



Desenvolvimento Web Ágil para a plataforma Java

Antonio Francisco Prado
Bruno Lorenço Lopes
Delano Medeiros Beder
(DC-UFSCar)

Big Ideas

Grails in Action
Glen Smith and Peter Ledbrook

- ▶ *#1: Convention over Configuration*
- ▶ *#2: Agility - Coding at the speed of thought*
- ▶ *#3: Rocks Solid Foundations*
- ▶ *#4: Scaffolding & Templating*
- ▶ *#5: Java integration: No Code left behind*
- ▶ *#6: Incredible Wetwear: A vibrant community*
- ▶ *#7 Productivity is the new black*

Fundação Sólida



O que é Grails ?

(1)

- ▶ Desenvolvimento Web Ágil
 - ▶ *Getting things done*
 - ▶ *Having fun doing it*
 - ▶ *Padrão arquitetural: Modelo-Visão-Controlador (MVC)*
 - ▶ *Open source*
 - ▶ Executa sobre a máquina Virtual Java (JVM)
-

O que é Grails ?

(2)



- ▶ Implemente suas aplicações em Groovy
 - ▶ JSR-241
 - ▶ *Groovy is an agile, dynamic programming language for the Java Virtual Machine. Groovy includes features found in Python, Ruby, and Smalltalk, but uses syntax similar to the Java programming language.*
- ▶ Totalmente Orientada a Objetos
- ▶ Dinâmica
- ▶ Sintaxe Familiar (Desenvolvedores Java)
- ▶ Integração perfeita com Java

O que é Grails ?

(3)

- ▶ **Convention over configuration**
 - ▶ Padronização: comportamento 'default'
 - ▶ Tudo tem seu lugar
- ▶ **DRY (Don't Repeat Yourself)**

Vamos começar (por mão na massa)?

- ▶ **Pré-requisito**

Instale JDK 1.4 (or mais recente)

Atualize a variável JAVA_HOME (.bashrc)

