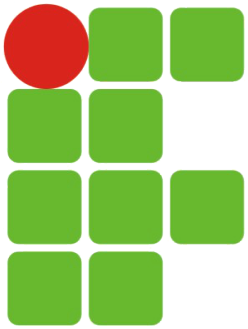


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Primeira Aplicação

Prof. Fellipe Aleixo (*fellipe.aleixo@ifrn.edu.br*)

Estrutura do Desenvolvimento

- Trata-se de uma aplicação exemplo:
  - "Armazém TADS" – sistema de compras
  - "*Depot Application*" (depósito ou armazém)
- Ilustra o desenvolvimento iterativo incremental – segundo a proposta ágil
  - Foco em: (i) desenvolvimento de funcionalidades + (ii) abraçar as mudanças + (iii) testes

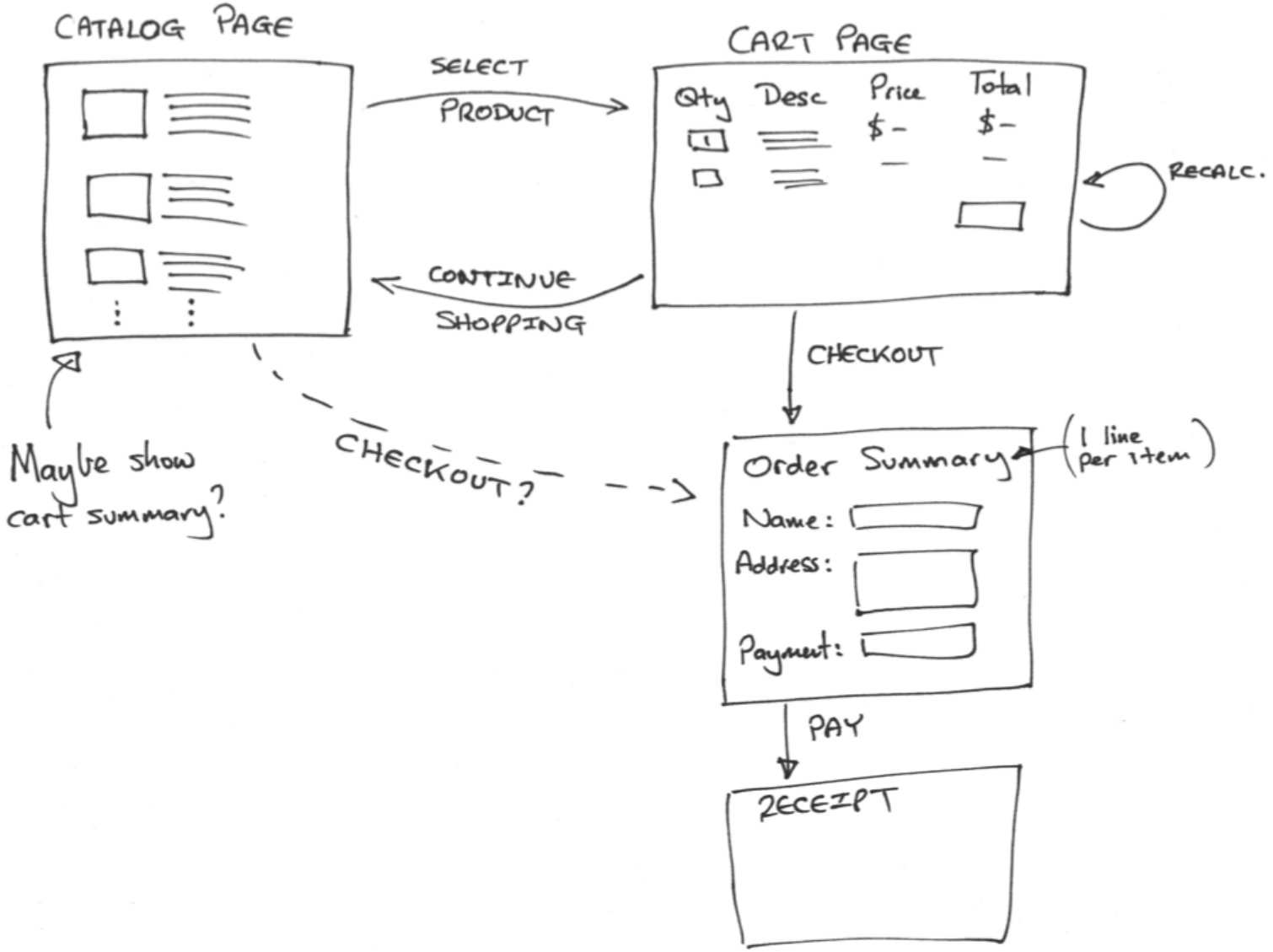
Aplicação: Armazém TADS

- **Atores**: (i) Comprador e (ii) Vendedor
- **Casos de uso**
  - Comprador: (i) navega pelos produtos; (ii) seleciona alguns para comprar e (iii) fornece informações para criar um pedido
  - Vendedor: (i) mantém a lista de produtos a venda; (ii) visualiza os pedidos aguardando entrega e (iii) marca um pedido como enviado (despachado)

