

GraiLs

Java produtivo e divertido

Lucas Teixeira

lucastex.com.br
lucastex.com
@lucastex

gr8forums.org

GR8Forums

Register • FAQ • Search • Login

View unanswered posts | View active topics

It is currently May 20th, 2010, 3:15 am

Board index

All times are UTC [DST]

► GRAILS

| | Forum | Topics | Posts | Last post |
|---|--------------------------------|--------|-------|--|
|  | Starter questions | 78 | 211 | Yesterday, 11:46 pm juliannoye  |
|  | Advanced questions | 34 | 85 | May 14th, 2010, 5:00 pm luistrigueiros  |
|  | Deployment | 15 | 38 | May 14th, 2010, 3:52 am mr.paul.woods  |
|  | Plugins | 30 | 74 | May 14th, 2010, 3:40 am mr.paul.woods  |
|  | Tips and Snippets | 5 | 14 | September 22nd, 2009, 4:01 pm rhyolight  |
|  | Announcements and Books | 0 | 0 | No posts |
|  | GORM related | 15 | 46 | May 18th, 2010, 6:51 pm jackson0810  |

► GROOVY

| | Forum | Topics | Posts | Last post |
|---|-------|--------|-------|-------------------------|
|  | | | | May 15th, 2010, 6:49 am |

`github.com/lucastex`

`gsolr`

`grails-stella`

`grails-correios-br`

`grails-featurable`

`grails-file-uploader`

`grails-tinyurl`

- Groovy
- Grails
- Comunidade
- Demo

Groovy

0 que te motiva?

Pensar a maior
parte do tempo



Pensar

```
import java.util.List;
import java.util.ArrayList;
```

Seletor.java

```
class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList();
        for (int i = 0; i < palavras.size(); i++) {
            String pal = (String) palavras.get(i);
            if (pal.length() <= tamanho) {
                resultado.add(pal);
            }
        }
        return resultado;
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList();
        linguagens.add("Java");
        linguagens.add("Scala");
        linguagens.add("Groovy");
        linguagens.add("Ruby");

        Seletor seletor = new Seletor();
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5);
        System.out.println("Total "+selecionadas.size())

        for (int i = 0; i < selecionadas.size(); i++) {
            String sel = (String) selecionadas.get(i);
            System.out.println(sel);
        }
    }
}
```

```
import java.util.List;
import java.util.ArrayList;

class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList();
        for (int i = 0; i < palavras.size(); i++) {
            String pal = (String) palavras.get(i);
            if (pal.length() <= tamanho) {
                resultado.add(pal);
            }
        }
        return resultado;
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList();
        linguagens.add("Java");
        linguagens.add("Scala");
        linguagens.add("Groovy");
        linguagens.add("Ruby");

        Seletor seletor = new Seletor();
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5);
        System.out.println("Total "+selecionadas.size())

        for (int i = 0; i < selecionadas.size(); i++) {
            String sel = (String) selecionadas.get(i);
            System.out.println(sel);
        }
    }
}
```

regra dos 80 - 20

```

import java.util.List;
import java.util.ArrayList;

class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList();
        for (int i = 0; i < palavras.size(); i++) {
            String pal = (String) palavras.get(i);
            if (pal.length() <= tamanho) {
                resultado.add(pal);
            }
        }
        return resultado;
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList();
        linguagens.add("Java");
        linguagens.add("Scala");
        linguagens.add("Groovy");
        linguagens.add("Ruby");

        Seletor seletor = new Seletor();
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5);
        System.out.println("Total "+selecionadas.size())

        for (int i = 0; i < selecionadas.size(); i++) {
            String sel = (String) selecionadas.get(i);
            System.out.println(sel);
        }
    }
}

```

```

class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList()
        palavras.each { String pal ->
            if (pal.length() <= tamanho) {
                resultado.add(pal)
            }
        }
        return resultado
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList()
        linguagens.add("Java")
        linguagens.add("Scala")
        linguagens.add("Groovy")
        linguagens.add("Ruby")

        Seletor seletor = new Seletor()
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5)
        System.out.println("Total "+selecionadas.size())

        selecionadas.each { String sel ->
            System.out.println(sel)
        }
    }
}

```

```

class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList()
        palavras.each { String pal ->
            if (pal.length() <= tamanho) {
                resultado.add(pal)
            }
        }
        return resultado
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList()
        linguagens.add("Java")
        linguagens.add("Scala")
        linguagens.add("Groovy")
        linguagens.add("Ruby")

        Seletor seletor = new Seletor()
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5)
        System.out.println("Total "+selecionadas.size())

        selecionadas.each { String sel ->
            System.out.println(sel)
        }
    }
}