- ▶ **Instalação**

`curl -s get.sdkman.io | bash`

`source "$HOME/.sdkman/bin/sdkman-init.sh"`

`sdk install grails`

- ▶ **Configure .bashrc**

`GRAILS_HOME=$HOME/.sdkman/candidates/grails/current`

`export GRAILS_HOME`

`PATH=$PATH:$GRAILS_HOME/bin`

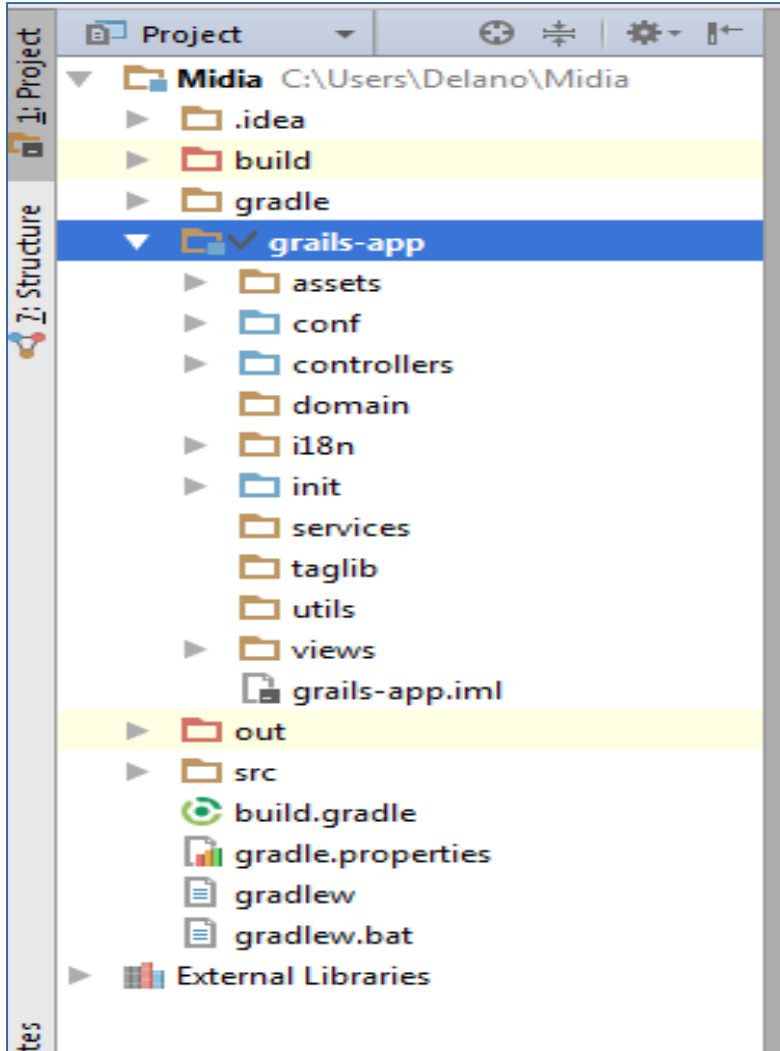
- ▶ **Execução**

`grails -version`

Aplicação Catálogo de Mídias

Demo

Hierarquia de Diretórios

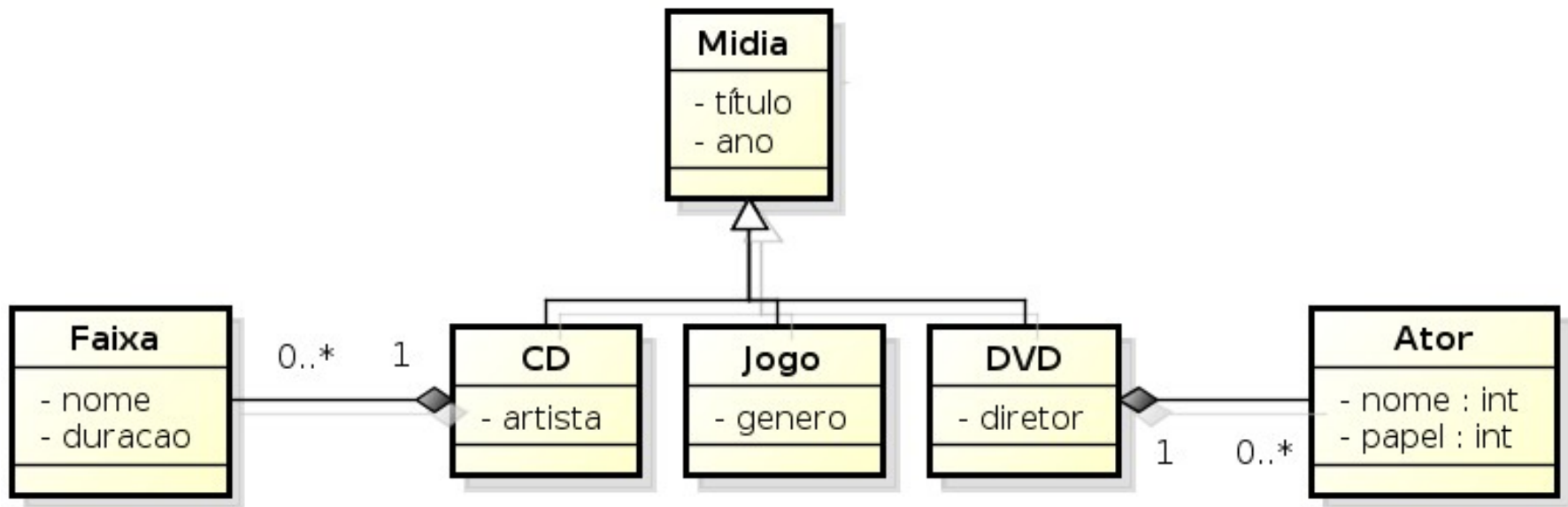


- ▶ Diretórios
 - ▶ Convenção
 - ▶ Nenhum arquivo xml
 - ▶ Cada 'artefato' tem seu lugar
 - ▶ MVC
 - ▶ Modelo
 - Leiautes
 - ▶ Controlador
 - ▶ Serviços
 - ▶ Internacionalization (i18n)

