



www.softblue.com.br

Java para Web: Java Server Faces

Exercícios Propostos

Modelo MVC e Estrutura das Aplicações

Carlos Tosin

Histórico

Data	Observação
29/12/2011	Criação do documento
20/03/2012	Adaptação devido ao uso de JPA

Sobre a Softblue

Fundada em 2003 na cidade de Curitiba-PR por André Milani e Carlos Tosin, inicialmente para dedicar-se à construção de soluções para web, a **Softblue** rapidamente expandiu seus negócios para a realização de cursos e treinamentos, devido ao fato de seus sócios serem altamente especializados em determinadas áreas de TI, certificados nas tecnologias em que atuam e autores de livros de informática sobre os mesmos temas.

Atualmente, a **Softblue** disponibiliza cursos e treinamentos on-line, permitindo que pessoas do Brasil inteiro possam realizá-los, estudando e aprendendo novas tecnologias, aprimorando seus conhecimentos para o mercado de trabalho, de acordo com sua disponibilidade de horários.

Exercícios Propostos

Exercício 1

Implemente um sistema de gerenciamento de impostos de veículos utilizando os conceitos de MVC.

A aplicação deve ter duas funcionalidades: cadastro de imposto de veículo e cadastro de pagamento de imposto.

No cadastro de imposto de veículo, o usuário deve procurar um veículo pela placa. O sistema deve então mostrar a marca e o modelo do veículo e permitir que um imposto seja cadastrado para ele. O imposto para o veículo é composto de um valor, número de parcelas para o pagamento e data do vencimento de cada parcela. Quando o imposto for criado, o sistema deve gerar automaticamente os registros de cada uma das parcelas, com valores e datas de vencimento corretos.

Veja um exemplo de como poderia ser a interface desta funcionalidade:

A interface web para o cadastro de imposto de veículo. No topo, há um link "Página Inicial" em azul. Abaixo, o título "Cadastrar Imposto" em negrito. O formulário contém: um campo "Placa:" com o valor "ABD4872" e botões "Pesquisar" e "Limpar"; uma linha de exibição com "Marca: Renault | Modelo: Clio | Ano: 2010"; um campo "Valor do imposto:" com o valor "560,00"; um campo "Número de parcelas:" com o valor "4" e uma seta para baixo; um campo "Dia de vencimento:" com o valor "10" e uma seta para baixo; e um botão "Cadastrar" no final.

No cadastro de pagamento de imposto, o usuário deve procurar um proprietário pelo seu CPF. Depois disso, o nome desse proprietário é exibido juntamente com a lista de veículos que ele possui (podem existir vários veículos). O usuário deve então escolher um dos veículos cuja parcela do imposto será pago e o sistema irá exibir a lista de parcelas referentes ao imposto daquele veículo. Cada parcela tem um número, um valor, uma data de vencimento e uma data de pagamento. Parcelas sem data de pagamento são consideradas pelo sistema como parcelas ainda em aberto. Para dar baixa no valor da parcela, basta que o usuário escolha a opção de pagar a parcela, e o sistema atribuirá a ela a data de pagamento, considerando a mesma como paga.

Veja um exemplo de como poderia ser a interface desta funcionalidade:

[Página Inicial](#)

Cadastrar Pagamento

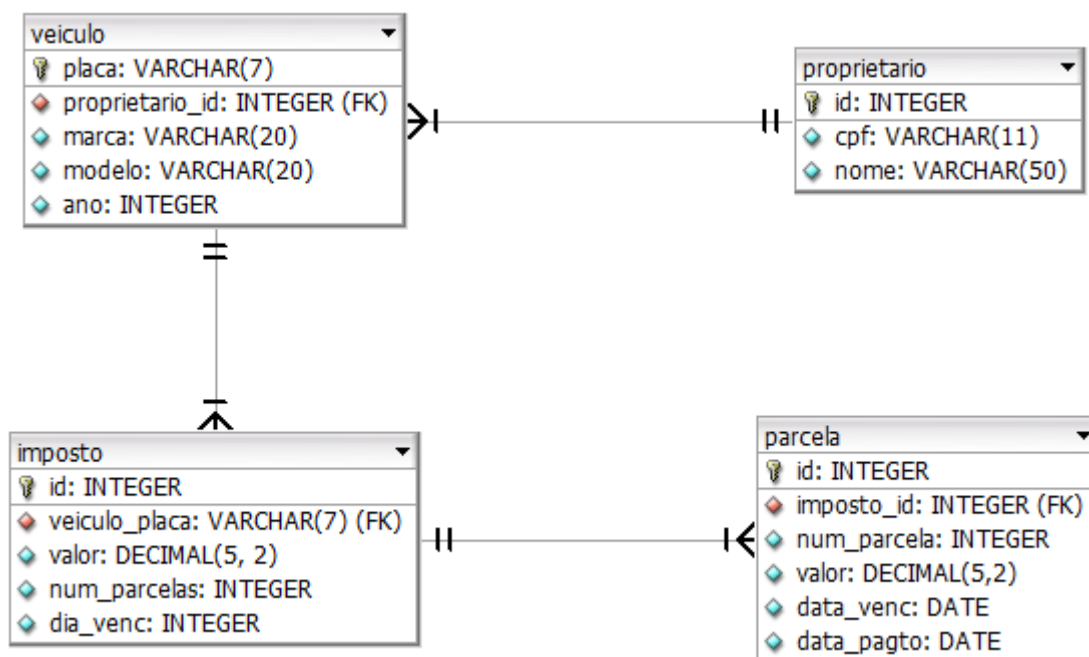
CPF:

Nome: José Oliveira

Veículo:

Núm.	Valor	Data Venc.	Data Pagto.
1	R\$ 140,00	10/02/2012	06/01/2012
2	R\$ 140,00	10/03/2012	<input type="button" value="Pagar"/>
3	R\$ 140,00	10/04/2012	<input type="button" value="Pagar"/>
4	R\$ 140,00	10/05/2012	<input type="button" value="Pagar"/>

Para auxiliar no desenvolvimento, você pode utilizar o modelo de dados proposto abaixo:



Preencha manualmente alguns dados nas tabelas *veiculo* e *proprietario*, já que o sistema não tem funcionalidades para cadastro desses dados. No projeto que apresenta a resolução deste exercício existe um script para inserir dados de teste. Com

relação à criação das tabelas, a própria JPA pode fazer isso para você (você precisa criar e configurar as entidades de forma correta).

Lembrando que um sistema desse tipo pode ser feito de várias formas. A resolução deste exercício mostra uma forma de fazer, que não é a única. O importante é seguir a filosofia do MVC e implementar um sistema que esteja de acordo com a especificação.