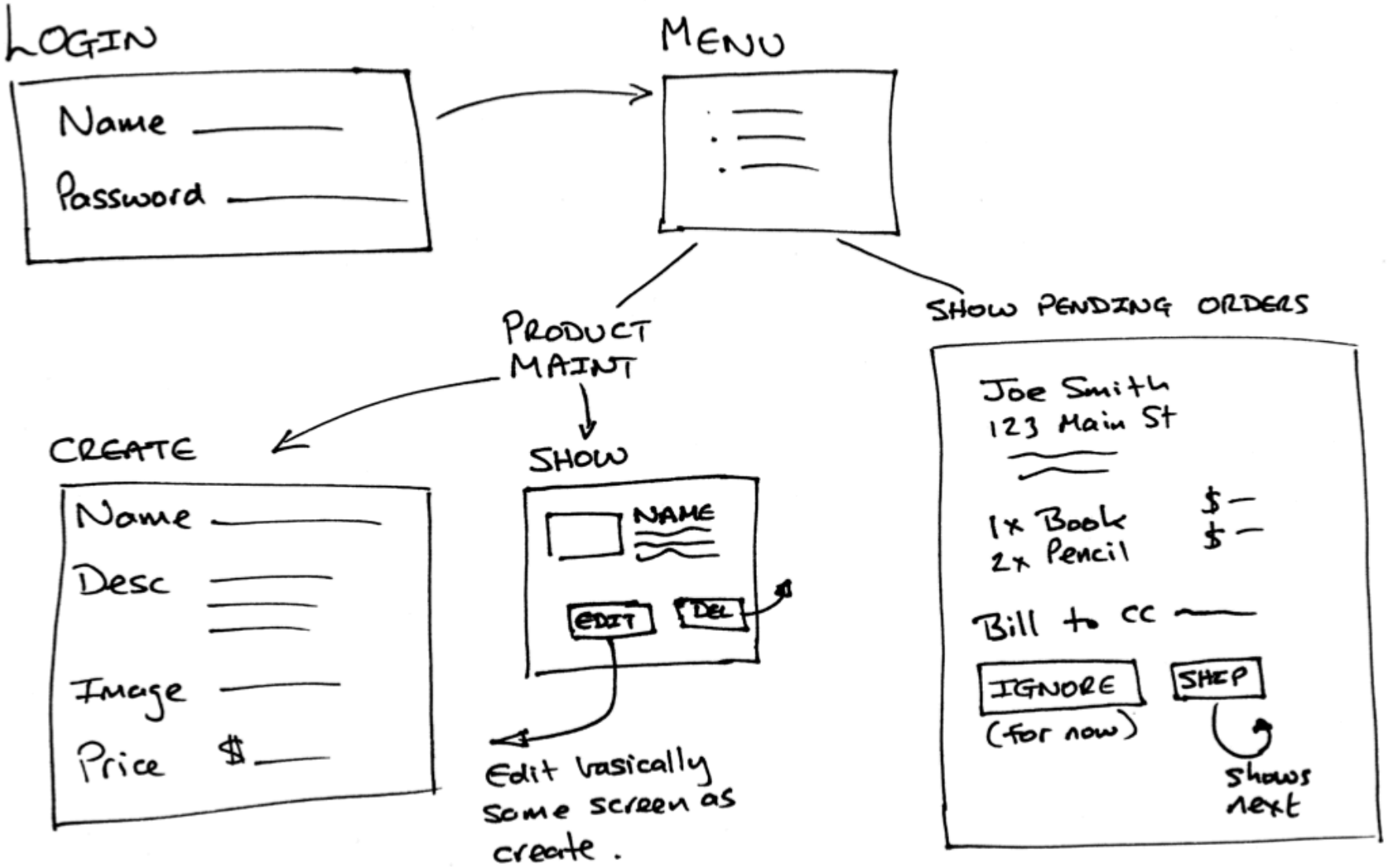
Desenvolvimento Ágil

- Essas informações são suficientes para se iniciar o desenvolvimento
- Algumas questões serão discutidas durante o desenvolvimento