Scaffolding

Aplicação Catálogo de Mídias

Classes de Domínio (Modelo – MVC)



GORM (Grails ORM)

- ▶ ***Zero-configuration persistence management***
 - ▶ ***Dynamic domain class methods***
 - ▶ save
 - ▶ delete
 - ▶ get
 - ▶ list
 - ▶ find
 - ▶ findAll
 - ▶ count
 - ▶ etc.
-

Classes de Domínio – Midia

Midia (classe abstrata)

```
package br.ufscar.dc.dsw
abstract class Midia {

    static constraints = {
        titulo size: 1..30
        ano min: 1900
    }

    String titulo
    int ano

    String toString() {
        return titulo
    }
}
```

Classes de Domínio – Cd

Relacionamento de Herança

```
package br.ufscar.dc.dsw

class Cd extends Midia {

    static hasMany = [faixas: Faixa]

    static constraints = {
        artista size:1..30
    }

    String artista
}
```

Classes de Domínio – Dvd

Relacionamento de Herança

```
package br.ufscar.dc.dsw

class Dvd extends Midia {

    static hasMany = [atores: Ator]

    static constraints = {
        diretor size:1..30
    }

    String diretor
}
```

Classes de Domínio – Jogo

Relacionamento de Herança

```
package br.ufscar.dc.dsw

class Jogo extends Midia {

    static constraints = {
        genero size:1..30
    }

    String genero
}
```


Mapeamento de Herança

- Uma tabela para toda a hierarquia de classes
 - Mapeamento *default*
- Uma tabela para cada classe de domínio
 - Adicione o seguinte trecho na raiz da hierarquia

```
static mapping = {  
    tablePerHierarchy = false  
}
```

Classes de Domínio – Faixa

Relacionamento 1 para N (belongsTo/hasMany)

```
package br.ufscar.dc.dsw
class Faixa {
    static belongsTo = [cd: Cd]
    static constraints = {
        nome size:1..30
        duracao min:1
    }

    String nome
    int duracao
    String toString() {
        return "[" + nome + ", " + duracao + "]"
    }
}
```

Classes de Domínio – Ator

Relacionamento 1 para N (belongsTo/hasMany)

```
package br.ufscar.dc.dsw
class Ator {
    static belongsTo = [dvd: Dvd]
    static constraints = {
        nome size:1..30
        papel size:1..30
    }

    String nome
    String papel
    String toString() {
        return "[" + nome + ", " + papel + "]"
    }
}
```

Constraints (Restrições)

Quick Reference

Constraint	Description	Example
blank	Validates that a String value is not blank	<code>login(blank:false)</code>
creditCard	Validates that a String value is a valid credit card number	<code>cardNumber(creditCard: true)</code>
email	Validates that a String value is a valid email address.	<code>homeEmail(email: true)</code>
inList	Validates that a value is within a range or collection of constrained values.	<code>name(inList: ["Joe", "Fred", "Bob"])</code>
matches	Validates that a String value matches a given regular expression.	<code>login(matches: "[a-zA-Z]+")</code>
max	Validates that a value does not exceed the given maximum value.	<code>age(max: new Date()) price(max: 999F)</code>
maxSize	Validates that a value's size does not exceed the given maximum value.	<code>children(maxSize: 25)</code>
min	Validates that a value does not fall below the given minimum value.	<code>age(min: new Date()) price(min: 0F)</code>
minSize	Validates that a value's size does not fall below the given minimum value.	<code>children(minSize: 25)</code>
notEqual	Validates that that a property is not equal to the specified value	<code>login(notEqual: "Bob")</code>
nullable	Allows a property to be set to <code>null</code> - defaults to <code>false</code> .	<code>age(nullable: true)</code>
range	Uses a Groovy range to ensure that a property's value occurs within a specified range	<code>age(range: 18..65)</code>
scale	Set to the desired scale for floating point numbers (i.e., the number of digits to the right of the decimal point).	<code>salary(scale: 2)</code>
size	Uses a Groovy range to restrict the size of a collection or number or the length of a String.	<code>children(size: 5..15)</code>
unique	Constrains a property as unique at the database level	<code>login(unique: true)</code>
url	Validates that a String value is a valid URL.	<code>homePage(url: true)</code>
validator	Adds custom validation to a field.	See documentation