```



```
def selecionaMenoresQue(palavras, tamanho) {  
  def resultado = new ArrayList()  
  palavras.each { pal ->  
    if (pal.size() <= tamanho) {  
      resultado.add(pal)  
    }  
  }  
  return resultado  
}
```

```
def linguagens = new ArrayList()  
linguagens.add("Java")  
linguagens.add("Scala")  
linguagens.add("Groovy")  
linguagens.add("Ruby")
```

```
def selecionadas = selecionaMenoresQue(linguagens, 5)  
System.out.println("Total "+selecionadas.size())  
selecionadas.each { sel ->  
  System.out.println(sel)  
}
```

```
def selecionaMenoresQue(palavras, tamanho) {  
    def resultado = new ArrayList()  
    palavras.each { pal ->  
        if (pal.size() <= tamanho) {  
            resultado.add(pal)  
        }  
    }  
    return resultado  
}  
  
def linguagens = new ArrayList()  
linguagens.add("Java")  
linguagens.add("Scala")  
linguagens.add("Groovy")  
linguagens.add("Ruby")  
  
def selecionadas = selecionaMenoresQue(linguagens, 5)  
System.out.println("Total "+selecionadas.size())  
selecionadas.each { sel ->  
    System.out.println(sel)  
}
```

```
def selecionaMenoresQue(palavras, tamanho) {  
  def resultado = []  
  palavras.each { pal ->  
    if (pal.size() <= tamanho) {  
      resultado.add(pal)  
    }  
  }  
  resultado  
}  
  
def linguagens = ["Java", "Scala", "Groovy", "Ruby"]  
def selecionadas = selecionaMenoresQue(linguagens, 5)  
println("Total ${selecionadas.size()}")  
selecionadas.each { sel ->  
  println(sel)  
}
```

```
def selecionaMenoresQue(palavras, tamanho) {  
  def resultado = []  
  palavras.each { pal ->  
    if (pal.size() <= tamanho) {  
      resultado.add(pal)  
    }  
  }  
  resultado  
}  
  
def linguagens = ["Java", "Scala", "Groovy", "Ruby"]  
def selecionadas = selecionaMenoresQue(linguagens, 5)  
println("Total ${selecionadas.size()}")  
selecionadas.each { sel ->  
  println(sel)  
}
```

```
def selecionaMenoresQue(palavras, tamanho) {  
    palavras.findAll { it.size() <= tamanho }  
}  
  
def linguagens = ["Java", "Scala", "Groovy", "Ruby"]  
def selecionadas = selecionaMenoresQue(linguagens, 5)  
println "Total ${selecionadas.size()}"  
selecionadas.each { sel ->  
    println sel  
}
```

```
def selecionaMenoresQue(palavras, tamanho) {  
    palavras.findAll { it.size() <= tamanho }  
}  
  
def linguagens = ["Java", "Scala", "Groovy", "Ruby"]  
def selecionadas = selecionaMenoresQue(linguagens, 5)  
println "Total ${selecionadas.size()}"  
selecionadas.each { sel ->  
    println sel  
}
```

```
def linguagens = ["Java","Scala","Groovy","Ruby"]
def seleccionadas = linguagens.findAll { it.size() <= 5 }
println "Total ${seleccionadas.size()}"
seleccionadas.each { sel ->
    println sel
}
```

Seletor.groovy

```
def linguagens = ["Java","Scala","Groovy","Ruby"]
def selecionadas = linguagens.findAll { it.size() <= 5 }
println "Total ${selecionadas.size()}"
selecionadas.each { sel ->
    println sel
}
```


Seletor.groovy

```
def linguagens = ["Java","Scala","Groovy","Ruby"]
def selecionadas = linguagens.findAll { it.size() <= 5 }
println "Total ${selecionadas.size()}"
selecionadas.each { sel ->
    println sel
}
```

```
import java.util.List;
import java.util.ArrayList;
```

Seletor.java

```
class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList();
        for (int i = 0; i < palavras.size(); i++) {
            String pal = (String) palavras.get(i);
            if (pal.length() <= tamanho) {
                resultado.add(pal);
            }
        }
        return resultado;
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList();
        linguagens.add("Java");
        linguagens.add("Scala");
        linguagens.add("Groovy");
        linguagens.add("Ruby");

        Seletor seletor = new Seletor();
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5);
        System.out.println("Total "+selecionadas.size())

        for (int i = 0; i < selecionadas.size(); i++) {
            String sel = (String) selecionadas.get(i);
            System.out.println(sel);
        }
    }
}
```

Groovy é Java!

```
import java.util.List;
import java.util.ArrayList;
```

Seletor.java

```
class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList();
        for (int i = 0; i < palavras.size(); i++) {
            String pal = (String) palavras.get(i);
            if (pal.length() <= tamanho) {
                resultado.add(pal);
            }
        }
        return resultado;
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList();
        linguagens.add("Java");
        linguagens.add("Scala");
        linguagens.add("Groovy");
        linguagens.add("Ruby");

        Seletor seletor = new Seletor();
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5);
        System.out.println("Total "+selecionadas.size())

        for (int i = 0; i < selecionadas.size(); i++) {
            String sel = (String) selecionadas.get(i);
            System.out.println(sel);
        }
    }
}
```