  - Ex.: o que significa em detalhes manter os produtos e o que constitui um pedido pronto para entrega?
  - Estratégia: esclarecer através das interações com o cliente (*feedback*)

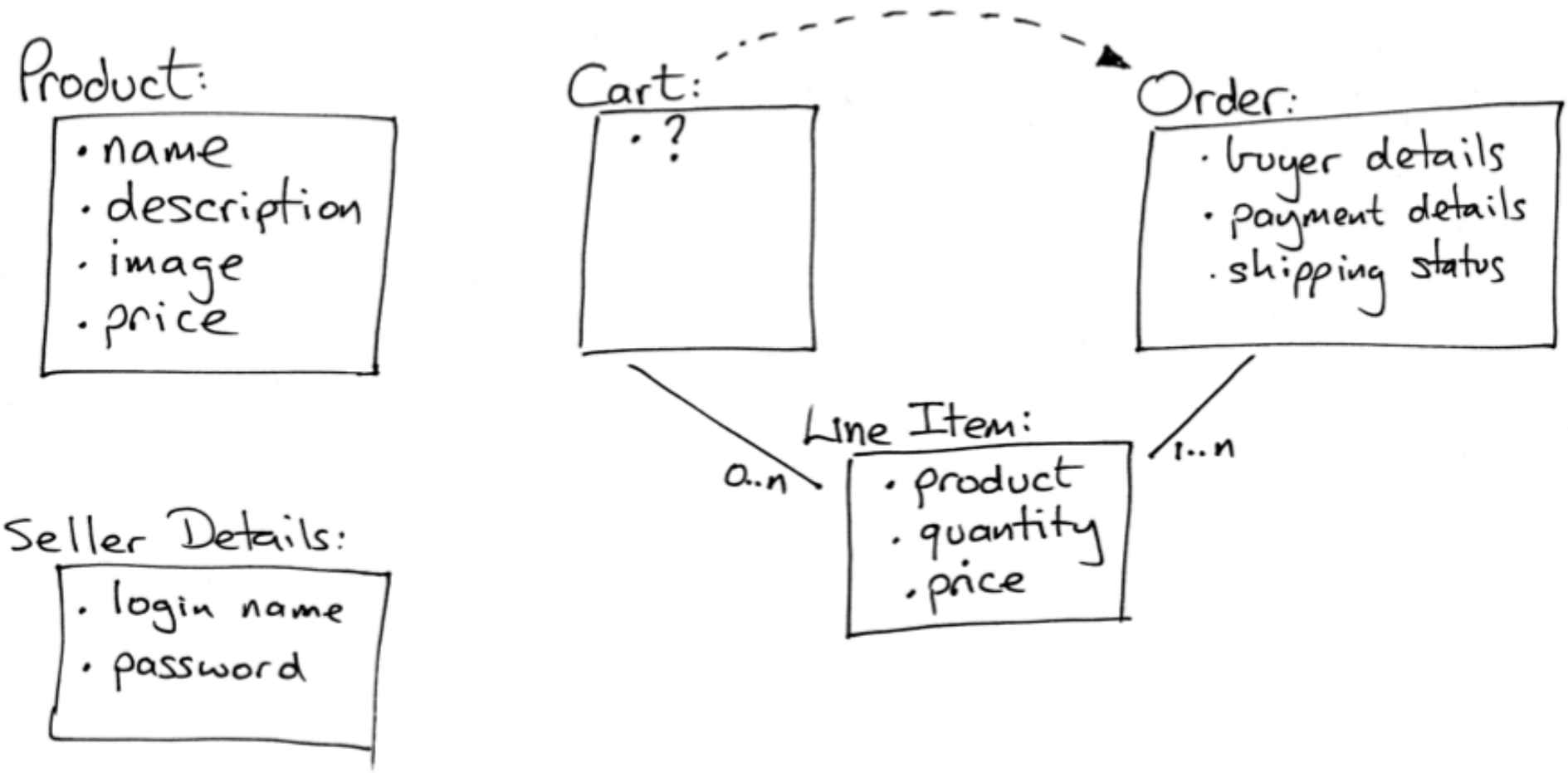
Fluxo de Páginas (Comprador)