Constraints (exemplos)

Midia

```
static constraints = {  
    titulo size: 1..30  
    ano min: 1900  
}
```

Cd

```
static constraints = {  
    artista size:1..30  
}
```

Dvd

```
static constraints = {  
    diretor size:1..30  
}
```

Jogo

```
static constraints = {  
    genero size:1..30  
}
```

Faixa

```
static constraints = {  
    nome size:1..30  
    duracao min:1  
}
```

Ator

```
static constraints = {  
    nome size:1..30  
    papel size:1..30  
}
```

Scaffolding Dinâmico x Estático

Dinâmico:

- ▶ `grails create-controller br.ufscar.dc.dsw.Cd`
`class CdController { static scaffold = true }`
ou
`class CdController { static scaffold = Cd }`

Estático:

- ▶ `grails generate-all br.ufscar.dc.dsw.Cd`
 - ▶ Gera as visões e o controlador associados a classe de domínio

Scaffolding Estático (demais classes)

- ▶ `grails generate-all br.ufscar.dc.dsw.Faixa`
- ▶ `grails generate-all br.ufscar.dc.dsw.Dvd`
- ▶ `grails generate-all br.ufscar.dc.dsw.Ator`
- ▶ `grails generate-all br.ufscar.dc.dsw.Jogo`

Convention over Configuration (MVC)

Aplicação Catálogo de Mídias

URL Mapping – Convention over Configuration

http://localhost:8080/cd/show/1

The screenshot shows an IDE with a project named 'Midia' at the path '~/Desktop/Midia'. The project structure on the left includes:

- Midia
├── .gradle
├── .idea
├── build
├── gradle
└── grails-app
 ├── assets
 ├── conf
 └── controllers
 └── br.ufscar.dc.dsw
 ├── AtorController
 ├── CdController
 ├── DvdController
 ├── FaixaController
 ├── JogoController
 ├── MidiaController
 ├── midia
 ├── domain
 └── i18n

The main editor displays the code for `CdController` in the package `br.ufscar.dc.dsw`. The code is as follows:

```
Groovy SDK is not configured for module 'grails-app'

package br.ufscar.dc.dsw

import ...

@Transactional(readOnly = true)
class CdController {

    static allowedMethods = [save: "POST", update: "PUT", delete: "DELETE"]

    def index(Integer max) {
        params.max = Math.min(max ?: 10, 100)
        respond Cd.list(params), model: [cdCount: Cd.count()]
    }

    def show(Cd cd) {
        respond cd
    }

    def create() {
        respond new Cd(params)
    }
}
```

Two blue arrows originate from the URL `http://localhost:8080/cd/show/1` above. One arrow points to the `CdController` entry in the project's `controllers` directory. The other arrow points to the `show` method definition in the `CdController` class.

Run Grails: Midia

MVC – Convention over Configuration

Controlador CdController

```
@Transactional(readOnly = true)
class CdController {

    static allowedMethods = [save: "POST", update: "PUT", delete: "DELETE"]

    def index(Integer max) {
        params.max = Math.min(max ?: 10, 100)
        respond Cd.list(params), model:[cdCount: Cd.count()]
    }

    def show(Cd cd) {
        respond cd
    }
}
```

Visão cd/show.gsp

```
<div id="show-cd" class="content scaffold-show" role="main">
    <h1>message code="default.show.label" args="[entityName]" /></h1>
    <g:if test="${flash.message}">
        <div class="message" role="status">${flash.message}</div>
    </g:if>
    <f:display bean="cd" />
    <g:form resource="${this.cd}" method="DELETE">
        <fieldset class="buttons">
            <g:link class="edit" action="edit" resource="${this.cd}"><g:message code="edit.label" /></g:link>
            <input class="delete" type="submit" value="${message(code: 'delete.label')}" />
        </fieldset>
    </g:form>
</div>
```