Seletor.groovy

```
import java.util.List;
import java.util.ArrayList;

class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList();
        for (int i = 0; i < palavras.size(); i++) {
            String pal = (String) palavras.get(i);
            if (pal.length() <= tamanho) {
                resultado.add(pal);
            }
        }
        return resultado;
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList();
        linguagens.add("Java");
        linguagens.add("Scala");
        linguagens.add("Groovy");
        linguagens.add("Ruby");

        Seletor seletor = new Seletor();
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5);
        System.out.println("Total "+selecionadas.size())

        for (int i = 0; i < selecionadas.size(); i++) {
            String sel = (String) selecionadas.get(i);
            System.out.println(sel);
        }
    }
}
```

Groovy é melhor
que Java!

```
import java.util.List;
import java.util.ArrayList;
```

Seletor.java

```
class Seletor {

    private List selecionaMenoresQue(List palavras, int tamanho) {
        List resultado = new ArrayList();
        for (int i = 0; i < palavras.size(); i++) {
            String pal = (String) palavras.get(i);
            if (pal.length() <= tamanho) {
                resultado.add(pal);
            }
        }
        return resultado;
    }

    public static void main(String[] args) {
        List linguagens = new ArrayList();
        linguagens.add("Java");
        linguagens.add("Scala");
        linguagens.add("Groovy");
        linguagens.add("Ruby");

        Seletor seletor = new Seletor();
        List selecionadas = seletor.selecionaMenoresQue(linguagens, 5);
        System.out.println("Total "+selecionadas.size())

        for (int i = 0; i < selecionadas.size(); i++) {
            String sel = (String) selecionadas.get(i);
            System.out.println(sel);
        }
    }
}
```

Seletor.groovy

```
def linguagens = ["Java","Scala","Groovy","Ruby"]
def selecionadas = linguagens.findAll { it.size() <= 5 }
println "Total ${selecionadas.size()}"
selecionadas.each { sel ->
    println sel
}
```


“When you program in groovy, in many ways you're writing a special kind of java”

(Dierk König)

Grails

cenário comum

- Spring

- Spring
- Hibernate

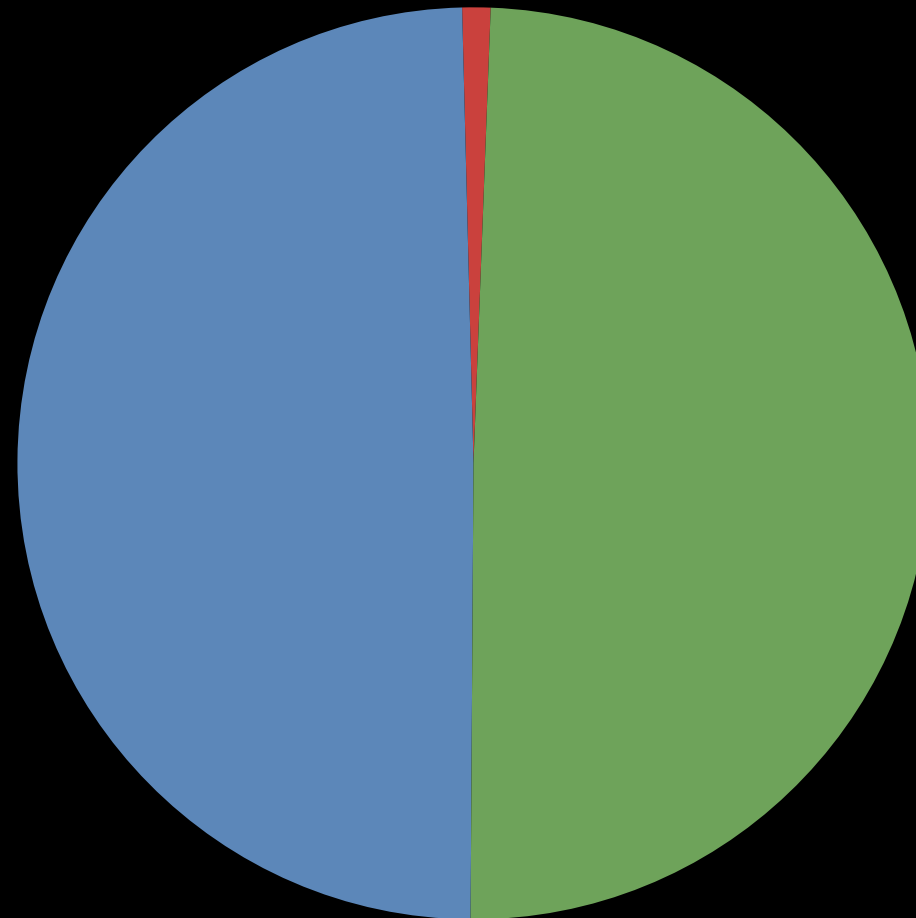
- Spring
- Hibernate
- JSF

- Spring
- Hibernate
- JSF
- DWR

- Spring
- Hibernate
- JSF
- DWR
- Velocity

- Spring
- Hibernate
- JSF
- DWR
- Velocity
- Tuckey





- Que você é incompetente
- Que você é mentiroso
- Que colocar todos os frameworks juntos toma tempo e pode gerar problemas