Fluxo de Páginas (Vendedor)



Dados


Product:
• name
• description
• image
• price
Cart:
• ?
Order:
• buyer details
• payment details
• shipping status
Line Item:
• product
• quantity
• price
0..n
1..n
Seller Details:
• login name
• password

TAREFA “A”:

CRIANDO A APLICAÇÃO

Iteração A1:

MANTER PRODUTOS

Criação da Aplicação

- Criação da aplicação a partir da linha de comando (*rails new*)

```
work> rails new depot
```

  - Resultado:

```
work> cd depot
depot> ls -p
app/  config/    db/      Gemfile.lock  log/     Rakefile     test/  vendor/
bin/  config.ru  Gemfile  lib/          public/  README.rdoc  tmp/
```

Criação do Banco de Dados

- O livro utiliza o banco de dados *open source* **SQLite**, mais precisamente a versão 3
  - O referido banco será utilizado pela aplicação
  - Não é necessário criar as tabelas inicialmente

- Para a aplicação utilizar o banco de dados, as informações do mesmo são adicionadas em: "config/database.yml"

Geração do “Esqueleto”

- Passos
  - Criar uma tabela no banco de dados
  - Criar um classe de modelo correspondente
  - Criar as visões responsáveis pela interface com o usuário
  - Criar um controlador para “orquestrar” a aplicação

Geração do “Esqueleto”

- Criação do esqueleto para as ações de CRUD referente à classe Produto

```
depot> rails generate scaffold Product \
         title:string description:text image_url:string price:decimal
  invoke  active_record
  create    db/migrate/20121130000001_create_products.rb
  create    app/models/product.rb
  invoke    test_unit
  create      test/models/product_test.rb
  create      test/fixtures/products.yml
  invoke  resource_route
   route    resources :products
  invoke  jbuilder_scaffold_controller
  create    app/controllers/products_controller.rb
```

Geração do “Esqueleto”

```
invoke    erb
create      app/views/products
create      app/views/products/index.html.erb
create      app/views/products/edit.html.erb
create      app/views/products/show.html.erb
create      app/views/products/new.html.erb
create      app/views/products/_form.html.erb
invoke    test_unit
create      test/controllers/products_controller_test.rb
invoke    helper
create      app/helpers/products_helper.rb
invoke      test_unit
create        test/helpers/products_helper_test.rb
invoke    jbuilder
 exist      app/views/products
create      app/views/products/index.json.jbuilder
create      app/views/products/show.json.jbuilder
```

Geração do “Esqueleto”

```
invoke  assets
invoke    coffee
create      app/assets/javascripts/products.js.coffee
invoke    scss
create      app/assets/stylesheets/products.css.scss
invoke  scss
create    app/assets/stylesheets/scaffolds.css.scss
```

Geração do “Esqueleto”

- Dentre os arquivos gerados, está um arquivo de “migração” (*migration*)
  - Uma “migração” representa uma alteração que desejamos realizar nos dados
    - Alterações no esquema do banco
    - Alterações nos dados nas tabelas do banco
  - Permite se adaptar mais facilmente à mudanças
  - A aplicação da “migração” atualiza o banco de dados – também podem ser “desaplicadas”

Geração do “Esqueleto”

- Alterando a “migração” gerada por padrão

Download rails40/depot_a/db/migrate/20121130000001_create_products.rb

```
class CreateProducts < ActiveRecord::Migration
  def change
    create_table :products do |t|
      t.string :title
      t.text :description
      t.string :image_url
      t.decimal :price, precision: 8, scale: 2

      t.timestamps
    end
  end
end
```