Listagem de Mídias

Aplicação Catálogo de Mídias

Controlador MidiaController

- `grails create-controller br.ufscar.dc.dsw.Midia`

```
package br.ufscar.dc.dsw

import grails.transaction.Transactional

@Transactional(readOnly = true)
class MidiaController {

    def index(Integer max) {
        params.max = Math.min(max ?: 10, 100)
        render view: 'index', model:[midiaList: Midia.list(params),
                                     midiaCount: Midia.count()]
    }

    ... // demais linhas de código
}
```

Visão midia/index.gsp

```
<table>
  <thead>
    <tr>
      <g:sortableColumn property="nome"
        title="${message(code: 'midia.titulo.label', default: 'Título')}" />
      <th><g:message code="midia.ano.label" default="Ano" /></th>
    </tr>
  </thead>
  <tbody>
    <g:each in="${midiaList}" status="i" var="midia">
      <tr class="${(i % 2) == 0 ? 'even' : 'odd'}">
        <td><g:link action="show"
          id="${midia.id}">${fieldValue(bean: midia, field:
            "titulo")}</g:link></td>
        <td>${fieldValue(bean: midia, field: "ano")}</td>
      </tr>
    </g:each>
  </tbody>
</table>
```

Controlador MidiaController (2)

```
package br.ufscar.dc.dsw
import grails.transaction.Transactional

@Transactional(readOnly = true)
class MidiaController {

    ... // demais linhas de código

    def show() {
        Midia midia = Midia.get(params.id)
        if (midia instanceof Cd) {
            redirect (controller: 'cd', action: 'show', id: params.id)
        } else if (midia instanceof Dvd) {
            redirect (controller: 'dvd', action: 'show', id: params.id)
        } else {
            redirect (controller: 'jogo', action: 'show', id: params.id)
        }
    }
}
```

Internacionalization (I18n)

Aplicação Catálogo de Mídias

Internacionalization (I18n)

i18n/messages.properties (idioma padrão)

cd.label = Cd
cd.titulo.label = Title
cd.ano.label = Year
cd.artista.label = Artist

dvd.label = Dvd
dvd.titulo.label = Title
dvd.ano.label = Year
dvd.diretor.label = Director

jogo.label = Game
jogo.titulo.label = Title
jogo.ano.label = Year
jogo.genero.label = Genre

faixa.label = Track
faixa.nome.label = Name
faixa.duracao.label = Duration

ator.label = Actor
ator.nome.label = Name
ator.papel.label = Role

Internacionalization (I18n)

i18n/messages.properties (idioma pt-BR)

cd.label = Cd
cd.titulo.label = Título
cd.ano.label = Ano
cd.artista.label = Artista

dvd.label = Dvd
dvd.titulo.label = Título
dvd.ano.label = Ano
dvd.diretor.label = Diretor

jogo.label = Jogo
jogo.titulo.label = Título
jogo.ano.label = Ano
jogo.genero.label = Gênero

faixa.label = Faixa
faixa.nome.label = Nome
faixa.duracao.label = Duração

ator.label = Ator
ator.nome.label = Nome
ator.papel.label = Papel

Execução da aplicação

Aplicação Catálogo de Mídias

Populando dados (BootStrap.groovy)

```
import br.ufscar.dc.dsw.Ator
import br.ufscar.dc.dsw.Cd
import br.ufscar.dc.dsw.Dvd
import br.ufscar.dc.dsw.Faixa
import br.ufscar.dc.dsw.Jogo

class BootStrap {

    def init = { servletContext ->
        def cd = new Cd(titulo: 'The Wall', ano: 1976, artista: 'Pink Floyd').save(failOnError: true)
        new Faixa(nome: 'Another Brick in the Wall', duracao: 190, cd: cd).save(failOnError: true)
        def dvd = new Dvd(titulo: 'O Senhor dos Anéis', ano: 2010,
                           diretor: 'Peter Jackson').save(failOnError: true)
        new Ator(nome: 'Elijah Wood', papel: 'Frodo Bolseiro', dvd: dvd).save(failOnError: true)
        new Jogo(titulo: 'Fifa 2016', ano: 2016, genero: 'Esporte').save(failOnError: true)
    }
    def destroy = {
    }
}
```