- Desenvolvimento rápido
 - Full stack
 - Fácil bootstrap
 - Convenções - CoC
 - Scaffolding
 - Plugins

- Desenvolvimento rápido
 - Full stack
 - Fácil bootstrap
 - Convenções - CoC
 - Scaffolding
 - Plugins

Full Stack

- mapeamento or - gorm
- log
- url rewrites
- view e decoration
- controladores
- controle transacional
- configuração por ambiente

Mapeamento OR – gorm

Mapeamento OR - gorm

```
Pessoa.findAllByNomeLikeAndIdadeGreaterThan("%lucas%", 25)
```


Mapeamento OR - gorm

```
Pessoa.findAllByNomeLikeAndIdadeGreaterThan("%lucas%", 25)
```

```
Pessoa.get(1234)
```

Mapeamento OR - gorm

```
Pessoa.findAllByNomeLikeAndIdadeGreaterThan("%lucas%", 25)
```

```
Pessoa.get(1234)
```

```
Date dateCreated
```

```
Date lastUpdated
```

Mapeamento OR - gorm

```
Pessoa.findAllByNomeLikeAndIdadeGreaterThan("%lucas%", 25)
```

```
Pessoa.get(1234)
```

```
Date dateCreated  
Date lastUpdated
```

```
Pessoa.findXXX(xxx, [max:10, offset:10, sort: "nome", order: "asc", cache: true])
```

Mapeamento OR - gorm

```
Pessoa.findAllByNomeLikeAndIdadeGreaterThan("%lucas%", 25)
```

```
Pessoa.get(1234)
```

```
Date dateCreated
```

```
Date lastUpdated
```

```
Pessoa.findXXX(xxx, [max:10, offset:10, sort: "nome", order: "asc", cache: true])
```

```
Pessoa.findAll("from Pessoa p where p.idade = 25")
```

Mapeamento OR - gorm

```
Pessoa.findAllByNomeLikeAndIdadeGreaterThan("%Lucas%", 25)
```

```
Pessoa.get(1234)
```

```
Date dateCreated
```

```
Date lastUpdated
```

```
Pessoa.findXXX(xxx, [max:10, offset:10, sort: "nome", order: "asc", cache: true])
```

```
Pessoa.findAll("from Pessoa p where p.idade = 25")
```

```
def onLoad() { ... }
```

```
def beforeInsert() { ... }
```

```
def beforeUpdate() { ... }
```

```
def beforeDelete() { ... }
```

- Desenvolvimento rápido
 - Full stack
 - Fácil bootstrap
 - Convenções - CoC
 - Scaffolding
 - Plugins

Fácil bootstrap

- ▶ `grails create-app umesp`
- ▶ `grails run-app`

Welcome to Grails

←

→

↺

✕

🏠

🌐

http://localhost:8080/umesp/

☆

▼

🌐

Google

🔍

⚙️


Most Visited ▾ Getting Started Orquestra Sinfônica ... Latest Headlines 📡 Apple Yahoo! Google Maps YouTube >>