Geração do “Esqueleto”

- Aplicando uma “migração”
  - Comando **rake**

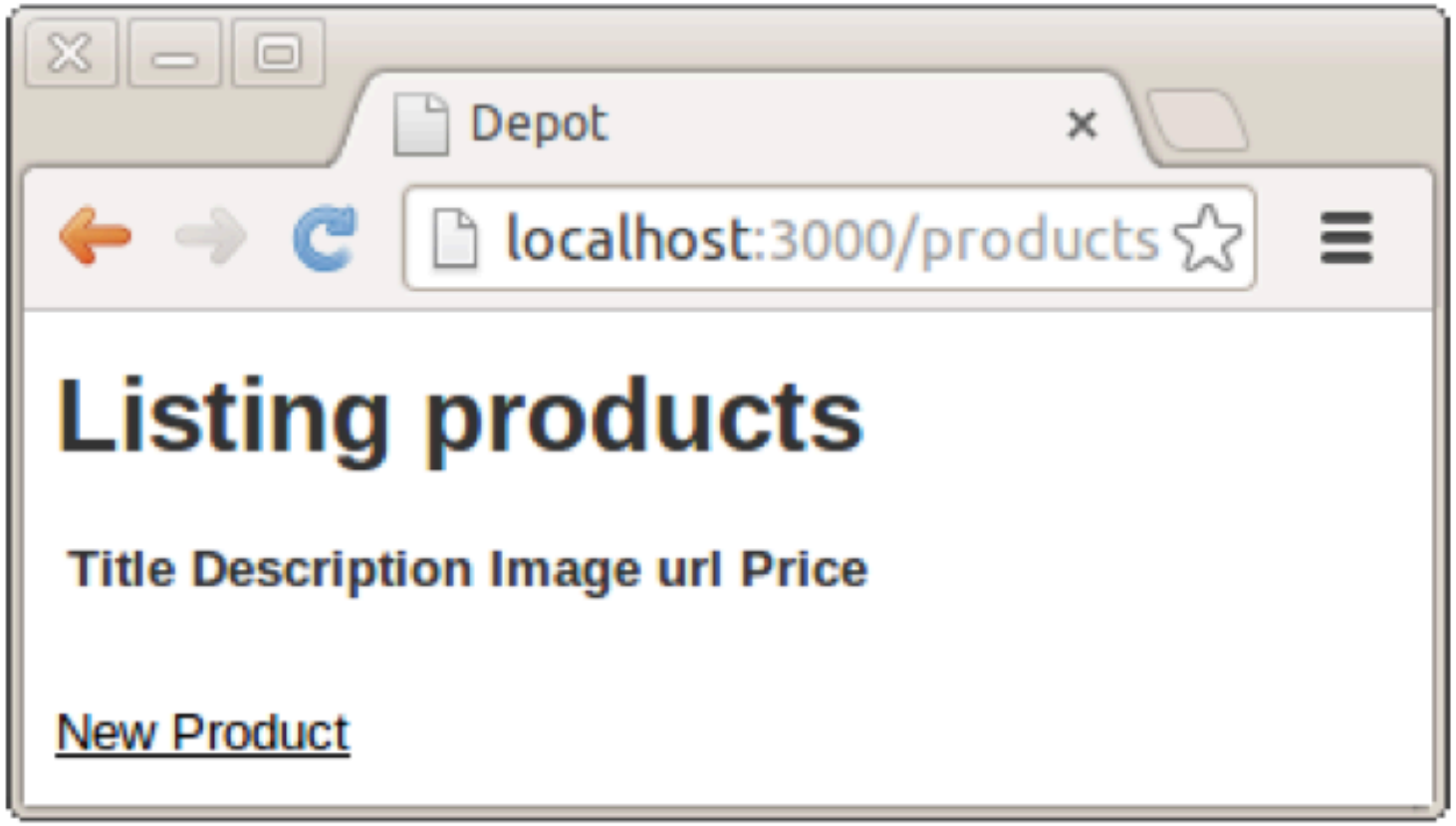
```
depot> rake db:migrate
==  CreateProducts: migrating ==============
-- create_table(:products)
   -> 0.0027s
==  CreateProducts: migrated (0.0023s) =====
```