Autenticação e Autorização

Aplicação Catálogo de Mídias

Plugin: Spring Security

Adicionar a seguinte linha em build.gradle

```
dependencies {  
    ...  
    compile "org.grails.plugins:spring-security-core:3.0.4"  
}
```

Executar o seguinte comando:

```
grails s2-quickstart br.ufscar.dc.dsw Usuario Papel
```

br.ufscar.dc.dsw ► nome do pacote

Usuario ► nome da classe que representa usuários (autenticação)

Papel ► nome da classe que representa papéis (autorização)

Adicionar a seguinte linha em application.groovy

```
grails.plugin.springsecurity.logout.postOnly = false
```

Usuario x Midia (Relacionamento 1 para N)

```
package br.ufscar.dc.dsw
class Usuario {
    static hasMany = [midias: Midia]
    ...
}
```

```
package br.ufscar.dc.dsw
class Midia {
    static belongsTo = [usuario: Usuario]
    ...
}
```

Personalização *Scaffolding*

Aplicação Catálogo de Mídias

Personalização *Scaffolding*

install-templates

Purpose

Copies the the templates used by Grails during code generation to your project directory

Examples

```
grails install-templates
```

Description

Usage:

```
grails install-templates
```

The `install-templates` command will copy the templates Grails uses for all code generation activities to the application's `src/main/templates/scaffolding` directory. The templates directories include:

```
src
  templates
    artifacts
    scaffolding
    war
```

The `artifacts` directory contains the templates used by the `create-*` commands. The `scaffolding` directory contains templates used by the `generate-*` commands. The `war` directory contains the `web.xml` template used to generate the deployment descriptor.

Personalização *Scaffolding*

Alterar o seguinte trecho nos templates create.gsp, edit.gsp, index.gsp, show.gsp

```
<div class="nav" role="navigation">
  <ul>
    <li><a class="home" href="\${createLink(uri: '/')}">
      <g:message code="default.home.label"/></a></li>
    <li><g:link class="list" action="index">
      <g:message code="default.list.label" args="[entityName]" /></g:link></li>
    <li><g:link controller="logout">Logout</g:link></li>
  </ul>
</div>
```

Alterar o seguinte trecho no template Controller.groovy

```
import grails.transaction.Transactional
import org.springframework.security.access.annotation.Secured

@Secured ('ROLE_USER')
@Transactional(readOnly = true)
```

Scaffolding ► Refletir personalizações

- ▶ `rails generate-all br.ufscar.dc.dsw.Cd --force`
- ▶ `rails generate-all br.ufscar.dc.dsw.Faixa --force`
- ▶ `rails generate-all br.ufscar.dc.dsw.Dvd --force`
- ▶ `rails generate-all br.ufscar.dc.dsw.Ator --force`
- ▶ `rails generate-all br.ufscar.dc.dsw.Jogo --force`
- ▶ `rails generate-all br.ufscar.dc.dsw.Usuario`

Como refletir as personalizações na entidade Midia?

Redirecionamento baseado em Papéis

Aplicação Catálogo de Mídias

Redirecionamento (baseado nos papéis)

► grails create-controller br.ufscar.dc.dsw.Main

```
package br.ufscar.dc.dsw
import org.springframework.security.access.annotation.Secured

@Secured (['ROLE_USER', 'ROLE_ADMIN'])
class MainController {

    def springSecurityService

    def index() {
        def usuario = springSecurityService.getCurrentUser()
        def authority = usuario.getAuthorities()[0].getAuthority()
        session.usuario = usuario
        if (authority.equals('ROLE_USER')) {
            redirect(controller: "midia")
        } else {
            redirect(controller: "usuario")
        }
    }
}
```

Redirecionamento (baseado nos papéis)