🌐

Welcome to Grails

+

🗑️

 **GRAILS**

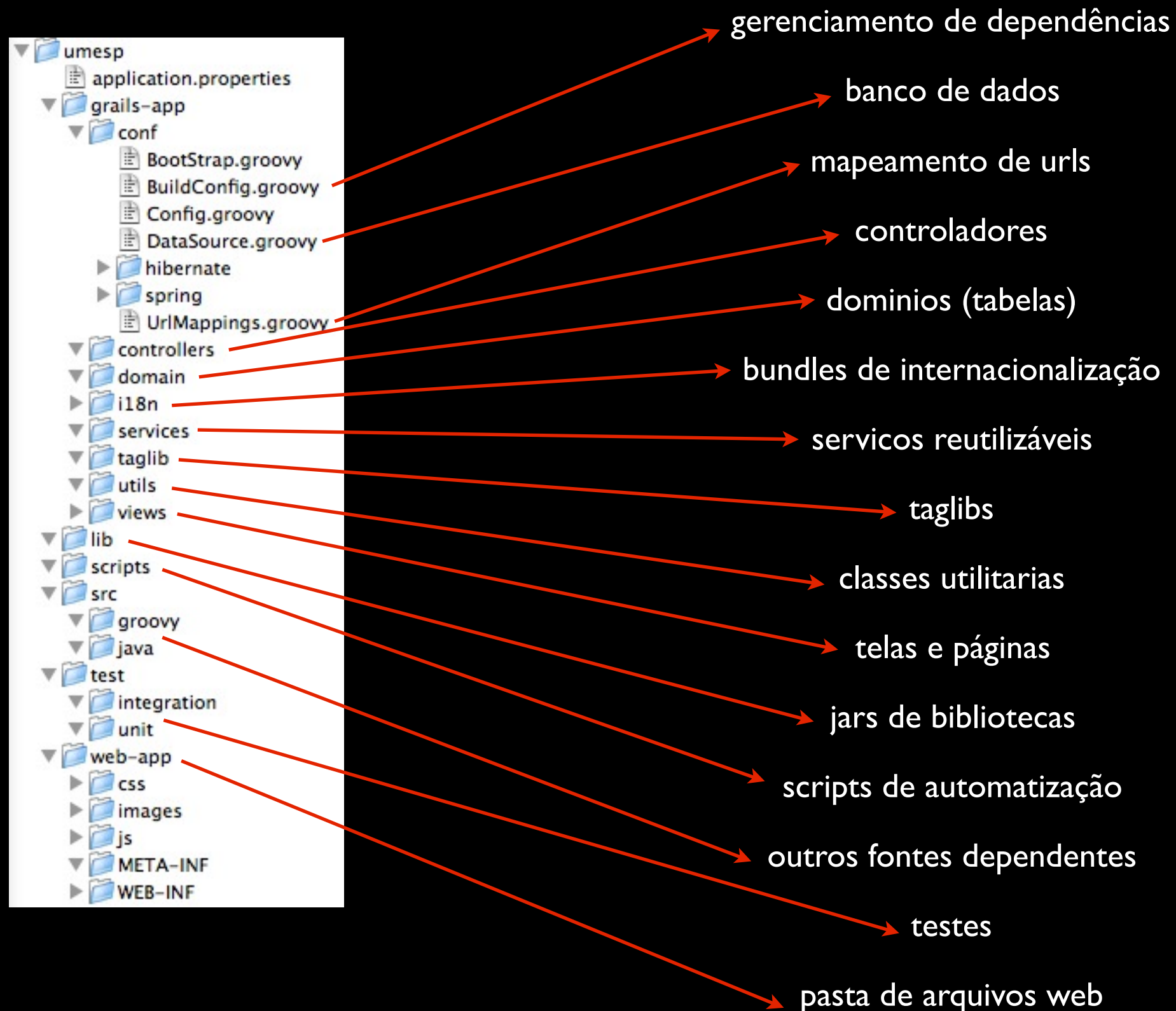
APPLICATION STATUS
App version: 0.1
Grails version: 1.3.1
JVM version: 1.6.0_20
Controllers: 0
Domains: 0
Services: 0
Tag Libraries: 9

INSTALLED PLUGINS
i18n - 1.3.1
logging - 1.3.1
filters - 1.3.1
core - 1.3.1
codecs - 1.3.1
tomcat - 1.3.1
urlMappings - 1.3.1
servlets - 1.3.1
dataSource - 1.3.1
groovyPages - 1.3.1
controllers - 1.3.1
mimeTypeNames - 1.3.1
domainClass - 1.3.1
scaffolding - 1.3.1
converters - 1.3.1
hibernate - 1.3.1
services - 1.3.1
validation - 1.3.1

Welcome to Grails
Congratulations, you have successfully started your first Grails application! At the moment this is the default page, feel free to modify it to either redirect to a controller or display whatever content you may choose. Below is a list of controllers that are currently deployed in this application, click on each to execute its default action:

Available Controllers:

Done



- Desenvolvimento rápido
 - Full stack
 - Fácil bootstrap
 - Convenções - CoC
 - Scaffolding
 - Plugins

CoC - Classes de domínio

```
class Pessoa {  
    String nome  
    String telefone  
    String email  
    String cpf  
  
    static constraints = {  
        nome(nullable:false)  
        telefone(nullable:true, maxSize: 8)  
        email(email:true)  
        cpf(cpf:true)  
    }  
}
```

CoC – Classes de domínio

```
class Pessoa {  
    String nome  
    String telefone  
    String email  
    String cpf  
  
    static constraints = {  
        nome(nullable:false)  
        telefone(nullable:true, maxSize: 8)  
        email(email:true)  
        cpf(cpf:true)  
    }  
  
    static mapping = {  
        table 'TBL_PESS'  
        nome column: 'STR_NM'  
        telefone column: 'STR_TLFN'  
        email column: 'STR_ML'  
        CPF column: 'STR_CPF'  
    }  
}
```