Iniciando a Aplicação

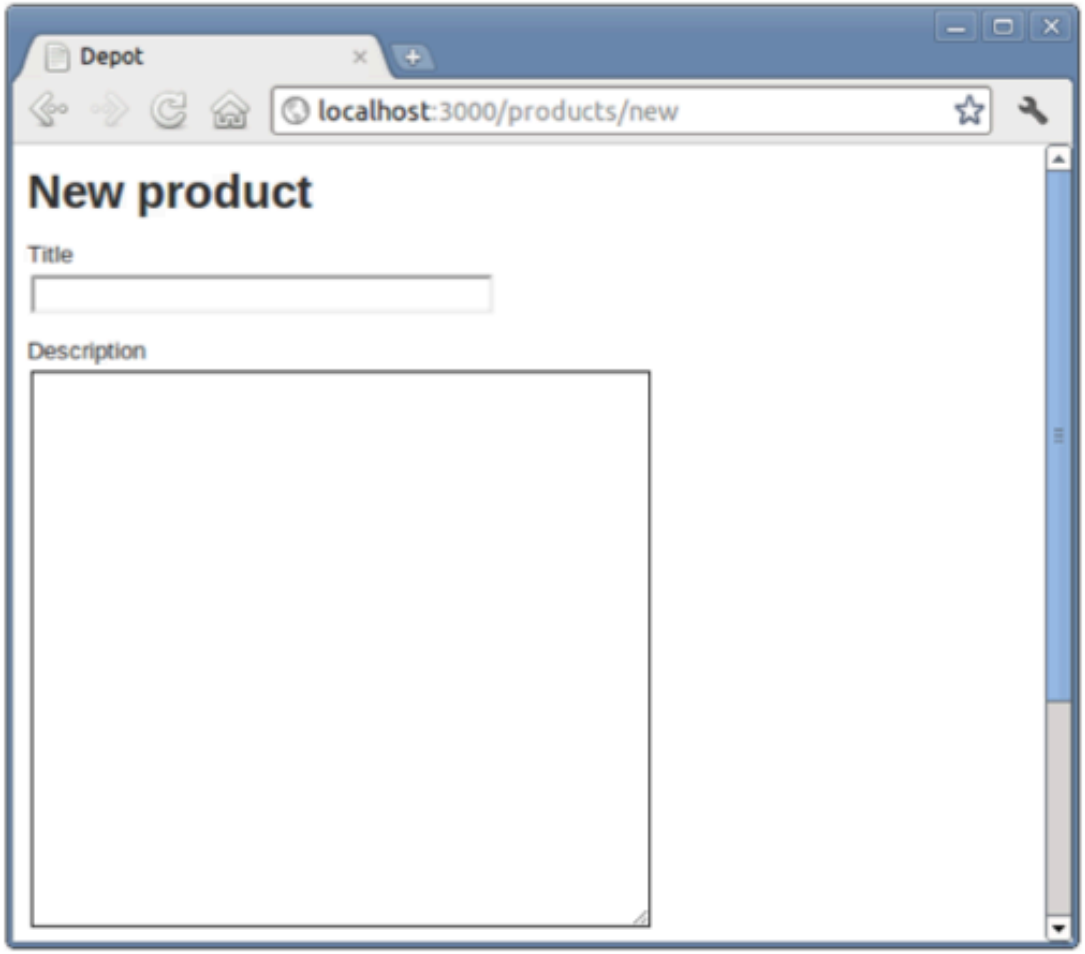
- Basta colocar o servidor para funcionar

```
depot> rails server
=> Booting WEBrick
=> Rails 4.0.0 application starting in development on http://0.0.0.0:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2013-04-18 17:45:38] INFO  WEBrick 1.3.1
[2013-04-18 17:45:38] INFO  ruby 2.0.0 (2013-02-24) [x86_64-linux]
[2013-04-18 17:45:43] INFO  WEBrick::HTTPServer#start: pid=24649 port=3000
```