► URLMapping.groovy

```
package midia

class UrlMappings {

    static mappings = {
        "/$controller/$action?/$id?(.$format)?"{
            constraints {
                // apply constraints here
            }
        }

        "/"(controller:"main") // redirecionamento ao MainController
        "500"(view:'/error')
        "404"(view:'/notFound')
    }
}
```

Listagem de Mídias (Controlador) (2)

Permissão de acesso ao usuários com o papel ROLE_USER

```
package br.ufscar.dc.dsw

import grails.transaction.Transactional
import org.springframework.security.access.annotation.Secured

@Secured ('ROLE_USER')
@Transactional(readOnly = true)
class MidiaController {

    def index(Integer max) {
        params.max = Math.min(max ?: 10, 100)
        render view: 'index', model:[midiaList: Midia.list(params),
                                     midiaCount: Midia.count()]
    }

    ... // demais linhas de código
}
```

UsuarioController

Permissão de acesso ao usuários com o papel ROLE_ADMIN

```
@Secured ('ROLE_ADMIN')
@Transactional(readOnly = true)
class UsuarioController {
    ...

    @Transactional
    def save(Usuario usuario) {
        if (usuario == null) {
            transactionStatus.setRollbackOnly()
            notFound()
            return
        }
        if (usuario.hasErrors()) {
            transactionStatus.setRollbackOnly()
            respond usuario.errors, view:'create'
            return
        }
        usuario.save flush:true
        UsuarioPapel.create(usuario,Papel.findByAuthority("ROLE_USER"))
        request.withFormat {
            form multipartForm {
                flash.message = message(code: 'default.created.message', args: [message(code: 'usuario.label',
                                                                                      default: 'Usuario'), usuario.id])

                redirect usuario
            }
            '*' { respond usuario, [status: CREATED] }
        }
    }
}
```

Populando dados (Bootstrap.groovy) (2)

```
class Bootstrap {
    def init = { servletContext ->
        def adminPapel = Papel.findByAuthority("ROLE_ADMIN") ?:
            new Papel(authority: "ROLE_ADMIN").save()
        def admin = new Usuario(username: "admin", password: "admin",
            nome: "Administrador", enabled: true).save(failOnError: true)
        UsuarioPapel.create(admin, adminPapel)
        def userPapel = Papel.findByAuthority("ROLE_USER")?:
            new Papel(authority: "ROLE_USER").save()
        def ana = new Usuario(username: 'ana', password: 'ana',
            nome: "Ana", enabled: true).save(failOnError: true)
        UsuarioPapel.create(ana, userPapel)
        def cd = new Cd(titulo: 'The Wall', ano: 1976, artista: 'Pink Floyd',
            usuario: ana).save(failOnError: true)
        new Faixa(nome: 'Another Brick in the Wall', duracao: 190, cd: cd).save(failOnError: true)
        def dvd = new Dvd(titulo: 'O Senhor dos Anéis', ano: 2010, diretor: 'Peter Jackson',
            usuario: ana).save(failOnError: true)
        new Ator(nome: 'Elijah Wood', papel: 'Frodo Bolseiro', dvd: dvd).save(failOnError: true)
        new Jogo(titulo: 'Fifa 2016', ano: 2016, genero: 'Esporte',
            usuario: ana).save(failOnError: true)
    }
    def destroy = {
    }
}
```



Executando aplicação

Acesso pelo usuário com papel ROLE_USER

Grails	
Home	
Midia List	
Título	Ano
The Wall	1976
O Senhor dos Anéis	2010
Fifa 2016	2016
0	

Executando aplicação

Acesso pelo usuário com papel ROLE_ADMIN

Grails		
Home New Usuario Logout		
Usuario List		
Password	Username	Account Locked
\$2a\$10\$k/r3/TKtP2lnWe5Hk5cOa.sJ1pUOTZFvC4s4T2KdxfY.Zh7mXTQ5O	admin	False
\$2a\$10\$zX7.mwBaeASibb8w20hqPeoTOXotbJiXmZ2EZexqK9JmSorcmwkSC	ana	False
		