CoC – Controllers

```
class PessoaController {  
    def criar = {  
        def pessoa = new Pessoa(params)  
        pessoa.save()  
    }  
}
```

CoC – Controllers

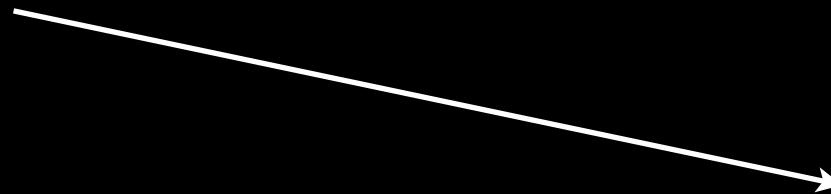
```
class PessoaController {
```

```
  def criar = {
```

```
    def pessoa = new Pessoa(params)  
    pessoa.save()
```

```
  }
```

```
}
```



<http://localhost:8080/app/pessoa/criar>

CoC – Controllers

```
class PessoaController {
```

```
    def criar = {
```

```
        def pessoa = new Pessoa(params)
        pessoa.save()
    }
```

```
}
```

<http://localhost:8080/app/pessoa/criar>

grails-app/
views/


pessoa/
criar.gsp

- Desenvolvimento rápido
 - Full stack
 - Fácil bootstrap
 - Convenções - CoC
 - Scaffolding
 - Plugins

Scaffolding

- Prototipação das Domain Classes
- Estático / Dinâmico (templates)
- Integridade relacional e dos tipos
- Validação

Scaffolding




[Home](#) [New Pessoa](#)

Pessoa List

| Id | Nome | Telefone | Email | Cpf |
|----|---------|----------|--------------------|-------------|
| 1 | Lucas | 12345678 | lucastex@gmail.com | 12345678901 |
| 2 | Fulano | 87654321 | fulano@email.com | 24680246800 |
| 3 | Ciclano | 34567890 | ciclano@email.com | 88877766655 |


Scaffolding



Home New Pessoa

Pessoa List

| Id | Nome | Telefone | Email | Cpf |
|----|---------|----------|--------------------|-------------|
| 1 | Lucas | 12345678 | lucastex@gmail.com | 12345678901 |
| 2 | Fulano | 87654321 | fulano@email.com | 24680246800 |
| 3 | Ciclano | 34567890 | ciclano@email.com | 88877766655 |



Home Pessoa List

Create Pessoa

Nome


Telefone

Email

Cpf

Create


Scaffolding



[Home](#) [New Pessoa](#)

Pessoa List

| Id | Nome | Telefone | Email | Cpf |
|----|---------|----------|--------------------|-------------|
| 1 | Lucas | 12345678 | lucastex@gmail.com | 12345678901 |
| 2 | Fulano | 87654321 | fulano@email.com | 24680246800 |
| 3 | Ciclano | 34567890 | ciclano@email.com | 88877766655 |



[Home](#) [Pessoa List](#)


Create Pessoa


Nome

Telefone

Email

Cpf

 Create



[Home](#) [Pessoa List](#) [New Pessoa](#)

Show Pessoa

Id

1

Nome

Lucas

Telefone


12345678


Email

lucastex@gmail.com

Cpf

12345678901

 Edit

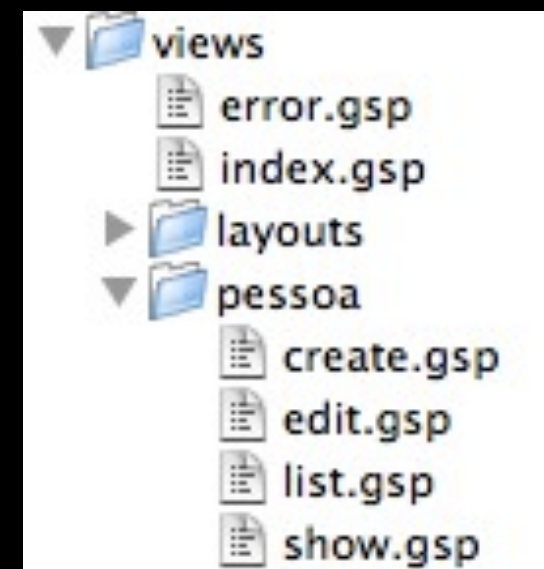
 Delete

Scaffolding estático

- ▶ `grails generate-all Pessoa`

Scaffolding estático


- `grails generate-all Pessoa`



Scaffolding dinâmico


```
class PessoaController {  
    def scaffold = true  
}
```

Validação




Home Pessoa List

Create Pessoa

 O campo [email] da classe [class umesp.Pessoa] com o valor [teste] não é um endereço de email válido.