Vendo a Lista de Produtos



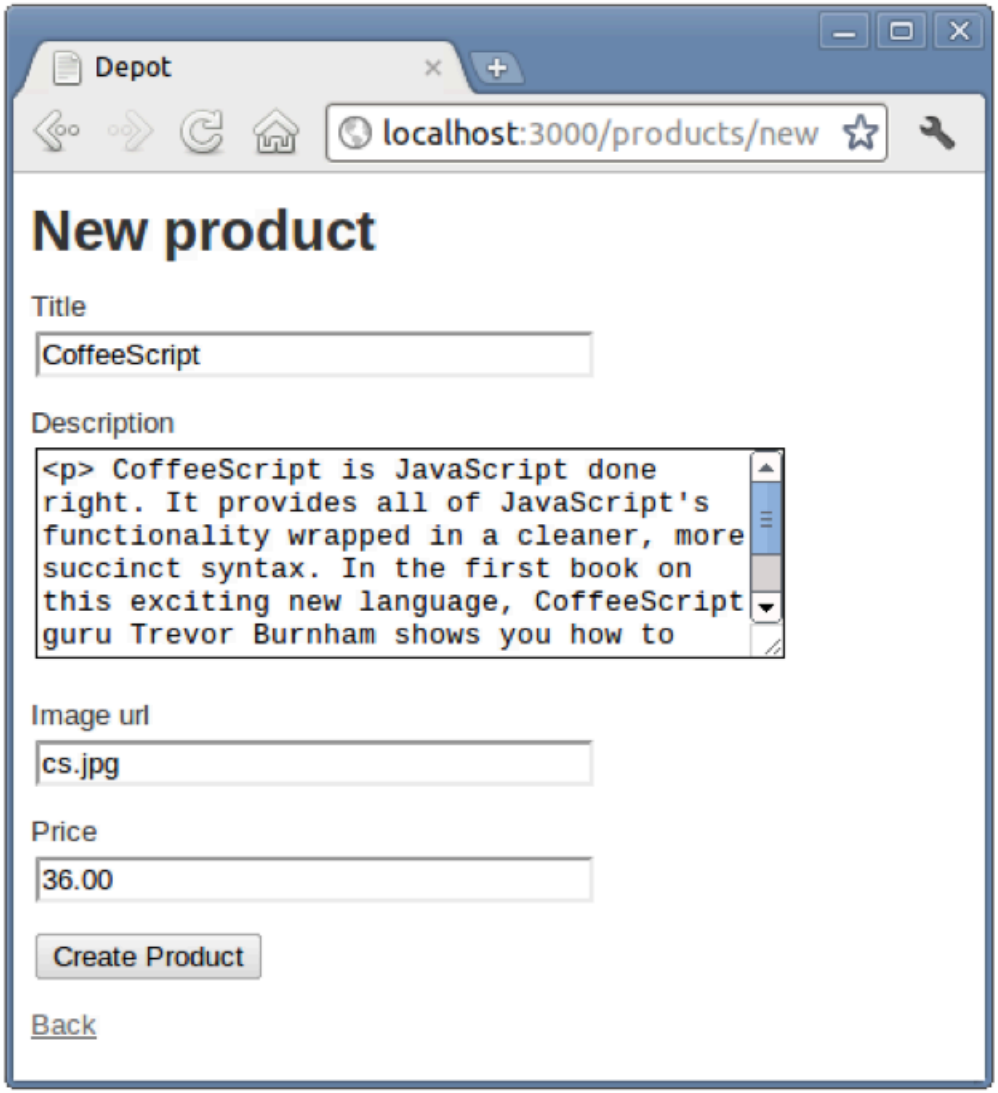
Inserindo um Novo Produto



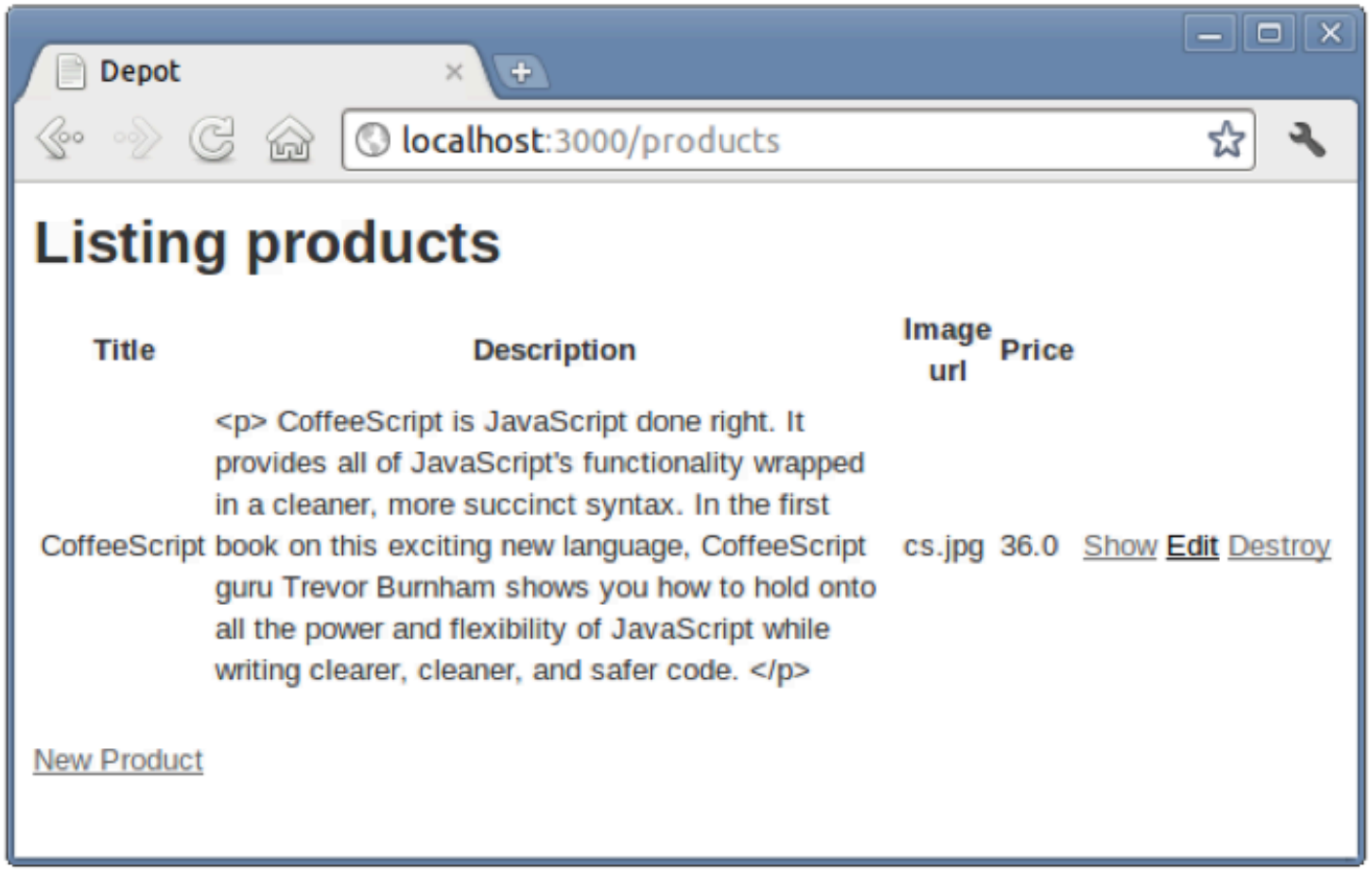
Download rails40/depot_a/app/views/products/_form.html.erb

```
<%= form_for(@product) do |f| %>
  <% if @product.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@product.errors.count, "error") %>
      prohibited this product from being saved:</h2>

      <ul>
      <% @product.errors.full_messages.each do |msg| %>
        <li><%= msg %></li>
      <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field">
    <%= f.label :title %><br>
    <%= f.text_field :title %>
  </div>
  <div class="field">
    <%= f.label :description %><br>
    <%= f.text_area :description, rows: 6 %>
  </div>
  <div class="field">
    <%= f.label :image_url %><br>
    <%= f.text_field :image_url %>
  </div>
  <div class="field">
    <%= f.label :price %><br>
    <%= f.text_field :price %>
  </div>
  <div class="actions">
    <%= f.submit %>
  </div>
<% end %>
```

Inserindo um Novo Produto



Listando os Produtos



Testando a Aplicação

- Utilização dos testes unitários
  - Aplicado às classes de modelo e controle
  - Comando: **rake test**
  - Resultado: “0 failures, 0 erros”