

#### Objetivo

Com enfoque totalmente prático permite que programadores Java possam ampliar seus conhecimentos no mundo Web na criação de sistemas profissionais. Utilizar os modelos de programação com Servlets, JSP, padrões de projeto (MVC, DAO, Front Controller, Command, Transfer Object e Intercepting Filter), servidor Web Apache Tomcat, Gerenciador de Banco de dados MySQL e IDE Eclipse. Conhecer o funcionamento de aplicações Web com os frameworks Hibernate 3 e Struts 2.

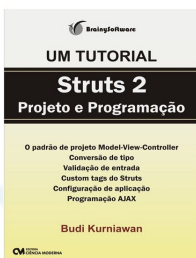
#### Pré-requisitos

Curso "Programação Java" ou conhecimento equivalente de Java (Envolvendo Anotação, Enums, Genéricos). Escrever aplicativos em Java, demonstrando capacidade de programação. Integrar código Java existente (por exemplo, reutilização classes). Utilização de navegador WEB e conhecimentos básicos de HTML e XML. Sólidos conhecimentos de Orientação a Objetos.

#### Público Alvo

Pessoas interessadas em iniciar a carreira de desenvolvimento de sistemas, ou para as que desejam se especializar no desenvolvimento WEB.

#### Material Didático



#### Livros:

**Struts 2 : Projeto e Programação**

Editora: Ciência Moderna

Autor: Antunes Budi Kurniawan

Número de páginas: 600

**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**

### 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**

### 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

### JSP

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

#### 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 e Delete) em uma das entidades do diagrama de classes
- **Exercício: CRUD para as demais entidades**

#### Hibernate

- Como funciona o Hibernate
- Vantagens em adicionar o Hibernate ao projeto
- Mapeamento de Objetos
  - Mapeamento utilizando XML
  - Mapeamento utilizando Anotações
    - @Entity
    - @Table
    - @Id
    - @GeneratedValue
    - @Column
    - @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 domínio do projeto modelo**
  - **Criar métodos para recuperar, listar, remover e inserir objetos**

#### Modelo MVC

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

#### 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 uma Action
  - ActionSupport
  - Prepareble

- 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**