| | |
|----------|--|
| Nome | <input type="text" value="Lucas"/> |
| Telefone | <input type="text" value="12345678"/> |
| Email | <input type="text" value="teste"/> |
| Cpf | <input type="text" value="12345678901"/> |

 Create

- Desenvolvimento rápido
 - Full stack
 - Fácil bootstrap
 - Convenções - CoC
 - Scaffolding
 - Plugins

Plugins

- fácil extensão
- modularização de aplicações
- adição de métodos / comportamento
- gerenciamento automático

Plugins

- 500+ plugins atualmente
- segurança
- manipulação de imagens
- cms / blog
- quartz (agendamento)
- ui
- lucene

Plugins

- mail
- gwt
- feed / rss
- geração de diagramas de classe
- gráficos
- jquery
- auditoria

Plugins

- performance
- behaviour
 - taggable
 - featurable
 - rateable
 - attachmentable
 - commentable

segredo?

Rock solid foundations

- Spring

Rock solid foundations

- Spring
- Hibernate

Rock solid foundations

- Spring
- Hibernate
- Sitemesh

Rock solid foundations

- Spring
- Hibernate
- Sitemesh
- Log4J

Rock solid foundations

- Spring
- Hibernate
- Sitemesh
- Log4J
- Plugins

Rock solid foundations

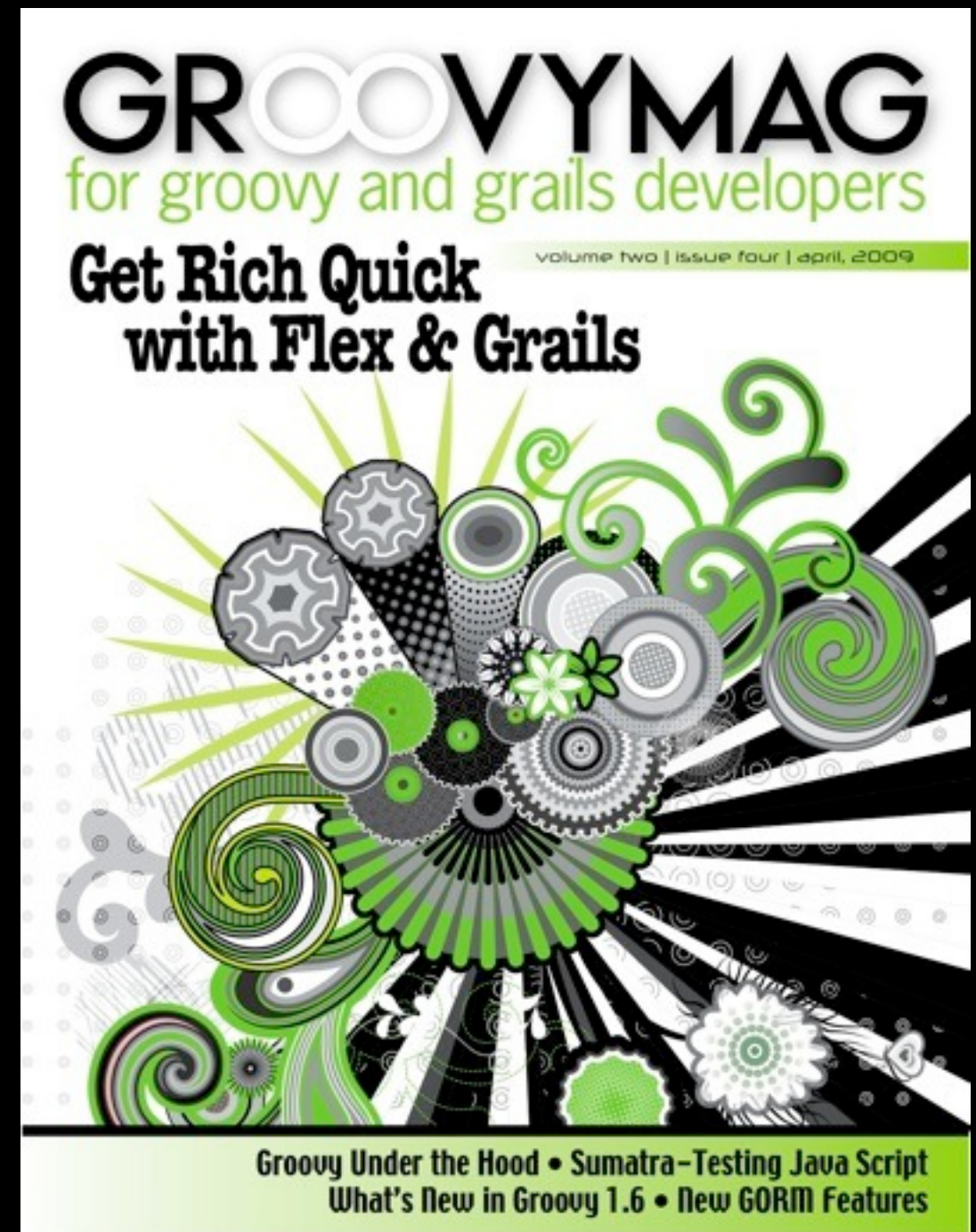
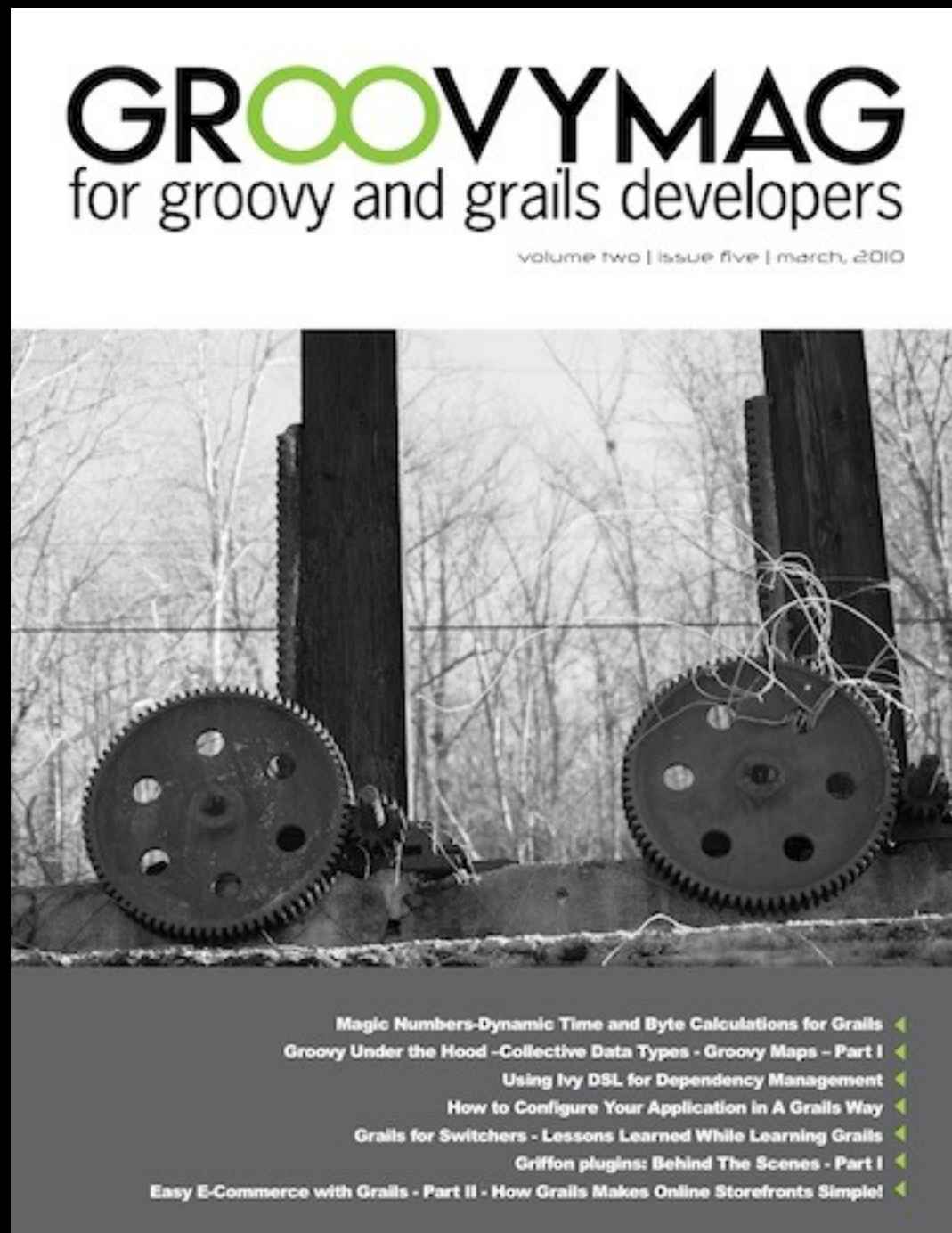
- Spring
- Hibernate
- Sitemesh
- Log4J
- Plugins
- JAVA / JVM

Comunidade

Fontes de informação

- `grails.org`
- `gr8forums.org` (en)
- `grailsbrasil.com` (pt)
- `springsource.com`
- `groovy.codehaus.org`
- twitter (`#groovy` e `#grails`)
- Listas de e-mail
 - `groovy-user`
 - `grails-user`

GroovyMag . com



brasilitfree

brasilit10

Dúvidas?

Demo !

GraiLs

Java produtivo e divertido

Lucas Teixeira

lucastex.com.br

lucastex.com

@lucastex