatement
 - Statement
 - ResultSet
- Aplicação de um CRUD (Create, Report, Update and Delete) em uma das entidades do diagrama de classes
- **Exercício: CRUD para as demais entidades**

Web Server e Tomcat, JSP, Servlet e HTML

- Sobre o Tomcat
 - Diretório onde está instalado o tomcat
 - Principal variável de ambiente
 - Problemas e soluções
- Estrutura de um projeto WEB
- **Exercício:**
 - **Criar a estrutura de diretórios para um projeto WEB**
 - **Criar um projeto WEB**

HTML

- Principais tags do HTML

- html
- head
- body
- h1, h2, h3, h4, h5 e h6
- p
- br
- form
- label
- input

- **Exercício: Construir os formulários de entrada de dados do projeto modelo**

JSP

- Delimitadores
- Diretivas
- Objetos implícitos
 - Exemplo
 - Projeto modelo
- **Exercício: Construir demais funcionalidades do projeto**

Servlets

- Classe HttpServlet e seus principais métodos
 - doGet e doPost
 - Como criar um Servlet
 - Vantagens e desvantagens de utilizar o Servlet
 - Como processar um formulário com o Servlet
- **Filtros**
 - Qual o papel de um filtro e quando o Tomcat executa um filtro
 - Como criar um filtro
 - Exemplo de utilização do filtro
 - Como criar um filtro para verificar autenticação do usuário

Modelo MVC

- Conceito do MVC e sua filosofia.
- Separação das camadas do projeto.
- Exemplo
 - Separando as camadas do projeto modelo

Hibernate

- Como funciona o Hibernate
- Vantagens em adicionar o Hibernate ao projeto
- Mapeamento de Objetos
 - Mapeamento utilizando XML
 - Mapeamento utilizando Anotações
 - Principais anotações
 - @Entity
 - @Table
 - @Id
 - @GeneratedValue
 - @Column
 - @Temporal
 - @Lob
 - @JoinColumn
 - @OneToOne
 - @OneToMany
 - @ManyToMany
 - @ManyToOne
- Utilitário Hibernate
- XML ou Anotações, qual o melhor formato?
- JPA
 - Outras implementações do JPA
 - Vantagens e desvantagens do JPA
 - Principais diferenças entre o Hibernate (Tradicional) e o Hibernate (JPA)
 - Objetos de conexão
 - Configuração
 - Relacionamentos
 - Principais relacionamentos;
 - Identificar os relacionamentos que o projeto possui;
 - Anotações
 - @OneToMany
 - @ManyToOne
 - @ManyToMany
 - @JoinColumn
 - relatedBy
 - @JoinTable
 - Como mapear objetos com herança
 - Tabela única
 - Junção
 - Tabela por classe

- Manipulando os dados de um objeto persistente
 - merge
 - persist
 - remove
 - createQuery
 - createNativeQuery
 - HQL
 - Manipulando dados com o HQL
 - Comparando HQL com o SQL ANSI
- **Exercício:**
 - **Mapear os objetos de dominio do projeto modelo**
 - **Criar métodos para recuperar, listar, remover e inserir objetos**

Struts 2

- O que é Struts 2
- Vantagens e desvantagens do Struts 2
- Adicionar o Struts 2 no projeto.
- Como o Struts 2 funciona
- Arquivo de configuração struts.xml
- Estrutura de um Action
 - ActionSupport
 - Prepareable
- Struts 2 e suas tags
 - s:form
 - s:head
 - s:include
 - s:textfield
 - s:textarea
 - s:submit
 - s:action
- **Exercício: Converter os formulários da aplicação, utilizando as tags do Struts**
- Validação de dados do formulário
- **Exercício: Validar os formulários do projeto modelo, com base no requisitos do projeto**
- Interceptor
 - Como funciona o interceptor
 - Como verificar a sessão do usuário com interceptor
 - Configurando o LoginInterceptor
- **Exercício: Remover o filtro do sistema e configurar o LoginInterceptor**
- Como tabular dados utilizando o displaytag
- **Exercício: Adicionar filtros e ordenação aos formulários**