Acesso às mídias do usuário logado

Aplicação Catálogo de Mídias

Controladores

```
class CdController {

    static allowedMethods = [save: "POST", update: "PUT", delete: "DELETE"]

    def index(Integer max) {
        params.max = Math.min(max ?: 10, 100)
        def criteria = Cd.createCriteria()
        def cdList = criteria.list(params) {
            eq('usuario', session.usuario)
        }
        respond cdList, model: [cdCount: cdList.totalCount]
    }



    def show(Cd cd) {
        if (cd != null && cd.usuario != session.usuario) {
            flash.message = message(code: 'springSecurity.denied.message',
                args: [message(code: 'cd.label', default: 'CD'), cd.id])
            redirect action: "index"
        }
        respond cd
    }

    // demais ações do controlador
}
```

Exercício: aplicar a mesma abordagem para as entidades DVD, Jogo e Midia

Executando aplicação

Lista de Cds do usuário logado

 Home
  New Cd
 Logout

Cd List

Title	Year	Artist	User	Tracks
The Wall	1976	Pink Floyd	ana	<ul style="list-style-type: none"> [Another Brick in the Wall, 190]

0

Usuário logado

Visão cd/create.gsp

Substituir o trecho abaixo:

```
<g:form action="save">
  <fieldset class="form">
    <f:all bean="cd"/>
  </fieldset>
  <fieldset class="buttons">
    <g:submitButton name="create" class="save" value="{message(code: 'default.button.create.label', default: 'Create')}" />
  </fieldset>
</g:form>
```

por:

```
<g:form action="save">
  <fieldset class="form">
    <f:all bean="cd" except="usuario"/>

    <g:render template="fields" />
  </fieldset>
  <fieldset class="buttons">
    <g:submitButton name="create" class="save" value="{message(code: 'default.button.create.label', default: 'Create')}" />
  </fieldset>
</g:form>
```

Exercício: aplicar a mesma abordagem para as entidades DVD, Jogo

Template cd/_fields.gsp

Crie o template com o seguinte conteúdo:

```
<div class="fieldcontain ${hasErrors(bean: cd, field: 'usuario', 'error')} required">
  <label for="usuario">
    <g:message code="cd.usuario.label" default="Usuario"/>
    <span class="required-indicator">*</span>
  </label>
  <g:select id="usuario" name="usuario.id" from="${session.usuario}" optionKey="id" required=""
    value="${cd?.usuario?.id}" class="many-to-one"/>
</div>
```

Observação: atualizar arquivos relacionados a I18n

```
=====
i18n/messages.properties (idioma padrão)
=====

cd.usuario.label = User

=====
i18n/messages_pt-BR.properties (Português)
=====

cd.usuario.label = Usuário
```

Exercício: aplicar a mesma abordagem para as entidades DVD, Jogo

Visão cd/edit.gsp

Substituir o trecho abaixo:

```
<g:form resource="${this.cd}" method="PUT">
  <g:hiddenField name="version" value="${this.cd?.version}" />
  <fieldset class="form">
    <f:all bean="cd"/>
  </fieldset>
  <fieldset class="buttons">
    <input class="save" type="submit" value="${message(code: 'default.button.update.label', default: 'Update')}" />
  </fieldset>
</g:form>
```

por:

```
<g:form resource="${this.cd}" method="PUT">
  <g:hiddenField name="version" value="${this.cd?.version}" />
  <fieldset class="form">
    <f:all bean="cd" except="usuario"/>

    <g:render template="fields" />
  </fieldset>
  <fieldset class="buttons">
    <input class="save" type="submit" value="${message(code: 'default.button.update.label', default: 'Update')}" />
  </fieldset>
</g:form>
```

Exercício: aplicar a mesma abordagem para as entidades DVD, Jogo

Executando aplicação

Criação de novo Cd

Grails

Home Cd List Logout

Create Cd

Title *

Year * 0

Artist *

Tracks

[Add Track](#)

User * ana

ana

Usuário logado

Create



Desenvolvimento Web Ágil para a plataforma Java

Antonio Francisco Prado
Bruno Lorenço Lopes
Delano Medeiros Beder